

# Heading Hierarchy Reconstruction for Complex and Long Documents with Level-Wise Multimodal Memory

Qian Chen, Xianyin Zhang, Lifan Guo, Chi Zhang

Qwen DianJin Team, Alibaba Cloud Computing

{xiaoyu.cq,zhangxianyin.zxy,lifan.lg,edward.zhang}@alibaba-inc.com

## Abstract

Vision-language models (VLMs) have been widely adopted for document parsing and optical character recognition (OCR), as they can accurately extract page-level elements. However, downstream applications such as retrieval-augmented generation (RAG) also require coherent document-level structure. Although these models indicate heading levels using different numbers of preceding “#” symbols, page-by-page prediction often lacks document-level consistency. In practice, input-resolution and context-length constraints make it infeasible for a VLM to jointly process all page images in documents containing tens or hundreds of pages. To address this limitation, we propose a VLM-based framework that incrementally reconstructs heading hierarchies with a *level-wise multimodal memory*. The memory retains a compact subset of all previously processed pages, selected by anchors. Our framework supports flexible anchor selection, and in this work we select the most recent node at each level, allowing the VLM to observe the visual, textual, and layout patterns of headings at every level. For comprehensive evaluation, we also introduce the Complex and Long Document Heading Hierarchy Reconstruction Benchmark (CLDoc-HHRBench), comprising 250 documents and 11,212 pages with expert-verified annotations. Experiments on CLDoc-HHRBench and three existing benchmarks demonstrate the effectiveness of our method.

## 1 Introduction

From scientific papers and technical manuals to laws and business reports, a massive amount of knowledge is stored in unstructured documents. To effectively manage, retrieve, and understand these documents, document parsing has become a critical research topic. Among the structural components of a document, its heading hierarchy provides a navigational backbone: it organizes section headings into a coherent tree and associates them with their

locations in the document. Recovering this hierarchy is important for downstream applications such as retrieval-augmented generation (RAG), where document structure supports structure-aware chunking, navigation, and full-text understanding. In this work, we focus specifically on *heading hierarchy reconstruction*, the task of assigning globally consistent hierarchy levels to document headings and recovering their tree-structured organization.

Heading hierarchy reconstruction remains challenging because document layouts, heading styles, and hierarchy depths vary widely. Early approaches relied on heuristic rules and regular expressions, while subsequent machine- and deep-learning methods combined CNNs, BERT (Devlin et al., 2019), and optical character recognition (OCR) engines to model visual, textual, and layout features. However, the reliance of these methods on multiple specialized components and limited-scale training data can hinder cross-modal alignment and generalization. Recently, methods based on vision-language models (VLMs) have reconstructed heading hierarchies from page-level parsing results. MinerU-Popo (Xu et al., 2026) uses dynamic chunking, vertically concatenates pages within each chunk, and synchronizes levels through overlaps. The resulting long images may dilute per-page visual detail, while offset-based synchronization cannot directly correct non-uniform within-chunk errors. LingDT-VL-OCR (Qian et al., 2026) aggregates heading crops into Pseudo-ToC images and prompts a VLM to recover heading levels. As the number of headings increases, however, both the Pseudo-ToC input and the generated hierarchy become longer, making predictions more susceptible to omissions, duplications, and inconsistent levels in ultra-long documents.

These limitations reveal a fundamental trade-off when applying VLMs to long documents. Providing more page images preserves fine-grained visual evidence, but tens or hundreds of images quickly

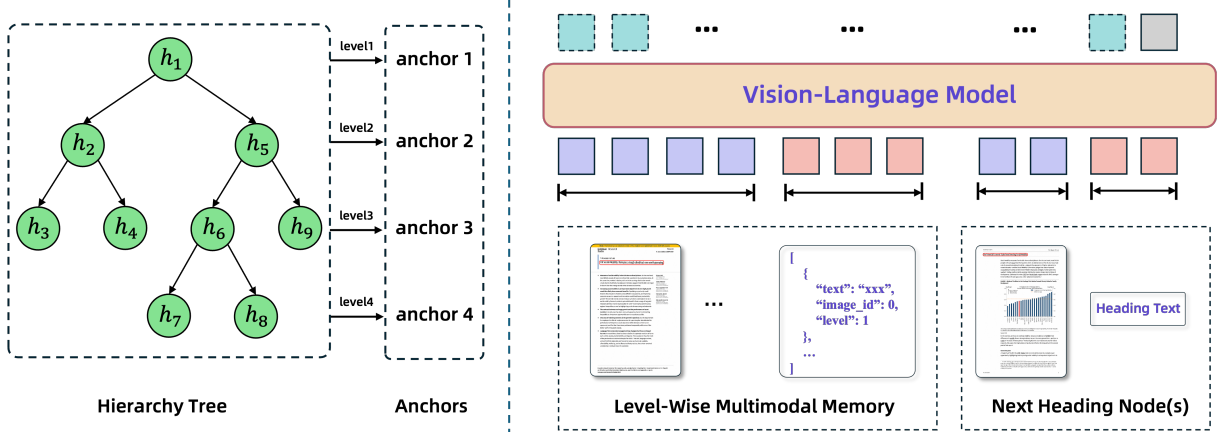


Figure 1: An overview of our proposed method. **Left:** a hierarchy tree constructed from the preceding predictions, where each circle denotes a heading node. **Right:** the memory and the next target heading are fed into the VLM, which is prompted to predict the target heading’s hierarchy level. Anchors guarantee level-wise coverage, while other headings on the selected pages provide additional references for prediction.

exceed the input-resolution and context-length limits of a VLM. Combining many pages into a single image reduces the image count, but may compress page-level visual cues. To balance visual fidelity and historical coverage, we ask whether heading hierarchies can be reconstructed without repeatedly providing the complete page history, while still preserving sufficient historical context. To answer this question, we propose a VLM-based framework that incrementally reconstructs heading hierarchies with a *level-wise multimodal memory*. The memory uses heading nodes as anchors to select a compact set of previously processed pages and supports flexible anchor-selection strategies. In this work, we retain the most recent node at each observed hierarchy level. The selected pages therefore preserve representative visual styles, heading texts, layout patterns, and predicted structural relationships across all levels encountered so far. At each prediction step, these historical references are provided together with the next target heading or target pages, enabling the VLM to infer new levels while maintaining document-level consistency. Because the historical memory remains fixed within each inference step, the framework supports prediction at different granularities, ranging from a single heading to headings on multiple consecutive pages.

In addition, existing benchmarks do not adequately capture both the length and structural complexity of real-world documents. Comp-HRDoc (Wang et al., 2024a) consists solely of scientific papers with similar layouts and hierarchical structures. Although DocHieNet (Xing et al., 2024) covers multiple document types, its document length

and average hierarchy complexity remain limited. To support the evaluation of heading hierarchy reconstruction on complex and long documents, we introduce the Complex and Long Document Heading Hierarchy Reconstruction Benchmark (CLDoc-HHRBench). It comprises 250 documents and 11,212 pages across 10 document categories, covering diverse layouts and larger heading hierarchies than existing benchmarks. Its longest document spans 387 pages, while some documents contain hundreds or even over a thousand headings.

Our main contributions can be summarized as follows:

- We propose a VLM-based framework with a *level-wise multimodal memory* for heading hierarchy reconstruction.
- We introduce CLDoc-HHRBench, a benchmark covering complex, long, and diverse documents.
- Extensive experiments validate our method and provide a comprehensive evaluation of heading hierarchy reconstruction on CLDoc-HHRBench.

## 2 Related Work

### 2.1 Vision-Language Models

#### 2.1.1 General VLMs

General VLMs have emerged as an alternative paradigm for document parsing and understanding. Closed-source models such as the GPT series (Achiam et al., 2023) and the Gemini family

(Team et al., 2023) demonstrate strong OCR capabilities, often outperforming traditional pipeline-based systems in recognizing text, formulas, and tables. Open-source models such as the Qwen collection (Wang et al., 2024b; Bai et al., 2025) and the InternVL family (Zhu et al., 2025; Wang et al., 2025) also show considerable promise for OCR applications.

### 2.1.2 OCR VLMs

OCR-oriented VLMs simplify document parsing by directly processing entire document pages or substantial regions through a unified framework, generating structured outputs without the need for multiple specialized models. End-to-end methods like DeepSeek-OCR (Wei et al., 2025) and FireRed-OCR (Wu et al., 2026) unify recognition within a single framework, while two-stage methods like PaddleOCR-VL (Cui et al., 2025) and MinerU 2.5 (Niu et al., 2026) first perform page-level layout analysis and then conduct efficient parallel parsing of identified regions. Although these models excel at layout analysis, reading order detection, and multiple element recognition, they do not explicitly address document-level heading hierarchy reconstruction.

## 2.2 Heading Hierarchy Reconstruction Methods

### 2.2.1 Rule-based Methods

Rule-based methods (Belaïd, 2001; Dresevic et al., 2008; Marinai et al., 2010; Wu et al., 2013) have been widely adopted, relying on predefined patterns, hand-crafted rules, or algorithmic heuristics. These methods typically follow a common pipeline: first identifying the pages that contain the table of contents, then extracting individual entries, and finally linking each entry to its corresponding section in the document.

### 2.2.2 ML-based/DL-based Methods

In contrast to rule-based methods, machine learning (ML) and deep learning (DL) methods (Déjean and Meunier, 2009; Klampfl et al., 2014; Bentabet et al., 2020; Wang et al., 2024a) introduce greater flexibility by leveraging textual, visual, and layout features. D-O-C (Wang et al., 2024a) encodes section headings with a Transformer, predicts parent-child and sibling scores, and serially inserts each heading as the rightmost child of a selected node on the current rightmost subtree.

### 2.2.3 VLM-based Methods

Benefiting from strong cross-modal capabilities, VLMs can jointly interpret the visual appearance, textual semantics, and spatial layout of document headings. MinerU-Popo (Xu et al., 2026) processes long documents through dynamic chunking and synchronization, while LingDT-VL-OCR (Qian et al., 2026) aggregates heading crops into Pseudo-ToC images for document-level hierarchy reconstruction.

## 2.3 Heading Hierarchy Reconstruction Benchmarks

Early benchmarks for heading hierarchy reconstruction were predominantly constructed from arXiv papers. arXivdocs (Rausch et al., 2021) is an early benchmark for hierarchical structure construction and contains 362 annotated pages. HRDoc (Ma et al., 2023) aims to recover the hierarchical structure of PDF documents and contains 2,500 multi-page documents. Comp-HRDoc (Wang et al., 2024a) is built upon the HRDoc-Hard dataset, which comprises 1,000 documents for training and 500 documents for testing. Nevertheless, these datasets are limited to scientific articles with similar structures. More recent benchmarks incorporate more complex layouts and a broader range of document types. E-Periodica (Rausch et al., 2023) mitigates this homogeneity and comprises 542 single pages from magazines. DocHieNet (Xing et al., 2024) contains 1,673 documents from domains including legal, financial, educational, technical, and scientific materials. MPDocBench-Parse (Zhou et al., 2026) contains 433 manually annotated documents with 3,246 pages across 15 document types (of which 420 documents with 3,135 pages are publicly released). However, their document length and hierarchy depth remain limited.

## 3 Method

We incrementally reconstruct a document’s heading hierarchy using a VLM equipped with a level-wise multimodal memory, as illustrated in Figure 1. This section first defines the task and then describes the memory construction and multimodal level prediction.

### 3.1 Problem Definition

Given a document  $D$ , we assume that its section headings have been detected, placed in reading order, and transcribed by upstream processing, yield-

ing a heading sequence

$$H = (h_1, h_2, \dots, h_n). \quad (1)$$

Each heading is represented as

$$h_i = (I_{p(h_i)}, t_i, b_i), \quad (2)$$

where  $p(h_i)$  denotes the index of the page containing  $h_i$ ,  $I_{p(h_i)}$  is the corresponding page image,  $t_i$  is the heading text, and  $b_i = (x_i^{\min}, y_i^{\min}, x_i^{\max}, y_i^{\max})$  is its bounding box on that page. Before a page image is provided to the VLM, the corresponding heading bounding box is drawn directly on the image. The marked page preserves the heading’s authentic visual and layout context while explicitly identifying the referenced heading. The numerical coordinates are not serialized in the textual prompt.

Heading hierarchy reconstruction aims to predict a hierarchy level

$$l_i \in \{1, 2, \dots, L\}, \quad (3)$$

for every heading, forming an ordered heading-level sequence

$$S = ((h_1, l_1), (h_2, l_2), \dots, (h_n, l_n)), \quad (4)$$

where  $L$  denotes the maximum hierarchy depth. Given  $S$ , the hierarchy tree can be constructed deterministically: processing the headings in reading order, each heading with level  $l_i > 1$  is attached as a child of the most recent preceding heading with level  $l_i - 1$ , and each heading with  $l_i = 1$  is attached to a virtual root node.

### 3.2 Level-Wise Multimodal Memory

To preserve document-level context within the VLM input limits, we construct a level-wise multimodal memory.

Specifically, after processing the first  $i$  headings, let  $d_i$  denote the maximum hierarchy level in the hierarchy tree constructed so far:

$$d_i = \max_{1 \leq j \leq i} \hat{l}_j, \quad (5)$$

where  $\hat{l}_j$  denotes the predicted hierarchy level of  $h_j$ . We introduce one anchor for each hierarchy level present in the constructed tree and denote the anchor set by

$$A_i = \{a_i^{(1)}, a_i^{(2)}, \dots, a_i^{(d_i)}\}, \quad (6)$$

where  $a_i^{(k)}$  is the representative heading node assigned level  $k$ . The historical pages containing these anchors constitute the selected page set

$$\mathcal{P}_i = \{p(h_j) \mid h_j \in A_i\}. \quad (7)$$

Using  $\mathcal{P}_i$ , the multimodal memory is defined as

$$M_i = \left\{ \left( I_q, \left\{ (t_j, b_j, \hat{l}_j) \right\}_{\substack{1 \leq j \leq i \\ p(h_j)=q}} \right) \mid q \in \mathcal{P}_i \right\}. \quad (8)$$

When multiple anchors lie on the same page, that page is selected only once. Anchors are used solely for memory-page selection, and all processed headings on the selected pages remain available as structural references. Accordingly, besides the anchors, the VLM input includes the text, bounding boxes, and predicted levels of all other processed headings on those pages. Since there is one anchor for every hierarchy level in the constructed tree,  $|\mathcal{P}_i| \leq |A_i| = d_i$ . Thus, at each inference step, the VLM receives only the historical pages selected by the anchors rather than the complete set of previously processed pages. The bound characterizes the number of selected historical pages in terms of the depth  $d_i$  of the currently constructed tree.

The memory is initialized with the first page that contains at least one heading. We assign level 1 to the first heading and treat the remaining headings in reading order as subsequent prediction targets. Our method directly predicts the level of each heading and remains applicable even if the document contains additional level-1 headings. In contrast, some prior relation-prediction methods (Wang et al., 2024a) treat the first heading as the root and cannot assign a later heading to the same level.

The framework is not tied to a specific anchor-selection strategy. Different strategies can select representative nodes for the observed levels while retaining the same memory construction and prediction procedure. In this work, we select the most recent node predicted at each level as the anchor, yielding a deterministic strategy that maintains one reference for every observed level. For example, when predicting the level of “Section 2.2,” “Chapter 2” is more informative than “Chapter 1,” while “Section 2.1” is more informative than “Section 1.1” or “Section 1.2.”

### 3.3 Level Prediction

We employ a VLM to jointly interpret the visual, textual, and structural information of the headings.

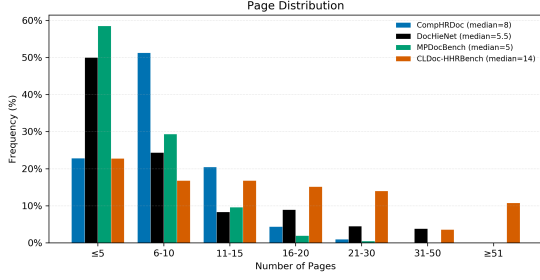


Figure 2: Distribution of document lengths (pages containing headings) across benchmarks. CLDoc-HHRBench has the highest median and a larger proportion of long documents.

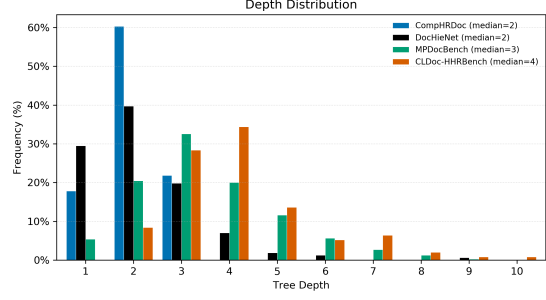


Figure 3: Distribution of tree depths across benchmarks. CLDoc-HHRBench has a higher median depth and a broader tail toward deeper hierarchies.

| Measure             | Comp-HRDoc | DocHieNet | MPDocBench-Parse | CLDoc-HHRBench |
|---------------------|------------|-----------|------------------|----------------|
| #Docs               | 500        | 161       | 420              | 250            |
| #Pages (Total)      | 7,043      | 2,311     | 3,135            | 11,212         |
| #Pages (Mean)       | 14.09      | 14.35     | 7.46             | 44.85          |
| #Pages (Maximum)    | 35         | 50        | 30               | 387            |
| #Headings (Total)   | 7,579      | 3,344     | 6,309            | 26,231         |
| #Headings (Mean)    | 15.16      | 20.77     | 15.02            | 104.92         |
| #Headings (Maximum) | 64         | 262       | 112              | 1,772          |
| Depth (Mean)        | 2.04       | 2.13      | 3.35             | 4.15           |
| Depth (Maximum)     | 3          | 9         | 9                | 10             |

Table 1: Comparison with existing datasets. Counts for Comp-HRDoc and DocHieNet refer to their test sets.

Given the memory  $M_{i-1}$  and the next heading  $h_i$ , the VLM predicts its level as

$$\hat{l}_i = \arg \max_{l \in \mathcal{L}_i} p_\theta(l \mid M_{i-1}, h_i), \quad (9)$$

where  $\theta$  denotes the model parameters. For single-heading prediction, the hierarchy level may remain unchanged, increase by one, or return to any shallower level. Accordingly, the candidate set is

$$\mathcal{L}_i = \{1, \dots, \hat{l}_{i-1} + 1\}. \quad (10)$$

After predicting  $\hat{l}_i = k$ , the new heading is attached to the most recent preceding heading at level  $k - 1$  when  $k > 1$ , or to the virtual root when  $k = 1$ .

Our framework also supports flexible prediction granularity. Because the historical page memory remains fixed within each inference step, the VLM can predict all headings on one or more consecutive target pages using the same memory context. The target pages are provided together with the memory pages, and the VLM outputs the heading levels in reading order. For a group beginning with  $h_i$ , the first predicted level is constrained by  $\hat{l}_{i-1}$ , and every subsequent prediction must satisfy  $\hat{l}_j \leq \hat{l}_{j-1} + 1$ . Grouping target pages reduces inference calls and may also provide useful look-ahead context, as later headings in the group can offer

visual, semantic, and structural cues for predicting earlier headings.

## 4 Benchmark

### 4.1 Benchmark Construction

To evaluate heading hierarchy reconstruction on complex and long documents, we construct **CLDoc-HHRBench**, comprising 250 documents and 11,212 pages. The source files are publicly available PDFs or Microsoft Word documents collected from the Internet, with no document-level overlap with the existing benchmarks used in this work. The benchmark spans 10 document types (Appendix A) and includes long documents and deep heading hierarchies. Figures 2 and 3 compare its document-length and tree-depth distributions with existing benchmarks, and Table 1 summarizes their statistics.

### 4.2 Annotation

We identify the headings in each document and annotate their bounding boxes, text, reading order, and hierarchy levels. Each document was labeled independently by two annotators and adjudicated by experts. Six annotators and three experts spent approximately 800 person-hours; the

| Methods<br>Name      | Comp-HRDoc        |                   | DocHieNet         |                   | MPDocBench-Parse  |                   | CLDoc-HHRBench    |                   | Overall           |                   |
|----------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|-------------------|
|                      | Macro             | Micro             | Macro             | Micro             | Macro             | Micro             | Macro             | Micro             | Macro             | Micro             |
| Different Strategies |                   |                   |                   |                   |                   |                   |                   |                   |                   |                   |
| PaddleX              | 91.37±0.00        | 90.20±0.00        | 48.66±0.00        | 43.51±0.00        | 42.77±0.00        | 43.30±0.00        | 48.55±0.00        | 39.68±0.00        | 62.83±0.00        | 49.60±0.00        |
| MinerU               | 94.33±0.13        | 94.10±0.03        | 74.86±0.57        | 73.59±0.15        | 71.26±0.55        | 71.45±0.42        | 80.62±0.20        | 73.10±2.47        | 82.12±0.22        | 76.66±1.38        |
| Dynamic Chunking     | 93.08±0.25        | 92.38±0.25        | 78.07±0.19        | 75.65±0.37        | 70.38±0.33        | 70.94±0.93        | 80.96±0.56        | 83.61±0.31        | 82.15±0.09        | 82.71±0.54        |
| Pseudo-ToC           | 94.44±0.20        | 94.04±0.21        | 74.61±0.34        | 76.17±0.20        | 72.94±0.11        | 75.13±0.12        | 83.90±0.33        | 79.09±1.71        | 83.64±0.10        | 81.03±0.98        |
| Ours                 | <b>95.73±0.26</b> | <b>95.49±0.19</b> | <b>78.98±0.41</b> | <b>79.03±0.40</b> | <b>73.07±0.62</b> | <b>76.60±0.32</b> | <b>86.01±0.24</b> | <b>91.07±0.15</b> | <b>84.73±0.32</b> | <b>88.68±0.17</b> |
| Separate SFT Results |                   |                   |                   |                   |                   |                   |                   |                   |                   |                   |
| MinerU-Popo* (4B)    | 96.32±0.12        | 95.86±0.20        | 76.66±0.27        | 70.97±0.10        | 71.87±0.09        | 71.13±0.10        | 84.18±0.06        | 85.64±0.42        | 83.95±0.02        | 84.06±0.25        |
| Ours (0.8B)          | 96.09±0.25        | 95.53±0.25        | 75.95±0.54        | 74.62±0.56        | 67.41±0.15        | 69.15±0.13        | 87.08±0.27        | 87.85±0.46        | 82.91±0.12        | 85.32±0.25        |

Table 2: Performance comparison of different methods using Macro-STEDS and Micro-STEDS. Scores are percentages reported as mean  $\pm$  standard deviation over three runs, except for deterministic PaddleX, which is evaluated once. Bold and underline indicate the best and second-best results, respectively, in each column within the Different Strategies group. The 0.8B SFT result illustrates the feasibility of fine-tuning our framework on a small model; the separately trained MinerU-Popo\* 4B model is shown only as an external reference, not a controlled comparator.

annotation protocol and validation are described in Appendix B.

## 5 Experiments

### 5.1 Experiment Setup

We compare our method with four baselines. The first is **PaddleX**<sup>1</sup>, which constructs the heading hierarchy using rule-based features, including heading text, numbering patterns, bounding-box height, and global reading order. The second is **MinerU**<sup>2</sup>, which employs an LLM to infer heading levels by jointly processing the ordered headings across the entire document. The remaining two are **Dynamic Chunking (MinerU-Popo)** (Xu et al., 2026) and **Pseudo-ToC (LingDT-VL-OCR)** (Qian et al., 2026).

A uniform comparison using the original implementations of both VLM-based baselines is not feasible: MinerU-Popo fine-tunes Qwen3-VL-4B<sup>3</sup> with 30K training instances, whereas LingDT-VL-OCR has not released its model. We therefore reproduce their heading hierarchy reconstruction strategies with Qwen3.6-Plus<sup>4</sup> as a shared backbone, thereby isolating differences among the reconstruction strategies. Apart from strategy-specific prompts, all reproduced VLM-based strategies use the same backbone and inference hyperparameters. Unless otherwise specified, our method

uses one-page prediction. An output is accepted only if it can be parsed as valid JSON and satisfies  $1 \leq \hat{l}_i \leq \hat{l}_{i-1} + 1$  for every target in reading order. If validation fails, we retry the prediction up to two times. For the rare case in which a single-heading prediction remains invalid, we project the predicted level onto the valid range.

To assess whether our framework can be used to fine-tune a small model, we conduct a separate SFT feasibility experiment. We collect documents from the training sets of Comp-HRDoc and DocHieNet and from publicly available documents found online. We manually annotate them to create 10K training instances and fine-tune Qwen3.5-0.8B<sup>5</sup> using our method. The documents underlying these training instances are strictly disjoint from all evaluation documents. We train with Llama-Factory (Zheng et al., 2024) on three NVIDIA A100 GPUs for one epoch, using a learning rate of  $5.0 \times 10^{-6}$ , a per-device batch size of 2, four gradient-accumulation steps, and an 8K maximum context length.

For each reproduced baseline, we follow the default settings specified in its paper or official GitHub repository. The experimental prompts and remaining implementation details are provided in Appendix C and Appendix D.

### 5.2 Evaluation Protocols

We compare hierarchy reconstruction strategies using the same ground-truth headings for every benchmark document. All methods receive annotated heading text, page indices, bounding boxes, and reading order, and predict only hierarchy lev-

<sup>1</sup>[https://github.com/PaddlePaddle/PaddleX/blob/release/3.7/paddlex/inference/pipelines/layout\\_parsing/title\\_level.py](https://github.com/PaddlePaddle/PaddleX/blob/release/3.7/paddlex/inference/pipelines/layout_parsing/title_level.py)

<sup>2</sup>[https://github.com/opendatalab/MinerU/blob/master/mineru/utils/llm\\_aided.py](https://github.com/opendatalab/MinerU/blob/master/mineru/utils/llm_aided.py)

<sup>3</sup><https://huggingface.co/Qwen/Qwen3-VL-4B-Instruct>

<sup>4</sup><https://qwen.ai/blog?id=qwen3.6>

<sup>5</sup><https://huggingface.co/Qwen/Qwen3.5-0.8B>

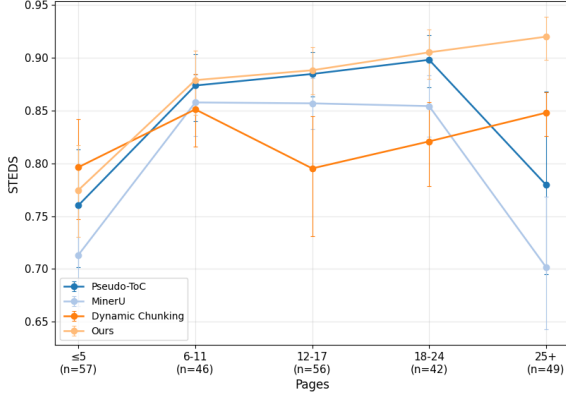


Figure 4: Micro-STEDS by the number of heading-bearing pages on CLDoc-HHRBench. Error bars show 95% document-bootstrap confidence intervals;  $n$  denotes documents per group.

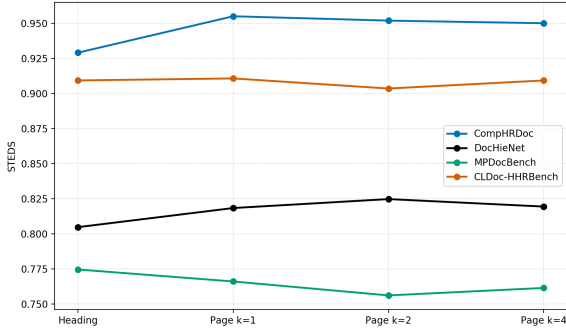


Figure 5: Effect of prediction granularity on Micro-STEDS across four benchmarks.

els. In the main comparison, ground-truth hierarchy levels are not provided as historical context: every incremental method updates its state using its own previous level predictions. This evaluation measures the hierarchy reconstruction strategies themselves; the impact of upstream errors on an end-to-end document parsing pipeline is outside the scope of this work.

Following prior work (Ma et al., 2023), we adopt STEDS as the primary evaluation metric. For document  $i$  among the  $N$  evaluated documents, let  $T_i$  and  $\hat{T}_i$  denote its ground-truth and predicted hierarchies, respectively, where  $ed_i = \text{EditDist}(T_i, \hat{T}_i)$  is their tree edit distance and  $n_i = \max(|T_i|, |\hat{T}_i|)$  is the larger number of nodes in the two trees. We report Macro-STEDS and Micro-STEDS:

$$\text{Macro-STEDS} = \frac{1}{N} \sum_{i=1}^N \left(1 - \frac{ed_i}{n_i}\right), \quad (11)$$

$$\text{Micro-STEDS} = 1 - \frac{\sum_{i=1}^N ed_i}{\sum_{i=1}^N n_i}. \quad (12)$$

A Micro-STEDS score higher than the corresponding Macro-STEDS score generally indicates better performance on longer documents or larger hierarchy trees. Unless otherwise specified, we report Micro-STEDS.

### 5.3 Experimental Results

Table 2 demonstrates the overall effectiveness of our method. In the comparison of different strategies, it achieves Overall Macro-STEDS and Micro-STEDS scores of 84.73 and 88.68, outperforming the strongest competing results by 1.09 and 5.97 points, respectively. The advantage is most pronounced on CLDoc-HHRBench, where our score of 91.07 exceeds the next-best score of 83.61 by 7.46 points, demonstrating the effectiveness of our method on long and structurally complex documents. In contrast, PaddleX remains competitive only on Comp-HRDoc and degrades markedly on the other benchmarks, indicating the limited generalization of rule-based methods. MinerU, which relies on textual heading sequences, underperforms the multimodal VLM-based methods overall, suggesting that page images provide more comprehensive visual and structural cues. Dynamic Chunking also performs worse overall, likely because vertical page concatenation weakens visual cues and overlap-based synchronization cannot fully correct non-uniform errors.

In the separate SFT feasibility experiment, fine-tuning Qwen3.5-0.8B with our framework yields 85.32 Overall Micro-STEDS, showing that the framework can also be applied to a small model.

Figure 4 groups CLDoc-HHRBench documents by the number of heading-bearing pages and reports their Micro-STEDS scores. Our advantage is most pronounced for documents with at least 25 such pages. As documents lengthen, Pseudo-ToC must process more page images and generate a longer hierarchy, making it harder for the VLM to integrate visual evidence and maintain consistent heading levels. Our level-wise memory selects historical pages by observed hierarchy level and keeps the context compact as page count grows. This may be one reason for our method’s larger gains on CLDoc-HHRBench.

### 5.4 Analysis

#### 5.4.1 Effect of Prediction Granularity

As our framework supports prediction at multiple granularities, we evaluate four settings under the same evaluation protocol. The prediction unit is set

| Strategy      | Comp-<br>HRDoc    | DocHieNet         | MPDocBench-<br>Parse | CLDoc-<br>HHRBench | Overall           |
|---------------|-------------------|-------------------|----------------------|--------------------|-------------------|
|               |                   |                   |                      |                    |                   |
| Random        | 94.62±0.14        | 74.86±0.24        | 73.53±0.61           | 74.28±1.77         | 78.22±0.75        |
| Previous- $n$ | 95.08±0.07        | 78.40±0.20        | 75.90±0.34           | 89.40±0.21         | 87.47±0.12        |
| Recent        | <b>95.49±0.19</b> | <b>79.03±0.40</b> | <b>76.60±0.32</b>    | <b>91.07±0.15</b>  | <b>88.68±0.17</b> |

Table 3: Micro-STEDS comparison of historical memory selection strategies. Previous- $n$  uses  $n = 3$ .

| Method    | Comp-<br>HRDoc    | DocHieNet         | MPDocBench-<br>Parse | CLDoc-<br>HHRBench | Overall           |
|-----------|-------------------|-------------------|----------------------|--------------------|-------------------|
|           |                   |                   |                      |                    |                   |
| Text-only | 91.23±0.62        | 70.96±0.61        | 66.78±0.84           | 86.71±0.32         | 83.47±0.12        |
| Ours      | <b>95.49±0.19</b> | <b>79.03±0.40</b> | <b>76.60±0.32</b>    | <b>91.07±0.15</b>  | <b>88.68±0.17</b> |

Table 4: Micro-STEDS comparison between the text-only variant and our multimodal method.

| Memory    | Comp-<br>HRDoc    | DocHieNet         | MPDocBench-<br>Parse | CLDoc-<br>HHRBench | Overall           |
|-----------|-------------------|-------------------|----------------------|--------------------|-------------------|
|           |                   |                   |                      |                    |                   |
| Predicted | 95.51±0.17        | 71.82±1.07        | 73.24±0.18           | 84.84±0.87         | 83.89±0.49        |
| GT-guided | <b>97.25±0.08</b> | <b>77.41±0.35</b> | <b>73.38±0.04</b>    | <b>91.59±0.15</b>  | <b>88.73±0.11</b> |

Table 5: Parent accuracy (%) with predicted or fully ground-truth-guided memory.

to a single heading, all headings on one page, or all headings on two or four consecutive pages. Figure 5 reports the results on four benchmarks. Performance remains relatively stable when a prediction unit spans multiple consecutive pages. These results suggest that the framework can accommodate multi-page prediction, which may benefit from look-ahead context, although its effect and the optimal granularity vary across benchmarks.

#### 5.4.2 Comparison of Memory Strategies

Table 3 compares our method with Previous- $n$  and Random on four benchmarks. Previous- $n$  retains the  $n$  headings immediately preceding the current target in reading order as historical context, whereas our Most Recent strategy selects the latest heading at each observed level. Random selects one processed node at each observed level at random. Overall, Most Recent achieves a Micro-STEDS of 88.68, compared with 87.47 for Previous- $n$  and 78.22 for Random, underscoring the importance of selecting useful historical context.

#### 5.4.3 Effect of Multimodal Context

To assess the contribution of visual information, we compare our method with a text-only variant under the same evaluation protocol. The text-only variant follows the same memory selection and prediction procedure but excludes page images, while retaining heading texts, bounding-box coordinates,

and predicted levels in textual form. Table 4 reports the results on four benchmarks. Our multimodal method consistently outperforms the text-only variant across all benchmarks, demonstrating the importance of visual context for reconstructing document hierarchies.

#### 5.4.4 Sensitivity to Historical Hierarchy Information

Because both anchor selection and the level labels exposed in memory depend on the predicted hierarchy, historical errors may affect subsequent predictions. We examine this sensitivity with a fully ground-truth-guided diagnostic: the ground-truth hierarchy is used to select the most recent anchor at each level, and all historical level labels exposed in the resulting memory are replaced with their ground-truth values; all other settings remain unchanged. Table 5 reports parent accuracy, the proportion of headings attached to the correct parent. The predicted-memory setting already achieves 83.89% overall parent accuracy, while replacing the entire historical state with ground-truth information improves it to 88.73%. This moderate overall gap suggests that imperfect historical predictions do not cause severe cascading degradation at the aggregate level.

## 6 Conclusion

We introduce level-wise multimodal memory for VLM-based heading hierarchy reconstruction. Retaining the most recent heading at each observed level provides compact visual and structural context for heading-level and multi-page prediction. We also present CLDoc-HHRBench, a diverse benchmark of long documents and deep hierarchies. Experiments on four benchmarks show particularly strong results on complex, long documents.

## Ethical Considerations

CLDoc-HHRBench is sourced from publicly available PDF and Microsoft Word documents found online and manually annotated by our annotators, with expert adjudication. We have secured the necessary permissions to include the source documents in the benchmark. CLDoc-HHRBench carries low ethical risk: we screen the documents and annotations to exclude sensitive or personally identifiable information. We will release the annotations and source information for the page images.

## References

- Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. GPT-4 technical report. *arXiv preprint arXiv:2303.08774*.
- Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. 2025. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*.
- Abdel Belaïd. 2001. Recognition of table of contents for electronic library consulting. *International Journal on Document Analysis and Recognition*, 4(1):35–45.
- Najah-Imane Bentabet, Rémi Juge, Ismaïl El Maarouf, Dialekti Valsamou-Stanislawski, and Sira Ferradans. 2020. Automatic table-of-contents generation for efficient information access. *SN Computer Science*, 1(5):283.
- Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, et al. 2025. PaddleOCR-VL: Boosting multilingual document parsing via a 0.9b ultra-compact vision-language model. *arXiv preprint arXiv:2510.14528*.
- Hervé Déjean and Jean-Luc Meunier. 2009. On tables of contents and how to recognize them. *International Journal of Document Analysis and Recognition (IJ-DAR)*, 12(1):1–20.
- Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics: human language technologies, volume 1 (long and short papers)*, pages 4171–4186.
- Bodin Drešević, Aleksandar Uzelac, Bogdan Radaković, and Nikola Todić. 2008. Book layout analysis: Toc structure extraction engine. In *International Workshop of the Initiative for the Evaluation of XML Retrieval*, pages 164–171. Springer.
- Stefan Klampfl, Michael Granitzer, Kris Jack, and Roman Kern. 2014. Unsupervised document structure analysis of digital scientific articles. *International journal on digital libraries*, 14(3):83–99.
- 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 AAAI Conference on Artificial Intelligence*, volume 37, pages 1870–1877.
- Simone Marinai, Emanuele Marino, and Giovanni Soda. 2010. Table of contents recognition for converting pdf documents in e-book formats. In *Proceedings of the 10th ACM symposium on Document engineering*, pages 73–76.
- Junbo Niu, Zheng Liu, Zhuangcheng Gu, Bin Wang, Linke Ouyang, Zhiyuan Zhao, Tao Chu, Tianyao He, Fan Wu, Qintong Zhang, et al. 2026. MinerU 2.5: A decoupled vision-language model for efficient high-resolution document parsing. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (ACL 2026)*, pages 13–42.
- Siyi Qian, Xiongfei Bai, Bingtao Fu, Yichen Lu, Gaoyang Zhang, Xudong Yang, and Peng Zhang. 2026. LingDT-VL-OCR: Structure-aware document-level parsing with fine-grained visual reference. *arXiv preprint arXiv:2603.11044*.
- Johannes Rausch, Octavio Martinez, Fabian Bissig, Ce Zhang, and Stefan Feuerriegel. 2021. DocParser: Hierarchical document structure parsing from renderings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4328–4338.
- Johannes Rausch, Gentiana Rashiti, Maxim Gusev, Ce Zhang, and Stefan Feuerriegel. 2023. DSG: An end-to-end document structure generator. In *2023 IEEE International Conference on Data Mining (ICDM)*, pages 518–527. IEEE.
- Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.
- Jiawei Wang, Kai Hu, Zhuoyao Zhong, Lei Sun, and Qiang Huo. 2024a. Detect-Order-Construct: A tree construction based approach for hierarchical document structure analysis. *Pattern Recognition*, 156:110836.
- Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. 2024b. Qwen2-VL: Enhancing vision-language model’s perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.
- Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. 2025. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.
- Haoran Wei, Yaofeng Sun, and Yukun Li. 2025. DeepSeek-OCR: Contexts optical compression. *arXiv preprint arXiv:2510.18234*.
- Hao Wu, Haoran Lou, Xinyue Li, Zuodong Zhong, Zhaojun Sun, Phellon Chen, Xuanhe Zhou, Kai Zuo, Yibo Chen, Xu Tang, et al. 2026. FireRed-OCR technical report. *arXiv preprint arXiv:2603.01840*.

- Zhaohui Wu, Prasenjit Mitra, and C Lee Giles. 2013. Table of contents recognition and extraction for heterogeneous book documents. In *2013 12th international conference on document analysis and recognition*, pages 1205–1209. IEEE.
- 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.
- Bangrui Xu, Ziyang Miao, Xuanhe Zhou, Yiming Lin, Zirui Tang, Xiaomeng Zhao, Fan Wu, Cheng Tan, Bin Wang, and Conghui He. 2026. MinerU-Popo: Universal post-processing model for structured document parsing. *arXiv preprint arXiv:2605.24973*.
- Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyang Luo, Zhangchi Feng, and Yongqiang Ma. 2024. LlamaFactory: Unified efficient fine-tuning of 100+ language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, Bangkok, Thailand. Association for Computational Linguistics.
- Bangbang Zhou, Hangdi Xing, Yifan Chen, Jianjun Xu, Qi Zheng, Feiyu Gao, Zhibo Yang, Shuai Bai, Ming Yan, Jieping Ye, et al. 2026. MPDocBench-Parser: Benchmarking practical multi-page document parsing. *arXiv preprint arXiv:2605.22100*.
- Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, et al. 2025. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*.

## A Benchmark Domain Breakdown

The benchmark covers 10 document types: Insurance Policy Terms (10.0%), Product Manuals (10.8%), Laws and Regulations (5.2%), Financial Reports (12.0%), Guidelines (3.6%), Prospectuses (3.6%), Rating Reports (3.6%), Academic Papers (4.0%), Industry Standards (3.6%), and Research Reports (43.6%). Research Reports cover documents that differ markedly in layout, length, and heading structure. This category can be further divided into five subcategories: In-Depth Research (11.2% of all documents), Initiation of Coverage (8.8%), Company Update (13.2%), Sector Research (5.6%), and Macro & Strategy Research (4.8%).

## B Annotation Protocol

CLDoc-HHRBench was manually annotated from the original pages using a shared protocol. We first identify which elements in each document are headings, then annotate each heading’s bounding box, text, reading order, and normalized hierarchy level. Skipped levels were prohibited: a heading could be at most one level deeper than its predecessor. We also developed a visual annotation platform that displayed pages alongside the hierarchy tree and supported direct editing of all targets.

Each document was independently labeled by two mutually blinded annotators. Before adjudication, agreement was 0.9611 F1 for heading detection, 0.9334 mean box IoU, 0.9761 normalized edit similarity for reading order, and 0.9673 hierarchy STEDS (Ma et al., 2023). The two complete annotation sets for each document were then compared. One of the three experts reviewed each missing or extra heading and each mismatch in box, text, order, or level against the original pages. Any remaining ambiguities were resolved by all three experts through consensus.

After adjudication, deterministic validators checked reading order indices, out-of-page boxes, empty text, skipped levels, and sequence-tree consistency. Flagged documents were corrected and revalidated until all checks passed. Six annotators and three experts spent approximately 800 person-hours. We will release the annotations and source information for the page images.

## C Prompts

This section provides the prompts used in our experiments, including the prompt of our method

and the strategy-specific prompts of the reproduced VLM-based baselines. The prompts of our method, Pseudo-ToC (LingDT-VL-OCR), and Dynamic Chunking (MinerU-Popo) are shown below, each instantiated with headings from the same example document.

In each prompt, <image> denotes the visual input of the corresponding strategy: our method provides the original page images with the heading bounding boxes drawn on them, LingDT-VL-OCR provides the pseudo-ToC images aggregated from heading crops, and MinerU-Popo provides the single image formed by vertically concatenating the pages within a chunk. All strategies share the same output format: the VLM outputs JSON only, mapping each target heading id to its hierarchy level. Because MinerU-Popo releases a fine-tuned model that follows its own output format, the original Dynamic Chunking prompt does not include the output instruction; we append it only when reproducing the strategy with the shared backbone, leaving the rest of the prompt unchanged.

For different prediction granularities, the prompt differs only in the number of target headings to be predicted and the pages containing them. For the alternative memory-selection strategies, Previous-*n* and Random, the prompt is similar to that of our method: only the given page images and the corresponding heading information are replaced according to the selected historical nodes, and everything else remains unchanged. For the text-only variant, the prompt differs only in that the page images are excluded and the bounding boxes are serialized in textual form. The ground-truth-guided diagnostic also reuses the prompt of our method: the given page images and the historical levels exposed in the memory are replaced according to the ground-truth hierarchy. The SFT model is fine-tuned and evaluated with the same prompt used by our method.

## D Implementation Details

This section describes the remaining implementation details, including method-specific configurations and retry and validation settings. The method-specific configurations are summarized as follows:

- **PaddleX:** We reproduce it using the source code at [https://github.com/PaddlePaddle/PaddleX/blob/release/3.7/paddlex/inference/pipelines/layout\\_parsing/title\\_level.py](https://github.com/PaddlePaddle/PaddleX/blob/release/3.7/paddlex/inference/pipelines/layout_parsing/title_level.py).

#### Prompt used for Our Method

```
Parse the titles' hierarchy.

Page Images (each image is preceded by its id [i]):

[1] <image>

[2] <image>

Known Title Levels:
[{"title": "I. INTRODUCTION", "level": "H1", "image_id": 1}, {"title": "II. RGE FOR  $\lambda$ H BELOW SCALE  $MP/\xi$ ", "level": "H1", "image_id": 1}]

Titles to Predict Hierarchy Level:
[{"id": 0, "title": "III. EMBEDDING HIGGS INFLATION INTO", "image_id": 2}, {"id": 1, "title": "SUPERGRAVITY", "image_id": 2}]

Output JSON only, mapping each id to its level, e.g. {"0": "H1", "1": "H2"}. No explanation.
```

#### Prompt used for Pseudo-ToC (LingDT-VL-OCR)

```
Parse the titles' hierarchy.

Page Images (each image is preceded by its id [i]):

[1] <image>

Titles to Predict Hierarchy Level:

[{"id": 0, "title": "I. INTRODUCTION"}, {"id": 1, "title": "II. RGE FOR  $\lambda$ H BELOW SCALE  $MP/\xi$ "}, {"id": 2, "title": "III. EMBEDDING HIGGS INFLATION INTO"}, {"id": 3, "title": "SUPERGRAVITY"}, {"id": 4, "title": "IV. CONSTRAINTS ON SUSY MASS SPECTRUM"}, {"id": 5, "title": "V. DISCUSSION AND SUMMARY"}, {"id": 6, "title": "ACKNOWLEDGMENTS"}]

Output JSON only, mapping each id to its level, e.g. {"0": "H1", "1": "H2"}. No explanation.
```

#### Prompt used for Dynamic Chunking (MinerU-Popo)

```
<image>

Title Level Analysis: <|id|>0<|page|>0<|box|>239 220 250 351<|content|>I. INTRODUCTION
<|id|>1<|page|>0<|box|>825 586 837 848<|content|>II. RGE FOR  $\lambda$ H BELOW SCALE  $MP/\xi$ 
<|id|>2<|page|>1<|box|>406 570 417 864<|content|>III. EMBEDDING HIGGS INFLATION INTO
<|id|>3<|page|>1<|box|>420 660 431 774<|content|>SUPERGRAVITY
<|id|>4<|page|>2<|box|>68 555 79 879<|content|>IV. CONSTRAINTS ON SUSY MASS SPECTRUM
<|id|>5<|page|>3<|box|>808 176 819 398<|content|>V. DISCUSSION AND SUMMARY
<|id|>6<|page|>3<|box|>476 637 487 797<|content|>ACKNOWLEDGMENTS

Output JSON only, mapping each id to its level, e.g. {"0": "H1", "1": "H2"}. No explanation.
```

with its default settings, including a chunk size of 50 and an overlap of 1.

- **Pseudo-ToC:** For the construction of the pseudo-ToC images, each canvas is white and has the same size as the first page containing a heading. Each heading crop is preceded by its rendered id and keeps its original horizontal position; the crops are tiled densely along the vertical axis from top to bottom with a fixed 12px gap, and a new blank canvas is started when the current one overflows.
- **Our Method:** The heading bounding boxes are drawn directly on the original page images, and all page images are provided at their original size.

We apply the same output validation and retry procedure to MinerU, Dynamic Chunking, Pseudo-ToC, and our method. Each output must be valid JSON, and every predicted heading level must satisfy  $1 \leq \hat{l}_i \leq \hat{l}_{i-1} + 1$ , since the four benchmarks contain no skipped levels. We retry invalid predictions up to two times. If a parseable single-heading prediction still violates the level constraint, we project it onto the valid range:  $\hat{l}_i \leftarrow \min(\max(\hat{l}_i, 1), \hat{l}_{i-1} + 1)$ .

- **MinerU:** We reproduce it using the source code at [https://github.com/opendatalab/MinerU/blob/master/mineru/utils/llm\\_aided.py](https://github.com/opendatalab/MinerU/blob/master/mineru/utils/llm_aided.py), with Qwen3.6-Plus as the LLM.
- **Dynamic Chunking:** We follow <https://github.com/opendatalab/MinerU-Popo>