MinerU2.5: A Decoupled Vision-Language Model for Efficient High-Resolution Document Parsing

Junbo Niu[1,2*], **Zheng Liu**[1,2*], **Zhuangcheng Gu**[1*], **Bin Wang**[1*‡], **Linke Ouyang**[1*], **Zhiyuan Zhao**[1*], **Tao Chu**[1*], **Tianyao He**[1*], **Fan Wu**[1*], **Qintong Zhang**[1,2*], **Zhenjiang Jin**[1*], **Guang Liang**[1], **Rui Zhang**[1], **Wenzheng Zhang**[1,2], **Yuan Qu**[1], **Zhifei Ren**[1], **Yuefeng Sun**[1], **Zirui Tang**[1,3], **Boyu Niu**[1,3],**Yuanhong Zheng**[1], **Dongsheng Ma**[1], **Ziyang Miao**[1], **Hejun Dong**[1], **Siyi Qian**[1,2], **Junyuan Zhang**[1], **Fangdong Wang**[1], **Jingzhou Chen**[1,2], **Xiaomeng Zhao**[1], **Liqun Wei**[1], **Wei Li**[1,4], **Shasha Wang**[1], **Ruiliang Xu**[1], **Yuanyuan Cao**[1], **Lu Chen**[1], **Qianqian Wu**[1], **Huaiyu Gu**[1], **Lindong Lu**[1], **Dechen Lin**[1], **Guanlin Shen**[1], **Xuanhe Zhou**[1,3], **Linfeng Zhang**[3], **Yuhang Zang**[1], **Xiaoyi Dong**[1], **Jiaqi Wang**[1], **Bo Zhang**[1], **Lei Bai**[1], **Pei Chu**[1], **Weijia Li**[1], **Jiang Wu**[1], **Lijun Wu**[1], **Zhenxiang Li**[1], **Guangyu Wang**[1], **Zhongying Tu**[1], **Chao Xu**[1], **Kai Chen**[1], **Bowen Zhou**[1], **Dahua Lin**[1✉], **Wentao Zhang**[1,2✉], **Conghui He**[1✉]

[1]Shanghai Artificial Intelligence Laboratory, [2]Peking University
[3]Shanghai Jiao Tong University , [4]East China Normal University
heconghui@pjlab.org.cn

Abstract

We introduce MinerU2.5, a 1.2B-parameter document parsing vision-language model that achieves state-of-the-art recognition accuracy while maintaining exceptional computational efficiency. Our approach employs a coarse-to-fine, two-stage parsing strategy that decouples global layout analysis from local content recognition. In the first stage, the model performs efficient layout analysis on downsampled images to identify structural elements, circumventing the computational overhead of processing high-resolution inputs. In the second stage, guided by the global layout, it performs targeted content recognition on native-resolution crops extracted from the original image, preserving fine-grained details in dense text, complex formulas, and tables. To support this strategy, we developed a comprehensive data engine that generates diverse, large-scale training corpora for both pretraining and fine-tuning. Ultimately, MinerU2.5 demonstrates strong document parsing ability, achieving state-of-the-art performance on multiple benchmarks, surpassing both general-purpose and domain-specific models across various recognition tasks, while maintaining significantly lower computational overhead.

1 Introduction

Document parsing (Zhang et al., 2024b) serves as a fundamental task in multimodal understanding, underpinning a variety of downstream applications such as information extraction (Liao et al., 2023; Wan et al., 2024), Retrieval-Augmented Generation (RAG) (Lin, 2024; Zhang et al., 2024a; Zhao et al., 2024a) and intelligent document analysis (Bai et al., 2022; Blecher et al., 2023; Tang et al., 2023). In contrast to natural images, document images are characterized by significantly higher resolutions, denser content, and more complex structural layouts (Liu et al., 2024a; Wang et al., 2023; Wei et al., 2024). These inherent properties introduce a unique set of challenges. Firstly, the high resolution and fine-grained layout structures necessitate models capable of processing images at their native resolution. Secondly, the text-dense and often lengthy nature of documents imposes stringent requirements on the parameter efficiency and robustness of the models. Thirdly, the success of OCR is contingent not only on precise text recognition but also heavily on reliable layout analysis and efficient inference.

Contemporary approaches to document parsing can be broadly categorized into two paradigms: pipeline-based approaches (Cui et al., 2025; Livathinos et al., 2025; Paruchuri, 2025; Wang et al., 2024c) and end-to-end approaches based on VLMs (Achiam et al., 2023; Bai et al., 2025; Comanici et al., 2025; rednote, 2025; Wei et al., 2024). The former employs a modular design, decomposing the task into discrete stages such as layout detection, reading order prediction, and recognition of text lines, formulas, and tables. Each stage is handled by a specialized model. While this approach offers interpretability, it suffers from a cumbersome workflow and the potential for error propagation across modules. The latter paradigm exhibits superior semantic modeling capabilities, yet it is still widely constrained by the hallucination

∗ indicates equal contribution. ✉ indicates corresponding author. ‡ indicates Project leader.

problem in long-document processing and suffers from severe efficiency bottlenecks when dealing with high-resolution inputs. A critical factor limiting the performance and efficiency of VLM-based parsing is token redundancy, arising from large blank or low-information regions within the document image.

In response to the aforementioned challenges, we introduce a new document parsing framework, **MinerU2.5**. The key innovation is a decoupled architecture that separates *global layout analysis* from *local content recognition* via an efficient coarse-to-fine, two-stage inference mechanism. In the first stage, the model conducts fast and holistic layout analysis on downsampled document images, capturing the global structural organization with minimal computational cost. In the second stage, guided by the detected layout, it crops key regions from the original high-resolution input and performs fine-grained recognition within local windows, thereby preserving native resolution and ensuring high accuracy. This decoupled strategy not only reduces computational cost by an order of magnitude, primarily by avoiding the enormous number of visual tokens with $\mathcal{O}(N^2)$ complexity inherent in end-to-end native-resolution approaches (Bai et al., 2025; Chen et al., 2025; rednote, 2025), but also brings multiple advantages: it significantly enhances the interpretability of parsing, effectively mitigates the common hallucination problem in VLMs, and allows the two stages to be independently optimized and iterated, resulting in more robust and efficient parsing capabilities. Ultimately, with its lightweight design of only 1.2B parameters, MinerU2.5 exhibits strong adaptability and efficiency in scenarios with long documents and high-density content while ensuring high parsing accuracy. Furthermore, to overcome the challenges of insufficient data diversity, sample imbalance, and inconsistent annotation quality in document parsing, we have developed a closed-loop data engine for complex documents. This engine systematically collects, processes, and generates large-scale, high-quality document corpora. This ensures that our model exhibits precise parsing capabilities and robustness across a wide spectrum of layouts, document types, and complex elements.

MinerU2.5 not only achieves state-of-the-art (SOTA) performance across a wide range of public benchmarks but also represents a qualitative leap in practical application and user experience over the previous MinerU2 version, as demonstrated by the examples in Appendix E.

2 Related Work

2.1 Traditional Pipelines

Early OCR systems (Cui et al., 2025; Livathinos et al., 2025; Paruchuri, 2025; Wang et al., 2024c) decompose document parsing into modular pipelines, sequentially executing layout detection (Wang et al., 2024a; Zhao et al., 2024b), text recognition (Cui et al., 2025), and reading order (Wang et al., 2021). For instance, Marker (Paruchuri, 2025) implements a sequential pipeline integrating Surya OCR (Paruchuri and Team, 2025) with layout analysis and reading order prediction modules to process diverse document types. MinerU (Wang et al., 2024c) leverages PDF-Extract-Kit (OpenDataLab, 2025) to orchestrate multiple specialized models for layout detection, formula recognition and table extraction. This modular architecture enables specialized optimization of individual components and facilitates targeted refinement of specific subtasks through well-defined module boundaries. However, pipeline-based methods are prone to error propagation across stages and exhibit limited robustness when confronted with complex layouts such as multi-column text or cross-page structures. Moreover, modular systems often entail multiple interdependencies in practice, rendering usage, maintenance, and updates cumbersome and less efficient.

2.2 General-Purpose Vision Language Models

General-purpose vision language models (VLMs) (Achiam et al., 2023; Bai et al., 2025; Comanici et al., 2025; Zhu et al., 2025) have emerged as an alternative paradigm for document understanding. Gemini2.5 Pro (Comanici et al., 2025) demonstrates strong OCR capabilities among general VLMs, surpassing traditional pipeline models like MinerU (Wang et al., 2024c) in text parsing and approaching specialized systems like UniMERNet (Wang et al., 2024b) in formula recognition, showcasing the potential of VLMs in OCR applications. Among open-source models, Qwen2.5-VL-72B (Bai et al., 2025) achieves the best results, using native-resolution vision encoders (Dehghani et al., 2023) to adapt to different image sizes, demonstrating the effectiveness of arbitrary-resolution processing in OCR tasks. However, these general models exhibit inherent limitations for document-centric tasks. Proprietary models like Gemini2.5 Pro (Co-

manici et al., 2025) are expensive and slow in processing, while open-source models require massive parameter scales for optimal performance, limiting practical deployment. Additionally, both types remain susceptible to hallucinations in densely populated text regions, affecting reliability in complex document layouts.

2.3 Domain-Specific Vision Language Models

End-to-End Approaches. Recent domain-specific models (Blecher et al., 2023; Chen et al., 2025; Kim et al., 2022; Liu et al., 2024b; Poznanski et al., 2025; rednote, 2025; Wei et al., 2024) adopt end-to-end architectures that unify document parsing within a single model, eliminating the need for cascaded processing stages. GOT (Wei et al., 2024), as an early representative of end-to-end approaches, pioneered the OCR 2.0 paradigm by establishing both model architecture and data methodology that unified recognition across diverse modalities—text, formulas, tables, and charts—within a single framework. Subsequent models like Ocean-OCR (Chen et al., 2025), olmOCR (Poznanski et al., 2025), and dots.ocr (rednote, 2025) leverage native resolution vision encoders to process documents and construct massive document corpora, further advancing the performance of end-to-end architectures. However, end-to-end designs face scalability challenges: joint optimization of layout and content often reduces accuracy on complex documents, while native-resolution processing introduces prohibitive $\mathcal{O}(N^2)$ complexity. Despite strengths in semantic modeling, these models suffer from hallucinations on long documents and severe inefficiency with high-resolution inputs, where token redundancy from blank or low-information regions becomes a major bottleneck.

Multi-Stage Approaches. Recently, multi-stage methods (Feng et al., 2025; Li et al., 2025) leveraging VLMs decouple layout analysis from content recognition, combining the efficiency of pipeline approaches with the accuracy of unified models. Dolphin (Feng et al., 2025) employs a Swin-Transformer VLM that first performs page-level layout, then conducts efficient parallel parsing of identified regions. However, Swin-Transformer's fixed resolution severely limits crop parsing—sub-regions with extreme aspect ratios suffer from distortion when resized to predetermined dimensions, degrading recognition quality while increasing computational overhead. MonkeyOCR (Li et al., 2025) adopts a similar multi-stage strategy but employs a native resolution vision encoder in its second stage, improving both performance and efficiency. However, MonkeyOCR requires multiple specialized models across different stages, increasing system complexity and deployment overhead. A single unified model with native resolution parsing presents a promising direction to address these limitations, which is precisely the goal that MinerU2.5 pursues.

3 MinerU2.5

3.1 Model Architecture

As illustrate in Figure 1, MinerU2.5 follows the Qwen2-VL framework (Wang et al., 2024d), using Qwen2-Instruct-0.5B (Team, 2024) as the decoder with M-RoPE (Wang et al., 2024d) (replacing 1D-RoPE (Su et al., 2024)) for better generalization across crop resolutions and aspect ratios, a 675M NaViT (Dehghani et al., 2023) vision encoder initialized from Qwen2-VL with dynamic-resolution 2D-RoPE encoding (as window attention in Qwen2.5-VL (Bai et al., 2025) degrades parsing quality), and a pixel-unshuffle based patch merger (Shi et al., 2016) that merges adjacent 2×2 vision tokens to reduce redundancy while balancing efficiency and accuracy.

3.2 Two-Stage Parsing Strategy

High-resolution documents contain large blank regions, causing severe visual token redundancy. Crop-based methods (Wei et al., 2024; Zhu et al., 2025) reduce cost but lose layout consistency, while native-resolution encoding (Bai et al., 2025; Guo et al., 2025; rednote, 2025; Niu et al., 2025) preserves details but is computationally impractical. We propose a two-stage strategy to decouple layout analysis and local recognition, improving interpretability and reducing hallucinations.

Stage I: Layout Analysis. We resize each page to a fixed 1036×1036 thumbnail for global layout detection. This size trades off visibility and efficiency and yields stable localization compared to native-aspect thumbnails.

Stage II: Content Recognition. We crop regions from the original high-resolution image according to Stage I results and parse them at native resolution, capped by $2048 \times 28 \times 28$ pixels to avoid both detail loss and redundant computation.

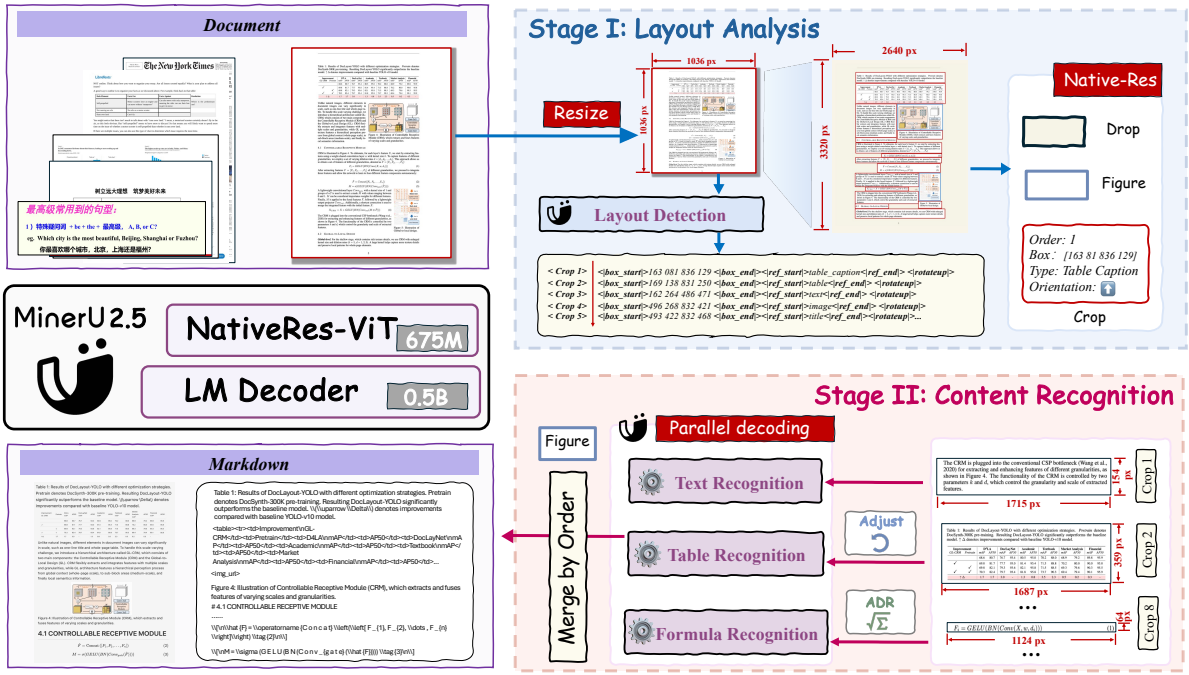

Figure 1: **The framework of MinerU2.5.** In stage I, MinerU2.5 performs rapid, global layout analysis on a downsampled page. In stage II, MinerU2.5 leverages the layout results to crop key regions from the original high-resolution document, performing fine-grained content recognition (e.g., text, table, and formula recognition) within these native-resolution local regions. The detailed prompts used in the inference are illustrated in Appendix D.

3.3 Training Recipe

MinerU2.5 initializes the vision encoder from Qwen2-VL-2B-Instruct and the LLM from Qwen2-Instruct-0.5B. Training consists of three stages.

3.3.1 Stage 0: Modality Alignment

Stage 0 aligns vision and language representations by first training only the two-layer patch-merger MLP on *LLaVA-Pretrain* with the vision encoder and LLM frozen, followed by full-parameter visual instruction tuning on *LLaVA-Instruct* to improve instruction-following and OCR capability; removing this stage leads to higher loss and degraded parsing performance.

3.3.2 Stage 1: Document Parsing Pre-training

Stage 1 jointly learns layout analysis and content recognition with all parameters trainable. Layout samples use resized full-page images with relative coordinates and the prompt "`Layout Detection:`". Content recognition uses cropped text/formula/table blocks with prompts "`Text/Formula/Table Recognition:`".

We train for 2 epochs, each with 6.9M samples: 2.3M layout, 2.4M text, 1.1M formula, and 1.1M table. The resulting model serves as a strong baseline and an efficient hard-sample miner for downstream annotation.

3.3.3 Stage 2: Document Parsing Fine-tuning

Stage 2 improves difficult cases while preserving core abilities. We mix diverse high-quality samples from Stage 1 with manually annotated hard cases mined from large-scale PDF corpora. We fine-tune for 3 epochs, each with 630K samples: 43K layout, 300K text, 147K formula, and 140K table.

3.4 Model Deployment

We build an offline pipeline based on vLLM (Kwon et al., 2023) with two optimizations: (1) asynchronous batching to overlap CPU/GPU workloads, and (2) decoupled Stage I/II inference to enable early downstream processing. To suppress degenerate repetition without harming structured outputs, we dynamically adjust `frequency_penalty` and `presence_penalty` based on detected layout types (e.g., higher for paragraphs, lower for tables).

We tune **max_num_batched_tokens**, **max_num_seqs**, and **cuda_graph_sizes** for higher utilization. On OmniDocBench (Ouyang et al., 2025) (1,355 pages), MinerU2.5 achieves 2.12 pages/s and 2337.25 tokens/s, outperforming MonkeyOCR-Pro-3B by $4\times$ and dots.ocr by $7\times$. Even without deployment optimizations, it reaches

0.95 pages/s and 1045.14 tokens/s.

4 Evaluation

In this section, we present a comprehensive quantitative evaluation of MinerU2.5 to demonstrate its effectiveness in document parsing tasks. Specifically, we compare MinerU2.5 against leading general-purpose VLMs including GPT-4o (Achiam et al., 2023), Gemini-2.5 Pro (Comanici et al., 2025), and Qwen2.5-VL (Bai et al., 2025), as well as state-of-the-art domain-specific VLMs such as dots.ocr (rednote, 2025), MonkeyOCR (Li et al., 2025), and olmOCR (Poznanski et al., 2025).

4.1 Full-Document Parsing Task

We evaluate MinerU2.5's full-document parsing performance on OmniDocBench (Ouyang et al., 2025), which provides a comprehensive evaluation across diverse document types, quality conditions, and parsing challenges.

4.1.1 Evaluation Details and Metrics

Following OmniDocBench (Ouyang et al., 2025), we use its latest version with three key updates: (1) increased resolution for Notes and Newspapers from 72 to 200 DPI; (2) added 374 pages to better balance Chinese-English content and enrich formula coverage (1,355 pages in total); and (3) adopted a hybrid matching-based evaluation protocol.

4.1.2 Evaluation Results

As shown in Table 1, MinerU2.5 achieves an overall score of 90.67 on OmniDocBench, outperforming the second-best model MonkeyOCR-pro-3B (Li et al., 2025) by 1.82 and dots.ocr (rednote, 2025) by 2.26 points. In text recognition tasks, MinerU2.5 achieves the lowest edit distance of 0.047, marginally better than dots.ocr at 0.048 and significantly outperforming PP-StructureV3 (Cui et al., 2025), which scores 0.073. For formula recognition, MinerU2.5 leads with a CDM score of 88.46, exceeding both Qwen2.5-VL-72B at 88.27 and MonkeyOCR-3B at 87.45. In table recognition tasks, MinerU2.5 achieves the highest TEDS score of 88.22 and TEDS-S score of 92.38. For reading order evaluation, it maintains the best edit distance of 0.044. The document-type specific results presented in Table 7 demonstrate that MinerU2.5 achieves best or second-best performance in 6 out of 9 categories. For textbooks, it delivers the best performance with an edit distance of 0.0499, substantially outperforming dots.ocr's 0.0788. For newspapers, MinerU2.5 leads with a score of 0.0540, surpassing all competing models. In both financial reports and slides categories, MinerU2.5 achieves second-best performance with scores of 0.0104 and 0.0294 respectively.

4.2 Element-Specific Parsing Task

4.2.1 Layout Analysis

We validate the effectiveness of our layout analysis via a fair zero-shot comparison on three public datasets: OmniDocBench (Ouyang et al., 2025), a recent document parsing benchmark with detailed layout annotations; D^4LA (Da et al., 2023), which contains 11,092 noisy document images annotated with 27 categories (we use its annotated test set); and DocLayNet (Pfitzmann et al., 2022), a large-scale dataset of 80,863 pages spanning seven document types with 11 categories (we use its annotated validation set).

We compare MinerU2.5 against recent baselines, including LayoutLMv3 (Huang et al., 2022), MinerU2-VLM (Wang et al., 2024c), DocLayout-YOLO (Zhao et al., 2024b), and PP-StructureV3 (Cui et al., 2025). To ensure an equitable evaluation, all models are tested without dataset-specific training. To address differences in label definitions and detection granularity, we map annotations into five coarse categories and adopt PageIoU for evaluation; the "Full Page" score measures spatial overlap regardless of category labels.

As shown in Table 2, MinerU2.5 consistently achieves the best Full Page F1-score@PageIoU across all benchmarks and leads on most individual element types, demonstrating strong generalization under zero-shot settings.

4.2.2 Table Recognition

We evaluate MinerU2.5 against representative baselines, including traditional table recognition systems, general multimodal large models, and document parsing models, on five benchmarks (Table 3). The benchmarks include PubTabNet (Zhong et al., 2020) , FinTabNet (Zheng et al., 2021) , CC-OCR (Yang et al., 2024) and OCRBench v2 (Fu et al., 2024) , and an in-house table recognition benchmark with ∼500 diverse tables spanning various layouts and attributes.

Overall, MinerU2.5 achieves state-of-the-art or competitive performance across all benchmarks. On PubTabNet, although trained with only 20% of the training set, MinerU2.5 remains comparable

Model Type	Methods	Parameters	Overall↑	Text^{Edit}↓	Formula^{CDM}↑	Table^{TEDS}↑	Table^{TEDS-S}↑	Read Order^{Edit}↓
Pipeline Tools	Marker-1.8.2 (Paruchuri, 2025)	-	71.30	0.206	76.66	57.88	71.17	0.250
	MinerU2-pipeline (Wang et al., 2024c)	-	75.51	0.209	76.55	70.90	79.11	0.225
	PP-StructureV3 (Cui et al., 2025)	-	86.73	0.073	85.79	81.68	89.48	0.073
General VLMs	GPT-4o (Achiam et al., 2023)	-	75.02	0.217	79.70	67.07	76.09	0.148
	InternVL3-76B (Zhu et al., 2025)	76B	80.33	0.131	83.42	70.64	77.74	0.113
	InternVL3.5-241B (Wang et al., 2025b)	241B	82.67	0.142	87.23	75.00	81.28	0.125
	Qwen2.5-VL-72B (Bai et al., 2025)	72B	87.02	0.094	88.27	82.15	86.22	0.102
	Gemini-2.5 Pro (Comanici et al., 2025)	-	88.03	0.075	85.82	85.71	90.29	0.097
Specialized VLMs	Dolphin (Feng et al., 2025)	322M	74.67	0.125	67.85	68.70	77.77	0.124
	OCRFlux (chatdoc com, 2025)	3B	74.82	0.193	68.03	75.75	80.23	0.202
	Mistral-OCR (Team, 2025)	-	78.83	0.164	82.84	70.03	78.04	0.144
	POINTS-Reader (Liu et al., 2025b)	3B	80.98	0.134	79.20	77.13	81.66	0.145
	olmOCR-7B (Poznanski et al., 2025)	7B	81.79	0.096	86.04	68.92	74.77	0.121
	MinerU2-VLM(Wang et al., 2024c)	0.9B	85.56	0.078	80.95	83.54	87.66	0.086
	Nanonets-OCR-s (Mandalm, 2025)	3.7B	85.59	0.093	85.90	80.14	85.57	0.108
	MonkeyOCR-pro-1.2B (Li et al., 2025)	1.9B	86.96	0.084	85.02	84.24	89.02	0.130
	MonkeyOCR-3B (Li et al., 2025)	3.7B	87.13	0.075	87.45	81.39	85.92	0.129
	dots.ocr (rednote, 2025)	3B	88.41	0.048	83.22	86.78	90.62	0.053
	MonkeyOCR-pro-3B (Li et al., 2025)	3.7B	88.85	0.075	87.25	86.78	90.63	0.128
	MinerU2.5	1.2B	**90.67**	**0.047**	**88.46**	**88.22**	**92.38**	**0.044**

Table 1: Performance comparison of document parsing methods on OmniDocBench across text, formula, table, and reading order extraction tasks.

Method	Textual			Image			Table			Equation			Page Margins			Full Page		
	P↑	R↑	F1↑	P↑	R↑	F1↑	P↑	R↑	F1↑	P↑	R↑	F1↑	P↑	R↑	F1↑	P↑	R↑	F1↑
OmniDocBench (Ouyang et al., 2025)																		
LayoutLMv3 (Huang et al., 2022)	90.4	48.2	58.1	72.1	51.2	57.2	72.6	55.1	61.0	-	36.9	-	-	-	-	-	-	-
MinerU2-VLM (Wang et al., 2024c)	90.3	95.6	91.9	87.2	91.0	90.9	96.0	97.1	97.8	87.4	95.8	90.5	-	-	-	-	-	-
DocLayout-YOLO (Zhao et al., 2024b)	95.4	**98.3**	96.5	87.6	**96.7**	94.7	94.9	98.1	98.4	95.3	90.6	93.8	-	**98.7**	-	92.3	**97.7**	94.1
PP-StructureV3 (Cui et al., 2025)	96.8	96.7	96.6	86.4	92.1	92.9	**96.6**	97.4	98.2	**96.5**	97.6	**96.7**	**92.9**	86.2	88.1	94.8	96.2	94.6
MinerU2.5	**97.2**	98.0	**97.5**	**89.6**	94.3	**95.0**	96.0	**98.1**	**98.4**	92.4	**99.6**	94.7	89.9	95.4	**91.4**	**95.8**	97.0	**95.9**
D^4LA (Da et al., 2023)																		
LayoutLMv3 (Huang et al., 2022)	86.9	41.2	52.4	**59.3**	32.0	31.4	59.3	41.8	43.3	-	50.5	-	-	-	-	-	-	-
MinerU2-VLM (Wang et al., 2024c)	88.3	88.9	87.9	56.7	35.0	38.1	89.1	84.1	90.6	38.3	99.4	79.1	-	-	-	-	-	-
DocLayout-YOLO (Zhao et al., 2024b)	86.3	97.8	90.8	41.5	92.9	62.6	87.6	**89.0**	89.8	31.9	80.2	**91.1**	-	95.0	-	82.6	**95.4**	87.3
PP-StructureV3 (Cui et al., 2025)	88.5	93.5	90.0	50.1	82.3	67.9	87.1	81.1	89.7	24.6	85.9	92.1	**76.8**	84.2	79.1	85.7	91.0	86.0
MinerU2.5	**91.8**	**98.3**	**94.6**	53.8	**94.3**	**72.8**	**91.9**	78.9	**91.4**	**46.0**	**100.0**	91.0	75.9	**97.6**	**84.2**	**90.4**	92.5	**90.2**
DocLaynet (Pfitzmann et al., 2022)																		
LayoutLMv3 (Huang et al., 2022)	88.8	59.3	67.9	79.0	50.3	61.9	75.2	54.9	61.8	-	31.9	-	-	-	-	-	-	-
MinerU2-VLM (Wang et al., 2024c)	88.1	96.1	91.7	85.5	78.1	91.3	94.9	94.4	95.6	83.9	97.0	90.0	-	-	-	-	-	-
DocLayout-YOLO (Zhao et al., 2024b)	86.9	96.8	91.2	85.8	96.2	91.3	92.0	**95.7**	94.8	80.5	86.9	82.8	-	97.7	-	88.0	96.3	90.9
PP-StructureV3 (Cui et al., 2025)	**90.9**	97.3	93.8	91.7	90.4	94.2	**96.4**	93.7	96.7	88.8	96.0	92.1	**76.8**	79.3	**77.4**	92.4	95.7	93.0
MinerU2.5	90.2	**99.6**	**94.8**	**92.5**	**96.3**	**95.9**	96.3	93.5	**97.1**	**88.9**	**98.6**	**93.5**	76.3	**98.9**	**86.3**	**92.8**	**97.7**	**94.6**

Table 2: Comparison of layout analysis performance (Precision@PageIoU, Recall@PageIoU, F1-score@PageIoU) across different methods and content types on multiple layout analysis benchmarks.

to the best-performing methods. On FinTabNet, MinerU2.5 achieves the best result with a clear margin, benefiting from large-scale high-quality financial table data. On CC-OCR and OCRBench v2, MinerU2.5 is competitive with strong multimodal baselines (e.g., Gemini-2.5 Pro and Qwen2.5-VL-72B), and significantly outperforms most other methods. On the in-house benchmark, MinerU2.5 achieves the best performance, slightly surpassing Gemini-2.5 Pro.

4.2.3 Formula Recognition

We compare MinerU2.5 with a diverse set of baselines, including specialized formula recognizers, document parsing models, and general vision-language models. We evaluate on UniMER-Test (Wang et al., 2024b) (CPE/HWE/SPE/SCE), LaTeX-80M^M), and an in-house benchmark comprising *Chinese* (formulas with Chinese characters), *Fuzzy Math* (low-quality textbook/exam scans with blur, degradation, and watermarks), and *Complex* (highly intricate formulas).

Results in Table 4 are evaluated using CDM (Wang et al., 2025a). MinerU2.5 achieves the best performance on four datasets and ranks second on one. It attains the highest CDM on SCE (96.4) and LaTeX-80M^M (90.6), demonstrating strong robustness to blurred screenshots and complex matrix structures. While slightly behind specialized for-

Method	PubTabNet		FinTabNet		CC-OCR		OCRBench v2		In-house TR Benchmark	
	TEDS↑	TEDS-S↑	TEDS↑	TEDS-S↑	TEDS↑	TEDS-S↑	TEDS↑	TEDS-S↑	TEDS↑	TEDS-S↑
RapidTable (RapidAI, 2024)	86.57	**96.43**	73.77	84.84	50.93	65.84	65.55	77.73	51.96	71.94
MiniCPM-V 4.5 (Yu et al., 2025)	80.30	87.67	85.41	89.18	68.49	77.55	80.28	85.65	55.47	69.61
InternVL3.5-241B (Wang et al., 2025b)	83.75	88.76	84.74	87.92	62.87	69.52	79.5	85.81	56.32	69.3
Qwen2.5-VL-7B (Bai et al., 2025)	81.60	86.78	82.58	87.46	78.29	84.26	77.44	84.71	57.34	73.17
Qwen2.5-VL-72B (Bai et al., 2025)	84.39	87.91	82.90	87.13	81.22	86.48	81.33	86.58	62.79	76.91
GPT-4o (Achiam et al., 2023)	76.53	86.16	83.94	87.00	66.98	79.04	70.51	79.55	46.99	70.29
Gemini-2.5 Pro (Comanici et al., 2025)	-	-	-	-	**85.56**	**90.07**	**88.94**	89.47	69.72	81.29
dots.ocr (rednote, 2025)	**90.65**	93.76	84.12	87.86	75.42	81.65	82.04	86.27	66.91	79.27
Nanonets-OCR-s (Mandalm, 2025)	63.58	75.68	68.06	73.6	66.15	71.33	69.66	76.28	54.35	66.12
MinerU2-VLM (Wang et al., 2024c)	88.11	90.85	78.49	83.03	64.61	71.8	73.22	78.24	63.54	76.66
MinerU2.5	89.07	93.11	**95.97**	**97.61**	79.76	85.16	87.13	**90.62**	**71.48**	**82.83**

Table 3: Table Recognition Performance. MinerU2.5 achieves SOTA performance on most benchmarks among TEDS and TEDS-S metrics, and the remaining ones are also generally competitive with the SOTA. (CCOCR and OCRBench v2 are OCR evaluation benchmarks, we only select the subsets that contain tables. PubTabNet and FinTabNet have a large number of images, so we have not evaluate Gemini-2.5 Pro on them.).

Method	Public Dataset					In-house Dataset		
	CPE	HWE	SCE	SPE	LaTeX-80M^M	Chinese	Fuzzy Math	Complex
UniMERNet* (Wang et al., 2024b)	**98.2**	**96.5**	95.4	**99.2**	83.9	84.0	84.3	67.9
PP-Formula_plus-L (Liu et al., 2025a)	98.2	94.7	95.7	99.2	85.9	84.0	86.5	76.5
Gemini-2.5-flash (Comanici et al., 2025)	89.2	90.0	85.1	97.5	78.7	88.1	89.4	80.1
Qwen2.5-VL-72B (Bai et al., 2025)	88.9	91.8	95.5	96.2	83.4	**90.8**	86.7	81.4
GPT-4o (Achiam et al., 2023)	82.7	85.9	87.8	96.7	73.4	88.3	85.0	78.6
InternVL3.5-241B (Wang et al., 2025b)	91.7	93.2	95.1	97.8	86.9	82.7	90.3	82.0
dots.ocr (rednote, 2025)	86.8	90.5	94.7	97.5	81.8	74.4	86.2	77.4
MinerU2.5	96.6	94.4	**96.4**	98.4	**90.6**	90.7	**92.6**	**82.2**

Table 4: Formula Recognition Performance (CDM metric used for evaluation). MinerU2.5 achieves 4 SOTA results and one second-best result across 7 benchmarks. Latex-80M^M denotes the matrix benchmark of Latex-80M dataset. * indicates that the UniMERNet results are based on an improved version compared to the publicly available open-source implementation.

mula recognizers on CPE/HWE/SPE, MinerU2.5 remains competitive. On our in-house benchmark, it matches Qwen2.5-VL-72B on *Chinese* (90.6) and achieves the best results on both *Fuzzy Math* and *Complex*.

5 Conclusion

In this paper, we present MinerU2.5, a 1.2B-parameter vision-language model for efficient document parsing. By adopting a decoupled coarse-to-fine strategy that separates global layout analysis from local recognition, MinerU2.5 achieves state-of-the-art performance with substantially reduced computation cost. Beyond standalone parsing, MinerU2.5 serves as a practical foundation for the LLM era, enabling rapid conversion of large-scale unstructured documents into structured, high-quality data. By better preserving tables, formulas, and layout semantics, it can further improve document understanding and enhance the reliability of retrieval-augmented generation systems.

Limitations

The proposed two-stage paradigm is sensitive to errors in the first-stage layout analysis, where misclassified or missing regions can propagate and cause unrecoverable failures. This problem is exacerbated by mismatches between training and real-world data distributions, especially under complex and noisy conditions. Therefore, improving the diversity and coverage of the training data—through augmentation, synthetic data, or domain adaptation—is crucial for enhancing the robustness of the first stage and the overall pipeline.

Acknowledgements

This project was supported by Shanghai Artifcial Intelligence Laboratory.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Haoli Bai, Zhiguang Liu, Xiaojun Meng, Wentao Li, Shuang Liu, Nian Xie, Rongfu Zheng, Liangwei Wang, Lu Hou, Jiansheng Wei, and 1 others. 2022. Wukong-reader: Multi-modal pre-training for fine-grained visual document understanding. *arXiv preprint arXiv:2212.09621*.

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

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

chatdoc com. 2025. Ocrflux. https://github.com/chatdoc-com/OCRFlux. Accessed:2025-09-25.

Song Chen, Xinyu Guo, Yadong Li, Tao Zhang, Mingan Lin, Dongdong Kuang, Youwei Zhang, Lingfeng Ming, Fengyu Zhang, Yuran Wang, and 1 others. 2025. Ocean-ocr: Towards general ocr application via a vision-language model. *arXiv preprint arXiv:2501.15558*.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, and 1 others. 2025. Paddleocr 3.0 technical report. *arXiv preprint arXiv:2507.05595*.

Cheng Da, Chuwei Luo, Qi Zheng, and Cong Yao. 2023. Vision grid transformer for document layout analysis. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 19462–19472.

Mostafa Dehghani, Basil Mustafa, Josip Djolonga, Jonathan Heek, Matthias Minderer, Mathilde Caron, Andreas Steiner, Joan Puigcerver, Robert Geirhos, Ibrahim M Alabdulmohsin, and 1 others. 2023. Patch n'pack: Navit, a vision transformer for any aspect ratio and resolution. *Advances in Neural Information Processing Systems*, 36:2252–2274.

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

Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, and 1 others. 2024. Ocrbench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. *arXiv preprint arXiv:2501.00321*.

Dong Guo, Faming Wu, Feida Zhu, Fuxing Leng, Guang Shi, Haobin Chen, Haoqi Fan, Jian Wang, Jianyu Jiang, Jiawei Wang, and 1 others. 2025. Seed1. 5-vl technical report. *arXiv preprint arXiv:2505.07062*.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM international conference on multimedia*, pages 4083–4091.

Geewook Kim, Teakgyu Hong, Moonbin Yim, JeongYeon Nam, Jinyoung Park, Jinyeong Yim, Wonseok Hwang, Sangdoo Yun, Dongyoon Han, and Seunghyun Park. 2022. Ocr-free document understanding transformer. In *European Conference on Computer Vision*, pages 498–517. Springer.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*.

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

Haofu Liao, Aruni RoyChowdhury, Weijian Li, Ankan Bansal, Yuting Zhang, Zhuowen Tu, Ravi Kumar Satzoda, R Manmatha, and Vijay Mahadevan. 2023. Doctr: Document transformer for structured information extraction in documents. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 19584–19594.

Demiao Lin. 2024. Revolutionizing retrieval-augmented generation with enhanced pdf structure recognition. *arXiv preprint arXiv:2401.12599*.

Chaohu Liu, Kun Yin, Haoyu Cao, Xinghua Jiang, Xin Li, Yinsong Liu, Deqiang Jiang, Xing Sun, and Linli Xu. 2024a. Hrvda: High-resolution visual document assistant. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15534–15545.

Hongen Liu, Cheng Cui, Yuning Du, Yi Liu, and Gang Pan. 2025a. Pp-formulanet: Bridging accuracy and efficiency in advanced formula recognition. *arXiv preprint arXiv:2503.18382*.

Yuan Liu, Zhongyin Zhao, Le Tian, Haicheng Wang, Xubing Ye, Yangxiu You, Zilin Yu, Chuhan Wu, Xiao

Zhou, Yang Yu, and 1 others. 2025b. Points-reader: Distillation-free adaptation of vision-language models for document conversion. *arXiv preprint arXiv:2509.01215*.

Yuliang Liu, Biao Yang, Qiang Liu, Zhang Li, Zhiyin Ma, Shuo Zhang, and Xiang Bai. 2024b. Textmonkey: An ocr-free large multimodal model for understanding document. *arXiv preprint arXiv:2403.04473*.

Nikolaos Livathinos, Christoph Auer, Maksym Lysak, Ahmed Nassar, Michele Dolfi, Panos Vagenas, Cesar Berrospi Ramis, Matteo Omenetti, Kasper Dinkla, Yusik Kim, and 1 others. 2025. Docling: An efficient open-source toolkit for ai-driven document conversion. *arXiv preprint arXiv:2501.17887*.

Maksym Lysak, Ahmed Nassar, Nikolaos Livathinos, Christoph Auer, and Peter Staar. 2023. Optimized table tokenization for table structure recognition. In *International Conference on Document Analysis and Recognition*, pages 37–50. Springer.

Souvik Mandalm. 2025. Nanonets-ocr-s. https://nanonets.com/research/nanonets-ocr-s/. Accessed:2025-09-25.

Mathpix. 2025. Mathpix. https://mathpix.com/. Accessed:2025-09-25.

Ahmed Nassar, Andres Marafioti, Matteo Omenetti, Maksym Lysak, Nikolaos Livathinos, Christoph Auer, Lucas Morin, Rafael Teixeira de Lima, Yusik Kim, A Said Gurbuz, and 1 others. 2025. Smoldocling: An ultra-compact vision-language model for end-to-end multi-modal document conversion. *arXiv preprint arXiv:2503.11576*.

Junbo Niu, Yuanhong Zheng, Ziyang Miao, Hejun Dong, Chunjiang Ge, Hao Liang, Ma Lu, Bohan Zeng, Qiahao Zheng, Conghui He, and 1 others. 2025. Native visual understanding: Resolving resolution dilemmas in vision-language models. *arXiv preprint arXiv:2506.12776*.

OpenDataLab. 2025. Pdf-extract-kit. https://github.com/opendatalab/PDF-Extract-Kit. Accessed:2025-09-25.

Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, and 1 others. 2025. Omnidocbench: Benchmarking diverse pdf document parsing with comprehensive annotations. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 24838–24848.

Vik Paruchuri. 2025. Marker. https://github.com/datalab-to/marker. Accessed:2025-09-25.

Vikas Paruchuri and Datalab Team. 2025. Surya: A lightweight document ocr and analysis toolkit. https://github.com/VikParuchuri/surya. Accessed:2025-09-25.

Birgit Pfitzmann, Christoph Auer, Michele Dolfi, Ahmed S Nassar, and Peter Staar. 2022. Doclaynet: A large human-annotated dataset for document-layout segmentation. In *Proceedings of the 28th ACM SIGKDD conference on knowledge discovery and data mining*, pages 3743–3751.

Jake Poznanski, Aman Rangapur, Jon Borchardt, Jason Dunkelberger, Regan Huff, Daniel Lin, Christopher Wilhelm, Kyle Lo, and Luca Soldaini. 2025. olmocr: Unlocking trillions of tokens in pdfs with vision language models. *arXiv preprint arXiv:2502.18443*.

RapidAI. 2024. Rapid table. https://github.com/RapidAI/RapidTable. Accessed: 2025-9-25.

rednote. 2025. dots.ocr: Multilingual document layout parsing in a single vision-language model. https://github.com/rednote-hilab/dots.ocr. Accessed:2025-09-25.

Wenzhe Shi, Jose Caballero, Ferenc Huszár, Johannes Totz, Andrew P Aitken, Rob Bishop, Daniel Rueckert, and Zehan Wang. 2016. Real-time single image and video super-resolution using an efficient sub-pixel convolutional neural network. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1874–1883.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. 2024. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063.

Zineng Tang, Ziyi Yang, Guoxin Wang, Yuwei Fang, Yang Liu, Chenguang Zhu, Michael Zeng, Cha Zhang, and Mohit Bansal. 2023. Unifying vision, text, and layout for universal document processing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 19254–19264.

Mistral AI Team. 2025. Mistral-ocr. https://mistral.ai/news/mistral-ocr?utm_source=ai-bot.cn. Accessed:2025-09-25.

Qwen Team. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Jianqiang Wan, Sibo Song, Wenwen Yu, Yuliang Liu, Wenqing Cheng, Fei Huang, Xiang Bai, Cong Yao, and Zhibo Yang. 2024. Omniparser: A unified framework for text spotting key information extraction and table recognition. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 15641–15653.

Ao Wang, Hui Chen, Lihao Liu, Kai Chen, Zijia Lin, Jungong Han, and 1 others. 2024a. Yolov10: Real-time end-to-end object detection. *Advances in Neural Information Processing Systems*, 37:107984–108011.

Bin Wang, Zhuangcheng Gu, Guang Liang, Chao Xu, Bo Zhang, Botian Shi, and Conghui He. 2024b. Unimernet: A universal network for real-world mathematical expression recognition. *arXiv preprint arXiv:2404.15254*.

Bin Wang, Fan Wu, Linke Ouyang, Zhuangcheng Gu, Rui Zhang, Renqiu Xia, Botian Shi, Bo Zhang, and Conghui He. 2025a. Image over text: Transforming formula recognition evaluation with character detection matching. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, pages 19681–19690.

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

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, and 1 others. 2024d. 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, and 1 others. 2025b. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*.

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

Zilong Wang, Yichao Zhou, Wei Wei, Chen-Yu Lee, and Sandeep Tata. 2023. Vrdu: A benchmark for visually-rich document understanding. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 5184–5193.

Haoran Wei, Chenglong Liu, Jinyue Chen, Jia Wang, Lingyu Kong, Yanming Xu, Zheng Ge, Liang Zhao, Jianjian Sun, Yuang Peng, and 1 others. 2024. General ocr theory: Towards ocr-2.0 via a unified end-to-end model. *arXiv preprint arXiv:2409.01704*.

Zhibo Yang, Jun Tang, Zhaohai Li, Pengfei Wang, Jianqiang Wan, Humen Zhong, Xuejing Liu, Mingkun Yang, Peng Wang, Shuai Bai, and 1 others. 2024. Cc-ocr: A comprehensive and challenging ocr benchmark for evaluating large multimodal models in literacy. *arXiv preprint arXiv:2412.02210*.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Haoyu Li, Weilin Zhao, Zhihui He, and 1 others. 2024. Minicpm-v: A gpt-4v level mllm on your phone. *arXiv preprint arXiv:2408.01800*.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Yuanqian Zhao, Bokai Xu, Junbo Cui, Yingjing Xu, Liqing Ruan, Luoyuan Zhang, Hanyu Liu, Jingkun Tang, Hongyuan Liu, Qining Guo, and 15 others. 2025. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. *arXiv preprint arXiv:2509.18154*.

Junyuan Zhang, Qintong Zhang, Bin Wang, Linke Ouyang, Zichen Wen, Ying Li, Ka-Ho Chow, Conghui He, and Wentao Zhang. 2024a. Ocr hinders rag: Evaluating the cascading impact of ocr on retrieval-augmented generation. *arXiv preprint arXiv:2412.02592*.

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

Penghao Zhao, Hailin Zhang, Qinhan Yu, Zhengren Wang, Yunteng Geng, Fangcheng Fu, Ling Yang, Wentao Zhang, Jie Jiang, and Bin Cui. 2024a. Retrieval-augmented generation for ai-generated content: A survey. *arXiv preprint arXiv:2402.19473*.

Zhiyuan Zhao, Hengrui Kang, Bin Wang, and Conghui He. 2024b. Doclayout-yolo: Enhancing document layout analysis through diverse synthetic data and global-to-local adaptive perception. *arXiv preprint arXiv:2410.12628*.

Xinyi Zheng, Douglas Burdick, Lucian Popa, Xu Zhong, and Nancy Xin Ru Wang. 2021. Global table extractor (gte): A framework for joint table identification and cell structure recognition using visual context. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 697–706.

Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. 2020. Image-based table recognition: data, model, and evaluation. In *European conference on computer vision*, pages 564–580. Springer.

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

A Training Details

As described in Section 3.1, MinerU2.5 consists of three core components: vision encoder, patch merger, and language model. Prior to the pre-training phase of MinerU2.5, the vision encoder is initialized from Qwen2-VL-2B-Instruct, while the language model is initialized from Qwen2-Instruct-0.5B. The overall training procedure of MinerU2.5 is divided into three stages, as summarized in Table 5.

A.0.1 Stage 0-Modality Alignment

To ensure that MinerU2.5 acquires the fundamental vision–language alignment ability as well as the OCR recognition capability, we first conduct two-stage modality alignment training on Visual Question Answering (VQA) datasets.

Language-Image Alignment. Only the two-layer MLP within the patch merger is trained, while both the vision encoder and the language model are frozen. We use image-caption pairs[1] for training to effectively project visual features into the LLM embedding space, thus achieving alignment of the modal representation.

Visual Instruction Tuning. All model parameters are unfrozen. The focus is on knowledge accumulation and ability expansion, particularly strengthening visual alignment and OCR capability. The training data[2] mainly covers image captioning, interleaved text-image pairs, visual alignment, and OCR data. The goal is to enable MinerU2.5 to follow instructions across diverse visual tasks and generate reasonable responses.

		Stage-0		**Stage-1**	**Stage-2**
		a	**b**		
Vision	**Max Resolution**	$2048 \times 28 \times 28$	$4096 \times 28 \times 28$	$2048 \times 28 \times 28$	$2048 \times 28 \times 28$
	#Tokens per Image	$4 \sim 2048$	$4 \sim 4096$	$4 \sim 2048$	$4 \sim 2048$
Data	**Dataset**	Image Caption	VQA	Layout&OCR	Layout&OCR
	#Samples	558K	665K	6.9M	630K
Model	**Trainable**	MLP Adaptor	All	All	All
	Sequence Length	4096	4096	8192	16384
	Data Augmentation	No	No	Yes	Yes
Training	**Batch Size**	128	64	256	256
	LR: ψ_{ViT}	1×10^{-3}	1×10^{-5}	4×10^{-6}	4×10^{-6}
	LR: $\{\theta_{\text{MLP}}, \phi_{\text{LM}}\}$	1×10^{-3}	1×10^{-5}	4×10^{-5}	4×10^{-5}
	Epoch	1	1	2	3

Table 5: Training setup and hyperparameters in three training stages.

Empirical results demonstrate that MinerU2.5, after VQA-based modality alignment training, exhibits significant improvements in tasks such as layout analysis and content recognition. Conversely, skipping this stage leads to higher losses and a clear drop in overall performance.

A.0.2 Stage 1-Document Parsing Pre-training

The objective of the document parsing pre-training stage is to enable MinerU2.5 to acquire two fundamental capabilities: **layout analysis** and **content recognition**. At this stage, all parameters of the model remain fully trainable.

[1]This dataset is sourced from *LLaVA-Pretrain*.
[2]This dataset is sourced from *LLaVA-Instruct*.

Training Data. We leveraged a large-scale mixture of model-labeled data and public datasets to ensure both sufficient scale and document diversity. For layout analysis, in consideration of training efficiency, full document images were resized to a fixed resolution with corresponding relative coordinates, and the prompt "`Layout Detection:`" was used. For content recognition, we employed single-element image samples of text blocks, formula blocks, and table blocks as inputs, with prompts "`Text Recognition:`", "`Formula Recognition:`", and "`Table Recognition:`" respectively. More details are shown in the Appendix D.

Training Configuration. The model, initialized from Stage 0, was trained for 2 epochs. Each epoch consisted of a total of 6.9M samples, including 2.3M for layout analysis, 2.4M for text blocks, 1.1M for formula blocks, and 1.1M for table blocks.

Through this document parsing pre-training, the model has acquired strong layout analysis and content recognition capabilities, demonstrating excellent performance across most simple and medium-level scenarios. The resulting model not only serves as a **strong baseline** for downstream fine-tuning, but also functions as an **efficient hard-sample miner** within our data engineering pipeline, facilitating the identification of challenging cases for human annotation and further improving document parsing performance.

A.0.3 Stage 2-Document Parsing Fine-tuning

The objective of the document parsing fine-tuning stage is to further enhance parsing performance in challenging scenarios, while maintaining the detection and parsing capabilities already acquired by MinerU2.5.

Training Data. To achieve this goal, it is crucial to construct a compact yet high-quality dataset:

- To preserve the model's fundamental capabilities, we sampled high-quality and diverse examples from the pre-training dataset via data engineering and incorporated them into Stage 2 training, ensuring broad coverage across different document element types.

- From a large-scale, multi-source PDF corpus, we employed data engineering to identify cases where the model still underperformed. We summarized these difficult scenarios and conducted targeted data collection with manual annotation to obtain high-quality samples representing challenging cases.

Training Configuration. We fine-tuned the pre-trained model for 3 epochs. Each epoch contained a total of 630K samples, consisting of 43K for layout analysis, 300K for text blocks, 147K for formula blocks, and 140K for table blocks.

With this targeted data iteration strategy, Stage 2 fine-tuning enables the model to not only retain its established document parsing abilities but also achieve significant improvements in previously challenging scenarios.

A.0.4 Data Augmentation Strategies

To enhance the model's robustness in handling diverse documents in an open-world setting, we designed a variety of targeted data augmentation strategies during both Stage 1 and Stage 2. These augmentations simulate common types of document interference, and can be categorized as shown in Table 6.

Note that spatial transformations are not applied to layout analysis samples. For different element types, we carefully design augmentation parameters and probabilities in order to strike a balance between model performance and robustness.

B Evaluation Details

This section provides additional experimental details of MinerU2.5, including dataset configurations, evaluation protocols, and implementation settings, to facilitate reproducibility and support the main results.

Augmentation Type	Operations
Spatial Transformations	Scaling, Grid Distortion, Rotation
Background Transformations	Texture, Weather effect, Image background, Watermark, Scanlines, Shadow
Color Transformations	Brightness Contrast, Illumination, RGB Shift
Degradation Transformations	PSF Blur, Vibration Blur, Gaussian Blur, Erosion / Dilation

Table 6: Data augmentation strategies for document parsing.

B.1 Full-Document Parsing Task

We evaluate MinerU2.5's full document parsing performance on three prominent benchmarks: OmniDocBench (Ouyang et al., 2025), Ocean-OCR (Chen et al., 2025) benchmarks, and olmOCR-bench (Poznanski et al., 2025). These benchmarks provide comprehensive evaluation from different dimensions, covering diverse document types, various quality conditions, and different parsing challenges to thoroughly assess the model's robustness and generalization capabilities.

- **OmniDocBench** (Ouyang et al., 2025): This evaluation dataset is designed for diverse document parsing in real-world scenarios, encompassing nine document types, four layout types, and three language types. It offers a comprehensive assessment of parsing scores for text, formulas, tables, and reading order in full-document parsing, as well as for element-specific parsing tasks.
- **olmOCR-bench** (Poznanski et al., 2025): This evaluation dataset comprises 1,402 PDF documents sourced from various repositories, organized into seven subsets. Certain test patterns are applicable across all document types (e.g., presence, absence, reading order), while others are specifically targeted at challenging yet crucial content extraction objectives (e.g., tables, mathematical formulas).
- **Ocean-OCR benchmark** (Chen et al., 2025): This evaluation dataset consists of 100 images from English papers and 100 images from Chinese papers. It primarily evaluates the ability of text parsing and employs several text OCR-related evaluation metrics, such as Normalized Edit Distance, F1 Score, Precision, Recall, BLEU, and METEOR.

B.1.1 Evaluation Details and Metrics

For OmniDocBench (Ouyang et al., 2025), we evaluate on the latest version with three key improvements:

- Enhanced resolution for Notes and Newspapers from 72 to 200 DPI, enabling more accurate evaluation of fine-grained text and handwritten content.
- An addition of 374 pages to balance Chinese-English content distribution and enrich mathematical formula coverage. Currently, it contains a total of 1,355 pages.
- Evaluation methodology updated to hybrid matching algorithm.

The Overall score combines three core metrics:

$$\begin{aligned}\textbf{Overall} = & \frac{(1 - \textbf{Text}^{\textbf{Edit}}) \times 100}{3} \\ & + \frac{\textbf{Table}^{\textbf{TEDS}} + \textbf{Formula}^{\textbf{CDM}}}{3}.\end{aligned} \tag{1}$$

For olmOCR-bench (Poznanski et al., 2025), we replace the formula scores of Arxiv Math (AR) and Old Scans Math (OSM) with the more reliable ExpRate of CDM (Wang et al., 2025a). The original evaluation compares LaTeX formulas by parsing them into abstract syntax trees and matching Unicode tokens, which is overly sensitive to syntax variations (e.g., $\backslash cdots$ vs. $\backslash dotsb$) that render identically but are scored as different. To avoid this bias, we adopt ExpRate, which directly compares rendered outputs, assigning 1 for exact matches and 0 otherwise.

Model Type	Models	Slides	Academic Papers	Book	Textbook	Exam Papers	Magazine	Newspaper	Notes	Financial Report
Pipeline Tools	Marker-1.8.2 (Paruchuri, 2025)	0.1796	0.0412	0.1010	0.2908	0.2958	0.1111	0.2717	0.4656	0.0341
	MinerU2-pipeline (Wang et al., 2024c)	0.4244	0.0230	0.2628	0.1224	0.0822	0.3950	0.0736	0.2603	0.0411
	PP-StructureV3 (Cui et al., 2025)	0.0794	0.0236	0.0415	0.1107	0.0945	0.0722	0.0617	0.1236	0.0181
General VLMs	GPT-4o (Achiam et al., 2023)	0.1019	0.1203	0.1288	0.1599	0.1939	0.1420	0.6254	0.2611	0.3343
	InternVL3-76B (Zhu et al., 2025)	0.0349	0.1052	0.0629	0.0827	0.1007	0.0406	0.5826	0.0924	0.0665
	InternVL3.5-241B (Wang et al., 2025b)	0.0475	0.0857	**0.0237**	0.1061	0.0933	0.0577	0.6403	0.1357	0.1117
	Qwen2.5-VL-72B (Bai et al., 2025)	0.0422	0.0801	0.0586	0.1146	0.0681	0.0964	0.2380	0.1232	0.0264
	Gemini-2.5 Pro (Comanici et al., 2025)	0.0326	0.0182	0.0694	0.1618	0.0937	**0.0161**	0.1347	0.1169	0.0169
Specialized VLMs	Dolphin (Feng et al., 2025)	0.0957	0.0453	0.0616	0.1333	0.1684	0.0702	0.2388	0.2561	0.0186
	OCRFlux (chatdoc com, 2025)	0.0870	0.0867	0.0818	0.1843	0.2072	0.1048	0.7304	0.1567	0.0193
	Mistral-OCR (Team, 2025)	0.0917	0.0531	0.0610	0.1349	0.1341	0.0581	0.5643	0.3097	0.0523
	POINTS-Reader (Liu et al., 2025b)	0.0334	0.0779	0.0671	0.1372	0.1901	0.1343	0.3789	0.0937	0.0951
	olmOCR-7B (Poznanski et al., 2025)	0.0497	0.0365	0.0539	0.1204	0.0728	0.0697	0.2916	0.1220	0.0459
	MinerU2-VLM(Wang et al., 2024c)	0.0745	**0.0104**	0.0357	0.1276	0.0698	0.0652	0.1831	**0.0803**	0.0236
	Nanonets-OCR-s (Mandalm, 2025)	0.0551	0.0578	0.0606	0.0931	0.0834	0.0917	0.1965	0.1606	0.0395
	MonkeyOCR-pro-1.2B (Li et al., 2025)	0.0961	0.0354	0.0530	0.1110	0.0887	0.0494	0.0995	0.1686	0.0198
	MonkeyOCR-3B (Li et al., 2025)	0.0904	0.0362	0.0489	0.1072	0.0745	0.0475	0.0962	0.1165	0.0196
	dots.ocr (rednote, 2025)	**0.0290**	0.0231	0.0433	0.0788	**0.0467**	0.0221	0.0667	0.1116	**0.0076**
	MonkeyOCR-pro-3B (Li et al., 2025)	0.0879	0.0459	0.0517	0.1067	0.0726	0.0482	0.0937	0.1141	0.0211
	MinerU2.5	0.0294	0.0235	0.0332	**0.0499**	0.0681	0.0316	**0.0540**	0.1161	0.0104

Table 7: Document Parsing Performance in Text Edit Distance on OmniDocBench: evaluation using edit distance across 9 PDF page types.

Model	Edit Distance ↓		F1-score ↑		Precision↑		Recall↑		BLEU↑		METEOR↑	
	en	zh	en	zh	en	zh	en	zh	en	zh	en	zh
Mathpix (Mathpix, 2025)	0.064	0.223	0.930	0.919	**0.950**	0.952	0.911	0.889	0.901	0.593	0.924	0.768
PP-StructureV3 (Cui et al., 2025)	0.068	0.210	0.871	0.929	0.856	0.924	0.892	0.935	0.796	0.570	0.902	0.802
MinerU2-pipeline (Wang et al., 2024c)	0.099	0.225	0.663	0.919	0.635	0.908	0.703	0.934	0.504	0.571	0.670	0.810
PaddleOCR (Cui et al., 2025)	0.323	0.649	0.707	0.864	0.690	0.912	0.730	0.821	0.517	0.537	0.674	0.699
Gemini-2.5 Pro (Comanici et al., 2025)	0.080	0.204	0.922	0.927	0.940	0.959	0.906	0.898	0.877	0.690	0.921	0.862
GPT-4o (Achiam et al., 2023)	0.085	0.450	0.919	0.686	0.929	0.694	0.910	0.703	0.870	0.354	0.922	0.495
Qwen2.5-VL-72B (Bai et al., 2025)	0.093	0.140	0.923	0.940	0.936	0.956	0.912	0.926	0.879	0.798	0.924	0.876
InternVL3-76B (Zhu et al., 2025)	0.125	0.282	0.828	0.871	0.842	0.889	0.817	0.856	0.728	0.527	0.829	0.759
Qwen2-VL-7B (Wang et al., 2024d)	0.165	0.270	0.849	0.883	0.834	0.847	0.873	0.942	0.795	0.578	0.859	0.763
MiniCPM-V2.6-8B (Yao et al., 2024)	0.244	0.437	0.804	0.778	0.793	0.721	0.837	0.875	0.695	0.431	0.640	0.642
MinerU2-VLM (Wang et al., 2024c)	0.048	0.182	0.936	0.941	0.926	0.927	0.947	0.958	0.893	0.611	**0.950**	0.837
Ocean-OCR (Chen et al., 2025)	0.057	**0.062**	0.937	0.962	0.932	0.956	**0.956**	**0.974**	0.906	**0.912**	0.945	**0.916**
MonkeyOCR-pro-1.2B (Li et al., 2025)	0.064	0.190	0.929	0.934	0.918	0.925	0.944	0.948	0.884	0.699	0.941	0.850
SmolDocling (Nassar et al., 2025)	0.080	0.878	0.899	0.157	0.895	0.140	0.912	0.268	0.839	0.048	0.907	0.151
dots.ocr (rednote, 2025)	0.083	0.179	0.904	0.931	0.920	0.951	0.890	0.913	0.849	0.639	0.911	0.842
GOT(Wei et al., 2024)	0.084	0.117	0.895	0.928	0.891	0.934	0.906	0.929	0.835	0.805	0.874	0.848
MinerU2.5	**0.033**	0.082	**0.945**	**0.965**	0.948	**0.966**	0.942	0.964	**0.909**	0.817	**0.950**	0.887

Table 8: Evaluation results on Ocean-OCR bench on dense English (en) and Chinese (zh) OCR for document-level pages. Some model results are sourced from the OceanOCR official reports.

B.1.2 Evaluation Results

MinerU2.5 demonstrates exceptional performance across all benchmarks, achieving state-of-the-art results in most metrics (Tables 1 and 7 to 9).

As shown in Table 1, MinerU2.5 achieves an overall score of 90.67 on OmniDocBench, outperforming the second-best model MonkeyOCR-pro-3B (Li et al., 2025) by 1.82 and dots.ocr (rednote, 2025) by 2.26 points. In text recognition tasks, MinerU2.5 achieves the lowest edit distance of 0.047, marginally better than dots.ocr at 0.048 and significantly outperforming PP-StructureV3 (Cui et al., 2025), which scores 0.073. For formula recognition, MinerU2.5 leads with a CDM score of 88.46, exceeding both Qwen2.5-VL-72B at 88.27 and MonkeyOCR-3B at 87.45. In table recognition tasks, MinerU2.5 achieves the highest TEDS score of 88.22 and TEDS-S score of 92.38. For reading order evaluation, it maintains the best edit distance of 0.044. The document-type specific results presented in Table 7 demonstrate that MinerU2.5 achieves best or second-best performance in 6 out of 9 categories. For textbooks, it delivers the best performance with an edit distance of 0.0499, substantially outperforming dots.ocr's 0.0788. For newspapers, MinerU2.5 leads with a score of 0.0540, surpassing all competing models. In both financial reports and slides categories, MinerU2.5 achieves second-best performance with scores of 0.0104 and

Model	Overall	AR	OSM	TA	OS	HF	MC	LTT	Base
MinerU2-pipeline(Wang et al., 2024c)	55.6	61.8	13.5	60.9	17.3	**96.6**	59.0	39.1	96.6
Nanonets-OCR-s(Mandalm, 2025)	60.7	63.9	41.0	77.7	39.5	40.7	69.9	53.4	99.3
GPT-4o(Achiam et al., 2023)	63.2	44.1	37.6	69.1	40.9	94.2	68.9	54.1	96.7
MonkeyOCR-pro-1.2B(Li et al., 2025)	64.3	65.4	26.9	60.3	31.2	93.3	66.2	81.7	89.5
Qwen2.5-VL-72B(Bai et al., 2025)	64.8	72.2	51.1	67.3	38.6	73.6	68.3	49.1	98.3
MonkeyOCR-pro-3B(Li et al., 2025)	68.8	67.7	28.4	74.6	36.1	91.2	76.6	80.1	95.3
olmOCR(Poznanski et al., 2025)	71.8	63.9	41.0	72.9	**43.9**	95.1	77.3	81.2	98.9
dots.ocr(rednote, 2025)	73.6	66.3	35.8	**88.3**	40.9	94.1	**82.4**	81.2	**99.5**
MinerU2.5	**75.2**	**76.6**	**54.6**	84.9	33.7	**96.6**	78.2	**83.5**	93.7

Table 9: Evaluation results on olmOCR-bench grouped by document types, including arXiv Math(AR), Old Scans Math (OSM), Tables (TA), Old Scans (OS), Headers Footers (HF), Multi Column (MC) and Long Tiny Text (LTT). Results on AR and OSM are replaced with ExpRate, and other results are sourced from the official reports of olmOCR-bench and dots.ocr. The Overall Score (Overall) represents the average across all document types.

0.0294 respectively.

For the results of the Ocean-OCR benchmark presented in Table 8, MinerU2.5 demonstrates exceptional performance in dense OCR tasks. On English documents, it achieves the lowest edit distance of 0.033 and the highest F1-score of 0.945, accompanied by best-in-class BLEU and METEOR scores of 0.909 and 0.950 respectively. For Chinese documents, MinerU2.5 achieves the highest F1-score of 0.965 and Precision of 0.966, while maintaining strong BLEU and METEOR scores of 0.817 and 0.887 respectively.

The results of olmOCR-bench are shown in Table 9, where MinerU2.5 achieves an overall score of 75.2, surpassing dots.ocr's 73.6 by 1.6 points. In the arXiv Math category, it leads with a score of 76.6, outperforming Qwen2.5-VL-72B (Bai et al., 2025)'s 72.2 by 4.4 points. For Old Scans Math, MinerU2.5 dominates with a score of 54.6, exceeding all other evaluated models. In the Long Tiny Text category, it achieves 83.5, surpassing MonkeyOCR-pro-1.2B (Li et al., 2025) which scores 81.7.

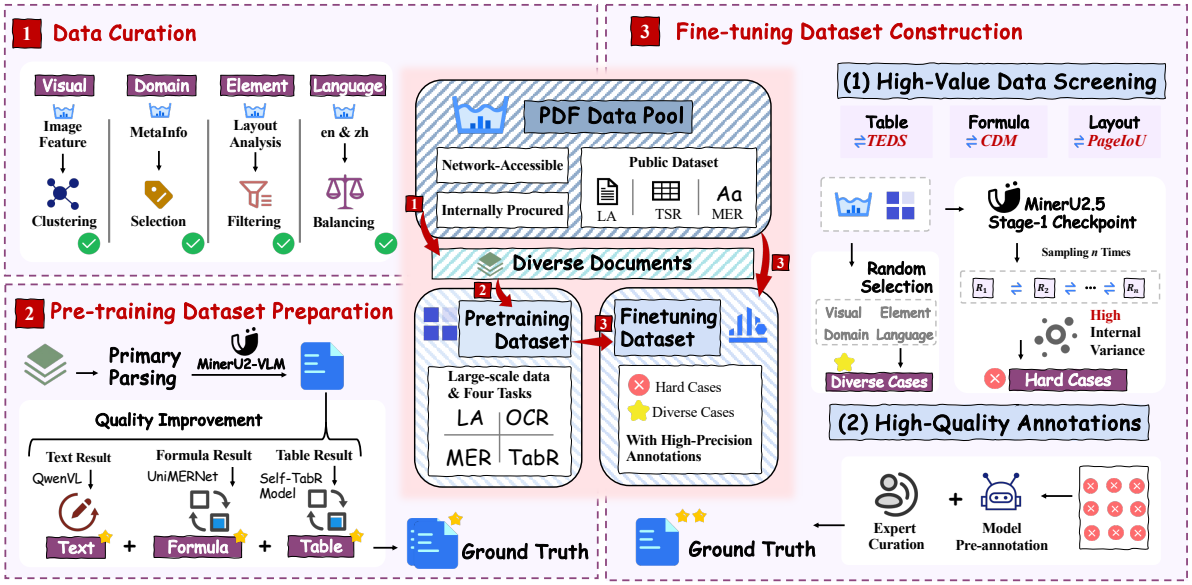

Figure 2: **Overview of the Data Engine.** Our data pipeline consists of three core stages. (1) **Data Curation:** We filter a massive, raw document pool to construct a diverse and balanced dataset based on layout, document type, element balance, and language. (2) **Pre-training Data Preparation:** We generate automated annotations for the curated data and then refine them using specialized, powerful models for text, tables, and formulas to ensure high quality. (3) **Fine-tuning Dataset Construction:** We employ our Iterative Mining via Inference Consistency (IMIC) strategy to automatically discover hard cases, which then undergo meticulous expert curation to create a high-quality SFT dataset.

C Data Engine

The state-of-the-art performance of MinerU2.5 is underpinned by a systematic Data Engine designed to generate large-scale, high-quality training data with uniform annotation standards. This engine first establishes a vast and diverse foundation through rigorous data curation and refined automated annotation for pre-training. Building upon this foundation, we introduce our novel Iterative Mining via Inference Consistency (IMIC) strategy, which efficiently identifies complex “hard cases" for targeted human annotation. This multi-stage approach creates a virtuous cycle of improvement, progressively enhancing the model’s capabilities. The entire process is illustrated in Figure 2.

C.1 Overall Workflow

C.1.1 Data Curation

Our process begins with a large-scale internal document pool comprising publicly available web data and commercially procured documents. While diverse, this raw pool suffers from a significant long-tail distribution. To mitigate this imbalance and enhance training robustness, we implement a rigorous curation process to build a balanced Chinese-English dataset with high diversity across multiple dimensions:

- **Layout Diversity:** We employ page-level image clustering to select exemplars from a wide spectrum of visual layouts and styles.
- **Document Type Diversity:** Using document metadata (e.g., discipline, tags), we perform stratified sampling to ensure a balanced representation of types such as academic papers, textbooks, reports, and presentations.
- **Element Balance:** A preliminary detection model helps ensure a balanced class distribution of key elements like titles, paragraphs, tables, formulas, and figures in the curated set.
- **Language Balance:** We filter the data to maintain a comparable volume of Chinese and English documents.

C.1.2 Pre-training Dataset Preparation

Initial annotations for the curated dataset are generated using our MinerU2-pipeline, establishing a baseline for subsequent refinement. To move beyond this baseline quality, we perform a multi-step refinement process using specialized, expert models for different content types:

- **Textual Content:** We leverage the powerful Qwen2.5-VL-72B-Instruct to verify and correct initial text recognition results on cropped text regions.
- **Formula Content:** Recognized formulas are substituted with higher-fidelity outputs from an in-house UniMERNet model, which we retrained on our extensive formula dataset to boost its accuracy.
- **Table Content:** All table structures are re-generated using an in-house, high-performance table parsing model.

This refinement workflow yields a high-quality pre-training dataset of image-annotation pairs, covering our four core tasks: layout analysis, text recognition, formula recognition, and table recognition.

C.1.3 Fine-tuning Dataset Construction

While pre-training ensures broad capabilities, the noise inherent in automated annotations creates a ceiling for model performance. To break through this ceiling, our fine-tuning strategy pivots to high-value, difficult examples. We designed an Iterative Mining via Inference Consistency (IMIC) strategy to automatically filter these hard cases from the large-scale data pool. To ensure annotation quality, these select samples are processed through an AI-assisted pipeline: they are first pre-annotated by a foundation model, such as Gemini-2.5-Pro for complex tables, and then meticulously reviewed and corrected by human experts[3].

[3]Human review is augmented by our open-source QA tool, Dingo, which applies both rule-based and model-based checks.

The final Supervised Fine-Tuning (SFT) dataset combines these high-quality hard cases with a smaller, randomly sampled set of regular examples, equipping MinerU2.5 to excel in complex, real-world parsing scenarios.

C.2 Task Reformulation and Enhancement

To move beyond the limitations of existing document analysis methods, we systematically reformulated the core tasks of layout analysis, formula recognition, and table recognition. This involved defining more robust standards, designing novel task paradigms, and introducing specialized metrics and representations.

C.2.1 Layout Analysis

A Unified Tagging System. A fundamental challenge in layout analysis is the lack of a standardized tagging system. Existing datasets suffer from widespread inconsistencies in element definitions, granularity, and scope. To address this, we engineered a hierarchical and comprehensive tagging system by analyzing a vast corpus of documents. Our system is defined by three key principles:

- **Comprehensive Coverage:** It includes non-body content often ignored by others, such as headers, footers, and page numbers, which is critical for downstream applications like RAG.
- **Fine Granularity:** It decomposes complex elements. For instance, figures are sub-categorized into image, chart, and chemical_structure, with distinct tags for their associated captions.
- **Semantic Distinction:** Visually distinct text blocks like code, algorithms, references, and lists are assigned their own categories to preserve crucial semantic information.

Table 10 presents a comparison with mainstream tagging systems, highlighting the superior coverage and granularity of our proposed system.

Category	MinerU2-pipeline	PaddleOCR	MinerU2.5
Textual	text	text, toc, abstract	text
	title	title, page_title	title
	×	×	phonetic
	image_caption	common_caption	image_caption
	image_footnote	common_footnote	image_footnote
	table_caption	common_caption	table_caption
	table_footnote	common_footnote	table_footnote
	×	code	code
	×	×	code_caption
	×	×	algorithm
	×	ref_text, ref_block	reference
	×	×	list
Image	image	image, seal, chart, molecular	image
Table	table	table	table
Equation	equation	equation	equation
	×	×	equation_block
Page Margins	×	header	header
	×	footer	footer
	×	aside_text	aside_text
	×	page_number	page_number
	×	page_footnote	page_footnote

Table 10: Comparison of category support across different OCR systems.

An Enhanced Multi-Task Paradigm. Traditional methods often treat layout analysis as a standard object detection task, which ignores element rotation and defers reading order prediction to downstream

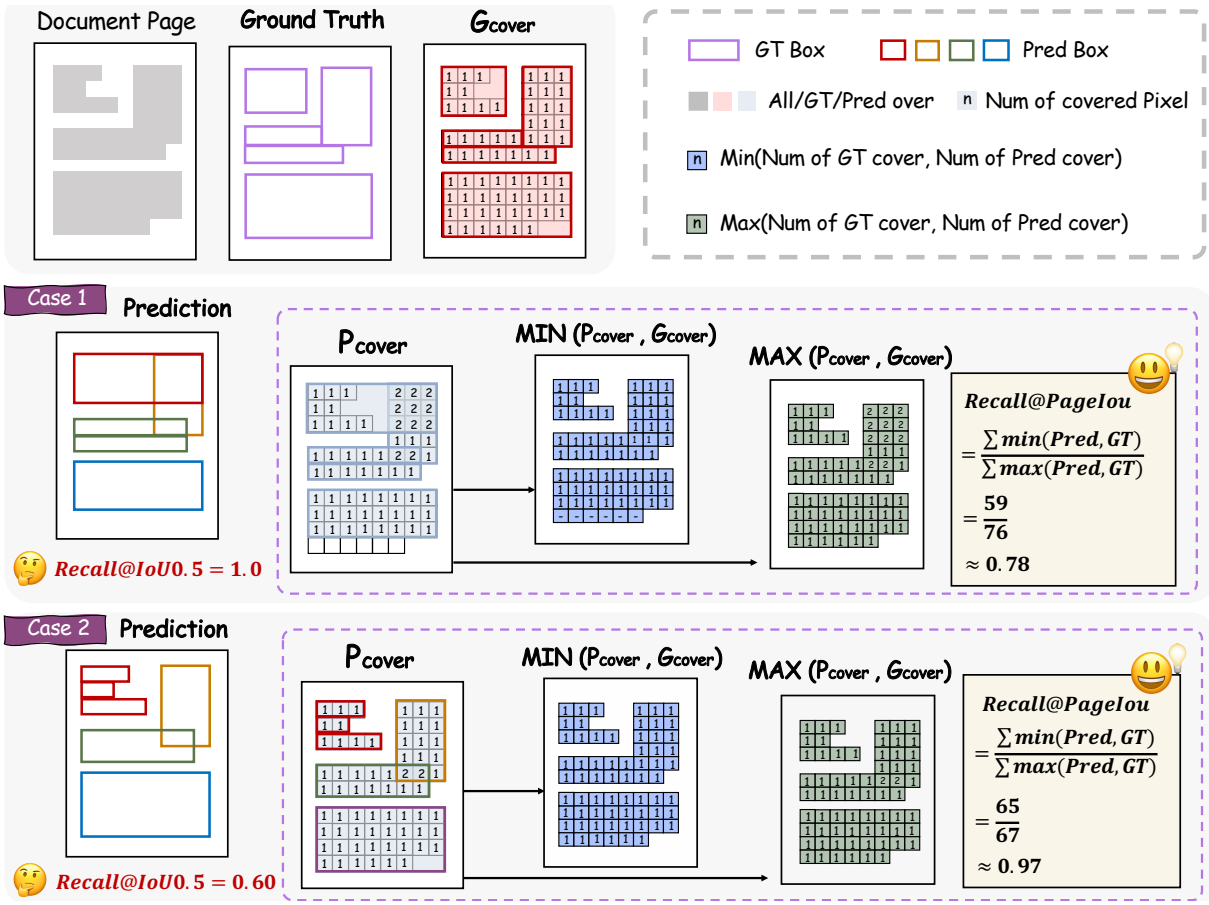

Figure 3: **Illustration of the proposed PageIoU metric.** Case 1 and Case 2 show that IoU-based recall may produce contradictory results compared with visual inspection, whereas PageIoU provides a page-level coverage score that aligns more closely with qualitative observations.

modules. This approach not only impairs the recognition of rotated elements but also increases system coupling. We propose an enhanced paradigm that redefines layout analysis as a multi-task problem. This paradigm simultaneously predicts four key attributes for each document element in a single inference pass: its **Position**, **Class**, **Rotation Angle**, and **Reading Order**. This integrated design effectively resolves the challenge of parsing rotated elements and streamlines the entire document analysis pipeline.

PageIoU: A New Metric for Layout Quality. Layout analysis is typically evaluated with object detection metrics like mAP, which rely on a fixed Intersection over Union (IoU) threshold. While effective for well-defined objects, this approach is ill-suited for document layouts where text block boundaries are often ambiguous. This can lead to a discrepancy where quantitative IoU-based scores do not align with qualitative visual assessment.

As illustrated in Figure 3, a prediction that coarsely covers a paragraph (Case 1) can achieve a perfect recall score (Recall@IoU0.5 = 1.0), while a more accurate line-by-line prediction (Case 2) is penalized for not matching the paragraph-level ground truth, yielding a lower score (Recall@IoU0.5 = 0.6). Visually, however, Case 2 is clearly a better fit.

To better evaluate document layout analysis, we introduce **PageIoU**, a page-level coverage metric that measures the spatial consistency between predicted layouts and ground-truth annotations. Let the predicted layout be

$$P = \{bbox_i \mid i = 1, 2, \ldots, n\},$$

and the ground truth be

$$G = \{bbox_j \mid j = 1, 2, \ldots, m\},$$

where each $bbox$ denotes a bounding box on the page. We first compute coverage maps for both prediction

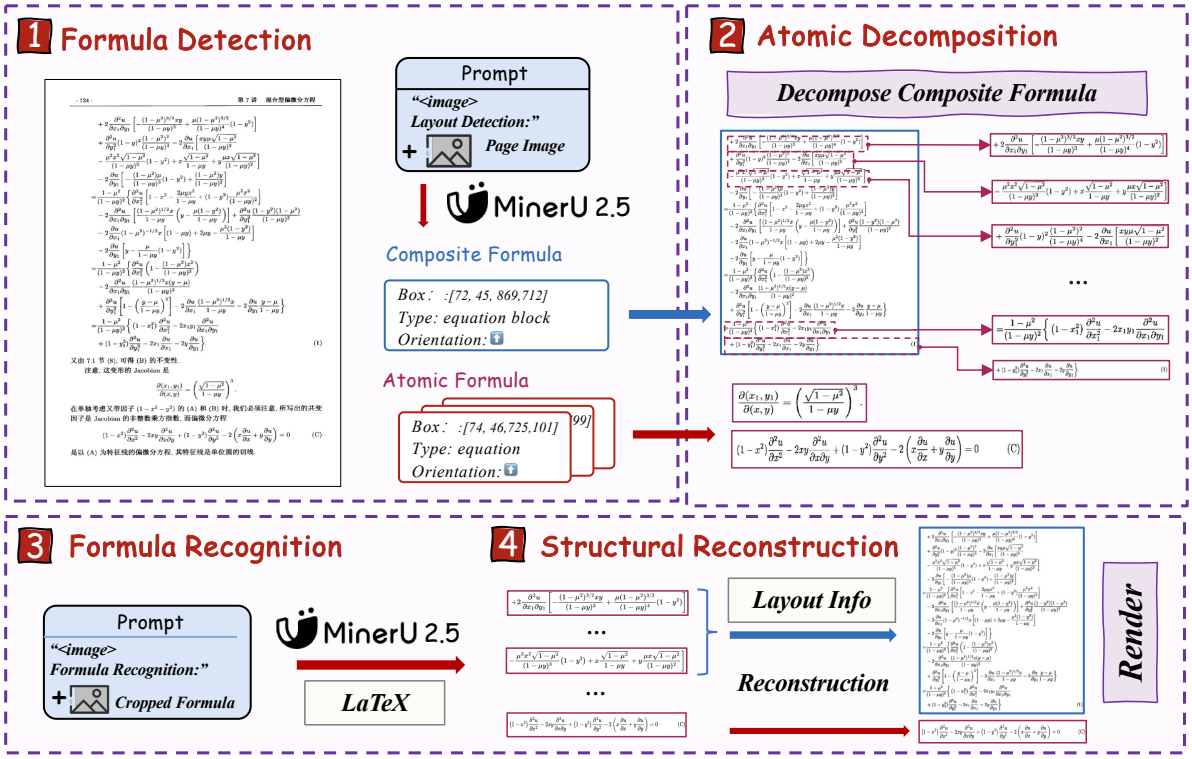

Figure 4: **The proposed ADR framework.** First, a compound formula is decomposed into atomic lines via layout analysis. Next, each line is individually recognized into LaTeX. Finally, the individual results are structurally recombined to produce the complete output.

and ground truth. For example, the ground-truth coverage map is defined as:

$$G_{cover} = \left\{ \sum_{j=1}^{m} 1_{p \in bbox_j} \,\middle|\, p \in M \right\},$$

where p is a page pixel and M denotes the non-background region of the page. Similarly, P_{cover} can be obtained. Based on these, PageIoU is defined as:

$$\begin{aligned} \text{PageIoU}(P, G) &= \frac{|P_{\text{cover}} \cap G_{\text{cover}}|}{|P_{\text{cover}} \cup G_{\text{cover}}|} \\ &= \frac{\sum_{p \in M} \min\{P_{\text{cover}}(p), G_{\text{cover}}(p)\}}{\sum_{p \in M} \max\{P_{\text{cover}}(p), G_{\text{cover}}(p)\}}. \end{aligned} \tag{2}$$

Here, $|\cdot|$ denotes the summation over all pixel values, while $\cap$ and $\cup$ correspond to the pixel-wise minimum and maximum of coverage counts, respectively. As shown in Figure 3, PageIoU aligns with human perception, scoring the qualitatively poor prediction 0.78 and the superior one 0.97.

C.2.2 Formula Recognition

Decoupling Atomic and Compound Formulas. Existing models struggle with long or multi-line formulas, and VLMs are prone to severe structural hallucinations. We identify the root cause as the tendency to treat all formulas as monolithic entities, failing to account for internal complexity. To this end, MinerU2.5 introduces a "whole-part" decoupling philosophy, classifying formulas into two types based on their structural and semantic integrity:

- **Atomic Formulas:** The smallest, indivisible semantic units with a tight 2D topology (e.g., a single fraction, a matrix).

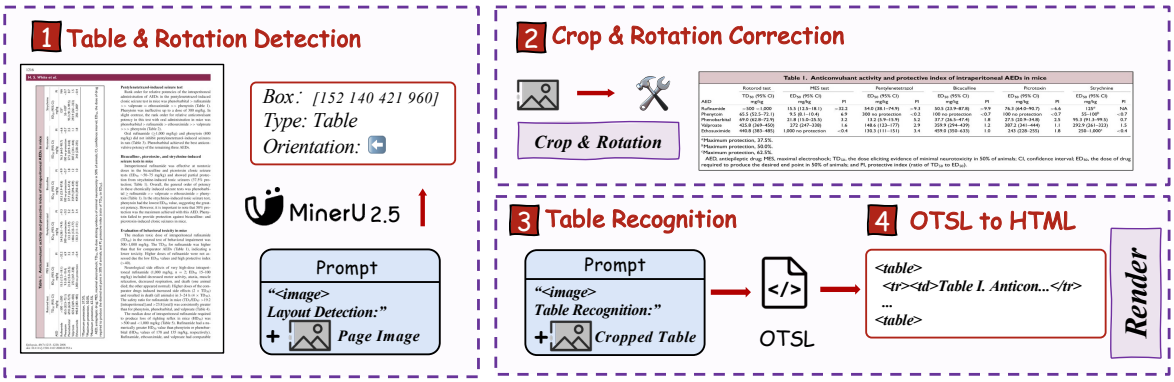

Figure 5: **The Table Recognition Pipeline.** The pipeline first detects a table and its rotation, then corrects its geometry. Next, the rectified image is recognized into the OTSL result, which is finally converted to standard HTML.

- **Compound Formulas:** An ordered set of atomic formulas composed vertically with specific alignment relationships (e.g., a multi-line derivation aligned at the equal signs).

The Atomic Decomposition & Recombination (ADR) Framework. To handle the complexity of compound formulas, we propose the ADR framework, which implements a multi-stage "divide and conquer" strategy. As illustrated in Figure 4, the ADR pipeline is powered by our versatile MinerU2.5 model, which acts as both a layout analyzer and a recognition engine, guided by task-specific prompts. The process begins with an initial layout analysis pass, where MinerU2.5, guided by a layout detection prompt, identifies and classifies all formula regions on the page as either atomic or compound. Next, in the decomposition stage, each identified compound formula is segmented into an ordered sequence of its constituent atomic formula lines, which are then cropped as individual images. In the third stage, these simple, semantically independent atomic formula images are fed back into the MinerU2.5 model. This time, using a formula recognition prompt, the model performs high-precision translation of each image into its corresponding LaTeX string. Finally, a lightweight recombination step uses the positional information from the initial layout pass to structurally reassemble the individual LaTeX strings into a single, coherent block, correctly formatting them within environments like align. This approach transforms a single, difficult recognition task into a series of simpler ones, ensuring both high-fidelity recognition of each component and the logical integrity of the overall structure.

C.2.3 Table Recognition

Overcoming Long-Sequence Dependencies. A primary challenge in table recognition is parsing complex, long tables, especially for VLM-based approaches that target HTML. We attribute this difficulty to two inherent weaknesses of the HTML representation: (1) its complex, non-visual syntax must be learned implicitly by the model; and (2) its high token redundancy results in excessively long sequences, degrading performance on large tables. (The issue of rotated tables is effectively handled by our enhanced layout paradigm.)

OTSL: An Optimized Table Structure Language. To robustly handle complex tables, we propose a four-stage recognition pipeline, as depicted in Figure 5. The first two stages handle geometric normalization: the system detects the table's bounding box and rotation angle, then corrects the image by cropping and rotating it to a canonical orientation. For the crucial third stage, table recognition, we leverage the Optimized Table-Structure Language (OTSL) (Lysak et al., 2023), an intermediate representation developed by IBM [citation, 2023]. We adopted OTSL for its significant advantages over HTML as a target for VLMs. Its minimalist design features a direct structural correspondence to a table's visual 2D matrix, reducing the number of structural tokens from over 28 to just 5 and shortening the average sequence length by approximately 50%. This makes it a far more effective target for model generation. The final stage is a straightforward conversion from the OTSL output into standard HTML.

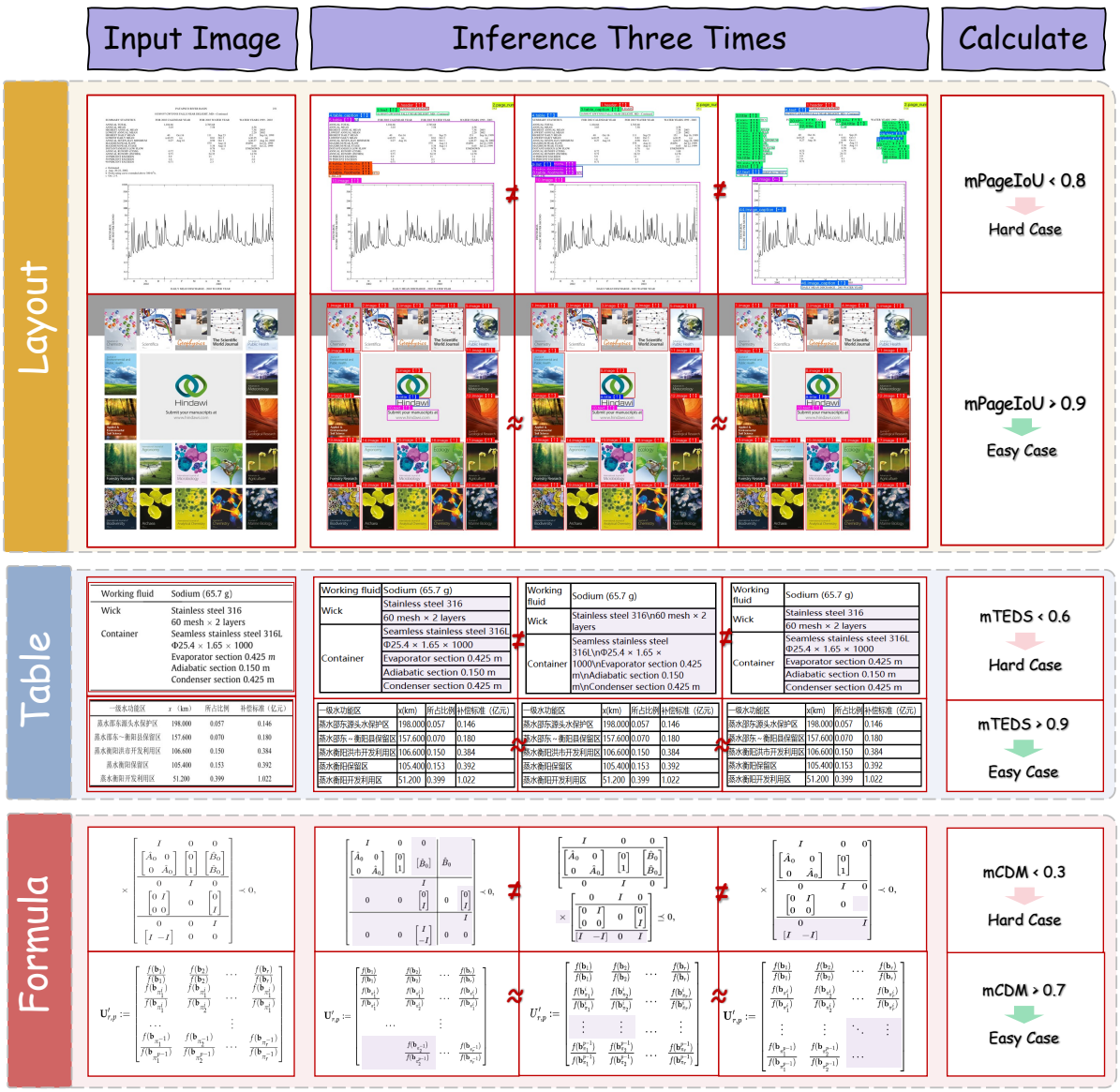

Figure 6: llustration of the proposed **IMIC (Iterative Mining via Inference Consistency) strategy**. From top to bottom: (a) Layout analysis, (b) Table recognition, and (c) Formula recognition. For each task, the model performs multiple stochastic inference runs, and the pairwise consistency between outputs is calculated with task-specific metrics (PageIoU, TEDS, CDM). Samples with low consistency are automatically identified as hard cases and prioritized for manual annotation.

C.3 Iterative Mining via Inference Consistency

To enable continuous model improvement and the efficient expansion of our high-quality training dataset, we introduce the IMIC (Iterative Mining via Inference Consistency) strategy. IMIC automatically identifies the most challenging samples—or "hard cases"—for the current model from a large corpus of unlabeled data. This allows us to direct limited human annotation efforts toward the data that offers the maximum value for model improvement.

The core principle of IMIC leverages the stochasticity inherent in model inference. For a given sample, if the model has learned its features robustly, multiple inference passes with stochastic sampling enabled should yield highly consistent outputs. Conversely, significant divergence across outputs suggests the sample lies near the model's decision boundary—a 'hard case' where its predictions are uncertain. Such samples are the most valuable candidates for manual annotation, as they directly target the model's specific weaknesses.

As illustrated in Figure 6, the implementation is tailored to each recognition task:

- **Layout analysis:** For full document pages, we perform multiple inference runs and measure consistency by calculating the pairwise PageIoU between the resulting layouts. Samples falling below a predefined similarity threshold are flagged as hard cases for precise manual annotation.
- **Formula Recognition:** For cropped formula images, consistency is assessed using the pairwise CDM (Wang et al., 2025a) across multiple outputs. Samples with low consistency are prioritized for manual correction.
- **Table Recognition:** For cropped table images, we use the TEDS (Tree-Edit-Distance-based Similarity) score to evaluate consistency across multiple recognized structures. Low-consistency samples are routed to the manual annotation workflow.

D Prompt Details

Here, we provide a detailed description of the different prompts used during the two-stage inference of MinerU2.5, along with their corresponding output formats.

D.1 Layout Detection

The layout detection output will include the relative coordinates, category, and rotation direction of each element in the document. Each element will be output in sequence, ensuring traceability for all layout data. The input image will be resized to a resolution of 1036×1036.

Output format:

- **Box Coordinates:** x1, y1, x2, y2
- **Document Element Category:** title, text, image, etc.
- **Rotation Direction:** up, down, left, right

Example:

```
<|box_start|>100 200 300 400<|box_end|><|ref_start|>title<|ref_end|><|rotate_up|>
<|box_start|>400 500 600 700<|box_end|><|ref_start|>text<|ref_end|><|rotate_up|>
```

D.2 Text Recognition

The output will contain the recognized text results. The input image will retain its native resolution; however, the number of image tokens will be limited to the range of 4 to 2048. If this limit is exceeded, the image will be scaled accordingly.

Output format:

- **OCR Results:** The raw OCR output

Example:

```
The results of the analyses of the uncertainty of the field data and related assumptions are shown in
     Figs 13 and 14.
```

D.3 Formula Recognition

Any formulas found in the image will be extracted and converted into LaTeX format. The input image will retain its native resolution; however, the number of image tokens will be limited to the range of 4 to 2048. If this limit is exceeded, the image will be scaled accordingly.

Output format:

- **LaTeX Format:** The LaTeX representation of the formula

Example:

```
\[
\hat{F} = \operatorname{Concat}\left(\left[ F_{1}, F_{2}, \dots, F_{n} \right]\right) \tag{2}
\]

\[
M = \sigma \bigl( \mathrm{GELU}(\mathrm{BN}(\mathrm{Conv}_{gate}(\hat{F}))) \bigr) \tag{3}
\]
```

D.4 Table Recognition

The output will include the recognized tables, structured in an OTSL (Open Table Structure Language) format for easy data processing. The input image will retain its native resolution; however, the number of image tokens will be limited to the range of 4 to 2048. If this limit is exceeded, the image will be scaled accordingly.

Output format:

- **OTSL Format:** The table represented in OTSL format

Example:

```
<fcel>Site<fcel>Cl<fcel>NO3<fcel>SO4<fcel>Na<fcel>Ca<fcel>K<fcel>Mg<fcel>NH4<fcel>References<nl>
<fcel>Cl dominance sites<lcel><lcel><lcel><lcel><lcel><lcel><lcel><lcel><lcel><nl>
<fcel>Comba<fcel>109.8<fcel>12.1<fcel>23.3<fcel>86.8<fcel>43.4<fcel>4.8<fcel>15.1<fcel>13.2<fcel>
    Present study<nl>
<fcel>Alibagh<fcel>236<fcel>9<fcel>36<fcel>220<fcel>46<fcel>5<fcel>64<fcel>8<fcel>Naik et al. (2002)<
    nl>
<fcel>Goa<fcel>113.4<fcel>5.5<fcel>27.4<fcel>97.2<fcel>41.5<fcel>2.5<fcel>24.5<fcel>5.5<fcel>Parashar
     et al. (2001)<nl>
<fcel>Bombay<fcel>138<fcel>-<fcel>10<fcel>115<fcel>36<fcel>3.6<fcel>24<fcel>-<fcel>Sequeira (1976)<nl
    >
<fcel>Na dominance sites<lcel><lcel><lcel><lcel><lcel><lcel><lcel><lcel><lcel><nl>
<fcel>Colaba<fcel>171<fcel>34<fcel>52<fcel>179<fcel>133<fcel>6<fcel>59<fcel>12<fcel>Naik et al.
    (2002)<nl>
<fcel>Silent Valley<fcel>43.0<fcel>21.0<fcel>20.0<fcel>46.0<fcel>43.0<fcel>4.0<fcel>14.0<fcel>3.0<
    fcel>Rao et al. (1995)<nl>
<fcel>Chembur<fcel>164.5<fcel>29.5<fcel>70.4<fcel>168.2<fcel>89.5<fcel>6.9<fcel>36.5<fcel>41.1<fcel>
    Khemani et al. (1994)<nl>
<fcel>Bhubaneswar<fcel>18<fcel>10<fcel>19.1<fcel>15<fcel>20.2<fcel>1.8<fcel>5.2<fcel>18.7<fcel>Das et
     al. (2005)<nl>
```

E Qualitative examples

This section presents qualitative examples illustrating the capabilities of the MinerU2.5 through document parsing outputs generated for various pages. This section illustrates the MinerU2.5's performance on Document Parsing, Table Recognition and Formula Recognition among all types of documents.

Examples demonstrating the Document Parsing performance among PDF types are provided in Figures 7 to 9, including Academic literature, Books, Textbooks, Research Report, Financial Report, Slides, Exam Paper, Note, Newspaper and Magazine.

Table Recognition performance among various types of tables is demonstrated in Figures 10 and 11, including the photograph of the table, table with colorful background, table with formula, table with empty cells, handwritten table, large table, rotated table, no-line table, three-line table, and full-line table.

The performance of Formula Recognition among types of formulas is demonstrated in Figures 12 and 13, including formula with background, formula with Chinese, formula with matrix, formula with condition and nested condition, handwritten formula, blurred formula, multi-column formula, degradation formula.

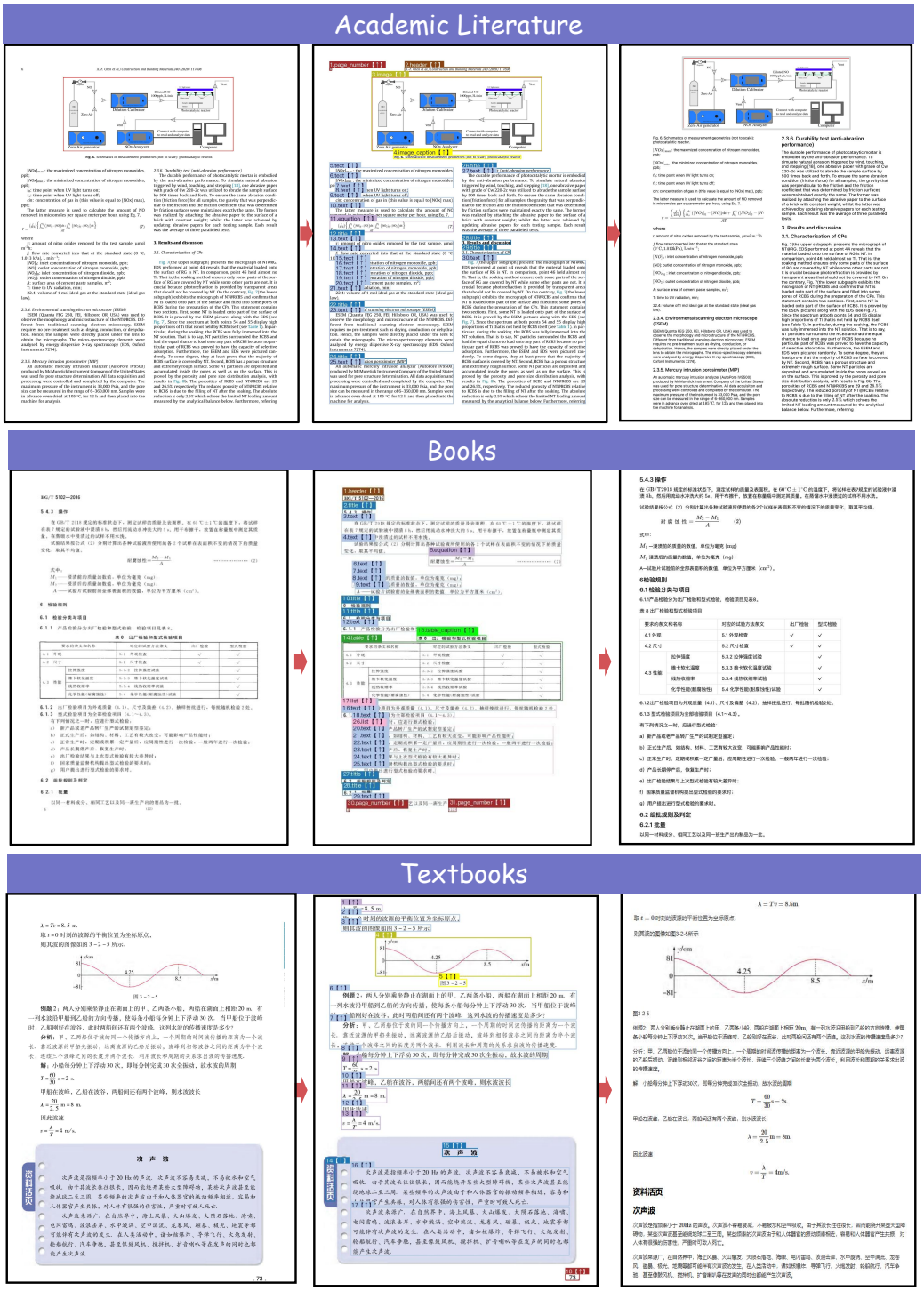

Figure 7: The Layout and rendered markdown output for Academic literature, Books, Textbooks.

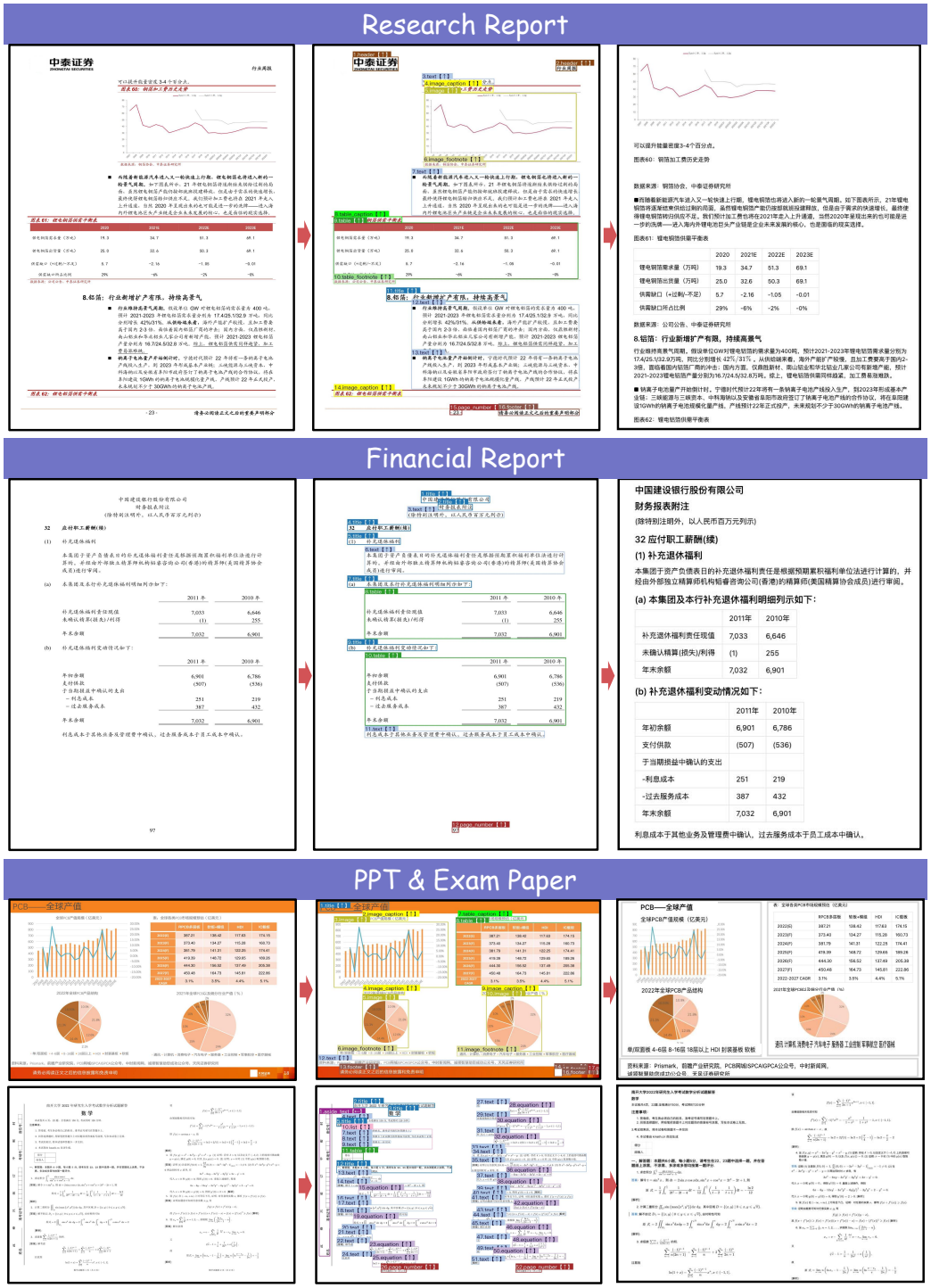

Figure 8: The Layout and rendered markdown output for Research Report, Financial Report, Slides and Exam Paper.

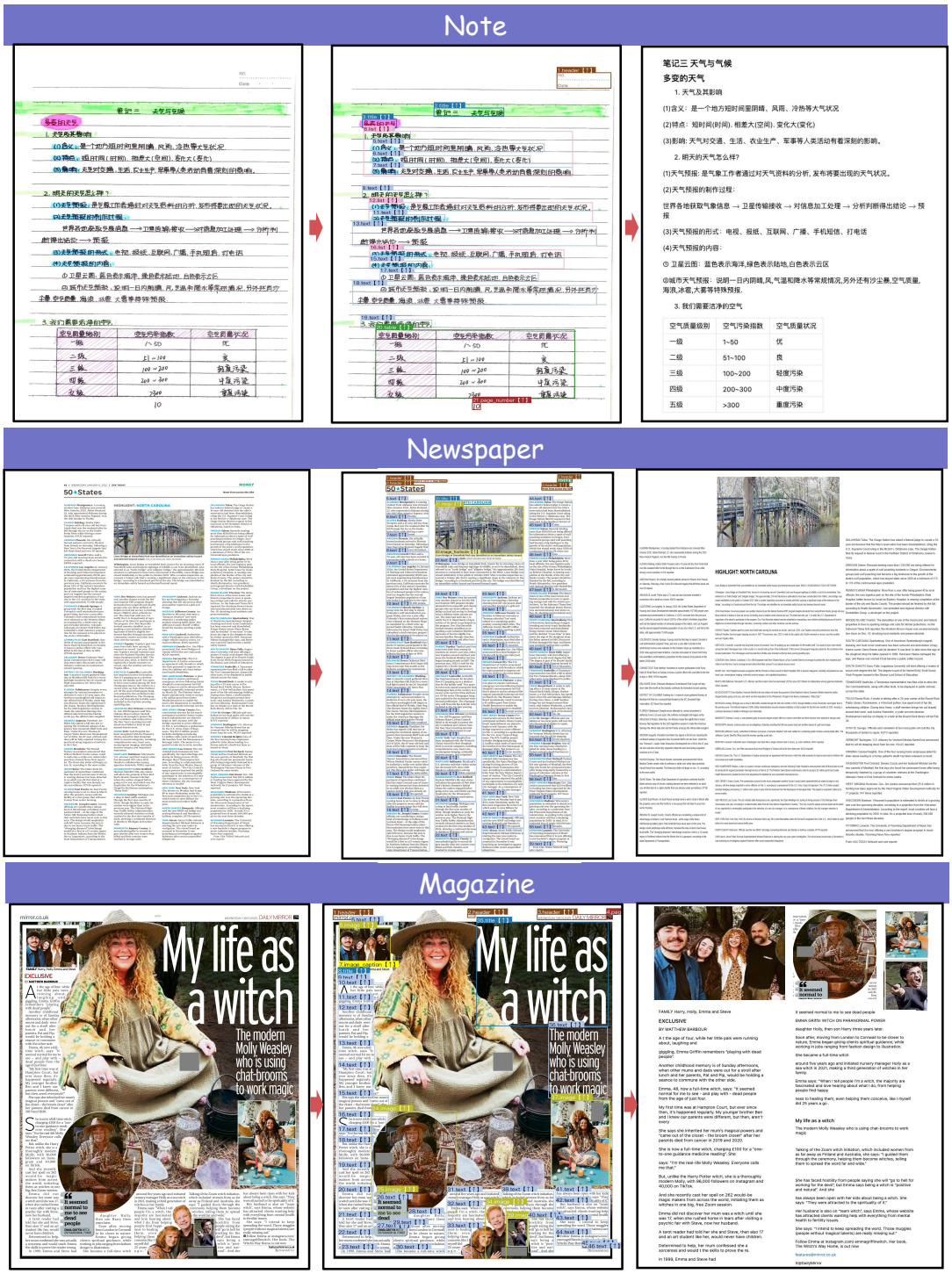

Figure 9: The Layout and rendered markdown output for Note, Newspaper and Magazine.

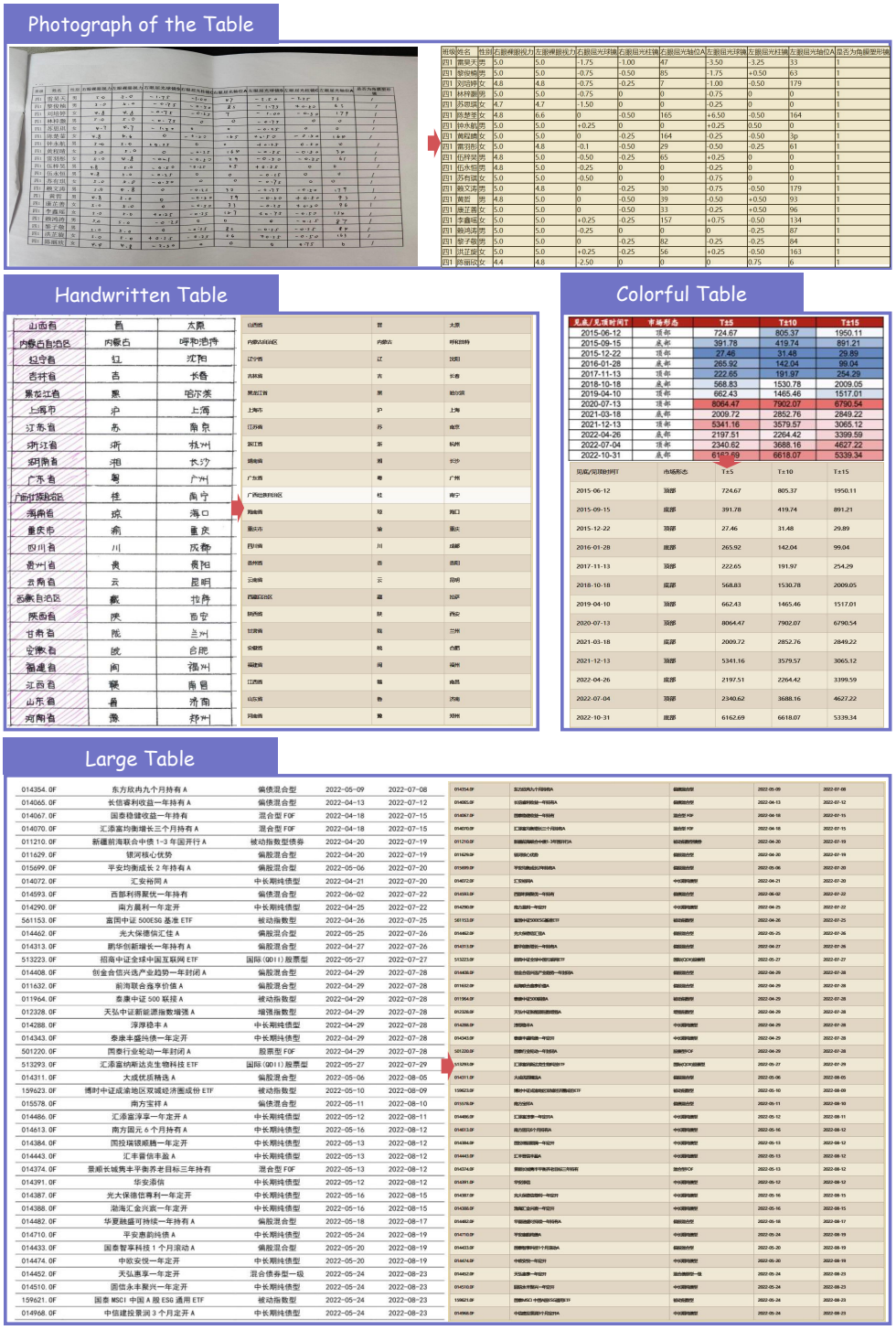

Figure 10: The rendered outputs for various types of Tables.

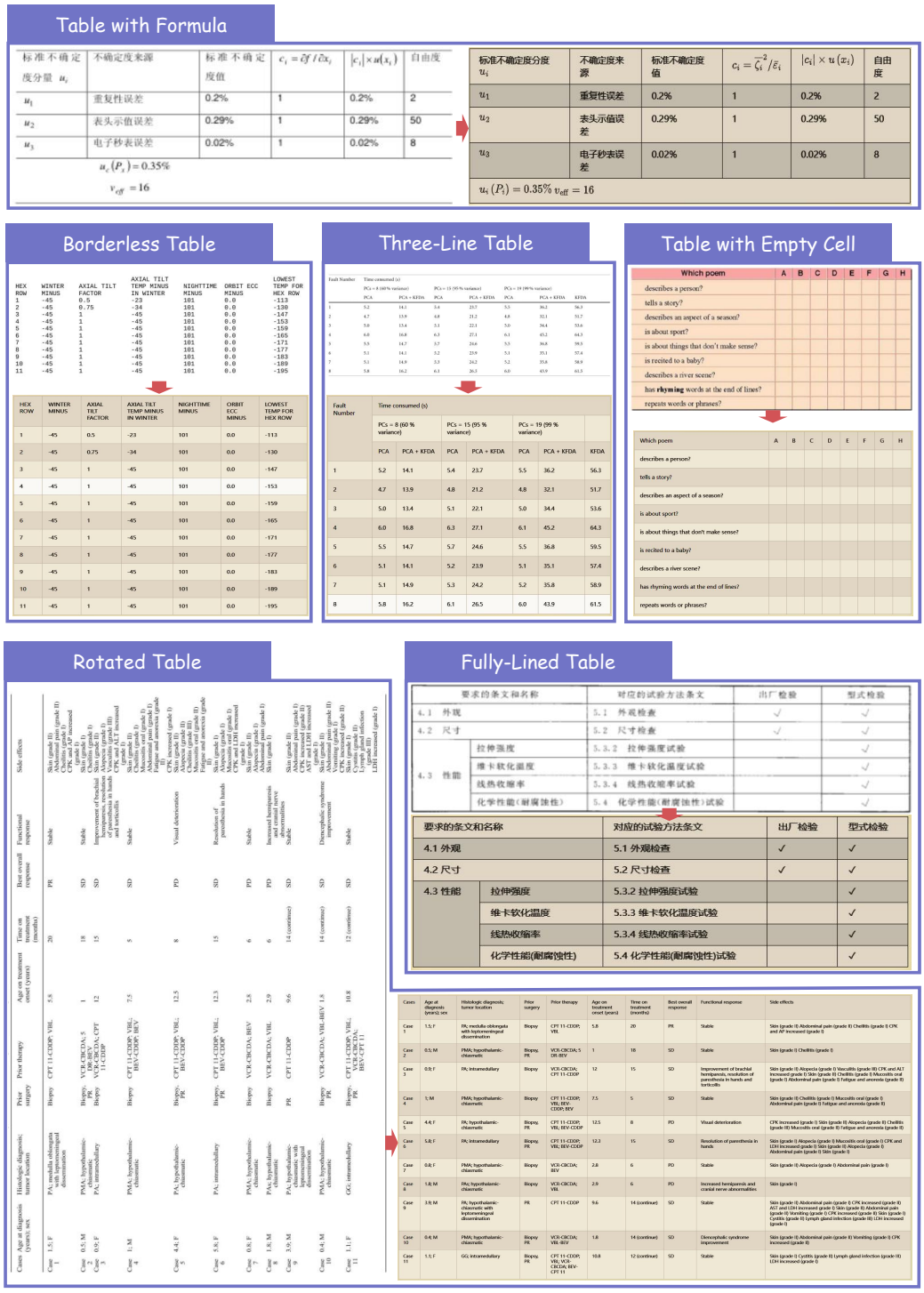

Figure 11: The rendered outputs for various types of Tables.

Formula with Backgroud

$$+\mathcal{M}_{\lambda,\lambda_1}^{\mathrm{cont}}(s,t,s_{12},\Omega^{\mathrm{H}}).$$

$$+\mathcal{M}_{\lambda,\lambda_1}^{\mathrm{cont}}(s,t,s_{12},\Omega^{\mathrm{H}}).$$

$$\begin{cases} d_i(j) = \begin{cases} d_i(j) & d_i(j) - d_i(j-1) \le T_{hi} \\ d_i(j-1) & d_i(j) - d_i(j-1) > T_{hi} \end{cases} \\ \bar{d}_i(j) = \left(d_i(j) - \min(d_{Ti})\right) / \left(\max(d_{Ti}) - \min(d_{Ti})\right) \end{cases} \quad (1)$$

$$\begin{cases} d_i(j) = \begin{cases} d_i(j) & d_i(j) - d_i(j-1) \le T_{hi} \\ d_i(j-1) & d_i(j) - d_i(j-1) > T_{hi} \end{cases} \\ \bar{d}_i(j) = (d_i(j) - \min(d_{Ti})) / (\max(d_{Ti}) - \min(d_{Ti})) \end{cases} \quad (1)$$

$$[\mathrm{CO}]_{\mathrm{d}}(P_{\mathrm{T}} - P_4)(1 + h_d) = P_5 \quad (126)$$

$$[\mathrm{CO}]_{\mathrm{d}}(P_{\mathrm{T}} - P_4)(1 + h_d) = P_5. \quad (126)$$

Handwritten Formula

解: 令 $t = x - \frac{1}{x}$ 则 $x^2 + \frac{1}{x^2} = t^2 + 2$

原式 $= x^2(x^2 - 4x + 2 + \frac{4}{x} + \frac{1}{x^2})$

$= x^2[x^2 + \frac{1}{x^2} - 4(x - \frac{1}{x}) + 2]$

$= x^2(t^2 + 2 - 4t + 2)$

$= x^2(t-2)^2$

$= [x(t-2)]^2$

$= [x(x - \frac{1}{x} - 2)]^2$

$= (x^2 - 2x - 1)^2$

解: 令 $t = x - \frac{1}{x}$ 则 $x^2 + \frac{1}{x^2} = t^2 + 2$

原式 $= x^2(x^2 - 4x + 2 + \frac{4}{x} + \frac{1}{x^2})$

$= x^2[x^2 + \frac{1}{x^2} - 4(x - \frac{1}{x}) + 2]$

$= x^2(t^2 + 2 - 4t + 2)$

$= x^2(t-2)^2$

$= [x(t-2)]^2$

$= [x(x - \frac{1}{x} - 2)]^2$

$= (x^2 - 2x - 1)^2$

$$2)\ \frac{dv}{dt} = \frac{\partial v(t,0)}{\partial t} + \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} f_i(x_i t) \le 0$$

$$2)\ \frac{dv}{dt} = \frac{\partial v(t,0)}{\partial t} + \sum_{i=1}^{n} \frac{\partial V}{\partial x_i} f_i(x_i t) \le 0$$

(6)

$R_3 \circ (R_1 \cap R_2) \subseteq (R_3 \circ R_1) \cap (R_3 \circ R_2)$

$(x,y) \in R_3 \circ (R_1 \cap R_2) \Rightarrow (x,z) \in R_3 \wedge (z;y) \in (R_1 \cap R_2) \Rightarrow (x,z) \in R_3 \wedge$

$\wedge (z;y) \in R_1 \Rightarrow (x,z) \in R_3 \wedge (x,z) \in R_3 \wedge (z;y) \in R_1 \wedge (z;y) \in R_2 \Rightarrow$

$\Rightarrow ((x,z) \in R_3 \wedge (z;y) \in R_1) \wedge (Cx,z) \in R_3 \wedge (z;y) \in R_1) \Rightarrow (x,y) \in (R_3 \circ R_1) \wedge$

$\wedge (x,y) \in (R_3 \circ R_2) \Rightarrow (x,y) \in (R_3 \circ R_1) \cap (R_3 \circ R_2)$ – чтоотпреба

(6)

$R_3 \circ (R_3 \cap R_1) \subseteq (R_3 \circ R_3) \cap (R_3 \circ R_1)$

$(x,y) \in R_3 \circ (R_3 \cap R_1) \Rightarrow (x,z) \in R_3 \wedge (z;y) \in (R_3 \cap R_1) \Rightarrow (x,z) \in R_3 \wedge$

$\wedge (z;y) \in R_3 \Rightarrow (x,z) \in R_3 \wedge (x,z) \in R_3 \wedge (z;y) \in R_1 \wedge (z;y) \in R_3 \Rightarrow$

$\Rightarrow ((x,z) \in R_3 \wedge (z;y) \in R_3) \wedge (Cx,z) \in R_3 \wedge (z;y) \in R_1) \Rightarrow (x,y) \in (R_3 \circ R_3) \wedge$

$\wedge (x,y) \in (R_3 \circ R_1) \Rightarrow (x,y) \in (R_3 \circ R_3) \cap (R_3 \circ R_1)$ – чтоотпреба

Blurred Formula

$$\mathbf{A}\begin{bmatrix}2\\2\end{bmatrix} = \begin{bmatrix}\frac{1}{2} & -\frac{\sqrt{3}}{2}\\ \frac{\sqrt{3}}{2} & \frac{1}{2}\end{bmatrix}\begin{bmatrix}2\\2\end{bmatrix} = \begin{bmatrix}1-\sqrt{3}\\1+\sqrt{3}\end{bmatrix},$$

$$\mathbf{A}\begin{pmatrix}2\\2\end{pmatrix} = \begin{pmatrix}\frac{1}{2} & -\frac{\sqrt{3}}{2}\\ \frac{\sqrt{3}}{2} & \frac{1}{2}\end{pmatrix}\begin{pmatrix}2\\2\end{pmatrix} = \begin{pmatrix}1-\sqrt{3}\\1+\sqrt{3}\end{pmatrix},$$

$$\Leftrightarrow \quad (a,\eta_j) = \sum_{i=1}^{m} k_i(\eta_i,\eta_j), j = 1,2,\cdots,m$$

$$\Leftrightarrow (a,\eta_i) = \sum_{i=1}^{n} k_i(\eta_i,\eta_i), j = 1,2,\ldots,m$$

$$(A^2B^2)(\vec{e}_1) = A^2B(\vec{e}_1) = A^2(-\vec{e}_1) = A(\vec{e}_1) = \vec{e}_1,$$

$$(A^2B^2)(\vec{e}_1) = A^2B(\vec{e}_1) = A^2(-\vec{e}_1) = A(\vec{e}_1) = \vec{e}_1,$$

$$\int_0^{\pi} 9\cos^8(\theta/4)d\theta = (9/2)\int_0^{\pi}(1+\cos(\theta/2))^4 d\theta/16 = (9/32)\int_0^{\pi}[1 + 4\cos(\theta/2) +$$

$$\int_0^{\pi} 9\cos^8(\theta/4)d\theta = (9/2)\int_0^{\pi}(1+\cos(\theta/2))^4 d\theta/16 = (9/32)\int_0^{\pi}[1 + 4\cos(\theta/2) +$$

Formula with Chinese

1 【答案】(1) 原式$= -2 \times \log_2 5 \times (-3) \times \log_3 2 \times (-2) \times \log_5 3 = -12$

(2) 原式$= \left(\log_2 5 - \frac{1}{2}\log_2 5\right)\left(\log_2 2 - \frac{1}{2}\log_2 2\right) = \frac{1}{4}$

(3) 原式$= 2 - \sqrt{3} + 2 + \sqrt{3} = 4$

(4) 原式$= \lg^2 5 - (\lg 2 - 1)^2 + 1 = 1$

2 【答案】(1) $\frac{3pq}{1+3pq}$; (2) $\frac{a+b}{2-a}$

1 [答案](1)原式 $= -2 \times \log_2 5 \times (-3) \times \log_3 2 \times (-2) \times \log_5 3 = -12$

(2)原式 $= \left(\log_2 5 - \frac{1}{2}\log_2 5\right)\left(\log_2 2 - \frac{1}{2}\log_5 2\right) = \frac{1}{4}$

(3)原式 $= 2 - \sqrt{3} + 2 + \sqrt{3} = 4$

(4)原式 $= \lg^2 5 - (\lg 2 - 1)^2 + 1 = 1$

2 [答案](1) $\frac{3pq}{1+3pq}$; (2) $\frac{a+b}{2-a}$

$$L_1 = \left\{ f(t) \left| \begin{array}{l} f(t) = \lim_{n\to\infty} e^{-iA_n t} \prod_{k=1}^{n} f_k\left(\frac{t}{B_n}\right),\ f(t)\ \text{是 c.} \\ \text{t.},\ \{B_n\}\ \text{是正数列},\ \{A_n\}\ \text{是实数列}, \\ \left\{f_k\left(\frac{t}{B_n}\right),\ k = 1, 2, \cdots, n\right\}\text{是 u.a.n. 体系.} \end{array} \right. \right\};$$

$$L_1 = \left\{ f(t) \left| \begin{array}{l} f(t) = \lim_{n\to\infty} e^{-iA_n t} \prod_{k=1}^{n} f_k\left(\frac{t}{B_n}\right),\ f(t)\ \text{是 c.} \\ \text{t.},\ \{B_n\}\ \text{是正数列},\ \{A_n\}\ \text{是实数列}, \\ \left\{f_k\left(\frac{t}{B_n}\right),\ k = 1, 2, \cdots, n\right\}\text{是 u.a.n. 体系.} \end{array} \right. \right\};$$

$$L_2 = \left\{ f(t) \left| \begin{array}{l} f(t) = \lim_{n\to\infty} e^{-iA_n t} \prod_{k=1}^{n} f_k\left(\frac{t}{B_n}\right), f(t)\text{是c.} \\ f(t), \{B_n\}\text{是正数列}, \{A_n\}\text{是实数列}, \\ \left\{f_k\left(\frac{t}{B_n}\right), k = 1, 2, \ldots, n\right\}\text{是 u.a.n. 体系.} \end{array} \right. \right\};$$

$$L_2 = \left\{ f(t) \left| \begin{array}{l} f(t) = \lim_{n\to\infty} e^{-iA_n t} \prod_{k=1}^{n} f_k\left(\frac{t}{B_n}\right), f(t)\text{是c.} \\ f(t), \{B_n\}\text{是正数列}, \{A_n\}\text{是实数列}, \\ \left\{f_k\left(\frac{t}{B_n}\right), k = 1, 2, \ldots, n\right\}\text{是 u.a.n. 体系.} \end{array} \right. \right\};$$

$$\left(\begin{array}{c}\textbf{在系统中的}\\ \textbf{平均等待时间}\end{array}\right) = \frac{W_s\text{的总和}}{\text{总的顾客数}} = \frac{689.44}{20} = 34.47$$

$$\left(\begin{array}{c}\text{在系统中的}\\ \text{平均等待时间}\end{array}\right) = \frac{W_s\text{的总和}}{\text{总的顾客数}} = \frac{689.44}{20} = 34.47$$

Matrices

$$\mathrm{Where}\ M = \begin{bmatrix} -\lambda & 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & p^2-1 \\ 1 & -\lambda & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & -\lambda & \cdots & 1 & 1 & 1 & \cdots & 1 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & \cdots & -\lambda & 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & 1 & \cdots & 1 & -\lambda & 1 & \cdots & 1 & p^2-1 \\ 1 & 1 & 1 & \cdots & 1 & 1 & -\lambda & \cdots & 1 & p^2-1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & -\lambda & p^2-1 \\ 1 & 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & -\lambda+p^2-2 \end{bmatrix}, N = \begin{bmatrix} 1 & \cdots & 1 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \\ 1 & \cdots & 1 \\ 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$$

$$\mathrm{Where}\ M = \begin{bmatrix} -\lambda & 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & p^2-1 \\ 1 & -\lambda & 1 & \cdots & 1 & 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & -\lambda & \cdots & 1 & 1 & 1 & \cdots & 1 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & \cdots & -\lambda & 1 & 1 & \cdots & 1 & 0 \\ 1 & 1 & 1 & \cdots & 1 & -\lambda & 1 & \cdots & 1 & p^2-1 \\ 1 & 1 & 1 & \cdots & 1 & 1 & -\lambda & \cdots & 1 & p^2-1 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 1 & 1 & \cdots & 1 & 1 & 1 & \cdots & -\lambda & p^2-1 \\ 1 & 0 & 0 & \cdots & 0 & 1 & 1 & \cdots & 1 & -\lambda+p^2-2 \end{bmatrix}, N = \begin{bmatrix} 1 & \cdots & 1 \\ 0 & \cdots & 0 \\ 0 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & 0 \\ 1 & \cdots & 1 \\ 1 & \cdots & 1 \\ \vdots & \ddots & \vdots \\ 1 & \cdots & 1 \end{bmatrix}$$

$$+ \underbrace{\begin{bmatrix} E_1 & 0 & 0 & \cdots & 0 \\ 0 & E_2 & 0 & \ddots & \vdots \\ 0 & 0 & E_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \cdots & 0 \\ 0 & \cdots & 0 & 0 & E_M \end{bmatrix}}_{E} \underbrace{\begin{bmatrix} d_{[1]} \\ d_{[2]} \\ d_{[3]} \\ \vdots \\ d_{[M]} \end{bmatrix}}_{y} \underbrace{\begin{bmatrix} y_{[1]} \\ y_{[2]} \\ y_{[3]} \\ \vdots \\ y_{[M]} \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} C_1 & 0 & 0 & \cdots & 0 \\ 0 & C_2 & 0 & \ddots & \vdots \\ 0 & 0 & C_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \cdots & 0 \\ 0 & \cdots & 0 & 0 & C_M \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} x_{[1]} \\ x_{[2]} \\ x_{[3]} \\ \vdots \\ x_{[M]} \end{bmatrix}}_{x}$$

$$+ \underbrace{\begin{bmatrix} E_1 & 0 & 0 & \cdots & 0 \\ 0 & E_2 & 0 & \ddots & \vdots \\ 0 & 0 & E_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \cdots & 0 \\ 0 & \cdots & 0 & 0 & E_M \end{bmatrix}}_{E} \underbrace{\begin{bmatrix} d_{[1]} \\ d_{[2]} \\ d_{[3]} \\ \vdots \\ d_{[M]} \end{bmatrix}}_{y} \underbrace{\begin{bmatrix} y_{[1]} \\ y_{[2]} \\ y_{[3]} \\ \vdots \\ y_{[M]} \end{bmatrix}}_{y} = \underbrace{\begin{bmatrix} C_1 & 0 & 0 & \cdots & 0 \\ 0 & C_2 & 0 & \ddots & \vdots \\ 0 & 0 & C_3 & \ddots & \vdots \\ \vdots & \ddots & \ddots & \cdots & 0 \\ 0 & \cdots & 0 & 0 & C_M \end{bmatrix}}_{C} \underbrace{\begin{bmatrix} x_{[1]} \\ x_{[2]} \\ x_{[3]} \\ \vdots \\ x_{[M]} \end{bmatrix}}_{x}$$

$$\mathbf{H}_{\mathrm{e}} = \begin{bmatrix} 1 & \frac{1}{3} & \cdots & \frac{1}{k-1} & 0 & \cdots & \cdots & 0 \\ \frac{1}{3} & \frac{1}{5} & \cdots & \frac{1}{k+1} & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{k-1} & \frac{1}{k+1} & \cdots & \frac{1}{2k-3} & 0 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & 0 & -\frac{1}{3} & -\frac{1}{5} & \cdots & -\frac{1}{k+1} \\ 0 & \cdots & \cdots & 0 & -\frac{1}{5} & -\frac{1}{7} & \cdots & -\frac{1}{k+3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & -\frac{1}{k+1} & -\frac{1}{k+3} & \cdots & -\frac{1}{2k-1} \end{bmatrix}.$$

$$\mathbf{H}_{\mathrm{e}} = \begin{bmatrix} 1 & \frac{1}{3} & \cdots & \frac{1}{k-1} & 0 & \cdots & \cdots & 0 \\ \frac{1}{3} & \frac{1}{5} & \cdots & \frac{1}{k+1} & 0 & \cdots & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ \frac{1}{k-1} & \frac{1}{k+1} & \cdots & \frac{1}{2k-3} & 0 & \cdots & \cdots & 0 \\ 0 & \cdots & \cdots & 0 & -\frac{1}{3} & -\frac{1}{5} & \cdots & -\frac{1}{k+1} \\ 0 & \cdots & \cdots & 0 & -\frac{1}{5} & -\frac{1}{7} & \cdots & -\frac{1}{k+3} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & \cdots & \cdots & 0 & -\frac{1}{k+1} & -\frac{1}{k+3} & \cdots & -\frac{1}{2k-1} \end{bmatrix}.$$

Figure 12: The rendered outputs for various types of Formulas.

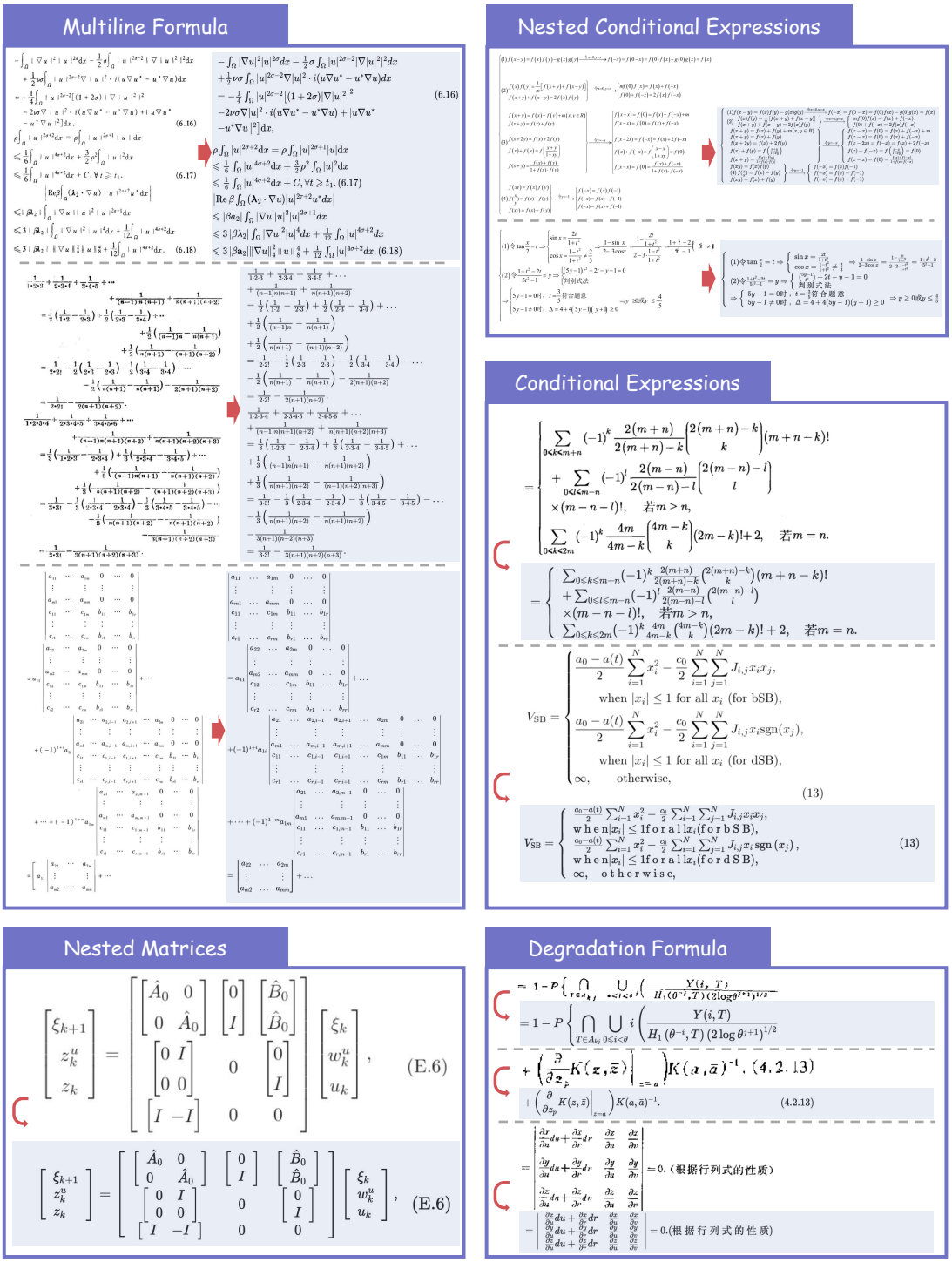

Figure 13: The rendered outputs for various types of Formulas.