PaddleOCR-VL: Boosting Multilingual Document Parsing via a 0.9B Ultra-Compact Vision-Language Model

Cheng Cui, Ting Sun, Suyin Liang, Tingquan Gao, Zelun Zhang, Jiaxuan Liu, Xueqing Wang, Changda Zhou, Hongen Liu, Manhui Lin, Yue Zhang, Yubo Zhang, Handong Zheng, Jing Zhang, Jun Zhang, Yi Liu, Dianhai Yu, Yanjun Ma

PaddlePaddle Team, Baidu Inc.
paddleocr@baidu.com

Source Code: https://github.com/PaddlePaddle/PaddleOCR
Models & Online Demo: https://huggingface.co/PaddlePaddle

Abstract

In this report, we propose PaddleOCR-VL, a SOTA and resource-efficient model tailored for document parsing. Its core component is PaddleOCR-VL-0.9B, a compact yet powerful vision-language model (VLM) that integrates a NaViT-style dynamic resolution visual encoder with the ERNIE-4.5-0.3B language model to enable accurate element recognition. This innovative model efficiently supports 109 languages and excels in recognizing complex elements (e.g., text, tables, formulas, and charts), while maintaining minimal resource consumption. Through comprehensive evaluations on widely used public benchmarks and in-house benchmarks, PaddleOCR-VL achieves SOTA performance in both page-level document parsing and element-level recognition. It significantly outperforms existing solutions, exhibits strong competitiveness against top-tier VLMs, and delivers fast inference speeds. These strengths make it highly suitable for practical deployment in real-world scenarios.

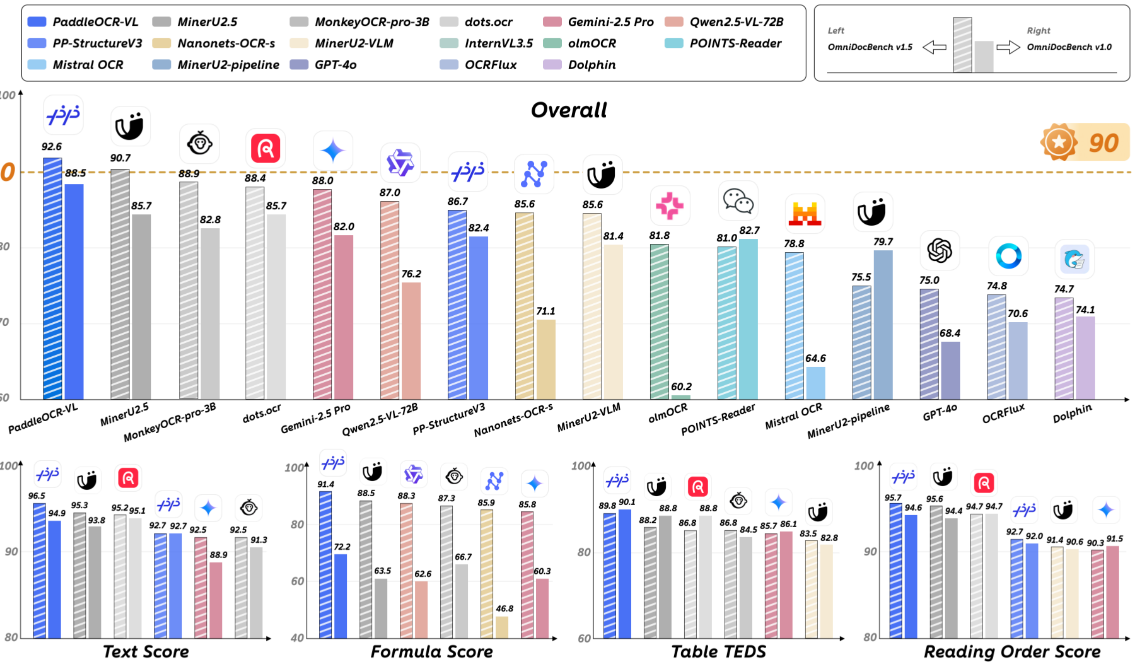


Figure 1 | Performance of PaddleOCR-VL on OmniDocBench v1.0 and v1.5.

Contents

1. Introduction

Documents serve as core information carriers, with their complexity and volume growing at an exponential rate, making document parsing an indispensable key technology. The primary goal of document parsing [1, 2, 3, 4] is to enable deep structural and semantic understanding of a document's layout. Specifically, it involves recognizing distinct text blocks and columns, distinguishing formulas, tables, charts, and images, determining the correct reading order, and detecting key elements (e.g., footnotes and image captions); these capabilities collectively lay a solid foundation for efficient information retrieval and data management. Furthermore, advanced document parsing enables large language models (LLMs) [5, 6, 7], especially when combined with Retrieval-Augmented Generation (RAG) [8], to access high-quality knowledge and enhance their practical applications.

The inherent complexity of modern documents presents unique challenges: they often combine dense text, complex tables or chart, mathematical expressions, multiple languages and handwritten texts, with deserve layout structures. Recent research [1, 9, 10, 11, 12] in the field of document parsing primarily following two technological approaches. The first approach [9, 10] employs pipeline methodologies based on specialized, modular expert models. Although these methods offer strong performance, they are increasingly hindered by integration complexity, cumulative error propagation, and inherent limitations when handling highly complex documents. Secondly, end-to-end approaches [12, 13, 14] leveraging multimodal models aim to simplify the workflow and enable joint optimization. However, these methods often struggle with correct text order and can even generate hallucinations when faced with lengthy or complex layouts, while also incurring substantial computational overhead for long sequence outputs, thereby restricting their practical deployment.

To address these advancements and challenges, we present PaddleOCR-VL, a high-performance, resource-efficient document parsing solution based on a vision-language model. This innovation paves the way for the widespread application of multimodal document parsing, particularly in resource-constrained environments. PaddleOCR-VL combines a robust layout analysis model with a compact yet powerful vision-language model, PaddleOCR-VL-0.9B.

Firstly, PaddleOCR-VL performs layout detection and reading order prediction to obtain the positional coordinates and reading order of elements (text blocks, tables, formulas, and charts). Compared to multimodal methods that rely on grounding and sequence output (e.g., MinerU2.5 [2], Dolphin [3]), our method offers faster inference speeds, lower training costs, and easier extensibility for new layout categories. Subsequently, the elements are segmented based on their positions and fed into PaddleOCR-VL-0.9B for recognition. This vision-language model is specifically designed for resource-efficient inference and excels at element recognition within document parsing. By integrating a NaViT-style [15] dynamic high-resolution visual encoder with the lightweight ERNIE-4.5-0.3B [5] language model, we have significantly enhanced the model's dense text recognition capabilities and decoding efficiency.

To train a powerful multimodal model, we have developed a high-quality training data construction pipeline. We collected over 30 million training samples through public data acquisition and data synthesis. We meticulously designed prompt engineering to guide the automatic labeling by general large models, based on the recognition results of expert models. Simultaneously, We performed data cleaning to remove low-quality or inconsistent annotations, such as those caused by model hallucinations. We designed an evaluation engine, which is an assessment collection that categorizes each element into more detailed categories. Through this automated evaluation, we can analyze the current model's training performance across different types. This allows us to conduct targeted hard sample mining based on element types and to construct similar challenging examples through data synthesis. Finally, we incorporated manual annotation for a small number of corner cases to complete the construction of the training data.

Comprehensive benchmarking on the public benchmarks, including OmniDocBench v1.0, v1.5 [16] and olmOCR-Bench [12], and in-house ones demonstrate that PaddleOCR-VL achieves SOTA performance in document parsing task, significantly outperforming existing pipeline-based solutions and exhibiting strong competitiveness against leading vision-language models (VLMs). Moreover, PaddleOCR-VL is optimized for efficiency, delivering substantially lower latency and higher throughput than competing approaches.

PaddleOCR-VL actively addresses current challenges in document processing with a high-performance, resource-efficient multimodal document parsing solution. Its key contributions include:

- **Compact yet Powerful VLM Architecture:** We present a novel vision-language model that is specifically designed for resource-efficient inference, achieving outstanding performance in element recognition. By integrating a NaViT-style dynamic high-resolution visual encoder with the lightweight ERNIE-4.5-0.3B language model, we significantly enhance the model's recognition capabilities and decoding efficiency. This integration maintains high accuracy while reducing computational demands, making it well-suited for efficient and practical document processing applications.
- **High-quality Data Construction Methodology:** We propose a systematic and comprehensive methodology for constructing high-quality datasets, providing a solid train data foundation for efficient and robust document parsing. This methodology not only enables us to construct high-quality data on demand, but also provides a new perspective on the automated generation of high-quality data.
- **SOTA Performance Document Parsing:** PaddleOCR-VL achieves state-of-the-art performance in document parsing task. It excels in recognizing complex document elements, such as **text, tables, formulas, and charts**, making it suitable for a wide range of challenging content types, including handwritten text and historical documents. Supporting **109 languages**, including major global languages and those with diverse scripts like Russian, Arabic, and Hindi, PaddleOCR-VL is highly applicable to multilingual and globalized document processing scenarios.

2. PaddleOCR-VL

2.1. Architecture

PaddleOCR-VL decomposes the complex task of document parsing into a two stages, as illustrated in Figure 2. The first stage, PP-DocLayoutV2, is responsible for layout analysis, where it localizes semantic regions and predicts their reading order. Subsequently, the second stage, PaddleOCR-VL-0.9B, leverages these layout predictions to perform fine-grained recognition of diverse content, including text, tables, formulas, and charts. Finally, a lightweight post-processing module aggregates the outputs from both stages and formats the final document into structured Markdown and JSON.

2.1.1. Layout Analysis

Considering that end-to-end approaches based on VLM rely on long-sequence autoregressive processes, which result in high latency and memory consumption, and increase the risk of

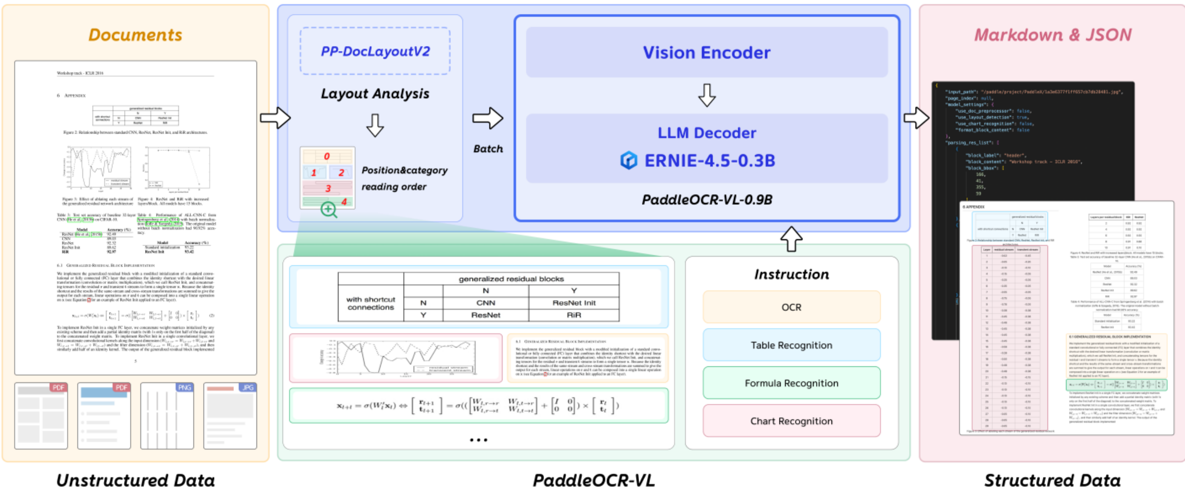


Figure 2 | The overview of PaddleOCR-VL.

unstable layout analysis and hallucinations—problems that are particularly pronounced in multi-column or mixed text–graphic layouts—we employ a dedicated lightweight model for layout analysis, focusing specifically on element detection, classification, and reading order prediction.

Specifically, we decouple the layout analysis process by introducing an independent model, PP-DocLayoutV2, dedicated solely to this task. PP-DocLayoutV2 consists of an object detection model (RT-DETR [17]) for elements localization and classification, as well as a lightweight pointer network [18] with six transformer layers to accurately predict the reading order of layout elements.

This separation enables us to fully leverage the advanced capabilities of the vision model, which typically requires lower input image resolution, and contains significantly fewer parameters. As a result, it achieves stable and accurate layout analysis, without the instability issues that may arise in end-to-end approaches.

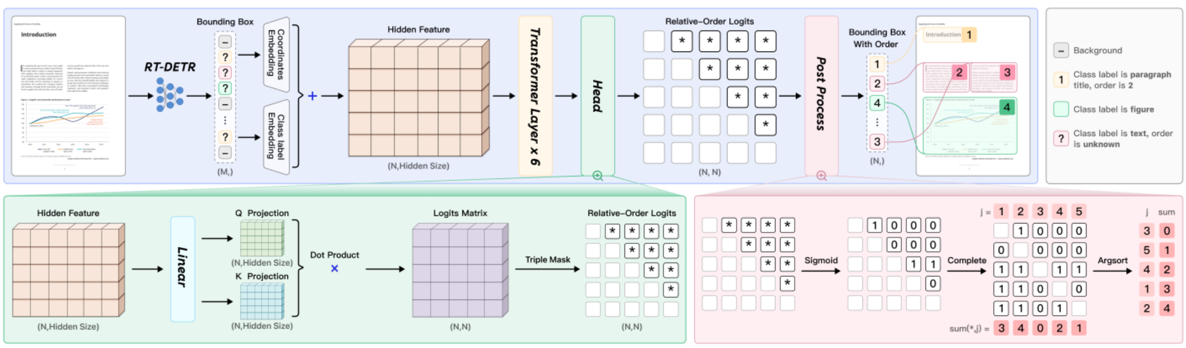


Figure 3 | Architecture of layout analysis model.

Architecturally, PP-DocLayoutV2 is composed of two sequentially connected networks, as shown in Figure 3. The first is an RT-DETR-based [17] detection model that performs layout element detection and classification. The detected bounding boxes and class labels are then passed to a subsequent pointer network, which is responsible for ordering these layout elements.

Specifically, we first apply per-class thresholds to select foreground proposals for the ordering network. The selected proposals are embedded using absolute 2D positional encodings and class label embeddings. Additionally, the encoder attention incorporates a geometric bias mechanism from Relation-DETR [18] to explicitly model pairwise geometric relationships among elements. The pairwise relation head linearly projects element representations into query and key vectors, then computes bilinear similarities to produce pairwise logits, resulting in an $N \times N$ matrix that represents the relative order between each pair of elements. Finally, a deterministic win-accumulation decoding algorithm recovers a topologically consistent reading order for the detected layout elements.

In comparison to other specialized models, such as LayoutReader [19], our model achieves higher performance with fewer parameters by efficiently extending RT-DETR [17] with a pointer network.

2.1.2. *Element-level Recognition*

We systematically explore architecture configurations optimized for high accuracy and low computational overhead, and propose the PaddleOCR-VL-0.9B as shown in Figure 4.

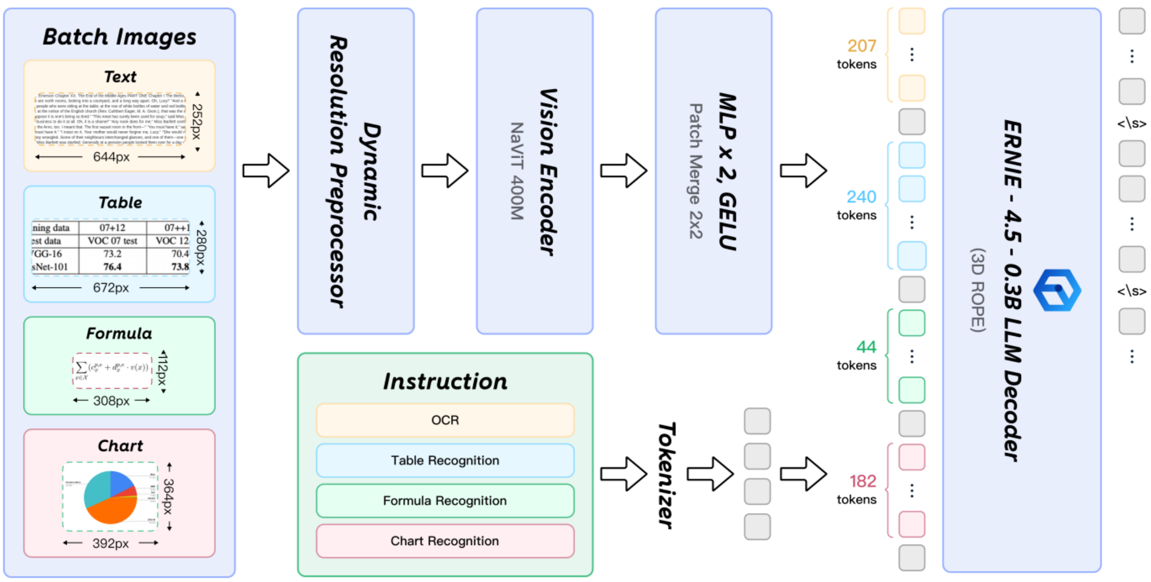


Figure 4 | Architecture of PaddleOCR-VL-0.9B.

We adopted an architectural style inspired by LLaVA [20], integrating a pre-trained vision encoder with a dynamic resolution preprocessor, a randomly initialized 2-layer MLP projector, and a pre-trained large language model. Our architecture achieves a balance the scale of vision and language models to optimize performance in multi-elements recognition tasks.

Compared to earlier document parsing models based on fixed-resolution or tiling-based approaches [4, 14, 21], our approach utilizes native dynamic high-resolution preprocessing. For the vision encoder, we employed a NaViT-style [15] encoder initialized from Keye-VL's [22] vision model, which support native-resolution inputs. This design enables the vision-language model to handle images of arbitrary resolution without distortion, yielding fewer hallucinations and stronger performance on text-intensive tasks.

The projector is a randomly initialized 2-layer MLP with GELU [23] activation, incorporating a merge size of 2 to efficiently bridge visual features from the encoder to the language model's embedding space.

In auto-regressive language models, the entire sequence is generated by predicting one token at a time. This approach means that the size of the decoder is directly linked to the overall inference latency, so a smaller model will decode faster. With this in mind, we use the ERNIE-4.5-0.3B [5] model, an open-source language model that balances a relatively small number of parameters with strong inference efficiency. In our implementation, we further enhance positional representation by incorporating a 3D-RoPE[24].

The combination of NaViT [15] with ERNIE-4.5-0.3B [5] has led to significant performance improvements in documents parsing, achieving minimal memory usage and faster inference speed.

2.2. Training Recipe

The following sections introduce the training details of these two modules: PP-DocLayoutV2 for layout analysis and PaddleOCR-VL-0.9B for element recognition.

2.2.1. Layout Analysis

We employ the PP-DocLayoutV2 model to perform layout element localization, classification, and reading order prediction. PP-DocLayoutV2 extends RT-DETR [17] by incorporating an additional pointer network [18], which is responsible for predicting the reading order of detected elements. The training process adopts a two-stage strategy: we first train the core RT-DETR [17] model for layout detection and classification. Afterward, we freeze its parameters and independently train the pointer network for reading order prediction.

For the first stage, we follow the training strategy of RT-DETR [17]. Specifically, we initialize the model with PP-DocLayout_Plus-L [25] pretrained weights and train it for 100 epochs on our self-constructed dataset comprising over 20,000 high-quality samples.

For the second stage, specifically, the model outputs a matrix representing the pairwise ordering relationships between any two elements, and the Generalized Cross Entropy Loss [26] is computed with respect to the ground truth labels, as this loss function demonstrates increased robustness in scenarios where pre-annotated data are mixed into the dataset. We utilize a constant learning rate 2e-4 and the AdamW optimizer to train 200 epochs.

2.2.2. Element-level Recognition

As described in Section 2.1.2, PaddleOCR-VL-0.9B consists of three modules: a vision encoder, a projector, and a language model. We adopt a post-adaptation strategy using pre-trained models. Specifically, the vision model is initialized with Keye-VL's weights, and the language model is initialized with ERNIE-4.5-0.3B's weights. The model is trained based on the ERNIEKit [27] repository and the training methodology for our VLM is divided into two stages, as outlined in Table 1.

Stage 1: The initial stage focuses on pre-training alignment, where the model learns to associate visual information from images with corresponding textual representations. This crucial step is performed on a massive dataset comprising 29 million high-quality image-text pairs. During this phase, which runs for one epoch, the model is trained to establish a coherent

Stages	Stage 1	Stage 2
Training Samples	29M	2.7M
Max Resolution	$1280 \times 28 \times 28$	$2048 \times 28 \times 28$
Sequence length	16384	16384
Trainable components	All	All
Batch sizes	128	128
Data Augmentation	Yes	Yes
Maximum LR	5×10^{-5}	5×10^{-6}
Minimum LR	5×10^{-6}	5×10^{-7}
Epoch	1	2

Table 1 | Training settings in stage 1 and stage 2.

understanding between diverse visual inputs and their semantic textual content. The training utilizes a batch size of 128, a sequence length of 16384, and supports a maximum image resolution of 1280×28×28, with data augmentation enabled to improve robustness. For optimization, the learning rate is scheduled between a maximum of 5×10^{-5} and a minimum of 5×10^{-6}. The primary objective is to align the feature spaces of the vision encoder and the language model, enabling them to jointly process multimodal information effectively. This large-scale pre-training allows the model to capture intricate visual patterns, common textual structures, and their interdependencies across a vast range of contexts, laying a strong foundation for subsequent specialized tasks.

Stage 2: Following pre-training, the model undergoes **instruction fine-tuning** to adapt its general multimodal understanding to specific downstream elements recognition tasks. This stage utilizes a meticulously curated dataset of 2.7 million samples, which is intentionally designed to be highly rich and diverse in its distribution. The training is conducted over two epochs, maintaining the 128 batch size and 16384 sequence length, but increasing the maximum resolution to 2048×28×28 to handle more detailed inputs. A finer learning rate is adopted, with the maximum and minimum values set to 5×10^{-6} and 5×10^{-7}, to carefully adjust the model on specialized data. The richness of this dataset encompasses a wide variety of document types, languages, writing systems, and visual complexities pertinent to real-world scenarios. During this fine-tuning phase, the model is trained with explicit instructions for four types of tasks:

1. **OCR:** This task fine-tunes the model to accurately identify and extract textual content from images, encompassing individual characters, words, text lines, text blocks and simple layout structure of page-level texts.
2. **Table Recognition:** The model learns to parse tabular structures within documents. This involves accurately extracting cell contents, identifying rows and columns, and recognize the logical relationships between different table elements, ultimately generating structured representations based on OTSL [28] format.
3. **Formula Recognition:** This instruction focuses on enabling the model to recognize and interpret mathematical and scientific formulas. It aims to convert their visual representation into a structured LaTeXformat and distinguishes between inline \(...\) and display \[...\] equations.
4. **Chart Recognition:** This task trains the model to recognition information from various types of charts, such as bar charts, line graphs, and pie charts and convert Markdown format tables.

3. Dataset

To build our high-quality and diverse training dataset, we propose a systematic methodology for constructing such datasets. As illustrated in Figure 5, we gather a diverse set of data from multiple sources to ensure comprehensive coverage. High-quality labels are then generated through automated annotation using large models, which guarantees precision and consistency. Additionally, we refine the training data by integrating challenging examples, which enhances the model's performance and robustness. Each of these crucial steps is detailed in the following sections.

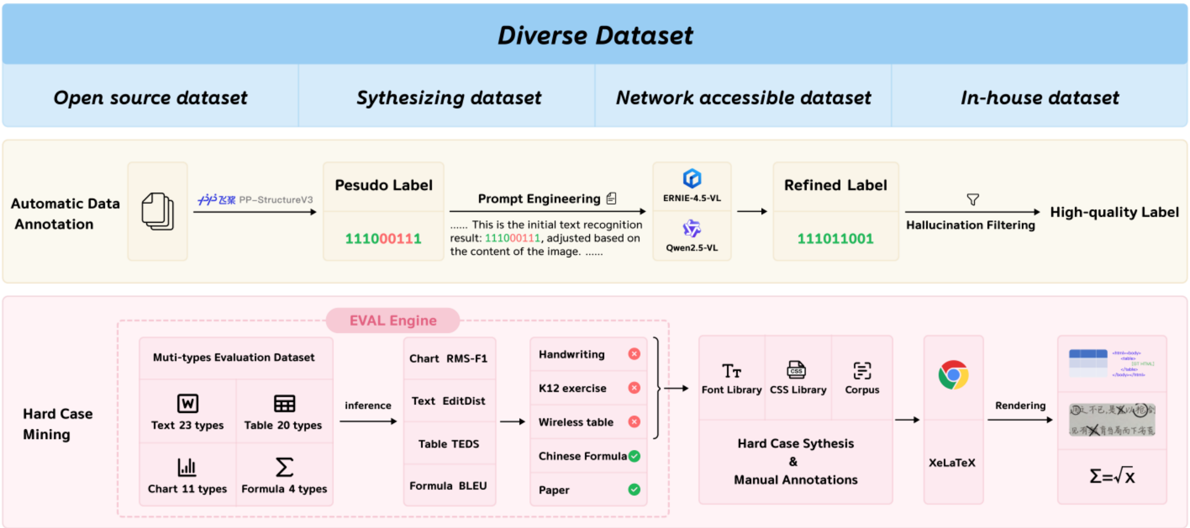


Figure 5 | The construction process of training data for PaddleOCR-VL-0.9B.

3.1. Data Curation

To ensure the breadth and diversity of the dataset, data is collected from four main sources: open-source dataset, synthesizing dataset, network accessible dataset, and in-house dataset.

1. **Open Source Dataset:** As the foundation of our dataset, we systematically aggregated and curated a wide array of established public datasets. For textual content, we sourced data from the canonical dataset CASIA-HWDB [29]. Our mathematical expression data is derived from UniMER-1M [30] and MathWriting [31]. To ensure comprehensive coverage of data visualizations, we incorporated a rich spectrum of chart and graph datasets, including ChartQA [32], PlotQA [33], Chart2Text [34], DVQA [35], Unichart [36], Beagle [37], ChartINFO [38], visText [39], and ExcelChart [40]. Each of these sources underwent an initial filtering and cleaning protocol to rectify or discard noisy and low-quality annotations.
2. **Data Synthesizing Dataset:** Due to the naturally imbalanced distribution of public data, we employed a data synthesizing strategy to produce large volumes of missing data types at low cost, providing our proposed model with the unbiased document parsing performance.
3. **Network Accessible Dataset:** To improve model generalization and robustness against the complexities of unstructured real-world documents, we amassed an extensive corpus of publicly accessible data harvested from the Internet. This public collection was deliberately curated to encompass a rich spectrum of document types and visual styles. It includes

academic papers, newspapers, formal scientific journal articles, scanned handwritten documents, diverse examination papers, and slides, etc. The integration of these varied sources proved instrumental in significantly broadening the stylistic, structural, and domain diversity of our training data, thereby mitigating the risk of overfitting to clean, canonical datasets.

4. **In-house Dataset:** Through years of research in the field of OCR, we have accumulated extensive datasets with diverse data types across all tasks of document parsing. We incorporate all in-house datasets into training with precisely controlled proportions, which have become unnecessary factors that enable our models to achieve outstanding performance.

3.2. Automatic Data Annotation

After acquiring the raw data, we utilize an automatic data annotations process for large-scale labeling. Initially, we employ the expert model, PP-StructureV3, to conduct preliminary processing on the data, generating pseudo labels that may contain some inaccuracies. Subsequently, through prompt engineering, we create prompts that include the original images and their associated pseudo labels, which are then submitted to more advanced multimodal large language models, ERNIE-4.5-VL [5] and Qwen2.5VL [24]. These sophisticated models refine and enhance the initial results by analyzing the image content, resulting in improved labels. Finally, to ensure the quality of the labels, the system performs a hallucination filtering step, which eliminates any potentially incorrect content generated by the large models, thereby producing reliable and high-quality labels.

3.3. Hard Cases Mining

To overcome performance bottlenecks in specific complex scenarios, we propose a hard case mining process for targeted performance improvement. We firstly develop a eval engine for various types. We created substantial evaluation data with precisely labeled data obtained through manual annotation. Theses evaluation datasets are categorized into several types: text data includes 23 categories such as Chinese, English, printed, handwritten, Japanese, Latin, and emojis; table data includes 20 categories such as limited tables, unlimited tables, handwritten tables, checklists, invoices, and rotated tables; formula data includes 4 categories such as Chinese and English formulas, handwritten and printed, simple, and complex; chart data includes 11 categories such as Chinese and English charts, line charts, and bar charts, sourced from diverse origins to cover different document. By inference on this evaluation set and using corresponding professional metrics (e.g., EditDist for Text, TEDS [41] for Tables, RMS-F1 [42] for Charts, and BLEU [43] for Formulas), we can accurately identify hard cases where the model performs poorly. Finally, for these identified weaknesses, the system utilizes a rich set of resources (such as Font Library, CSS Library, Corpus) and rendering tools (like XeLaTeX and web browsers) to synthetically generate a large volume of new, high-quality hard cases.

4. Evaluation

To thoroughly assess the effectiveness of PaddleOCR-VL, we compared it against leading general vision language models and specialized document parsing models across multiple public benchmarks and in-house benchmarks. We conducted comprehensive performance comparisons in two aspects: page-level document parsing and element-level recognition, which are detailed in Sections 4.1 and 4.2. Page-level involves analyzing entire pages of a document to parsing their overall content, structure and layout, while element-level is dedicated exclusively to assessing the recognition of specific elements, such as text, tables, formulas, and charts, within the document.

4.1. Page-level Evaluation

This section details the evaluation of end-to-end document parsing capabilities using the following three benchmarks, aiming to measure its overall performance in real-world document scenarios.

OmniDocBench v1.5 To comprehensively evaluate the document parsing capabilities, we conducted extensive experiments on the OmniDocBench v1.5 [2] benchmark. It is an expansion of version v1.0, adding 374 new documents for a total of 1,355 document pages. It features a more balanced distribution of data in both Chinese and English, as well as a richer inclusion of formulas and other elements. The evaluation method has been updated, with formulas assessed using the CDM method. The overall metric is a weighted combination of the metrics for text, formulas, and tables.

Table 2 demonstrate that PaddleOCR-VL achieves SOTA performance, outperforming existing pipeline tools, general VLMs, and other specialized document parsing models across all key metrics. Specifically, our model achieves a top-ranking overall score of 92.56, surpassing the next best model, MinerU2.5-1.2B (90.67). Moreover, our model establishes new SOTA results in the sub-tasks, including the lowest Text-Edit distance [44] of 0.035, the highest Formula-CDM score of 91.43, the leading scores of 89.76 and 93.52 in Table-TEDS and Table-TEDS-S, and the best readering ordering scores of 0.043, respectively. These results underscore its superior accuracy in text recognition, formula recognition, and complex table structure analysis.

Model Type	Methods	Parameters	Overall↑	$Text^{Edit}$↓	$Formula^{CDM}$↑	$Table^{TEDS}$↑	$Table^{TEDS\text{-}S}$↑	Reading $Order^{Edit}$↓
Pipeline Tools	Marker-1.8.2 [45]	-	71.30	0.206	76.66	57.88	71.17	0.250
	Mineru2-pipeline [14]	-	75.51	0.209	76.55	70.90	79.11	0.225
	PP-StructureV3 [10]	-	86.73	0.073	85.79	81.68	89.48	0.073
General VLMs	GPT-4o [7]	-	75.02	0.217	79.70	67.07	76.09	0.148
	InternVL3-76B [46]	76B	80.33	0.131	83.42	70.64	77.74	0.113
	InternVL3.5-241B [47]	241B	82.67	0.142	87.23	75.00	81.28	0.125
	Qwen2.5-VL-72B [24]	72B	87.02	0.094	88.27	82.15	86.22	0.102
	Gemini-2.5 Pro [48]	-	88.03	0.075	85.82	85.71	90.29	0.097
Specialized VLMs	Dolphin [3]	322M	74.67	0.125	67.85	68.70	77.77	0.124
	OCRFlux-3B [49]	3B	74.82	0.193	68.03	75.75	80.23	0.202
	Mistral OCR [50]	-	78.83	0.164	82.84	70.03	78.04	0.144
	POINTS-Reader [4]	3B	80.98	0.134	79.20	77.13	81.66	0.145
	olmOCR-7B [12]	7B	81.79	0.096	86.04	68.92	74.77	0.121
	MinerU2-VLM [14]	0.9B	85.56	0.078	80.95	83.54	87.66	0.086
	Nanonets-OCR-s [51]	3B	85.59	0.093	85.90	80.14	85.57	0.108
	MonkeyOCR-pro-1.2B [1]	1.9B	86.96	0.084	85.02	84.24	89.02	0.130
	MonkeyOCR-3B [1]	3.7B	87.13	0.075	87.45	81.39	85.92	0.129
	dots.ocr [52]	3B	88.41	0.048	83.22	86.78	90.62	0.053
	MonkeyOCR-pro-3B [1]	3.7B	88.85	0.075	87.25	86.78	90.63	0.128
	MinerU2.5 [2]	1.2B	90.67	0.047	88.46	88.22	92.38	0.044
	PaddleOCR-VL	0.9B	**92.56**	**0.035**	**91.43**	**89.76**	**93.52**	**0.043**

Table 2 | Comprehensive evaluation of document parsing on OmniDocBench v1.5. Results are reported by OmniDocBench [16] unless Ours.

OmniDocBench v1.0 A publicly available benchmark dataset specifically is designed to evaluate real-world document parsing capabilities. It comprises 981 PDF pages, spanning 9 distinct document types, 4 layout styles, and 3 language categories.

Based on the experimental results presented in Table 3, PaddleOCR-VL demonstrates superior performance with an average overall edit distance of 0.115, demonstrating its superior capability in document parsing. The model excels in formula edit distance (0.241 EN, 0.316 ZH), and achieves the SOTA performance (0.062) and a comparable SOTA performance (0.041) for Chinese and English text edit distance respectively, showcasing its accuracy in handling textual and formulaic data. Although the model exhibits slightly lower performance in the English Table TEDS (88.0), this can be largely attributed to typo-related annotation errors in OmniDocBench v1.0. Nevertheless, it demonstrates a clear advantage in the Chinese Table TEDS (92.14). Regarding the reading order edit distance, the model achieves the best performance in Chinese (0.063) and a comparable SOTA result in English (0.045), emphasizing its capability to maintain structural integrity and logical document flow.

Method Type	Methods	AvgOverall^{Edit}↓	Overall^{Edit}↓		Text^{Edit}↓		Formula^{Edit}↓		Table^{TEDS}↑		Table^{Edit}↓		Reading Order^{Edit}↓	
			EN	ZH	EN	ZH	EN	ZH	EN	ZH	EN	ZH	EN	ZH
Pipeline Tools	Docling-2.14.0 [11]	0.749	0.589	0.909	0.416	0.987	0.999	1	61.3	25.0	0.627	0.810	0.313	0.837
	OpenParse-0.7.0 [53]	0.730	0.646	0.814	0.681	0.974	0.996	1	64.8	27.5	0.284	0.639	0.595	0.641
	Unstructured-0.17.2 [54]	0.651	0.586	0.716	0.198	0.481	0.999	1	0	0.1	1	0.998	0.145	0.387
	Pix2Text-1.1.2.3 [55]	0.424	0.320	0.528	0.138	0.356	0.276	0.611	73.6	66.2	0.584	0.645	0.281	0.499
	Marker-1.7.1 [45]	0.397	0.296	0.497	0.085	0.293	0.374	0.688	67.6	54.0	0.609	0.678	0.116	0.329
	Mathpix [56]	0.278	0.191	0.364	0.105	0.381	0.306	0.454	77.0	67.1	0.243	0.320	0.108	0.304
	MinerU-pipeline [9]	0.203	0.162	0.244	0.072	0.111	0.313	0.581	77.4	79.5	0.166	0.150	0.097	0.136
	PP-StructureV3 [10]	0.176	0.145	0.206	0.058	0.088	0.295	0.535	77.2	83.9	0.159	0.109	0.069	0.091
General VLMs	InternVL2-76B [57]	0.442	0.440	0.443	0.353	0.290	0.543	0.701	63.0	60.2	0.547	0.555	0.317	0.228
	GPT-4o [7]	0.316	0.233	0.399	0.144	0.409	0.425	0.606	72.0	62.9	0.234	0.329	0.128	0.251
	InternVL3-78B [46]	0.257	0.218	0.296	0.117	0.210	0.380	0.533	69.0	73.9	0.279	0.282	0.095	0.161
	Qwen2.5-VL-72B [24]	0.238	0.214	0.261	0.092	0.180	0.315	0.434	81.4	83.0	0.341	0.262	0.106	0.168
	Gemini2.5-Pro [48]	0.180	0.148	0.212	0.055	0.168	0.356	0.439	85.8	86.4	0.130	0.119	0.049	0.121
Specialized VLMs	Nougat [58]	0.713	0.452	0.973	0.365	0.998	0.488	0.941	39.9	0.0	0.572	1	0.382	0.954
	SmolDocling-256M [13]	0.655	0.493	0.816	0.262	0.838	0.753	0.997	44.9	16.5	0.729	0.907	0.227	0.522
	olmOCR-7B [12]	0.398	0.326	0.469	0.097	0.293	0.455	0.655	68.1	61.3	0.608	0.652	0.145	0.277
	GOT [21]	0.349	0.287	0.411	0.189	0.315	0.360	0.528	53.2	47.2	0.459	0.520	0.141	0.280
	OCRFlux-3B [49]	0.294	0.238	0.349	0.112	0.256	0.447	0.716	69.0	80.0	0.269	0.162	0.126	0.263
	Nanonets-OCR-s [51]	0.289	0.283	0.295	0.134	0.231	0.518	0.546	76.8	79.4	0.343	0.201	0.135	0.200
	Dolphin [3]	0.259	0.205	0.313	0.092	0.204	0.447	0.606	76.1	66.9	0.193	0.282	0.088	0.160
	MinerU2-VLM [14]	0.186	0.133	0.238	0.045	0.115	0.273	0.506	82.1	83.4	0.150	0.209	0.066	0.122
	MonkeyOCR-pro-1.2B [1]	0.184	0.146	0.221	0.068	0.118	0.272	0.452	81.3	85.5	0.149	0.134	0.093	0.179
	MonkeyOCR-pro-3B [1]	0.172	0.138	0.206	0.067	0.107	0.246	0.421	81.5	87.5	0.139	0.111	0.100	0.185
	dots.ocr [52]	0.143	0.125	0.160	**0.032**	0.066	0.329	0.416	**88.6**	89.0	0.099	0.092	**0.040**	0.067
	MinerU2.5 [2]	0.143	0.111	0.174	0.050	0.074	0.258	0.473	88.3	89.2	**0.089**	0.083	0.045	0.068
	PaddleOCR-VL	**0.115**	**0.105**	**0.126**	0.041	**0.062**	**0.241**	**0.316**	88.0	**92.1**	0.093	**0.062**	0.045	**0.063**

Table 3 | Comprehensive evaluation of document parsing on OmniDocBench v1.0. Results are reported by OmniDocBench [16] unless MinerU2.5 and Ours.

olmOCR-Bench olmOCR-Bench [12] includes 1,402 PDF documents and 7,010 test cases, addressing diverse document types and extraction challenges. It offers a detailed evaluation framework for PDF content extraction by assessing tools and models through simple, clear, and machine-verifiable unit tests. This approach avoids biased evaluations and soft metric comparisons, allowing for the detection of subtle but significant extraction errors.

Table 4 highlights the outstanding performance of PaddleOCR-VL in the olmOCR-Bench evaluation, achieving the highest overall score of 80.0 ± 1.0. It excels in various categories, leading in ArXiv (85.7), Headers and Footers (97.0) and securing second place in Multi-column text (79.9), Long Tiny Text (85.7). These results highlight the proposed model's capability to effectively manage diverse document types, reinforcing its status as a top solution in document parsing and its reliability in complex OCR tasks.

Methods	Unit Test Pass Rate ↑								
	Overall	ArXiv	Old Scans Math	Tables	Old Scans	Headers and Footers	Multi column	Long Tiny Text	Base
GOT [21]	48.3 ± 1.1	52.7	52.0	0.2	22.1	93.6	42.0	29.9	94.0
Gemini Flash 2 (No Anchor) [48]	57.8 ± 1.1	32.1	56.3	61.4	27.8	48.0	58.7	84.4	94.0
MinerU-pipeline [9]	61.5 ± 1.1	75.4	47.4	60.9	17.3	96.6	59.0	39.1	96.6
Gemini Flash 2 (Anchored) [48]	63.8 ± 1.2	54.5	56.1	72.1	34.2	64.7	61.5	71.5	95.6
Nanonets-OCR-s [51]	64.5 ± 1.1	67.0	68.6	77.7	39.5	40.7	69.9	53.4	99.3
Qwen2.5-VL-7B (No Anchor) [24]	65.5 ± 1.2	63.1	65.7	67.3	38.6	73.6	68.3	49.1	98.3
GPT-4o (No Anchor) [7]	68.9 ± 1.1	51.5	**75.5**	69.1	40.9	94.2	68.9	54.1	96.7
GPT-4o (Anchored) [7]	69.9 ± 1.1	53.5	74.5	70.0	40.7	93.8	69.3	60.6	96.8
Marker-1.8.2 [45]	70.1 ± 1.1	76.0	57.9	57.6	27.8	84.9	72.9	84.6	99.1
olmOCR v0.1.75 (No Anchor) [12]	74.7 ± 1.1	71.5	71.4	71.4	**42.8**	94.1	77.7	71.0	97.8
olmOCR v0.1.75 (Anchored) [12]	75.5 ± 1.0	74.9	71.2	71.0	42.2	94.5	78.3	73.3	98.3
MonkeyOCR-pro-3B [1]	75.8 ± 1.0	83.8	68.8	74.6	36.1	91.2	76.6	80.1	95.3
MinerU2.5 [2]	77.5 ± 1.0	81.1	74.0	85.1	33.8	96.3	65.5	**89.8**	94.4
dots.ocr [52]	79.1 ± 1.0	82.1	64.2	**88.3**	40.9	94.1	**82.4**	81.2	**99.5**
PaddleOCR-VL	**80.0 ± 1.0**	**85.7**	71.0	84.1	37.8	**97.0**	79.9	85.7	98.5

Table 4 | Comprehensive evaluation of document parsing on olmOCR-Bench. Results are reported by olmOCR-Bench [12] unless MinerU2.5 and Ours.

4.2. Element-level Evaluation

This section centers on evaluating the element-level capabilities of PaddleOCR VL 0.9B. We thoroughly assessed four tasks: text, tables, formulas, and charts using both public competition data and in-house data.

4.2.1. *Text Recognition*

For text recognition, we utilize three benchmarks to validate the effectiveness of models based on the edit distance metric.

OmniDocBench-OCR-block: From the 1355 images of OmniDocBench v1.5, we extracted all text-related sub-images based on layout detection labels, removing any with null annotations. This process resulted in a total of 17,148 block-level images. This evaluation set is named OmniDocBench-OCR-block, with the ground truth still sourced from OmniDocBench. This evaluation set can more accurately assess the model's text recognition performance on without being affected by layout detection. We use the average normalized edit distance for evaluation.

In Table 5, we present a comprehensive comparison of performance across various document types using different models. Our model, PaddleOCR-VL, consistently demonstrates superior performance, achieving the lowest error rates in almost all categories. Specifically, PaddleOCR-VL achieves the best results in the PPT2PDF (0.049), Academic Literature (0.021), Book (0.045), Colorful Textbook (0.081), Exam Paper (0.115), Magazine (0.020), Newspaper (0.034), Note (0.081), and Research Report (0.033) categories. These results highlight PaddleOCR-VL's robust and versatile capability in handling diverse document types, establishing it as the leading method in the OmniDocBench-OCR-block performance evaluation.

Methods	Edit Distance ↓								
	PPT2PDF	Academic Literature	Book	Colorful Textbook	Exam Paper	Magazine	Newspaper	Note	Research Report
Qwen2.5-VL-72B [24]	0.054	0.023	0.061	0.084	0.195	0.032	0.056	0.118	0.040
MonkeyOCR-pro-3B [1]	0.058	**0.021**	0.064	0.096	0.116	0.023	0.058	0.124	0.052
MinerU2.5 [2]	0.195	0.089	0.111	0.234	0.194	0.147	0.056	0.142	0.094
Dolphin [3]	0.237	0.095	0.135	0.347	0.248	0.233	0.121	0.309	0.213
PaddleOCR-VL	**0.049**	**0.021**	**0.045**	**0.081**	**0.115**	**0.020**	**0.034**	**0.081**	**0.033**

Table 5 | Overall Comparison of OmniDocBench-OCR-block Performance.

In-house-OCR: This is our self-built line-level text evaluation dataset which contains 107452 samples with high-quality labels. The dataset includes various text types such as handwritten Chinese, handwritten English, printed Chinese, printed English, traditional Chinese, ancient texts, general scenarios, Pinyin, obscure characters, vertical text, single characters, emojis, and artistic fonts. It also comprises evaluation sets for 109 languages, such as Latin and Japanese.

Table 6 provides a detailed evaluation of performance across multiple languages and text types. In the Multilingual Metrics (Table 6a), the model demonstrates outstanding accuracy with the lowest edit distances in all evaluated scripts: Arabic(0.122), Korean(0.052), Tamil(0.043), Greek(0.135), Thai(0.081), Telugu (0.114), Devanagari (0.097), Cyrillic (0.109), Latin (0.013), and Japanese (0.096), indicating superior capability in handling diverse languages. Similarly, in the Text Type Metrics (Table 6b), it excels in various text types, achieving the lowest error rates in categories like Handwritten CN (0.089), Handwritten EN (0.042), Printed CN (0.035), Printed EN (0.016), Traditional Chinese (0.048), Ancient Texts(0.198), General Scene (0.067), Pinyin (0.113), Rare Characters (0.001), Vertical Text (0.005), Single Characters (0.027), Emoji (0.057), and Art Font (0.165). These impressive results underscore the model's robust performance and versatility, establishing it as the leading OCR solution in this benchmark comparison.

Methods	Edit Distance ↓									
	Arabic	Korean	Tamil	Greek	Thai	Telugu	Devanagari	Cyrillic	Latin	Japanese
Qwen2.5-VL-72B [24]	0.405	0.056	0.389	0.165	0.194	0.758	0.164	0.220	0.021	0.181
Dolphin [3]	0.682	0.699	0.912	0.691	0.709	0.832	0.818	0.549	0.037	0.309
MonkeyOCR-pro-3B [1]	0.601	0.182	0.921	0.449	0.876	0.909	0.896	0.387	0.036	0.262
MinerU2.5 [2]	0.978	0.917	0.957	0.661	0.880	0.937	0.915	0.832	0.063	0.588
PaddleOCR-VL	**0.122**	**0.052**	**0.043**	**0.135**	**0.081**	**0.011**	**0.097**	**0.109**	**0.013**	**0.086**

(a) Multilingual Metrics.

Methods	Edit Distance ↓												
	Hand-written CN	Hand-written EN	Printed CN	Printed EN	Trad. Chinese	Ancient Texts	General Scene	Pinyin	Rare Char.	Vertical Text	Single Char.	Emoji	Art Font
Dolphin [3]	0.236	0.145	0.074	0.025	0.095	0.218	0.113	0.183	0.092	0.190	0.202	0.225	0.230
MonkeyOCR-pro-3B [1]	0.253	0.071	0.048	0.023	0.295	0.529	0.144	0.165	0.063	0.086	0.110	0.184	0.263
Qwen2.5-VL-72B [24]	0.188	0.047	0.037	0.018	0.100	0.387	0.122	0.186	0.034	0.090	0.041	0.134	0.220
MinerU2.5 [2]	0.370	0.088	0.041	0.023	0.232	0.950	0.179	0.256	0.048	0.962	0.097	0.174	0.337
PaddleOCR-VL	**0.089**	**0.042**	**0.035**	**0.016**	**0.048**	**0.198**	**0.067**	**0.113**	**0.001**	**0.005**	**0.027**	**0.057**	**0.165**

(b) Text Type Metrics.

Table 6 | Comparison of In-house-OCR Edit Distance Performance.

Ocean-OCR-Handwritten: This is a line and paragraph levels handwritten evaluation dataset designed for comprehensive handwriting recognition assessment. It contains 400 samples, evenly divided into four subsets of 100 images each. The dataset covers both real and synthetic handwriting in Chinese and English. Real samples are collected from established handwriting datasets such as CASIA-HWDB [29], GNHK [59], and BRUSH [60], while synthetic samples are generated to simulate diverse writing styles, character densities, and layouts. The benchmark aims to provide balanced and fine-grained evaluation for handwritten text recognition across different scripts and writing conditions.

Table 7 presents a comparison of OCR performance for handwritten English and Chinese text on the Ocean-OCR-Bench. Our model demonstrates superior performance across all metrics in both languages. For English, it achieves the best edit distance of 0.118 and excels in F1-score, Precision, Recall, BLEU, and METEOR, establishing itself as the leading model. In Chinese, PaddleOCR-VL sets a benchmark with an edit distance of 0.034 and leads in all other metrics, showcasing its outstanding precision and reliability.

Methods	Edit Distance ↓		F1-score ↑		Precision↑		Recall↑		BLEU↑		METEOR↑	
	EN	ZH	EN	ZH	EN	ZH	EN	ZH	EN	ZH	EN	ZH
InternVL2.5-4B [57]	0.197	0.240	0.661	0.741	0.674	0.754	0.655	0.734	0.406	0.473	0.652	0.687
MiniCPM-V2.6-8B [61]	0.147	0.175	0.727	0.810	0.747	0.811	0.714	0.812	0.443	0.583	0.727	0.774
Qwen2-VL-7B [62]	0.127	0.113	0.760	0.881	0.773	0.884	0.754	0.884	0.490	0.666	0.756	0.859
GOT [21]	0.616	0.402	0.283	0.568	0.309	0.618	0.273	0.544	0.151	0.295	0.255	0.492
PaddleOCR [10]	0.418	0.325	0.237	0.664	0.232	0.646	0.263	0.700	0.069	0.431	0.236	0.648
TextIn	0.358	0.180	0.362	0.840	0.368	0.869	0.362	0.822	0.098	0.567	0.337	0.751
Ocean-OCR [63]	0.145	0.106	**0.774**	0.885	**0.780**	0.912	**0.782**	0.862	0.532	0.736	0.772	0.885
MinerU2.5 [2]	0.238	0.356	0.558	0.619	0.547	0.623	0.574	0.622	0.344	0.489	0.553	0.601
PaddleOCR-VL	**0.118**	**0.034**	0.750	**0.957**	0.748	**0.959**	0.753	**0.957**	**0.551**	**0.856**	**0.787**	**0.936**

Table 7 | Comparison of performance on English(EN) and Chinese(ZH) OCR for handwritten recognition on Ocean-OCR-Bench. Results are reported by Ocean-OCR [63] unless MinerU2.5 and Ours.

4.2.2. Table Recognition.

For table recognition, we utilize two benchmarks to validate the effectiveness of PaddleOCR-VL-0.9B based on TEDS [41] and Edit Distance.

OmniDocBench-Table-block: To evaluate the table recognition performance of PaddleOCR-VL, we crop 512 tables from OmniDocBench v1.5 datasets.

As shown in Table 8, our PaddleOCR-VL leads in the OmniDocBench-Table-block benchmark, surpassing all competitors. It achieves a top overall TEDS of 0.9195, reflecting high accuracy in capturing table structure and content. Its structural TEDS of 0.9543 highlights its ability to parse complex structures, while the lowest Overall Edit Distance of 0.0561 indicates minimal recognition errors. These results confirm PaddleOCR-VL's superior performance and establish it as the benchmark for accurate table recognition.

Methods	Overall TEDS↑	Structural TEDS↑	Overall Edit Dist↓
MinerU2-VLM [14]	0.9002	0.9369	0.0734
Seed1.6	0.9079	0.9489	0.0652
dots.ocr [52]	0.8194	0.8442	0.1508
MinerU2.5 [2]	0.9005	0.9539	0.0693
PaddleOCR-VL	**0.9195**	**0.9543**	**0.0561**

Table 8 | Comparison of OmniDocBench-Table-block Performance

In-house-Table: Our self-built evaluation set contains diverse array of table images with comprehensive annotations and type classifications. It includes 20 different table types such as Chinese, English, mixed Chinese-English, and tables with various characteristics like full, partial, or no borders. The collection also covers tables with formulas, dense data, book/manual formats, lists, academic papers, merged cells, as well as low-quality, watermarked, registration forms, statistical forms, research reports, financial reports, images, invoices, and handwritten tables, among others.

Table 9 provides a comparison of different methods on the In-house-Table task, highlighting their performance across various metrics. We achieves the highest scores in Overall TEDS (0.8699), Structural TEDS (0.9066), Overall Edit Distance (0.9066) and Structural Edit Distance (0.9339). These results underscore PaddleOCR-VL's effectiveness and reliability in table recognition tasks.

Methods	**Overall TEDS↑**	**Structural TEDS↑**	**Overall Edit Dist↑**	**Structural Edit Dist↑**
MinerU2-VLM [14]	0.8286	0.8730	0.8757	0.9088
MonkeyOCR [1]	0.7396	0.7824	0.8174	0.8537
Nanonets-OCR-s [51]	0.7824	0.8190	0.8377	0.8692
OCRFlux-3B [49]	0.7741	0.8071	0.8238	0.8617
Qwen2.5-VL-3B [24]	0.7398	0.7765	0.8132	0.8701
Qwen2.5-VL-7B [24]	0.7549	0.7926	0.8251	0.8819
Qwen2.5-VL-72B [24]	0.7762	0.8361	0.843	0.8987
dots.ocr [52]	0.7547	0.7914	0.8047	0.8361
MinerU2.5 [2]	0.8469	0.8955	0.8896	0.9239
PaddleOCR-VL	**0.8699**	**0.9066**	**0.9066**	**0.9339**

Table 9 | Comparison of In-house-Table Performance

4.2.3. *Formula Recognition.*

For formula recognition, we validate the effectiveness our model based on the Character Detection Matching (CDM) [64] metric on OmniDocBench-Formula-block and In-house-Formula datasets.

OmniDocBench-Formula-block Using the formula bounding boxes from OmniDocBench v1.5, 1050 formula sub-images were cropped. This step was taken to minimize the influence of layout detection on formula recognition. As shown in Table 10, the model achieved state-of-the-art CDM score of 0.9453.

Methods	**Overall CDM ↑**	**EN CDM ↑**	**ZH CDM ↑**
dots.ocr [52]	0.4641	0.4868	0.4414
MinerU2-VLM [14]	0.8286	0.9616	0.6956
MonkeyOCR-pro-1.2B [1]	0.8531	0.9642	0.7419
MonkeyOCR-3B [1]	0.8621	0.9718	0.7524
Qwen2.5-VL-72B [24]	0.8747	0.9574	0.7920
MinerU2.5 [2]	0.9187	**0.9751**	0.8623
PaddleOCR-VL	**0.9453**	0.9677	**0.9228**

Table 10 | Comparison of OmniDocBench v1.5 Formula-block Performance. Due to dots.ocr [52] easily recognizing cropped formulas as images, the score is relatively low.

In-house-Formula: The self-constructed formula evaluation set contains 34,816 samples, covering common formula recognition scenarios such as academic papers, mathematics books, and primary and secondary school exam papers. Among them, there are 498 Chinese formulas and 34,318 English formulas. As shown in Table 11, our model obtains the best performance of

0.9882 CDM score on the In-house-Formula dataset. These results collectively demonstrate the powerful recognition capability of PaddleOCR-VL in real-world complex formula scenarios.

Methods	Overall CDM ↑	EN CDM ↑	ZH CDM ↑
dots.ocr [52]	0.6737	0.8066	0.5408
MinerU2-VLM [14]	0.9237	0.9764	0.8709
MonkeyOCR-pro-1.2B [1]	0.9537	0.9656	0.9417
MonkeyOCR-3B [1]	0.9566	0.9761	0.9371
Qwen2.5-VL-72B [24]	0.9412	0.9519	0.9304
MinerU2.5 [2]	0.9770	0.9832	0.9708
PaddleOCR-VL	**0.9882**	**0.9914**	**0.9849**

Table 11 | Comparison of In-house-Formula Performance. Due to dots.ocr [52] easily recognizing cropped formulas as images, the score is relatively low.

4.2.4. *Chart Recognition.*

For chart recognition, considering the limitations in dataset size, the imbalanced distribution of chart categories, and the poor annotation quality of publicly available test sets, we only utilize a in-house benchmark to validate the effectiveness of PaddleOCR-VL-0.9B based on the RMS-F1 [42] score metric. As shown in Table 12, the proposed PaddleOCR-VL not only outperforms expert OCR VLMs but also surpasses some 72B-level multimodal language models.

In-house-Chart: This in-house chart recognition evaluation set comprises 1,801 samples, all of which have underwent a rigorous manual review to ensure annotation correctness. The evaluation set is broadly categorized into 11 chart categories, including bar-line hybrid, pie, 100% stacked bar, area, bar, bubble, histogram, line, scatterplot, stacked area, and stacked bar. Of these samples, 851 are in English and 950 are in Chinese. Prior to evaluation, both the predicted data table and the ground truth data table are normalized to a uniform markdown format to eliminate expression ambiguities.

Methods	RMS-F1 ↑		
	Overall	EN	ZH
TinyChart [65]	0.2159	0.4726	0.0876
GOT [21]	0.3160	0.1100	0.4190
OneChart [66]	0.3716	0.1384	0.4882
Qwen2.5-VL-3B [24]	0.5942	0.5619	0.6103
Qwen2.5-VL-7B [24]	0.6821	0.5876	0.7293
Qwen2.5-VL-72B [24]	0.7300	0.6972	0.7464
PP-StructureV3 [10]	0.8060	0.7963	0.8109
PaddleOCR-VL	**0.8440**	**0.8222**	**0.8549**

Table 12 | Comparison of In-house-Chart Performance

4.3. Inference Performance

To improve the inference performance of PaddleOCR-VL, we introduce multi-threading asynchronous execution into the inference workflow. The process is divided into three main stages—data loading (e.g., rendering PDF pages as images), layout model processing, and VLM inference—each running in a separate thread. Data is transferred between adjacent stages via queues, enabling concurrent execution for higher efficiency. In particular, for VLM inference, batch processing is only triggered when either the number of items in the queue reaches a predefined threshold or the waiting time for queued data exceeds a specified limit. This design allows blocks across different pages to be aggregated and processed together, thereby maximizing parallelism, especially when handling large volumes of files. We further deploy PaddleOCR-VL-0.9B on high-throughput inference and serving engines [67, 68, 69], tuning parameters like max-num-batched-tokens and gpu-memory-utilization to balance inference throughput with GPU memory consumption.

We measured the end-to-end inference speed and GPU usage on the OmniDocBench v1.0 dataset, processing PDF files in batches of 512 on a single NVIDIA A100 GPU. By "end-to-end", we mean that the inference time was measured from providing the PDF file path as input to the complete generation of the Markdown text. For MonkeyOCR, dots.ocr, and MinerU, inference was run with the vLLM backend and the default configuration (including the KV cache settings). The generated Markdown text was tokenized with the "cl100k_base" tokenizer to compute the number of output tokens. For dots.ocr specifically, 200 threads were used for concurrent page processing, and the Base64-encoded image content in the produced Markdown text was replaced with a dummy path (UUID-based, prefixed with "images/" and suffixed with ".png") to ensure a reasonable token count.

Table 13 provides a comprehensive comparison of inference efficiency across different methods. The proposed PaddleOCR-VL demonstrates clear and consistent advantages in both processing speed and memory efficiency. When deployed with the vLLM backend, it achieves 15.8% higher page throughput and 14.2% higher token throughput than the leading baseline, MinerU2.5, establishing itself as the most efficient solution overall. In addition, PaddleOCR-VL achieves notable memory savings, using roughly 40% less GPU memory than dots.ocr while sustaining significantly faster processing. These results collectively confirm that PaddleOCR-VL attains state-of-the-art inference efficiency through a balanced optimization of speed and memory usage, making it highly suitable for real-world, high-throughput document understanding scenarios.

Methods	Total Time (s)↓	Pages/s↑	Tokens/s↑	Avg. VRAM Usage (GB)↓
MonkeyOCR-pro-1.2B† [1]	1456.4	0.6730	1120.3	75.5
dots.ocr† [52]	2784.6	0.3522	532.9	78.5
MinerU2.5† [2]	927.3	1.0574	<u>1647.9</u>	**41.9**
PaddleOCR-VL†	**800.9**	**1.2241**	**1881.2**	<u>43.7</u>
PaddleOCR-VL‡	<u>917.6</u>	<u>1.0684</u>	1641.5	49.8

Table 13 | End-to-End Inference Performance Comparison. † denotes the vLLM backend, and ‡ denotes the SGLang backend.

5. Conclusion

This report introduces PaddleOCR-VL, an advanced and efficient model for document parsing that excels at both element-level and page-level recognition. Its core componets, PaddleOCR-VL-0.9B, built with a NaViT-style visual encoder and ERNIE-4.5-0.3B language model, it accurately recognizes complex elements such as text, tables, formulas, and charts in over 100 languages. PaddleOCR-VL achieves fast inference and low resource consumption, making it practical for real-world deployment. It outperforms existing pipeline solutions on many benchmarks and effectively handles challenging content including handwriting and historical documents, as well as converting chart visuals into structured data. Its broad multilingual support and strong performance have the potential to advance the application and development of multimodal document processing technologies, bringing innovation to automated analysis and information retrieval. This will significantly enhance the performance and stability of RAG systems, making information extraction from complex documents more efficient, thereby providing more reliable data support for future AI applications.

References

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

[2] Junbo Niu, Zheng Liu, Zhuangcheng Gu, Bin Wang, Linke Ouyang, Zhiyuan Zhao, Tao Chu, Tianyao He, Fan Wu, Qintong Zhang, et al. Mineru2. 5: A decoupled vision-language model for efficient high-resolution document parsing. arXiv preprint arXiv:2509.22186, 2025.

[3] Hao Feng, Shu Wei, Xiang Fei, Wei Shi, Yingdong Han, Lei Liao, Jinghui Lu, Binghong Wu, Qi Liu, Chunhui Lin, et al. Dolphin: Document image parsing via heterogeneous anchor prompting. arXiv preprint arXiv:2505.14059, 2025.

[4] Yuan Liu, Zhongyin Zhao, Le Tian, Haicheng Wang, Xubing Ye, Yangxiu You, Zilin Yu, Chuhan Wu, Xiao Zhou, Yang Yu, et al. Points-reader: Distillation-free adaptation of vision-language models for document conversion. arXiv preprint arXiv:2509.01215, 2025.

[5] Baidu-ERNIE-Team. Ernie 4.5 technical report, 2025.

[6] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. arXiv preprint arXiv:2505.09388, 2025.

[7] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. arXiv preprint arXiv:2303.08774, 2023.

[8] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in neural information processing systems, 33:9459–9474, 2020.

[9] Bin Wang, Chao Xu, Xiaomeng Zhao, Linke Ouyang, Fan Wu, Zhiyuan Zhao, Rui Xu, Kaiwen Liu, Yuan Qu, Fukai Shang, et al. Mineru: An open-source solution for precise document content extraction. arXiv preprint arXiv:2409.18839, 2024.

[10] Cheng Cui, Ting Sun, Manhui Lin, Tingquan Gao, Yubo Zhang, Jiaxuan Liu, Xueqing Wang, Zelun Zhang, Changda Zhou, Hongen Liu, et al. Paddleocr 3.0 technical report. arXiv preprint arXiv:2507.05595, 2025.

[11] Docling Team. Docling. https://github.com/docling-project/docling, 2024. Accessed: 2025-06-23.

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

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

[14] opendatalab. Mineru2.0-2505-0.9b. https://huggingface.co/opendatalab/MinerU2.0-2505-0.9B, 2025.

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

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

[17] Yian Zhao, Wenyu Lv, Shangliang Xu, Jinman Wei, Guanzhong Wang, Qingqing Dang, Yi Liu, and Jie Chen. Detrs beat yolos on real-time object detection. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 16965–16974, 2024.

[18] Xiuquan Hou, Meiqin Liu, Senlin Zhang, Ping Wei, Badong Chen, and Xuguang Lan. Relation detr: Exploring explicit position relation prior for object detection. In European Conference on Computer Vision, pages 89–105. Springer, 2024.

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

[20] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. Advances in neural information processing systems, 36:34892–34916, 2023.

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

[22] Kwai Keye Team, Biao Yang, Bin Wen, Changyi Liu, Chenglong Chu, Chengru Song, Chongling Rao, Chuan Yi, Da Li, Dunju Zang, et al. Kwai keye-vl technical report. arXiv preprint arXiv:2507.01949, 2025.

[23] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). arXiv preprint arXiv:1606.08415, 2016.

[24] Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, et al. Qwen2. 5-vl technical report. arXiv preprint arXiv:2502.13923, 2025.

[25] Ting Sun, Cheng Cui, Yuning Du, and Yi Liu. Pp-doclayout: A unified document layout detection model to accelerate large-scale data construction. arXiv preprint arXiv:2503.17213, 2025.

[26] Zhilu Zhang and Mert Sabuncu. Generalized cross entropy loss for training deep neural networks with noisy labels. Advances in neural information processing systems, 31, 2018.

[27] PaddlePaddle Authors. Erniekit. https://github.com/PaddlePaddle/ERNIE, 2025.

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

[29] Cheng-Lin Liu, Fei Yin, Da-Han Wang, and Qiu-Feng Wang. Casia online and offline chinese handwriting databases. In 2011 international conference on document analysis and recognition, pages 37–41. IEEE, 2011.

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

[31] Philippe Gervais, Anastasiia Fadeeva, and Andrii Maksai. Mathwriting: A dataset for handwritten mathematical expression recognition. In Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2, pages 5459–5469, 2025.

[32] Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. arXiv preprint arXiv:2203.10244, 2022.

[33] Nitesh Methani, Pritha Ganguly, Mitesh M Khapra, and Pratyush Kumar. Plotqa: Reasoning over scientific plots. In Proceedings of the ieee/cvf winter conference on applications of computer vision, pages 1527–1536, 2020.

[34] Shankar Kantharaj, Rixie Tiffany Ko Leong, Xiang Lin, Ahmed Masry, Megh Thakkar, Enamul Hoque, and Shafiq Joty. Chart-to-text: A large-scale benchmark for chart summarization. arXiv preprint arXiv:2203.06486, 2022.

[35] Kushal Kafle, Brian Price, Scott Cohen, and Christopher Kanan. Dvqa: Understanding data visualizations via question answering. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 5648–5656, 2018.

[36] Ahmed Masry, Parsa Kavehzadeh, Xuan Long Do, Enamul Hoque, and Shafiq Joty. Unichart: A universal vision-language pretrained model for chart comprehension and reasoning. arXiv preprint arXiv:2305.14761, 2023.

[37] Leilani Battle, Peitong Duan, Zachery Miranda, Dana Mukusheva, Remco Chang, and Michael Stonebraker. Beagle: Automated extraction and interpretation of visualizations from the web. In Proceedings of the 2018 CHI conference on human factors in computing systems, pages 1–8, 2018.

[38] Kenny Davila, Bhargava Urala Kota, Srirangaraj Setlur, Venu Govindaraju, Christopher Tensmeyer, Sumit Shekhar, and Ritwick Chaudhry. Icdar 2019 competition on harvesting raw tables from infographics (chart-infographics). In 2019 International Conference on Document Analysis and Recognition (ICDAR), pages 1594–1599. IEEE, 2019.

[39] Benny J Tang, Angie Boggust, and Arvind Satyanarayan. Vistext: A benchmark for semantically rich chart captioning. arXiv preprint arXiv:2307.05356, 2023.

[40] Junyu Luo, Zekun Li, Jinpeng Wang, and Chin-Yew Lin. Chartocr: Data extraction from charts images via a deep hybrid framework. In Proceedings of the IEEE/CVF winter conference on applications of computer vision, pages 1917–1925, 2021.

[41] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. Image-based table recognition: data, model, and evaluation, 2020.

[42] Fangyu Liu, Julian Martin Eisenschlos, Francesco Piccinno, Syrine Krichene, Chenxi Pang, Kenton Lee, Mandar Joshi, Wenhu Chen, Nigel Collier, and Yasemin Altun. Deplot: One-shot visual language reasoning by plot-to-table translation. arXiv preprint arXiv:2212.10505, 2022.

[43] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In Proceedings of the 40th annual meeting of the Association for Computational Linguistics, pages 311–318, 2002.

[44] VI Lcvenshtcin. Binary coors capable or 'correcting deletions, insertions, and reversals. In Soviet physics-doklady, volume 10, 1966.

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

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

[47] Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265, 2025.

[48] Google DeepMind. Gemini 2.5. https://blog.google/technology/google-deepmind/gemini-model-thinking-updates-march-2025/, 2025.

[49] chatdoc com. Ocrflux. https://github.com/chatdoc-com/OCRFlux, 2025. Accessed:2025-09-25.

[50] Mistral AI Team. Mistral-ocr. https://mistral.ai/news/mistral-ocr?utm_source=ai-bot.cn, 2025.

[51] Souvik Mandal, Ashish Talewar, Paras Ahuja, and Prathamesh Juvatkar. Nanonets-ocr-s: A model for transforming documents into structured markdown with intelligent content recognition and semantic tagging, 2025.

[52] rednote-hilab. dots.ocr: Multilingual document layout parsing in a single vision-language model, 2025.

[53] Filimoa. open-parse. https://github.com/Filimoa/open-parse, 2024. Accessed: 2025-06-23.

[54] Unstructured-IO. unstructured. https://github.com/Unstructured-IO/unstructured, 2022. Accessed: 2025-06-23.

[55] breezedeus. Pix2text. https://github.com/breezedeus/Pix2Text, 2022. Accessed: 2025-06-23.

[56] Mathpix. Mathpix snip: Convert images and pdfs to latex, docx, and more. https://mathpix.com/, 2025.

[57] Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 24185–24198, 2024.

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

[59] Alex WC Lee, Jonathan Chung, and Marco Lee. Gnhk: a dataset for english handwriting in the wild. In International Conference on Document Analysis and Recognition, pages 399–412. Springer, 2021.

[60] Atsunobu Kotani, Stefanie Tellex, and James Tompkin. Generating handwriting via decoupled style descriptors. In European Conference on Computer Vision, pages 764–780. Springer, 2020.

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

[62] Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. arXiv preprint arXiv:2409.12191, 2024.

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

[64] Bin Wang, Fan Wu, Linke Ouyang, Zhuangcheng Gu, Rui Zhang, Renqiu Xia, Botian Shi, Bo Zhang, and Conghui He. Image over text: Transforming formula recognition evaluation with character detection matching. In 2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 19681–19690, 2025.

[65] Liang Zhang, Anwen Hu, Haiyang Xu, Ming Yan, Yichen Xu, Qin Jin, Ji Zhang, and Fei Huang. Tinychart: Efficient chart understanding with visual token merging and program-of-thoughts learning. arXiv preprint arXiv:2404.16635, 2024.

[66] Jinyue Chen, Lingyu Kong, Haoran Wei, Chenglong Liu, Zheng Ge, Liang Zhao, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. Onechart: Purify the chart structural extraction via one auxiliary token. In Proceedings of the 32nd ACM International Conference on Multimedia, pages 147–155, 2024.

[67] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In Proceedings of the 29th symposium on operating systems principles, pages 611–626, 2023.

[68] Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Livia Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E Gonzalez, et al. Sglang: Efficient execution of structured language model programs. Advances in neural information processing systems, 37:62557–62583, 2024.

[69] PaddlePaddle Authors. Fastdeploy. https://github.com/PaddlePaddle/FastDeploy, 2025.

[70] Qian Chen, Xianyin Zhang, Lifan Guo, Feng Chen, and Chi Zhang. Dianjin-ocr-r1: Enhancing ocr capabilities via a reasoning-and-tool interleaved vision-language model. arXiv preprint arXiv:2508.13238, 2025.

[71] Lukas Blecher. pix2tex - latex ocr. https://github.com/lukas-blecher/LaTeX-OCR, 2022. Accessed: 2025-06-23.

[72] Harold Mouchere, Christian Viard-Gaudin, Richard Zanibbi, and Utpal Garain. Icfhr 2014 competition on recognition of on-line handwritten mathematical expressions (crohme 2014). In 2014 14th international conference on frontiers in handwriting recognition, pages 791–796. IEEE, 2014.

[73] Harold Mouchère, Christian Viard-Gaudin, Richard Zanibbi, and Utpal Garain. Icfhr2016 crohme: Competition on recognition of online handwritten mathematical expressions. In 2016 15th International Conference on Frontiers in Handwriting Recognition (ICFHR), pages 607–612. IEEE, 2016.

[74] Mahshad Mahdavi, Richard Zanibbi, Harold Mouchere, Christian Viard-Gaudin, and Utpal Garain. Icdar 2019 crohme+ tfd: Competition on recognition of handwritten mathematical expressions and typeset formula detection. In 2019 International Conference on Document Analysis and Recognition (ICDAR), pages 1533–1538. IEEE, 2019.

[75] Ye Yuan, Xiao Liu, Wondimu Dikubab, Hui Liu, Zhilong Ji, Zhongqin Wu, and Xiang Bai. Syntax-aware network for handwritten mathematical expression recognition. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pages 4553–4562, 2022.

[76] Tao Ge, Xin Chan, Xiaoyang Wang, Dian Yu, Haitao Mi, and Dong Yu. Scaling synthetic data creation with 1,000,000,000 personas. arXiv preprint arXiv:2406.20094, 2024.

Appendix

A. Training Dataset Details

This two-stage approach offers unique advantages in terms of data collection, as obtaining isolated element imagesalong with their annotations is more feasible than collecting complete document pages containing different elements. In the following sections, we will elaborate on the construction of multimodal model training data for text, tables, formulas, and charts.

A.1. Text

We have curated a large-scale dataset comprising 20 Million High-Quality Image-Text Pairs. As shown in Figure A1, the dataset generation follows a rigorous multi-stage pipeline which primarily involves:

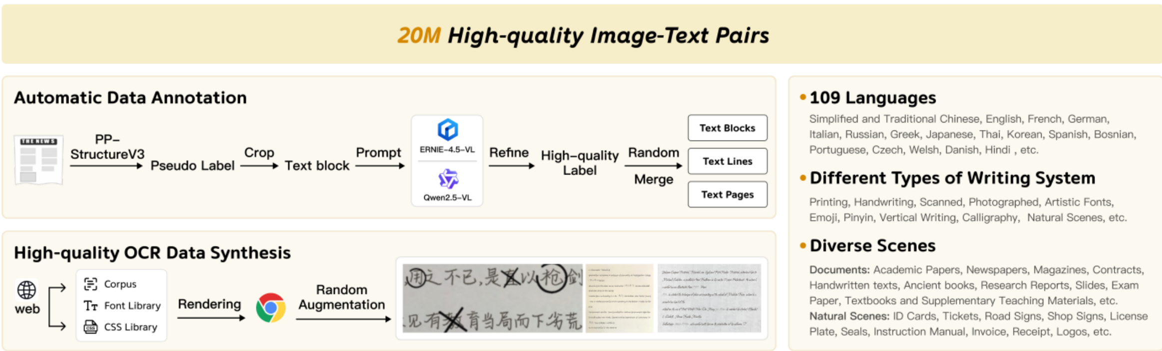


Figure A1 | The construction method and characteristics of the text training data for PaddleOCR-VL-0.9B.

1. **Automatic Data Annotation:** We design an automatic annotation pipeline that integrates lightweight document-structure models with large multimodal language models. Specifically, PP-StructureV3 is employed as an expert model to perform layout analysis and text recognition, generating pseudo labels that are converted into prompts for multimodal models such as ERNIE-4.5-VL and Qwen2.5-VL to refine. Finally, the refined labels are aggregated and randomly merged at multiple granularities to produce 20 million high-quality image–text training samples.
2. **High-quality OCR Data Synthesis:** During data distillation, low label quality in challenging scenarios like messy handwriting and dense blurry text was addressed by expanding the dataset through synthetic generation. Utilizing diverse CSS styles, over 200 fonts, and various corpora, we rendered a large amount of images, thereby enhancing the model's capabilities in these difficult scenarios.

Ultimately, the data is meticulously annotated at three distinct hierarchical levels: text lines, text blocks, and text pages. With extensive language coverage of 109 languages, including major global ones like Chinese, English, French, and Hindi. It includes diverse scenes including Academic Papers, Newspapers, Handwritten texts, Ancient books, Id cards, tickets, seals, etc. Additionally, the dataset addresses compatibility with a variety of writing systems and text styles, covering Printing, Handwriting, Scanned text, Artistic Fonts, etc.

A.2. Table

As shown in Figure A2, we constructed a large-scale dataset of over 5 million high-quality image-table pairs. Our dataset construction employs three key strategies: automatic data annotation, potential annotation mining, and high-quality data synthesis. For coding efficiency, we adopt OTSL [28] as the model's target format instead of conventional HTML. The main dataset construction process is as follows:

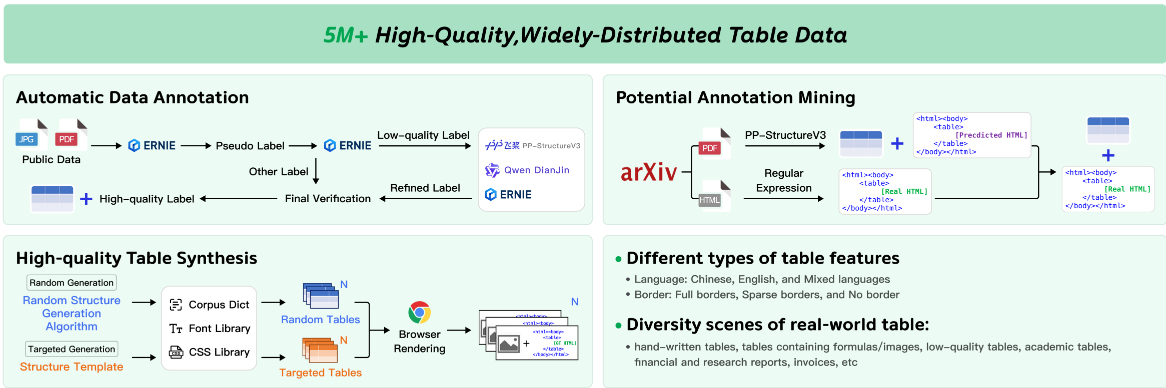


Figure A2 | The construction method and characteristics of the table training data for PaddleOCR-VL-0.9B.

1. **Automatic Data Annotation:** To enhance the performance of PaddleOCR-VL in table recognition, we built a large-scale, diverse dataset covering various languages, border styles, and table types. Tables are first located using PP-StructureV3 [10]. For unlabeled images, we employed a multi-stage annotation pipeline: ERNIE-4.5-VL [5] first generates pseudo-labels, which are then validated by a ERNIE-4.5-VL-28B-A3B [5] as discriminative model. Rejected annotations are refined using DianJin-OCR-R1 [70] (for tools, we use ERNIE-4.5-VL and PP-StructureV3 [10]). Finally, all annotations undergo rigorous rule-based verification, including n-gram analysis and HTML validation, to ensure only high-quality samples are used for training.
2. **Potential Annotation Mining:**
For public data with potential annotations (e.g., from arXiv), we extract tables and their corresponding official-supported HTML source code. We then employ a mechanism combining regular expression matching with contextual and sequential alignment to construct accurate table-HTML pairs. The extracted HTML subsequently undergoes rule-based filtering, yielding high-quality data samples ready for model training.
3. **High-quality Table Synthesis:**
To overcome data imbalance and high annotation costs, we introduce an innovative high-quality table synthesis tool which constitutes the cornerstone of our table data collection pipeline. This tool enables both randomized synthesis for comprehensive data supplement and targeted synthesis to enhance recognition of specific table categories. Specifically, we first leverage LLMs to gather a diverse and extensive corpus.Then, our tool generates table training pairs through randomized configurations of structures, fonts, CSS styles, and textual content, while also supporting customized synthesis by specifying particular parameters to accurately simulate specialized table types. With a synthesis speed of 10,000 samples per hour, our tool has produced over 5,500,000 training instances, substantially enhancing our model's generalization capability and comprehensive performance in table

recognition.

Through the aforementioned data construction strategies, we build a comprehensive table dataset encompassing diverse table categories and recognition scenarios, thereby providing robust support for training our model in the table recognition task.

A.3. Formula

As shown in Figure A3, this dataset was developed using a range of strategies, including source code rendering, automatic data annotation, targeted synthesis of long-tail data, and public data collection. It encompasses a variety of formula scenarios, such as educational supplementary materials, test papers for primary and secondary schools, mathematical papers, PowerPoint courseware, university theses, financial research reports, and handwritten mathematical notes. The dataset features four types of formulas: Simple Printed Expressions, Complex Printed Expressions, Screen-Captured Expressions, and Handwritten Expressions, available in both Chinese and English. The main process for constructing the dataset is as follows:

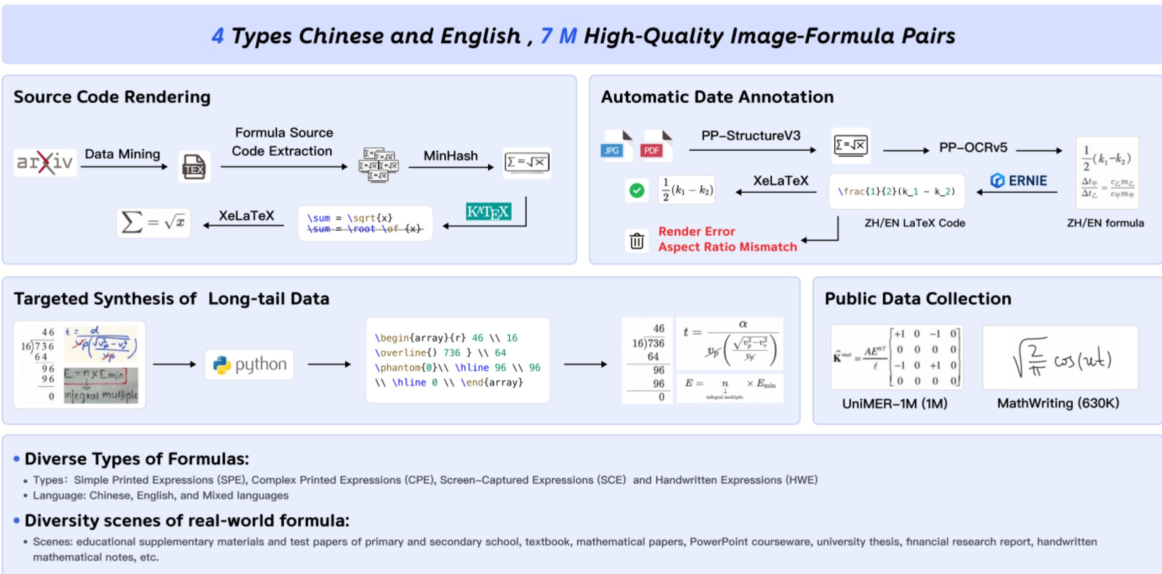


Figure A3 | The construction method and characteristics of the formula training data for PaddleOCR-VL-0.9B.

1. **Source Code Rendering:** To enhance the model's adaptability to a wide variety of unusual formula structures, a large amount of paper source code was scraped from arXiv, and LaTeX code for the formulas was extracted using regular expressions. Then, MinHash was used to remove duplicate and highly similar formula source codes, and KaTeX was employed to normalize the formula source codes, thereby reducing their ambiguity. Finally, the formulas were re-rendered into images using a formula rendering engine.
2. **Automatic Data Annotation:** For real-world formula data from exam papers, educational materials, and handwritten notes, the process begins with the use of the layout analysis method PP-StructureV3 [10] to identify the bounding boxes for formulas. Based on these bounding boxes, formula regions are cropped from the images. Subsequently, large multimodal language models, such as ERNIE-4.5-VL-28B-A3B [5], are employed to

generate the LaTeX source code for these formulas. Given the rarity of Chinese formulas in real-world scenarios—where approximately 1 out of 100 formulas contains Chinese characters—PP-OCRv5 [10] is utilized to recognize characters within the cropped regions, enabling targeted optimization when Chinese characters are detected. Due to the complex and diverse nature of real-world formulas, recognition errors may occur with existing large models. To address this, a LaTeX rendering engine is used to filter the formulas generated by these models. Specifically, image-formula pairs that cannot be successfully rendered by xelatex are discarded. For those that render successfully, a more in-depth screening is conducted by comparing metrics such as the aspect ratio between the recognized image and the rendered image.

3. **Targeted Synthesis of Long-tail Data:** For certain long-tail formula structures, such as elementary school vertical calculations, formulas with strikethroughs, and handwritten formulas with explanatory arrows, existing multimodal large models struggle to accurately recognize them due to data distribution issues. To address this, LaTeX code is synthetically generated based on rules and inverse rendering is performed using a LaTeX rendering engine, thereby constructing image-formula matching pairs for these long-tail scenarios.
4. **Public Data Collection:** In order to enable the model to learn high-quality formula representations, a substantial amount of data has been collected from existing public datasets, including UniMER-1M [30] and MathWriting [31]. Specifically, UniMER-1M is oriented towards real document scenarios and has gathered 1 million formula data from arXiv, Pix2tex [71], CROHME [72, 73, 74], and HME100K [75]. On the other hand, MathWriting is currently the largest handwritten mathematical formula dataset, comprising 230,000 real handwritten formula samples and 400,000 synthetic handwritten formula samples.

A.4. Chart

We constructed a large-scale, bilingual (Chinese and English) dataset of over 0.8 million high-quality image-chart pairs. Our dataset construction employs four key strategies: public data collection and cleaning, automatic data annotation, data synthesis, and targeted long-tail data augmentation. The dataset covers a wide array of chart types from diverse sources, including academic papers, financial reports, and web pages. The main dataset construction process is as follows:

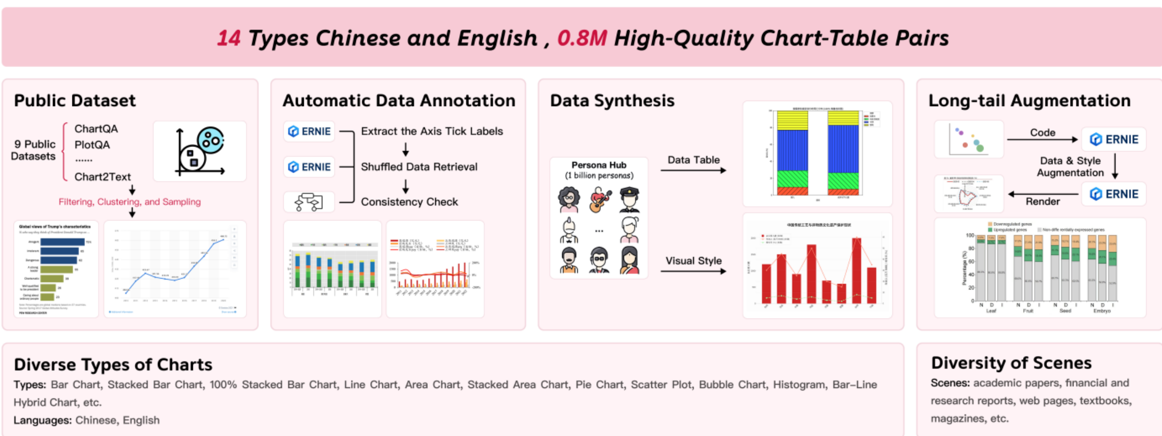


Figure A4 | The construction method and characteristics of the chart training data for PaddleOCR-VL-0.9B.

1. **Public Data Collection and Cleaning:** We collected a large number of samples from public datasets, including ChartQA [32], PlotQA [33], Chart2Text [34], DVQA [35], Unichart [36], Beagle [37], ChartINFO [38], visText [39], and ExcelChart [40]. However, the raw datasets suffered from poor annotation quality and extremely imbalanced data distributions. Thus, a meticulous data cleaning and filtering pipeline was implemented to remove noisy samples and ensure balanced clustering, resulting in a high-quality dataset of 220k samples.
2. **Automatic Data Annotation:** To annotate our large collection of unlabeled public and in-house data, we developed a two-stage annotation pipeline based on the Vision Large Language Model ERNIE-4.5-VL [5]. In the first stage, the model extracts tick labels from the x- and y-axes; in the second, random permutations of these labels are used to query corresponding data points, framing annotation as a data retrieval task. A final consistency check ensures that only verified annotations are included in the training set, guaranteeing high reliability.
3. **Data Synthesis:** To capture diverse visual styles and enhance model generalization, we designed a three-stage data synthesis pipeline. It begins with a large collection of base data tables, followed by an LLM Persona [76] strategy using ERNIE-X1 [5], which diversifies table content and generates persona-specific rendering code. This enables control over chart aesthetics such as color, font, and layout. Leveraging a billion distinct personas, the pipeline produces highly varied data structures and visual styles, substantially improving PaddleOCR-VL's generalization across real-world charts. For rendering, we employ matplotlib and seaborn.
4. **Targeted Long-tail Data Augmentation:** To improve generalization on real-world long-tail samples, we designed a data augmentation pipeline based on seed charts. It first selects long-tail samples by their distinctive visual features, then uses ERNIE-4.5-VL [5] to replicate their rendering code. ERNIE-X1 [5], guided by a specific persona [76], further diversifies the code by altering data tables and visual styles. Executing the modified code produces new augmented charts with corresponding data tables.

Through the four data construction strategies mentioned above, the final chart dataset covers a wide range of application scenarios and a rich variety of chart styles, providing strong support for the training of chart models.

B. Supported Languages

PaddleOCR-VL supports a total of 109 languages. Table 6 in the main text shows the text line recognition accuracy for different languages. Table A1 lists the correspondence between each language category and the specific supported languages.

Language Category	Specific Languages
Chinese	Chinese
English	English
Korean	Korean
Japanese	Japanese
Thai	Thai
Greek	Greek
Tamil	Tamil
Telugu	Telugu
Arabic	Arabic, Persian, Uyghur, Urdu, Pashto, Kurdish, Sindhi, Balochi
Latin	French, German, Afrikaans, Italian, Spanish, Bosnian, Portuguese, Czech, Welsh, Danish, Estonian, Irish, Croatian, Uzbek, Hungarian, Serbian (Latin), Indonesian, Occitan, Icelandic, Lithuanian, Maori, Malay, Dutch, Norwegian, Polish, Slovak, Slovenian, Albanian, Swedish, Swahili, Tagalog, Turkish, Latin, Azerbaijani, Kurdish, Latvian, Maltese, Pali, Romanian, Vietnamese, Finnish, Basque, Galician, Luxembourgish, Romansh, Catalan, Quechua
Cyrillic	Russian, Belarusian, Ukrainian, Serbian (Cyrillic), Bulgarian, Mongolian, Abkhazian, Adyghe, Kabardian, Avar, Dargin, Ingush, Chechen, Lak, Lezgin, Tabasaran, Kazakh, Kyrgyz, Tajik, Macedonian, Tatar, Chuvash, Bashkir, Malian, Moldovan, Udmurt, Komi, Ossetian, Buryat, Kalmyk, Tuvan, Sakha, Karakalpak
Devanagari	Hindi, Marathi, Nepali, Bihari, Maithili, Angika, Bhojpuri, Magahi, Santali, Newari, Konkani, Sanskrit, Haryanvi

Table A1 | Supported Languages

C. Inference Performance on Different Hardware Configurations

We measured the inference performance of PaddleOCR-VL on different hardware configurations, as summarized in Table A2. As observed, PaddleOCR-VL demonstrates stable and efficient inference performance across a wide range of hardware and backend configurations, showing that the system can flexibly adapt to diverse computing environments. Moreover, we are currently integrating the FastDeploy backend, which is expected to further enhance inference efficiency in future releases.

Hardware	Backend	Total Time (s)↓	Pages/s↑	Tokens/s↑	Avg. VRAM Usage (GB)↓
A100	vLLM	800.9	1.2241	1881.2	43.7
	SGLang	917.6	1.0684	1641.5	49.8
A10	vLLM	1238.0	0.7921	1217.2	14.1
	SGLang	1429.9	0.6858	1055.8	20.0
RTX 3060	vLLM	2749.1	0.3568	548.2	11.9
	SGLang	2792.4	0.3513	540.8	11.8
RTX 5070	vLLM	1292.9	0.7584	1165.5	8.9
RTX 4090D	vLLM	845.3	1.1597	1781.8	16.7
	SGLang	951.8	1.0303	1586.1	21.8

Table A2 | End-to-End Inference Performance

D. Real-world Samples

This appendix showcases the parsing and recognition capabilities of our proposed algorithm across a variety of challenging scenarios.

Section D.1 demonstrates the overall document parsing capability of PaddleOCR-VL. Figures A5-A8 are examples of parsing different types of documents in Markdown format.

Figures A9-A11 in section D.2 illustrate the superior ability of PaddleOCR-VL to process pages featuring intricate or challenging layouts.

Figures A12 and A13 in section D.3 demonstrate that PaddleOCR-VL maintains excellent reading order when faced with complex layouts, such as those found in various reports, textbooks, newspapers, magazines, and even vertical documents.

Section D.4 highlights the robust text recognition performance of PaddleOCR-VL in challenging cases, including multilingual text, handwriting text, and vertical text, which are presented in Figures A14-A22.

The model's table recognition abilities are demonstrated in section D.5. Figures A23 and A24 showcase its robust handling of a wide array of table formats, including tables from academic papers, tables from financial reports, tables with watermark, tables with image, tables with formulas and photograph of tables.

Figures in section D.6 detail the formula recognition performance. Figure A25 demonstrates the ability to handle various types of english formulas including complex printed expressions, handwritten expressions screen-captured expressions and vertical formula, while Figure A26 focuses on the ability to handle formulas that contain Chinese characters.

In section D.7, PaddleOCR-VL demonstrates impressive chart recognition capabilities, a feature currently lacking in many expert OCR VLMs like MinerU2.5 [14], dots.ocr [52] or MonkeyOCR [1]. Figures A27-A29 showcase our ability to parse various chart types, including pie charts, bar charts, line charts, bar-line hybrid charts and heatmap.

D.1. Comprehensive Document Parsing

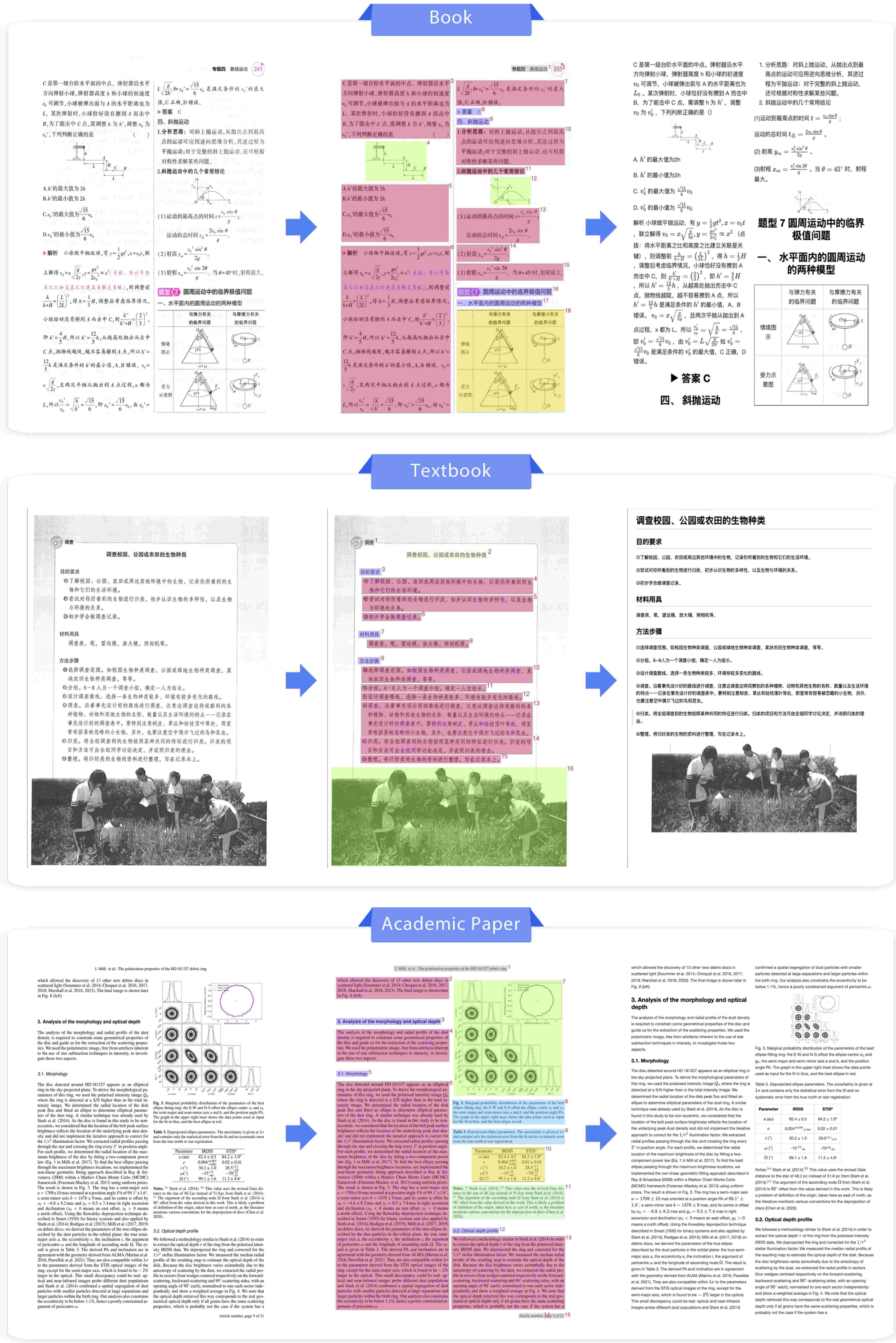


Figure A5 | The Layout and Markdown Output for Book, Textbook and Academic Paper.

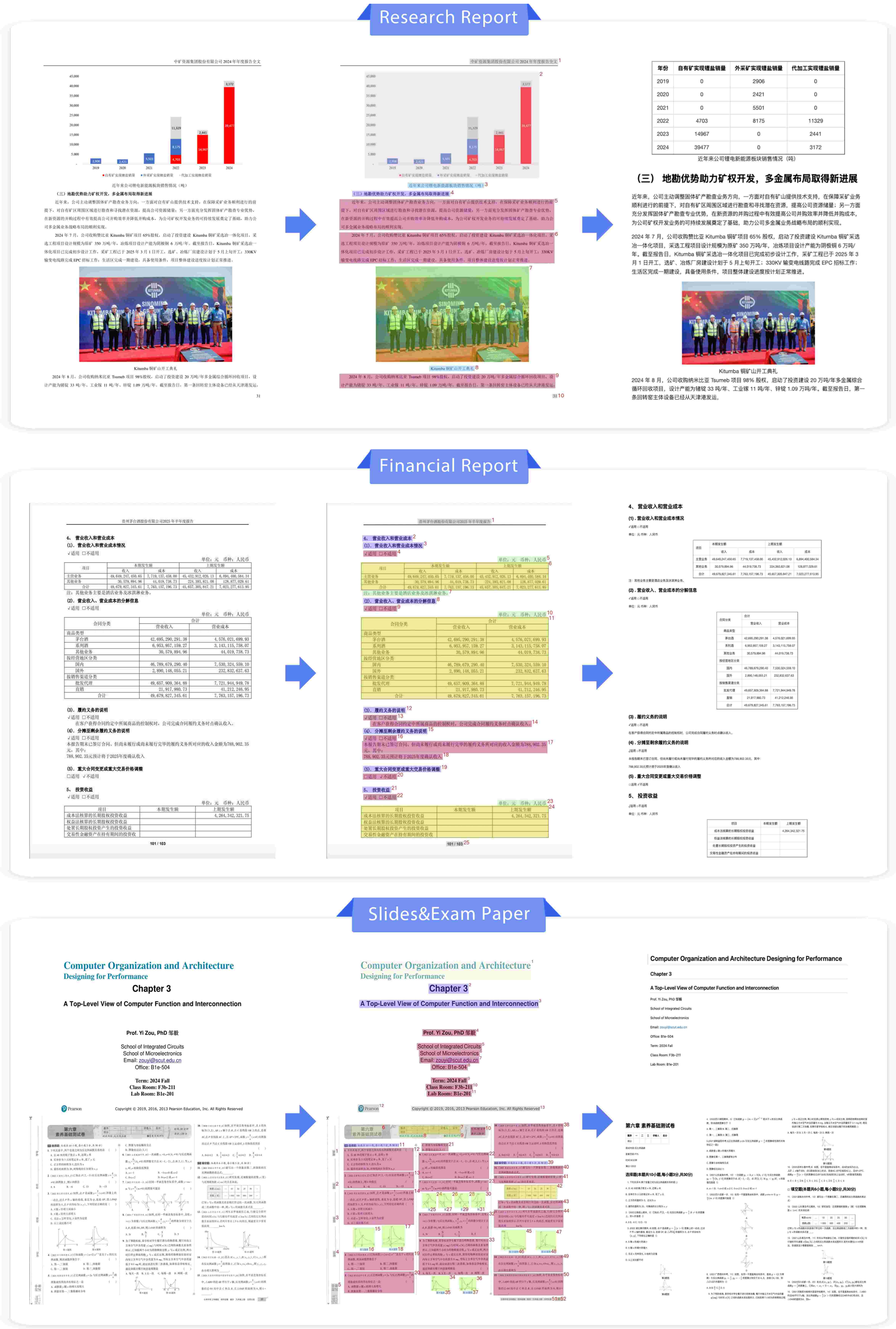


Figure A6 | The Layout and Markdown Output for Research Report(with chart recognition enabled), Financial Report, Slides and Exam Paper.

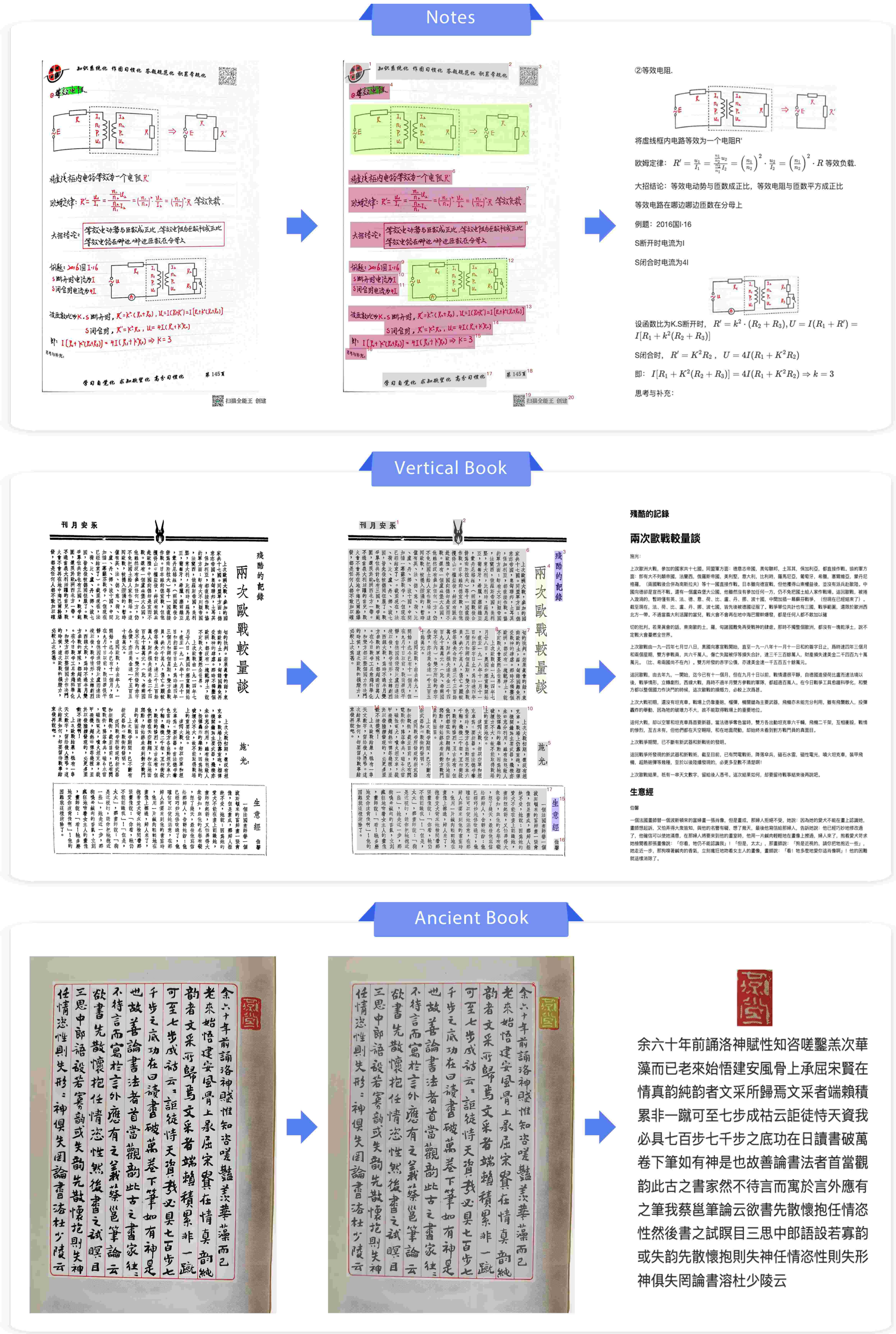


Figure A7 | The Layout and Markdown Output for Notes, Vertical Book and Ancient Book.

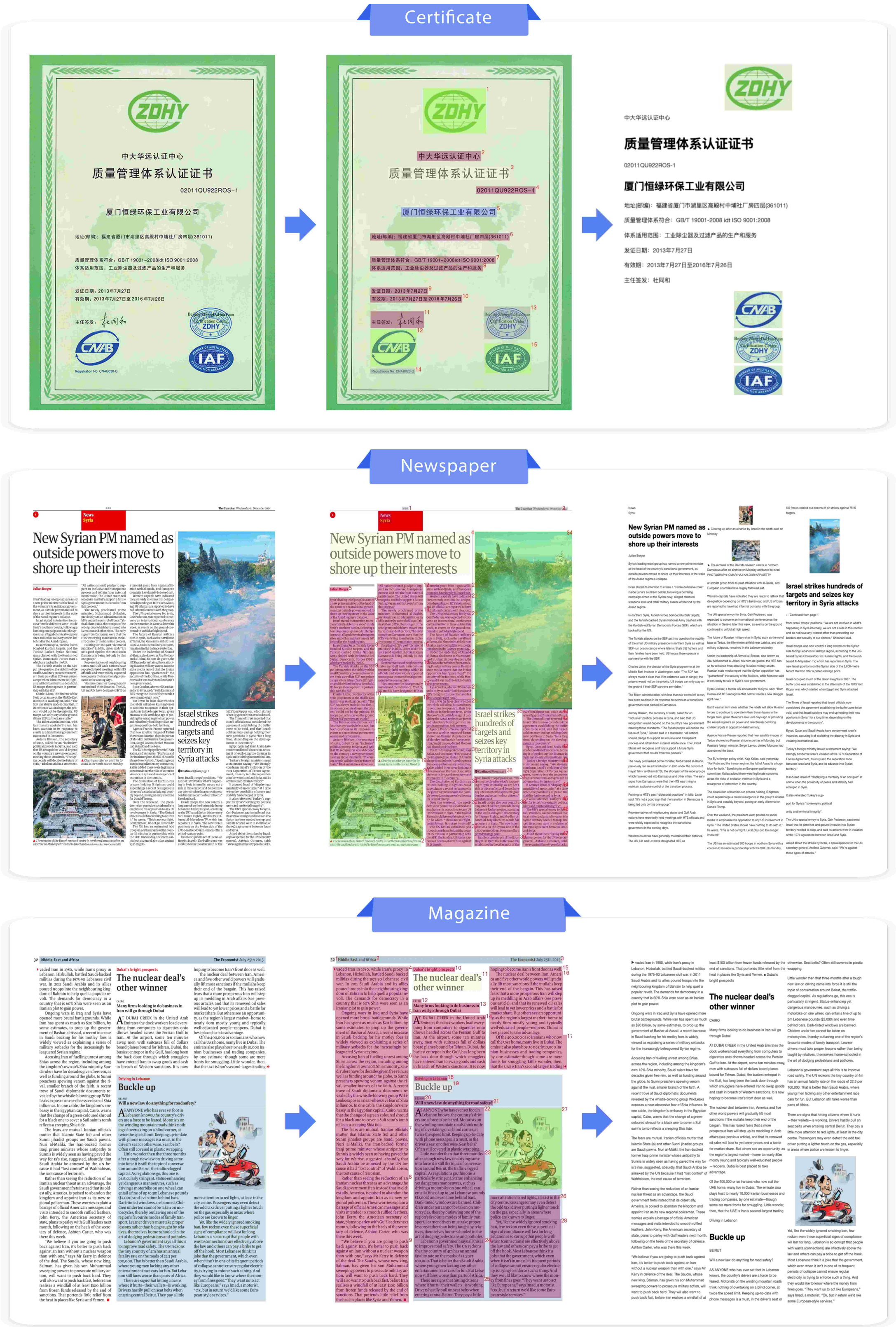


Figure A8 | The Layout and Markdown Output for Certificate, Newspaper and Magazine.

D.2. Layout Detection

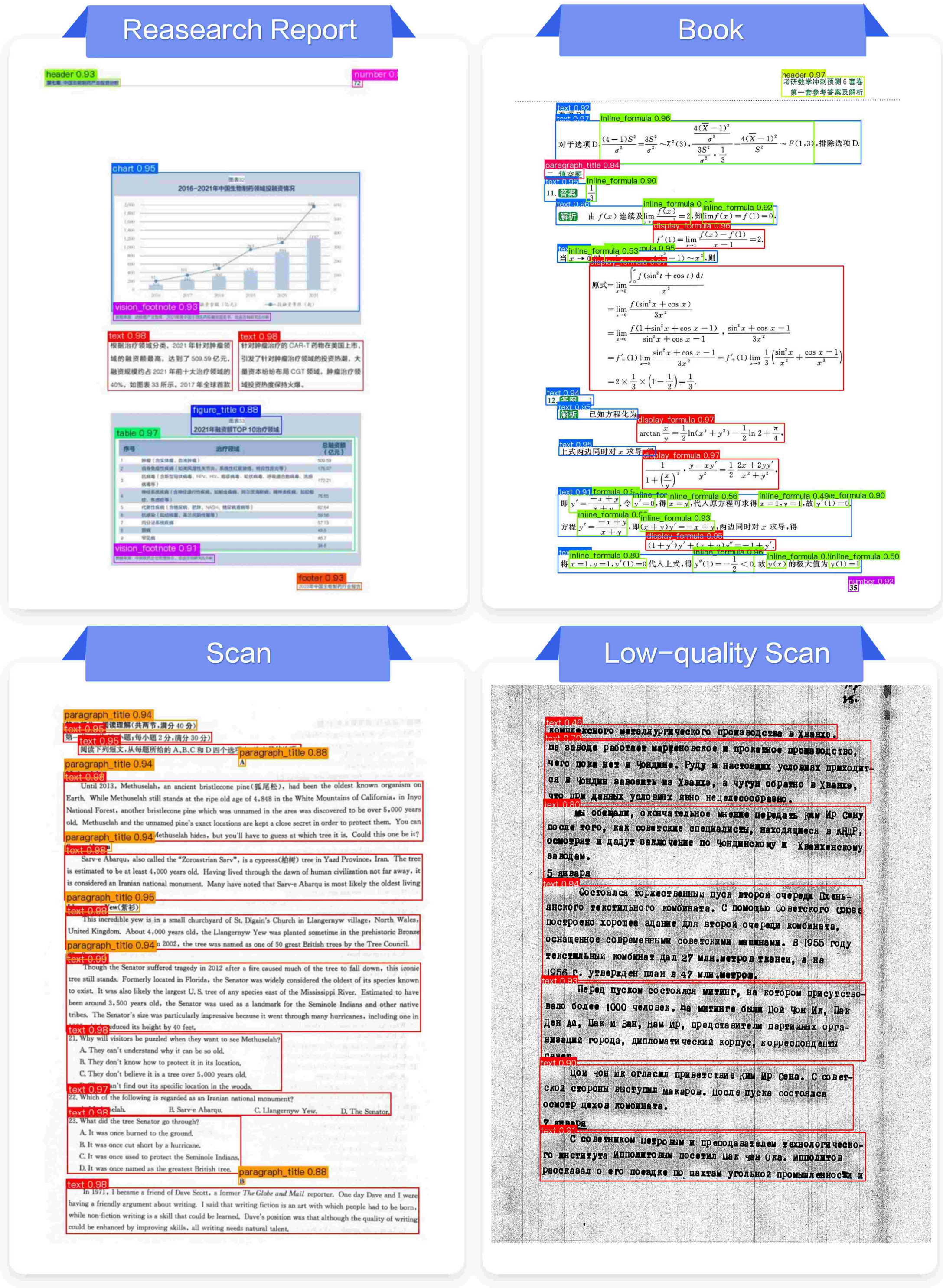


Figure A9 | The Layout Detection results for various types of documents.

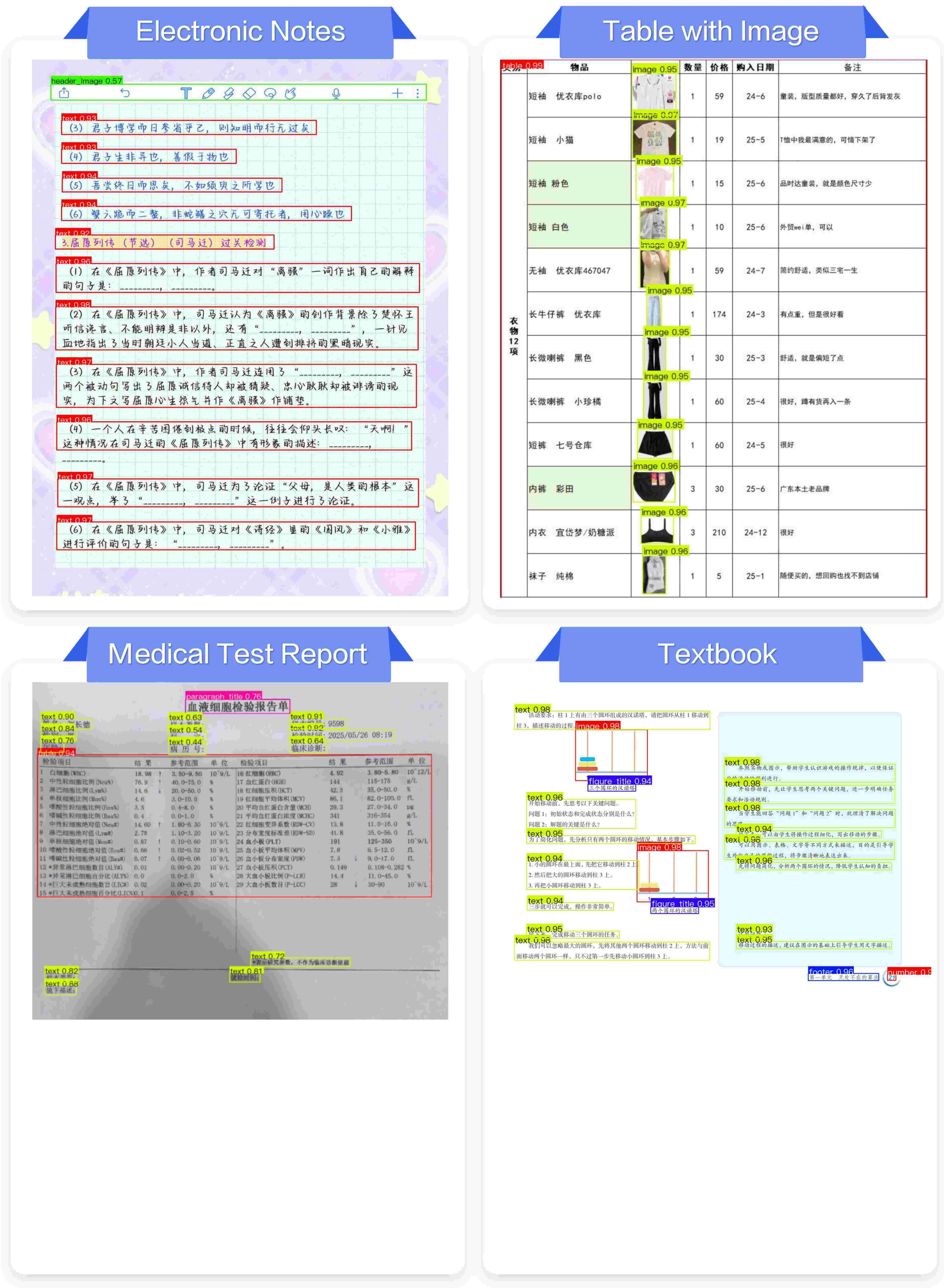


Figure A10 | The Layout Detection results for various types of documents.

Figure A11 | The Layout Detection results for various types of documents.

D.3. Reading Order

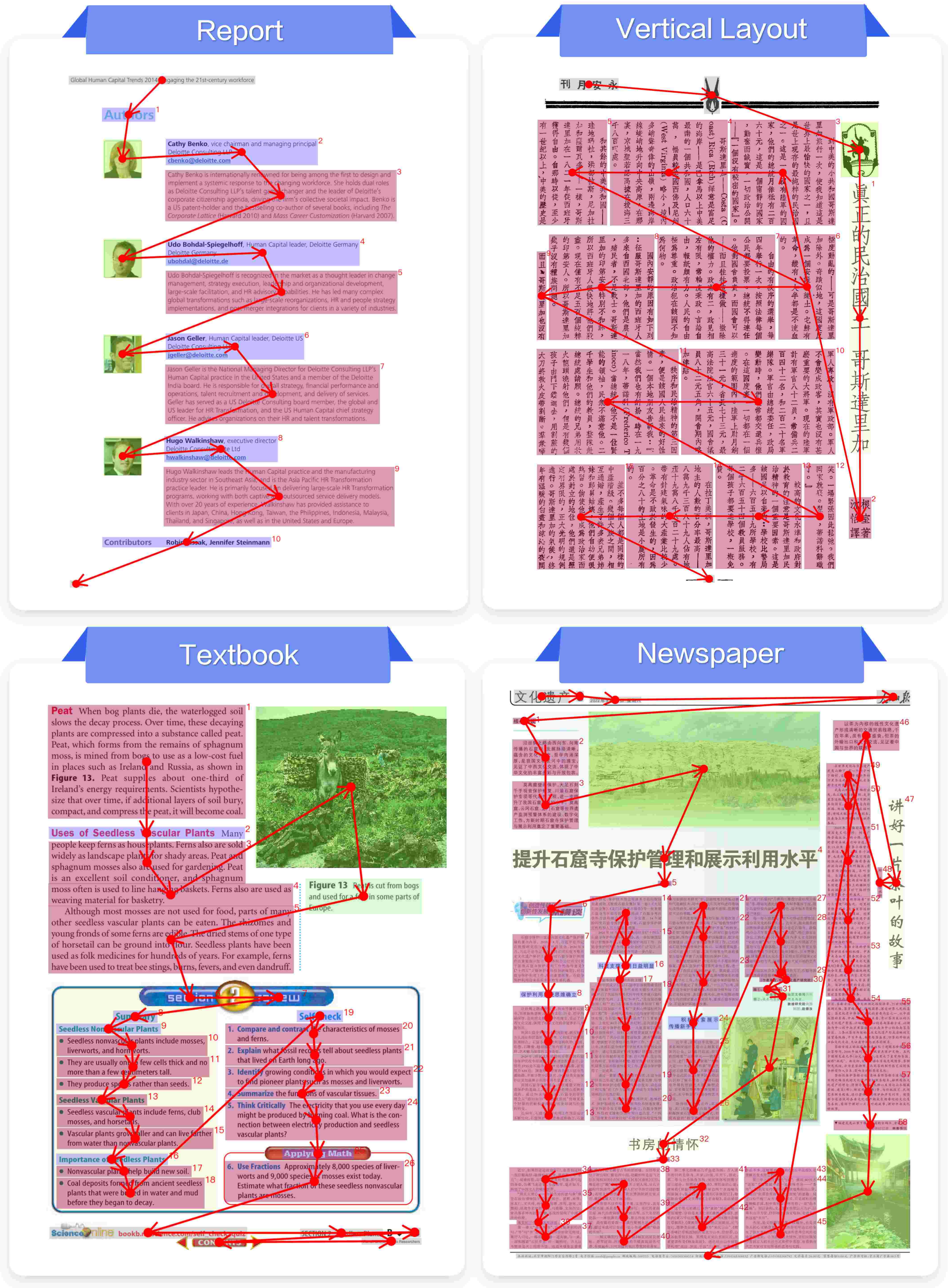


Figure A12 | The Reading Order results for various types of documents.

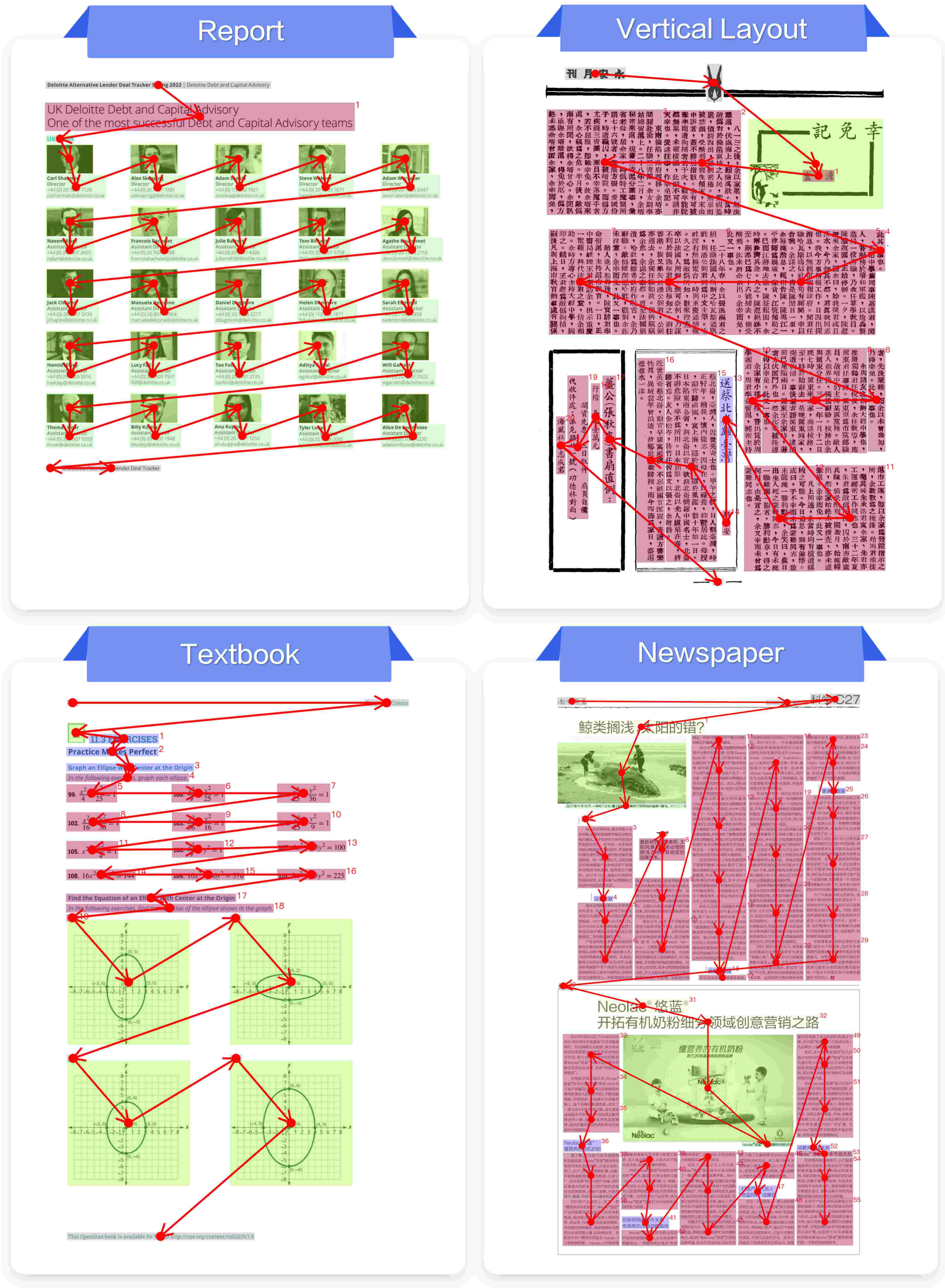


Figure A13 | The Reading Order results for various types of documents.

D.4. Text Recognition

D.4.1. Multilingual Text Recognition

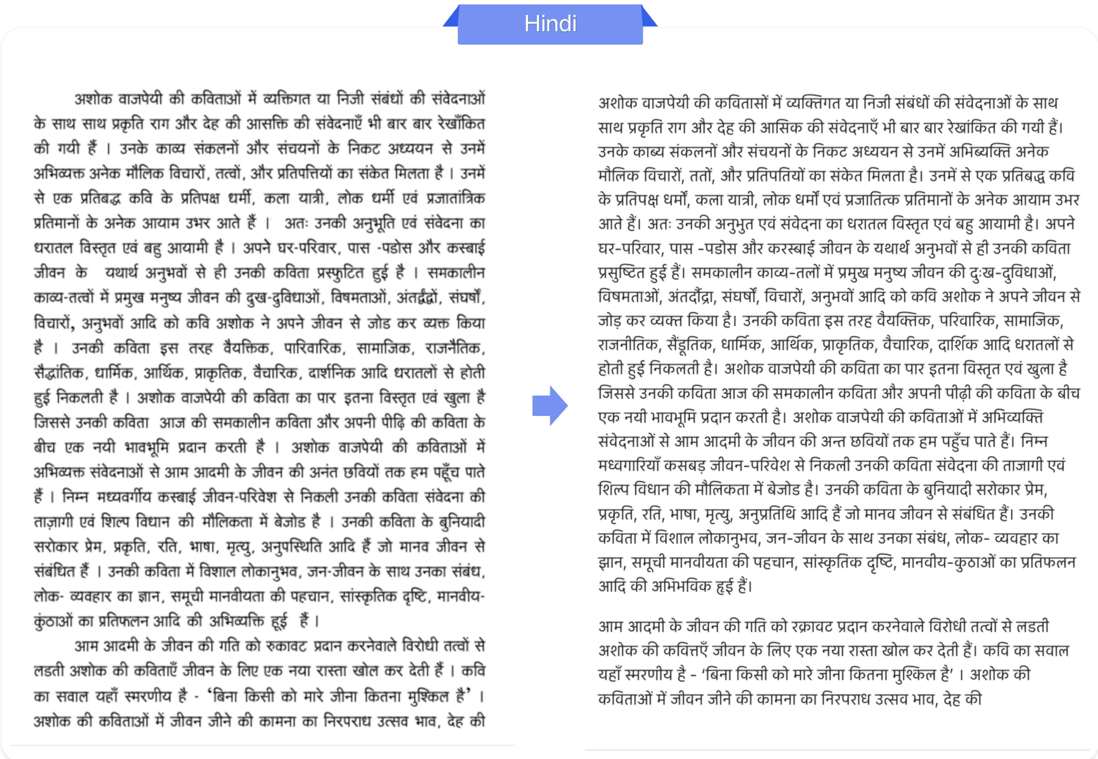


Figure A14 | The markdown output for French and Hindi documents.

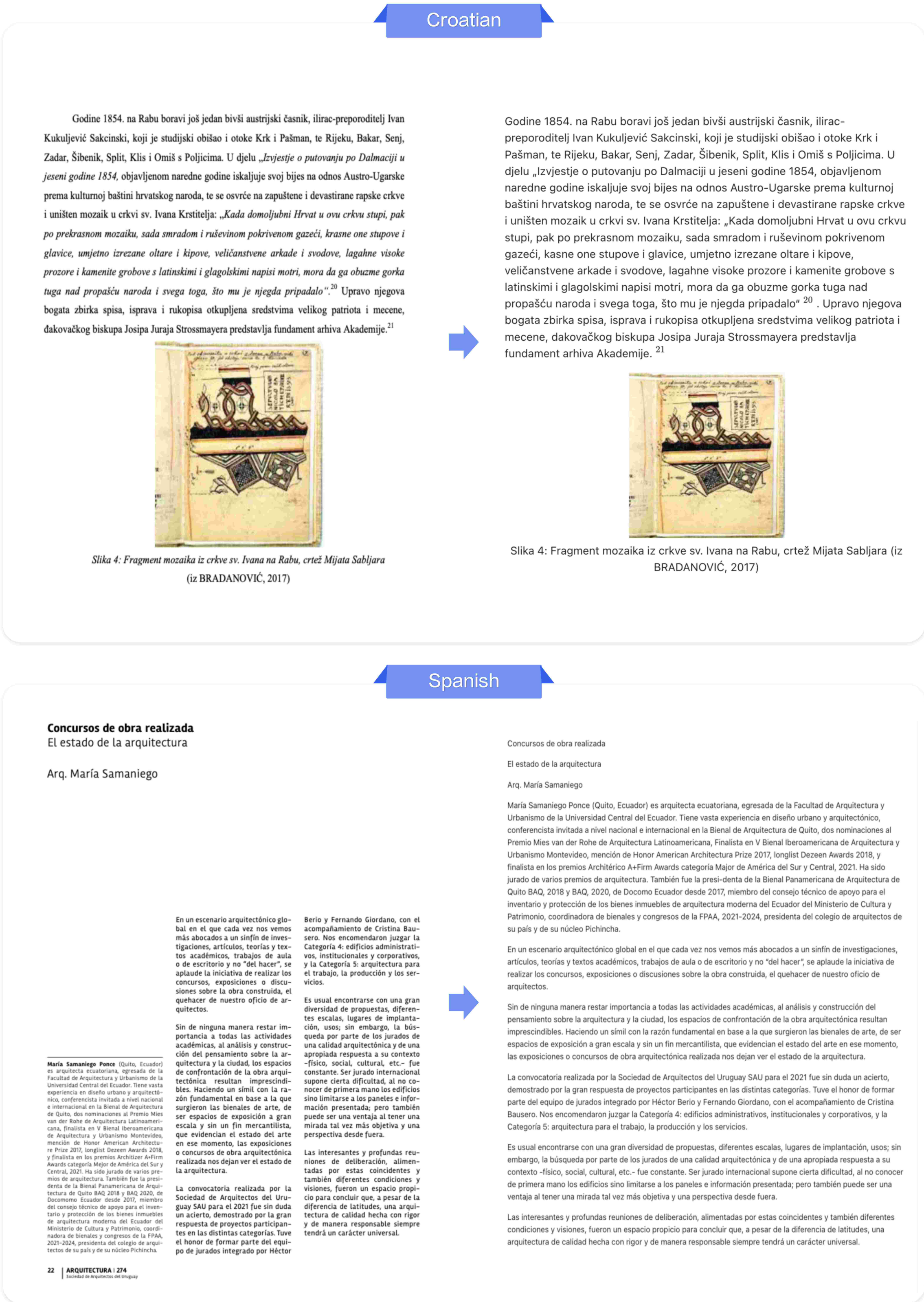
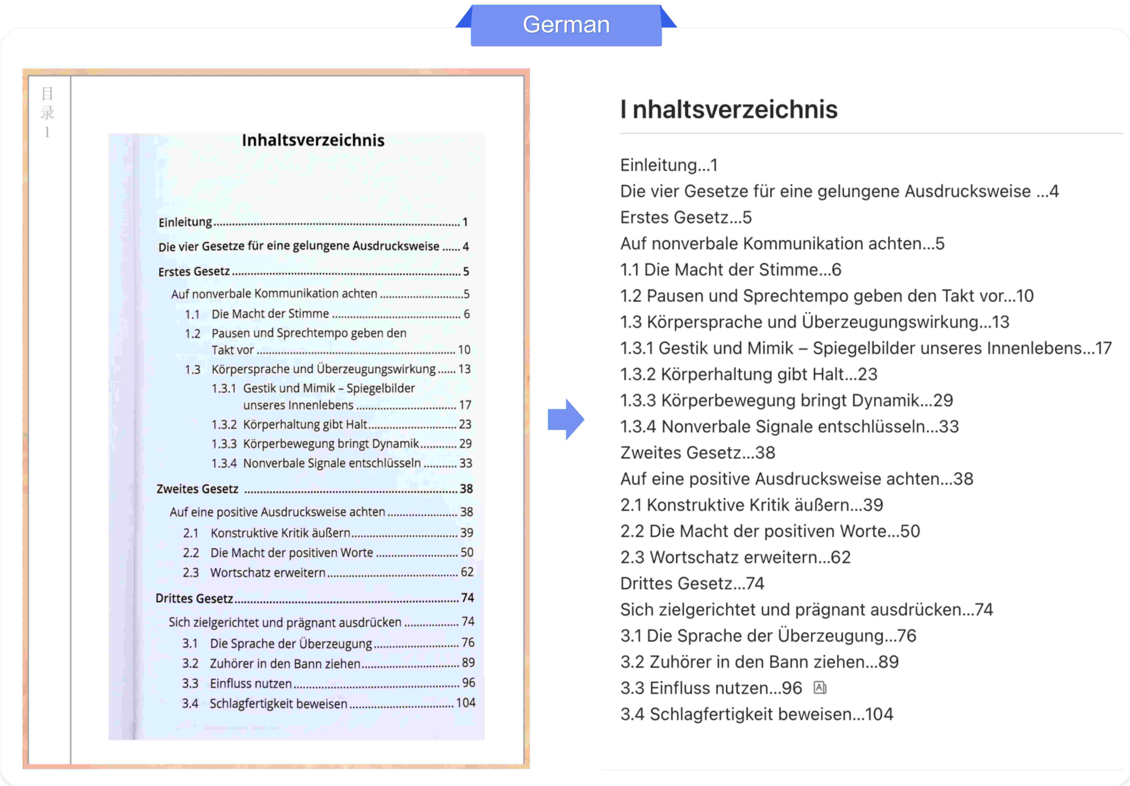


Figure A15 | The markdown output for Croatian and Spanish documents.

Figure A16 | The markdown output for English and Arabic documents.



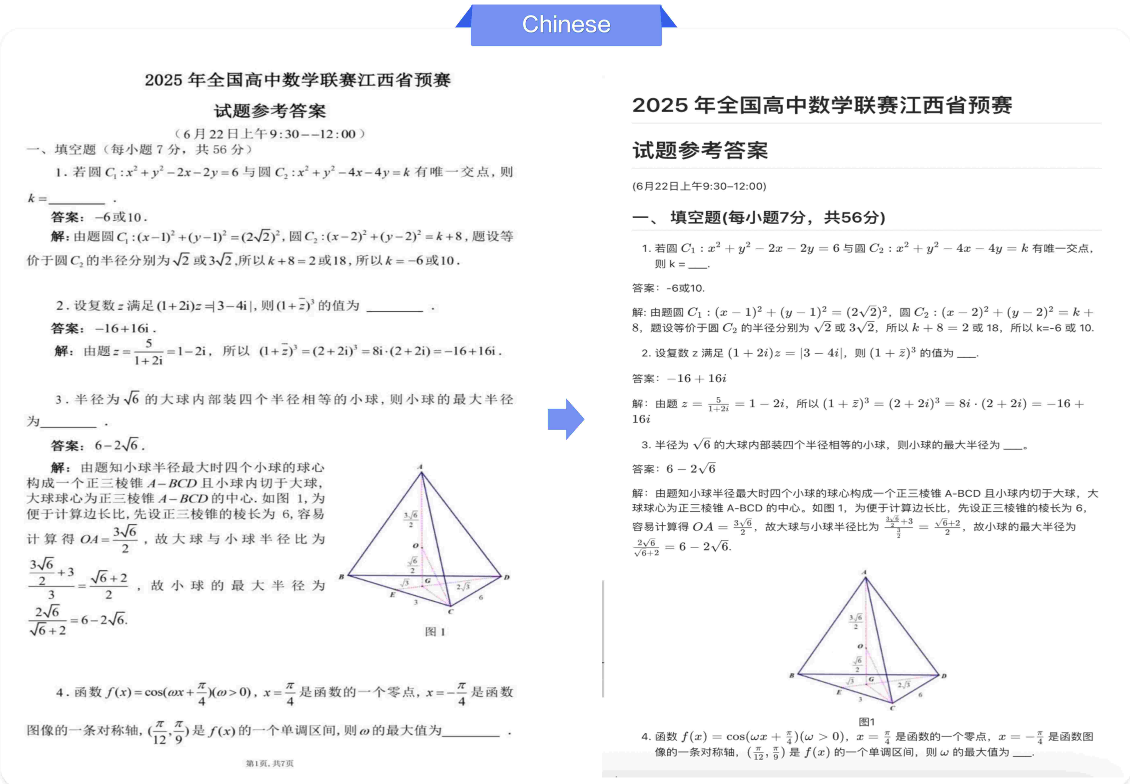


Figure A17 | The markdown output for German and Chinese documents.

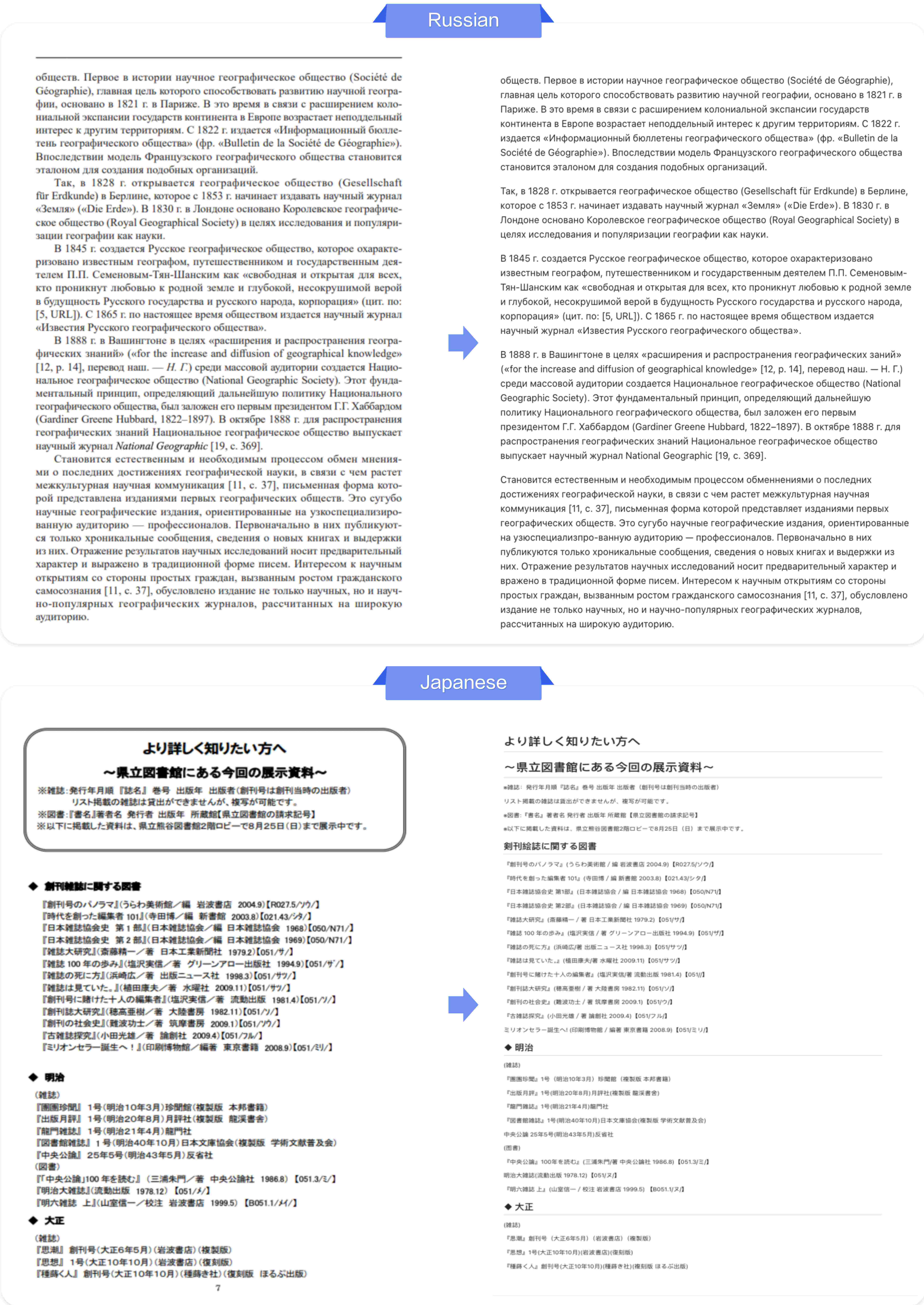
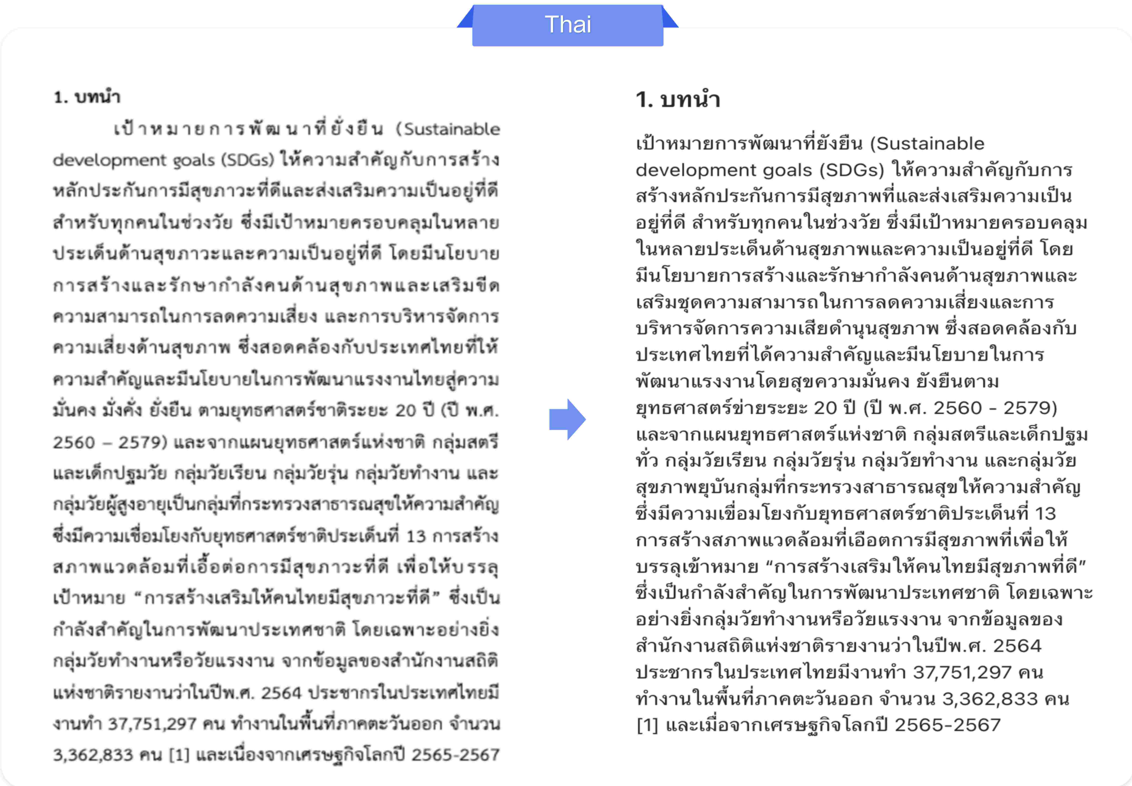


Figure A18 | The markdown output for Russian and Japanese documents.



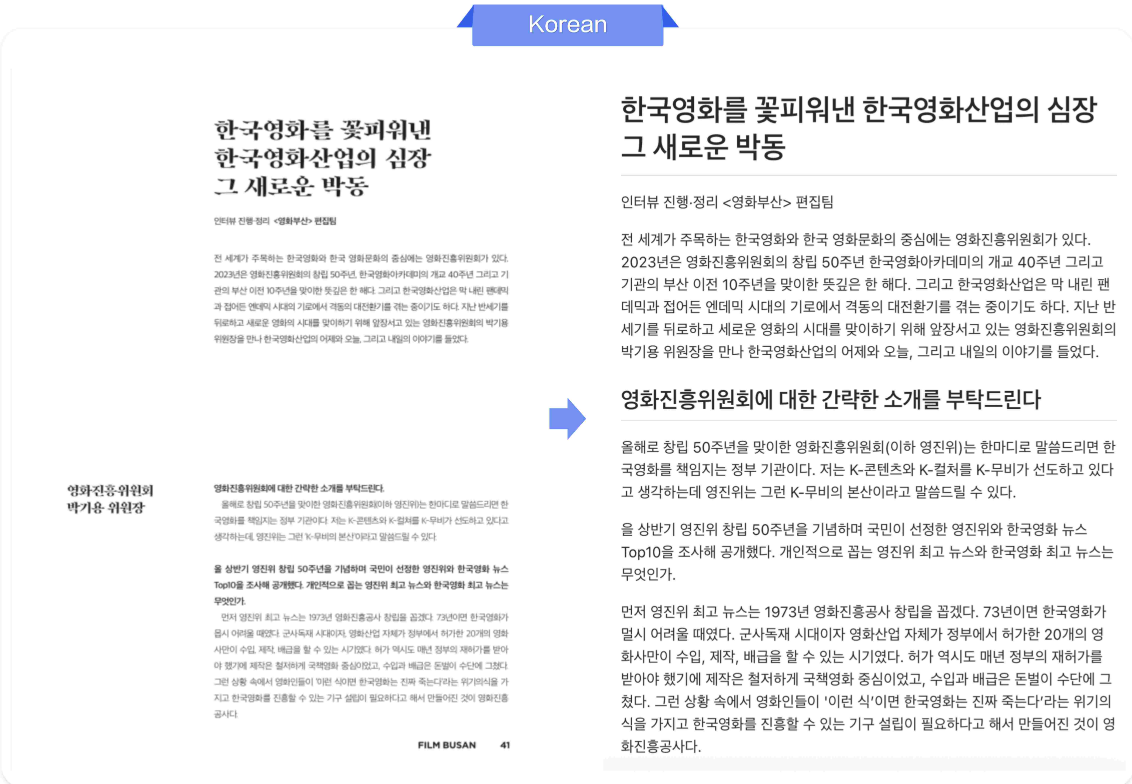


Figure A19 | The markdown output for Thai and Korean documents.

D.4.2. Handwriting Text Recognition

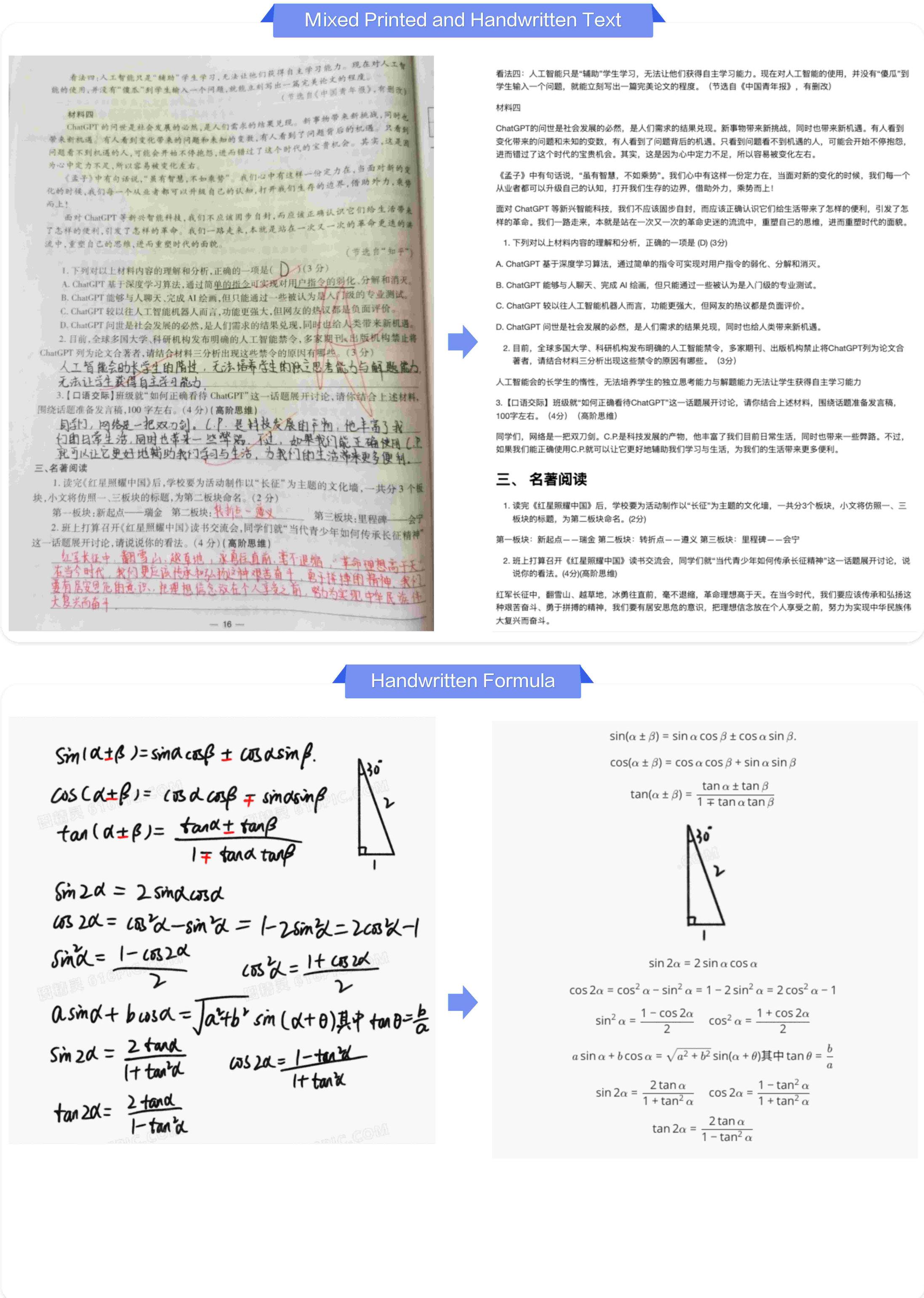
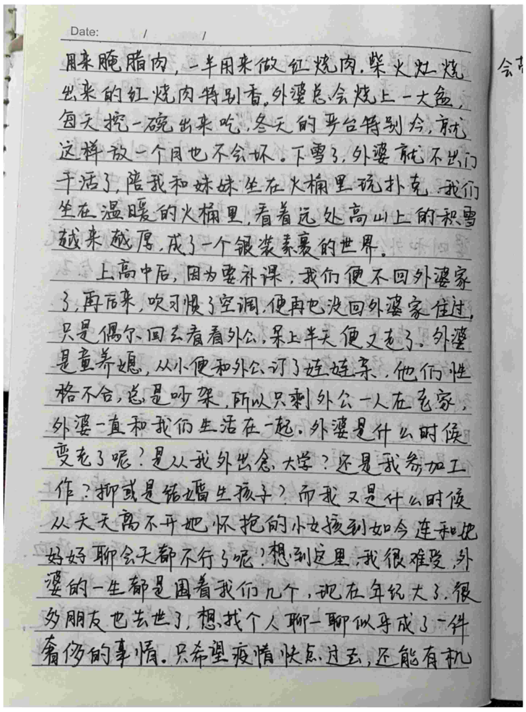


Figure A20 | The markdown output for Mixed Printed Handwritten Text and Handwritten Formula documents.

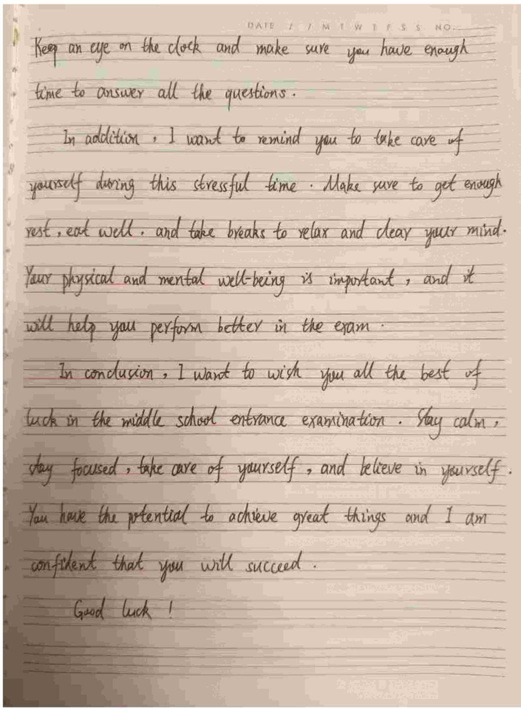
Handwriting Chinese



用来腌腊肉，一半用来做红烧肉。柴火灶烧出来的红烧肉特别香。外婆总会烧上一大盆，每天挖一碗出来吃，冬天的严台特别冷，就这样放一个月也不会坏。下雪了，外婆就不出门干活了，陪我和妹妹坐在火桶里玩扑克。我们坐在温暖的火桶里，看着远处高山上的积雪越来越厚，成了一个银装素裹的世界。

上高中后，因为要补课，我们便不回外婆家了，再后来，吹习惯了空调，便再也没回外婆家住过。只是偶尔回去看看外公，呆上半天便又走了。外婆是童养媳，从小便和外公订了娃娃亲，他们性格不合，总是吵架，所以只剩外公一人在老家，外婆一直和我们生活在一起。外婆是什么时候变老了呢？是从我外出念大学？还是我参加工作？抑或是结婚生孩子？而我又是什么时候从天天离不开她怀抱的小女孩到如今连和她好好聊会天都不行了呢？想到这里，我很难受，外婆的一生都是围着我们几个，现在年纪大了，很多朋友也去世了，想找个人聊一聊似乎成了一件奢侈的事情。只希望疫情快点过去，还能有机

Handwriting English



Keep an eye on the clock and make sure you have enough time to answer all the questions.

In addition, I want to remind you to take care of yourself during this stressful time. Make sure to get enough rest, eat well, and take breaks to relax and clear your mind. Your physical and mental well-being is important, and it will help you perform better in the exam.

In conclusion, I want to wish you all the best of luck in the middle school entrance examination. Stay calm, stay focused, take care of yourself, and believe in yourself. You have the potential to achieve great things and I am confident that you will succeed.

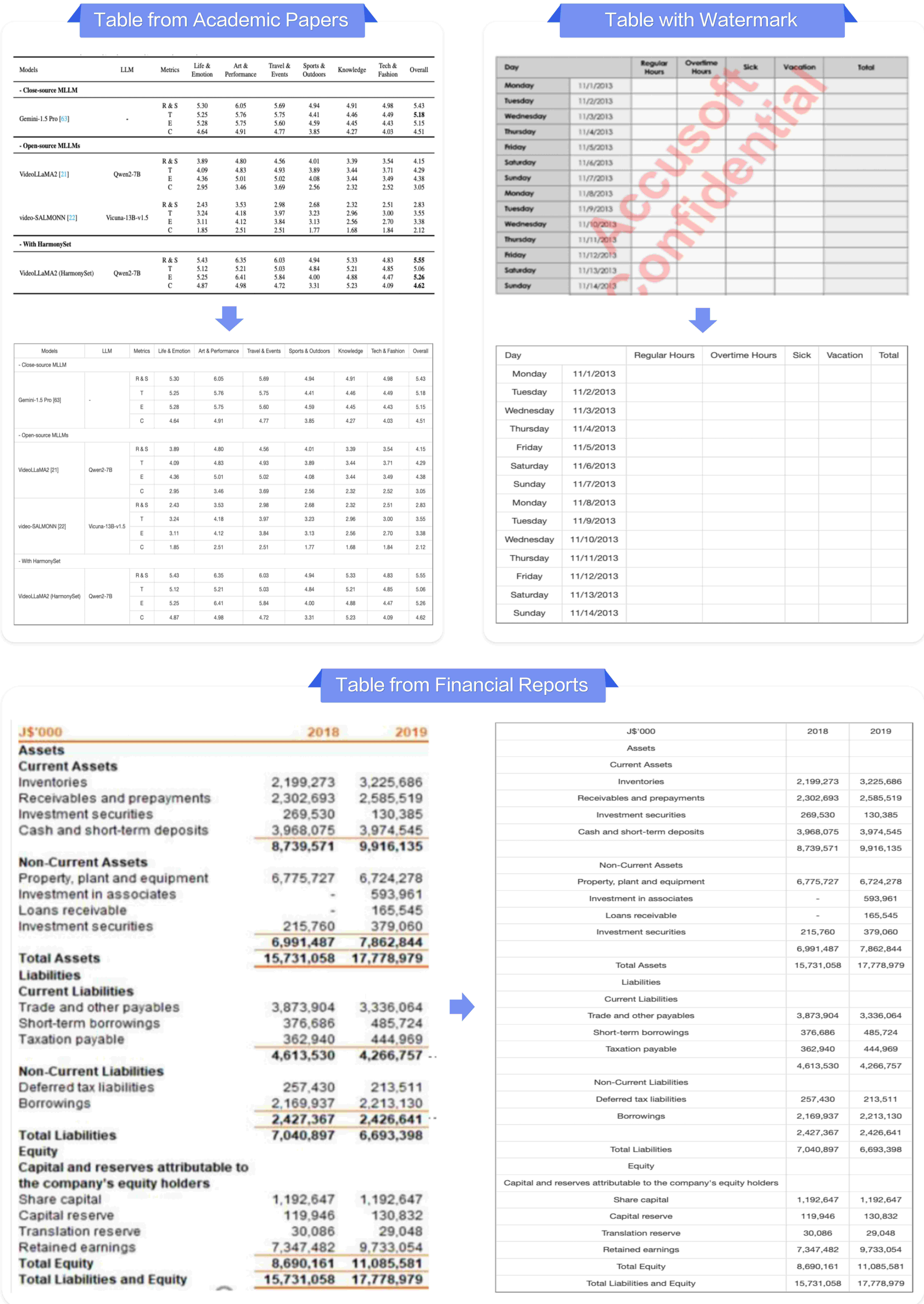
Good luck!

Figure A21 | The markdown output for Handwriting Chinese and Handwriting English documents.

D.4.3. *Vertical Text Recognition*

Figure A22 | The markdown output for various types of vertical documents.

D.5. Table Recognition

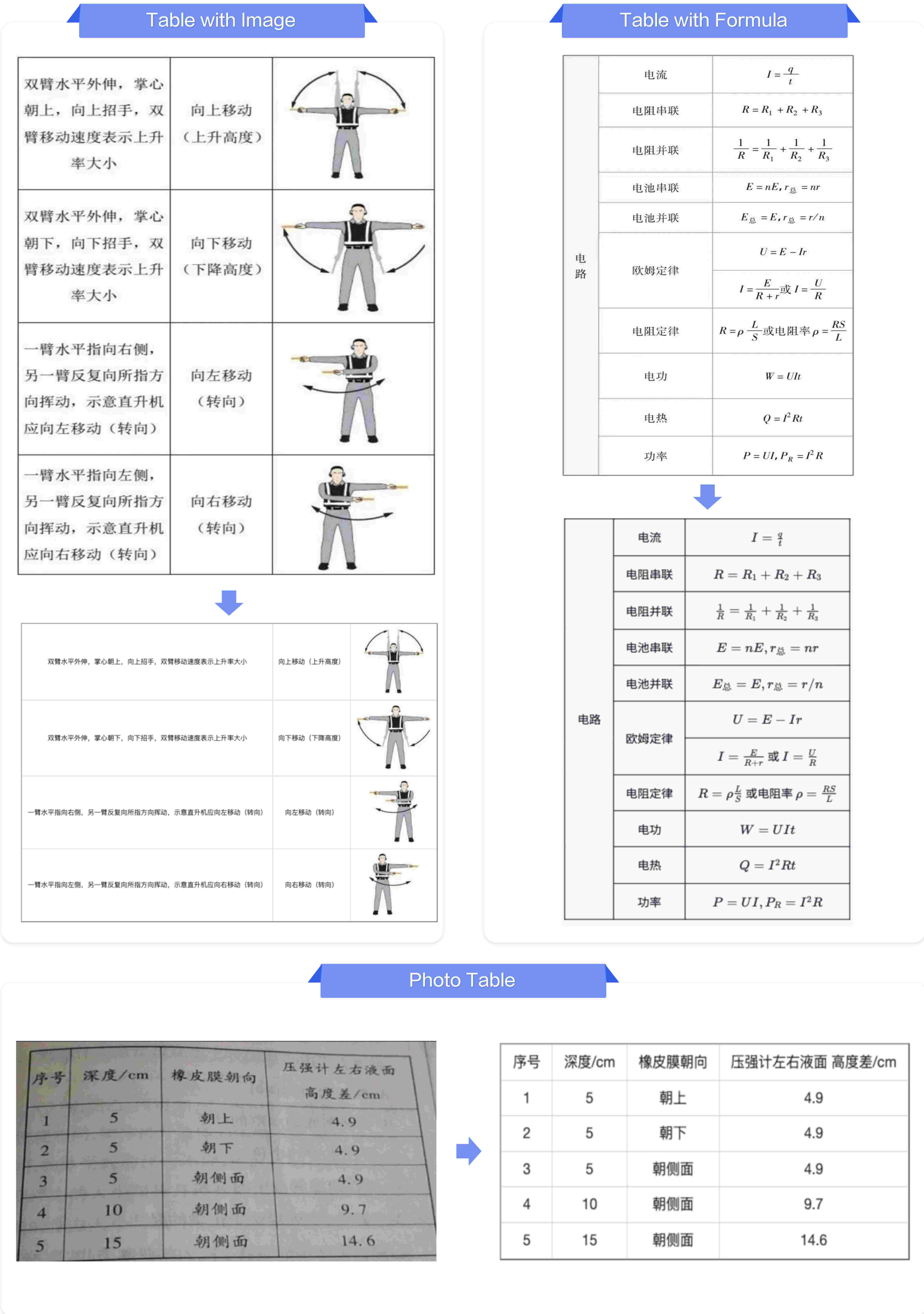


Models	LLM	Metrics	Life & Emotion	Art & Performance	Travel & Events	Sports & Outdoors	Knowledge	Tech & Fashion	Overall
- Close-source MLLM									
Gemini-1.5 Pro [63]	-	R & S	5.30	6.05	5.69	4.94	4.91	4.98	5.43
		T	5.25	5.76	5.75	4.41	4.46	4.49	**5.18**
		E	5.28	5.75	5.60	4.59	4.45	4.43	5.15
		C	4.64	4.91	4.77	3.85	4.27	4.03	4.51
- Open-source MLLMs									
VideoLLaMA2 [21]	Qwen2-7B	R & S	3.89	4.80	4.56	4.01	3.39	3.54	4.15
		T	4.09	4.83	4.93	3.89	3.44	3.71	4.29
		E	4.36	5.01	5.02	4.08	3.44	3.49	4.38
		C	2.95	3.46	3.69	2.56	2.32	2.52	3.05
video-SALMONN [22]	Vicuna-13B-v1.5	R & S	2.43	3.53	2.98	2.68	2.32	2.51	2.83
		T	3.24	4.18	3.97	3.23	2.96	3.00	3.55
		E	3.11	4.12	3.84	3.13	2.56	2.70	3.38
		C	1.85	2.51	2.51	1.77	1.68	1.84	2.12
- With HarmonySet									
VideoLLaMA2 (HarmonySet)	Qwen2-7B	R & S	5.43	6.35	6.03	4.94	5.33	4.83	**5.55**
		T	5.12	5.21	5.03	4.84	5.21	4.85	5.06
		E	5.25	6.41	5.84	4.00	4.88	4.47	**5.26**
		C	4.87	4.98	4.72	3.31	5.23	4.09	**4.62**

Day		Regular Hours	Overtime Hours	Sick	Vacation	Total
Monday	11/1/2013					
Tuesday	11/2/2013					
Wednesday	11/3/2013					
Thursday	11/4/2013					
Friday	11/5/2013					
Saturday	11/6/2013					
Sunday	11/7/2013					
Monday	11/8/2013					
Tuesday	11/9/2013					
Wednesday	11/10/2013					
Thursday	11/11/2013					
Friday	11/12/2013					
Saturday	11/13/2013					
Sunday	11/14/2013					

J$'000	2018	2019
Assets		
Current Assets		
Inventories	2,199,273	3,225,686
Receivables and prepayments	2,302,693	2,585,519
Investment securities	269,530	130,385
Cash and short-term deposits	3,968,075	3,974,545
	8,739,571	9,916,135
Non-Current Assets		
Property, plant and equipment	6,775,727	6,724,278
Investment in associates	-	593,961
Loans receivable	-	165,545
Investment securities	215,760	379,060
	6,991,487	7,862,844
Total Assets	15,731,058	17,778,979
Liabilities		
Current Liabilities		
Trade and other payables	3,873,904	3,336,064
Short-term borrowings	376,686	485,724
Taxation payable	362,940	444,969
	4,613,530	4,266,757
Non-Current Liabilities		
Deferred tax liabilities	257,430	213,511
Borrowings	2,169,937	2,213,130
	2,427,367	2,426,641
Total Liabilities	7,040,897	6,693,398
Equity		
Capital and reserves attributable to the company's equity holders		
Share capital	1,192,647	1,192,647
Capital reserve	119,946	130,832
Translation reserve	30,086	29,048
Retained earnings	7,347,482	9,733,054
Total Equity	8,690,161	11,085,581
Total Liabilities and Equity	15,731,058	17,778,979

Figure A23 | The markdown output for various types of Tables.



双臂水平外伸，掌心朝上，向上招手，双臂移动速度表示上升率大小	向上移动（上升高度）	
双臂水平外伸，掌心朝下，向下招手，双臂移动速度表示上升率大小	向下移动（下降高度）	
一臂水平指向右侧，另一臂反复向所指方向挥动，示意直升机应向左移动（转向）	向左移动（转向）	
一臂水平指向左侧，另一臂反复向所指方向挥动，示意直升机应向右移动（转向）	向右移动（转向）	

电路	电流	$I = \frac{q}{t}$
	电阻串联	$R = R_1 + R_2 + R_3$
	电阻并联	$\frac{1}{R} = \frac{1}{R_1} + \frac{1}{R_2} + \frac{1}{R_3}$
	电池串联	$E = nE, r_{总} = nr$
	电池并联	$E_{总} = E, r_{总} = r/n$
	欧姆定律	$U = E - Ir$
		$I = \frac{E}{R+r}$ 或 $I = \frac{U}{R}$
	电阻定律	$R = \rho\frac{L}{S}$ 或电阻率 $\rho = \frac{RS}{L}$
	电功	$W = UIt$
	电热	$Q = I^2Rt$
	功率	$P = UI, P_R = I^2R$

序号	深度/cm	橡皮膜朝向	压强计左右液面 高度差/cm
1	5	朝上	4.9
2	5	朝下	4.9
3	5	朝侧面	4.9
4	10	朝侧面	9.7
5	15	朝侧面	14.6

Figure A24 | The markdown output for various types of Tables.

D.6. Formula Recognition

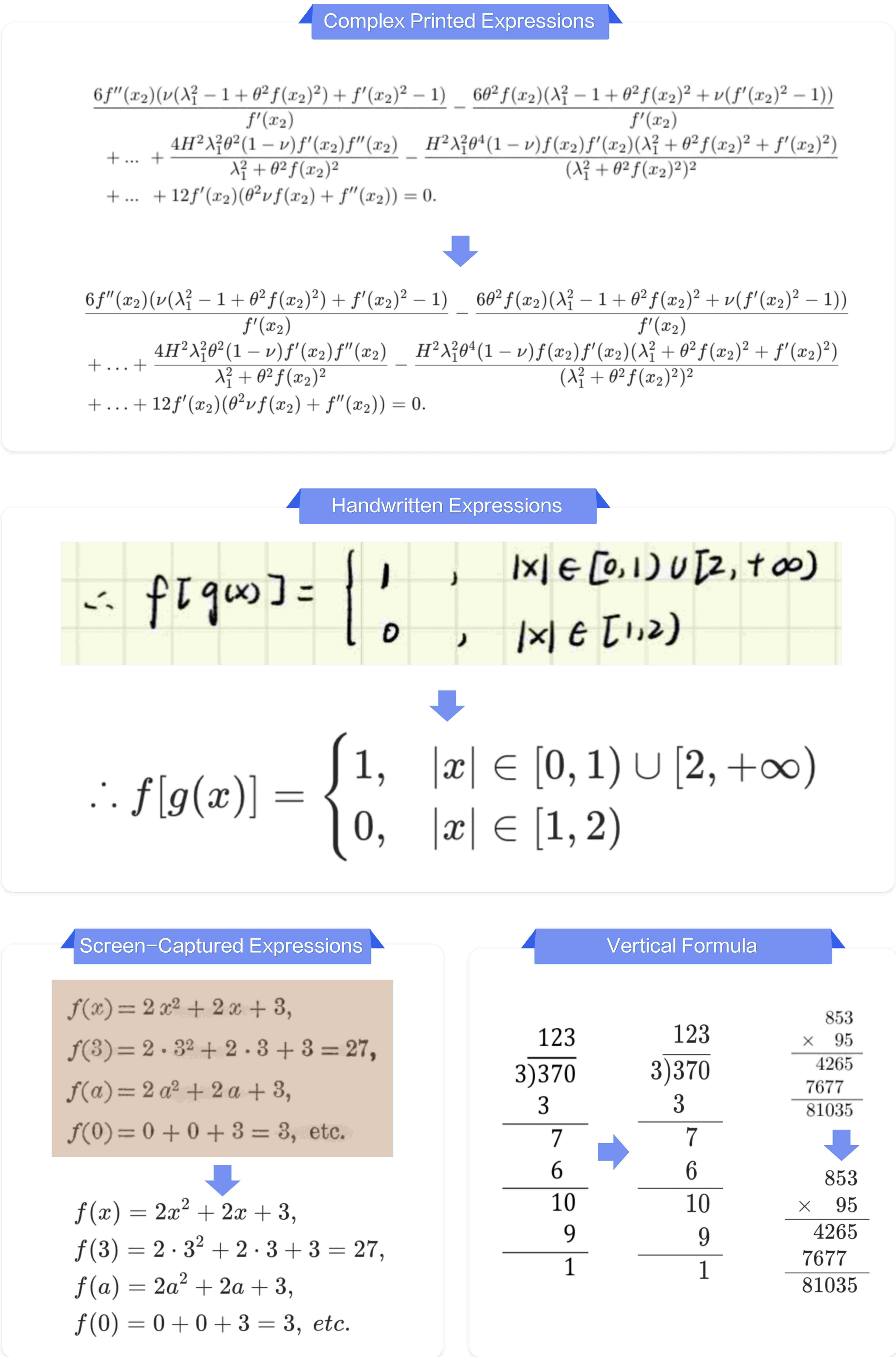


Figure A25 | The markdown output for various types of Formulas.

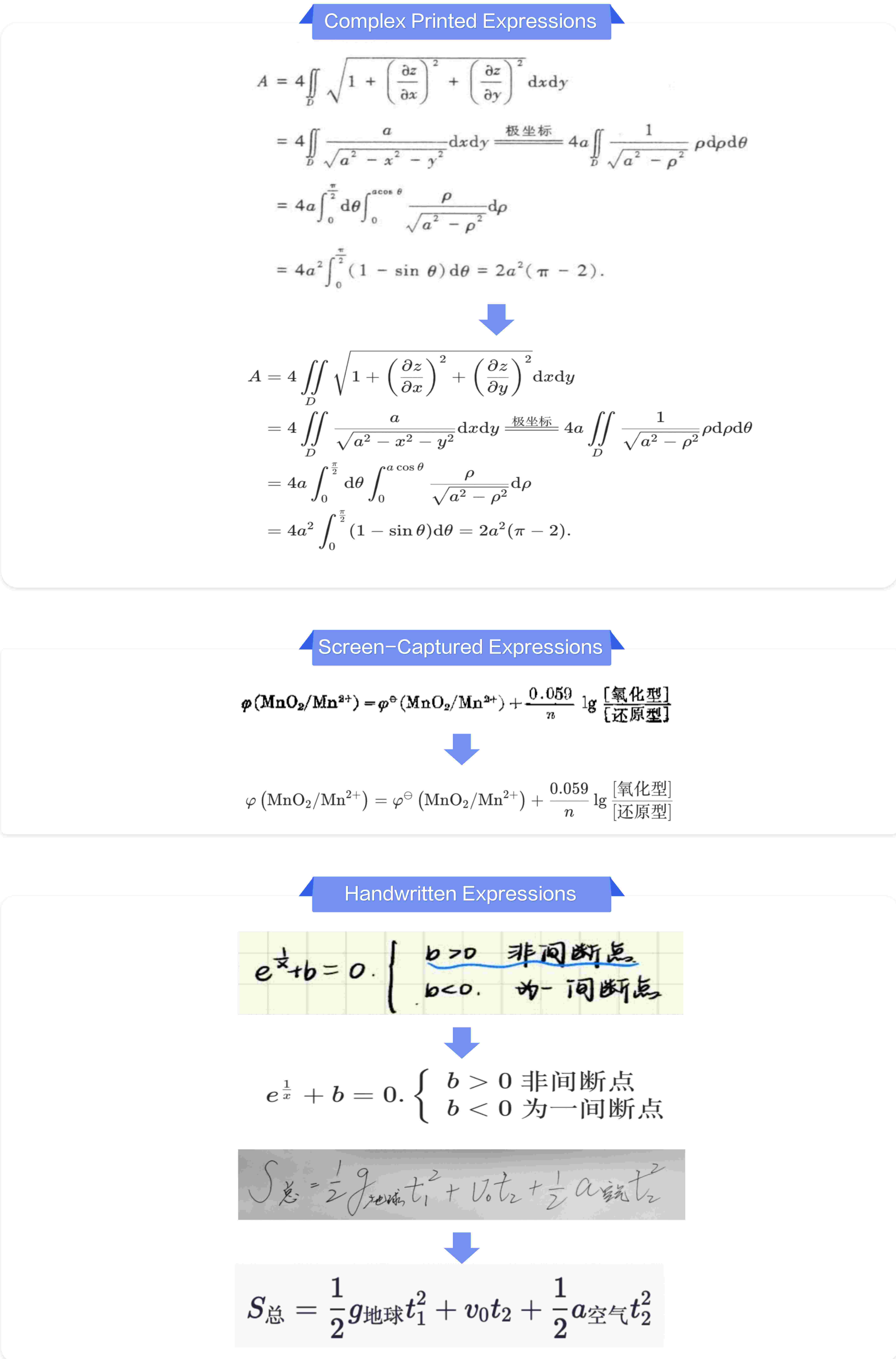
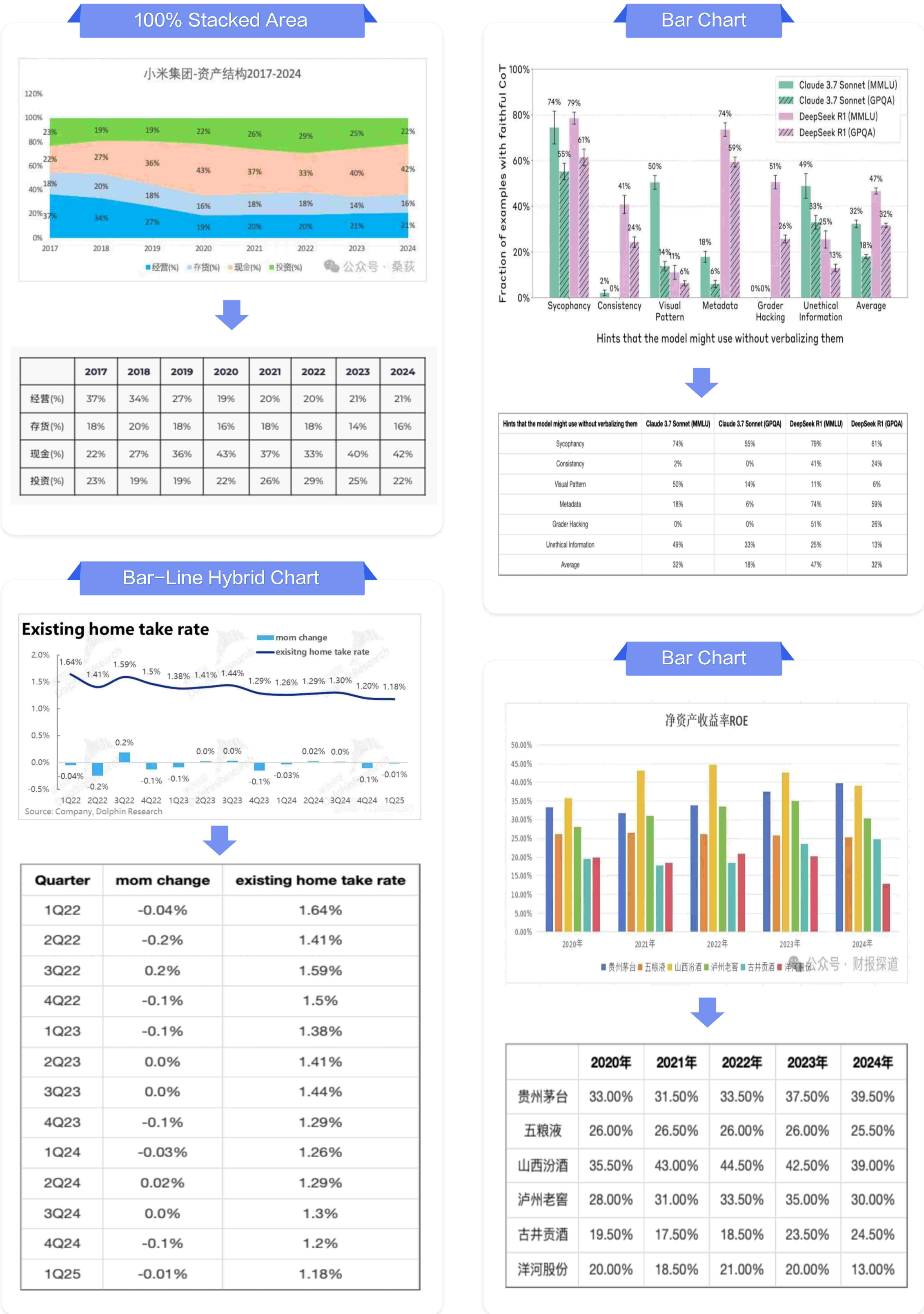


Figure A26 | The markdown output for various types of Formulas.

D.7. Chart Recognition



	2017	2018	2019	2020	2021	2022	2023	2024
经营(%)	37%	34%	27%	19%	20%	20%	21%	21%
存货(%)	18%	20%	18%	16%	18%	18%	14%	16%
现金(%)	22%	27%	36%	43%	37%	33%	40%	42%
投资(%)	23%	19%	19%	22%	26%	29%	25%	22%

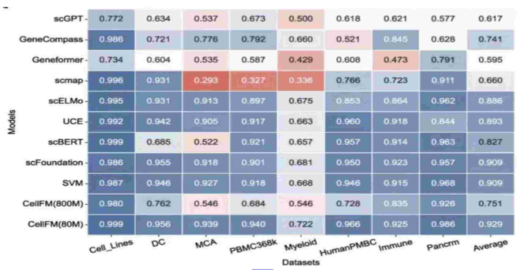
Hints that the model might use without verbalizing them	Claude 3.7 Sonnet (MMLU)	Claude 3.7 Sonnet (GPQA)	DeepSeek R1 (MMLU)	DeepSeek R1 (GPQA)
Sycophancy	74%	55%	79%	61%
Consistency	2%	0%	41%	24%
Visual Pattern	50%	14%	11%	6%
Metadata	18%	6%	74%	59%
Grader Hacking	0%	0%	51%	26%
Unethical Information	49%	33%	25%	13%
Average	32%	18%	47%	32%

Quarter	mom change	existing home take rate
1Q22	-0.04%	1.64%
2Q22	-0.2%	1.41%
3Q22	0.2%	1.59%
4Q22	-0.1%	1.5%
1Q23	-0.1%	1.38%
2Q23	0.0%	1.41%
3Q23	0.0%	1.44%
4Q23	-0.1%	1.29%
1Q24	-0.03%	1.26%
2Q24	0.02%	1.29%
3Q24	0.0%	1.3%
4Q24	-0.1%	1.2%
1Q25	-0.01%	1.18%

	2020年	2021年	2022年	2023年	2024年
贵州茅台	33.00%	31.50%	33.50%	37.50%	39.50%
五粮液	26.00%	26.50%	26.00%	26.00%	25.50%
山西汾酒	35.50%	43.00%	44.50%	42.50%	39.00%
泸州老窖	28.00%	31.00%	33.50%	35.00%	30.00%
古井贡酒	19.50%	17.50%	18.50%	23.50%	24.50%
洋河股份	20.00%	18.50%	21.00%	20.00%	13.00%

Figure A27 | The markdown output for various types of Charts.

Heatmap

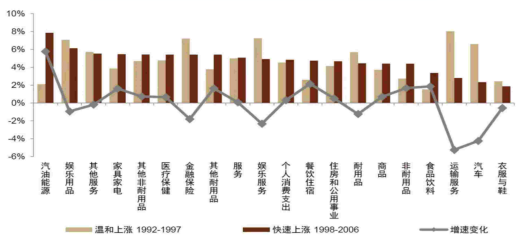


Models	Cell_Lines	DC	MCA	PBMC368k	Myeloid	HumanPMBC	Immune	Pancrm	Average
scGPT	0.772	0.634	0.537	0.673	0.500	0.618	0.621	0.577	0.617
GeneCompass	0.986	0.721	0.776	0.792	0.660	0.521	0.845	0.628	0.741
Geneformer	0.734	0.604	0.535	0.587	0.429	0.608	0.473	0.791	0.595
scmap	0.996	0.931	0.293	0.327	0.336	0.766	0.723	0.911	0.660
scELMo	0.995	0.931	0.913	0.897	0.675	0.853	0.864	0.962	0.886
UCE	0.992	0.942	0.905	0.917	0.663	0.960	0.918	0.844	0.893
scBERT	0.999	0.685	0.522	0.921	0.657	0.957	0.914	0.963	0.827
scFoundation	0.986	0.955	0.918	0.901	0.681	0.950	0.923	0.957	0.909
SVM	0.987	0.946	0.927	0.918	0.668	0.946	0.915	0.968	0.909
CellFM(800M)	0.980	0.762	0.546	0.684	0.546	0.728	0.835	0.926	0.751
CellFM(80M)	0.999	0.956	0.939	0.940	0.722	0.966	0.925	0.986	0.929

Line Chart

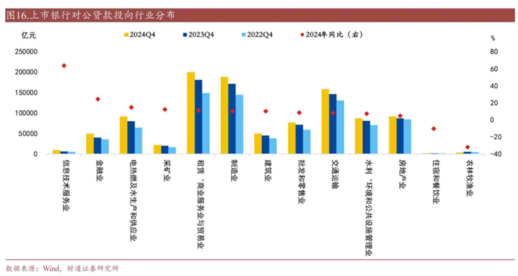
年份	爱尔眼科 - 净资产收益率(ROE)	医疗服务(申万) - 净资产收益率(ROE)
2015	19%	13.25%
2016	20.86%	14.79%
2017	18.85%	7.25%
2018	18.63%	8.96%
2019	22.03%	-0.79%
2020	21.24%	11.70%
2021	21.59%	12.31%
2022	18.02%	12.43%
2023	19.43%	11.11%
2024	17.86%	7.86%

Bar–Line Hybrid Chart



	温和上涨 1992-1997	快速上涨 1998-2006	增速变化
汽油能源	2.2%	7.8%	5.8%
娱乐用品	7.0%	6.2%	-1.2%
其他服务	5.8%	5.4%	-0.2%
家具家电	3.8%	5.4%	1.8%
其他非耐用品	4.8%	5.4%	0.8%
医疗保健	4.8%	5.4%	0.8%
金融保险	7.2%	5.4%	-2.0%
其他耐用品	3.8%	5.4%	1.8%
服务	5.0%	5.2%	0.0%
娱乐服务	7.2%	4.8%	-2.5%
个人消费支出	4.4%	4.8%	0.2%
餐饮住宿	2.8%	4.8%	2.2%
住房和公用事业	4.2%	4.8%	0.2%
耐用品	5.8%	4.4%	-1.5%
商品	3.8%	4.4%	0.8%
非耐用品	2.8%	4.4%	1.8%
食品饮料	1.8%	3.2%	2.0%
运输服务	8.0%	2.8%	-5.5%
汽车	6.4%	2.4%	-4.2%
衣服与鞋	2.4%	1.8%	-0.8%

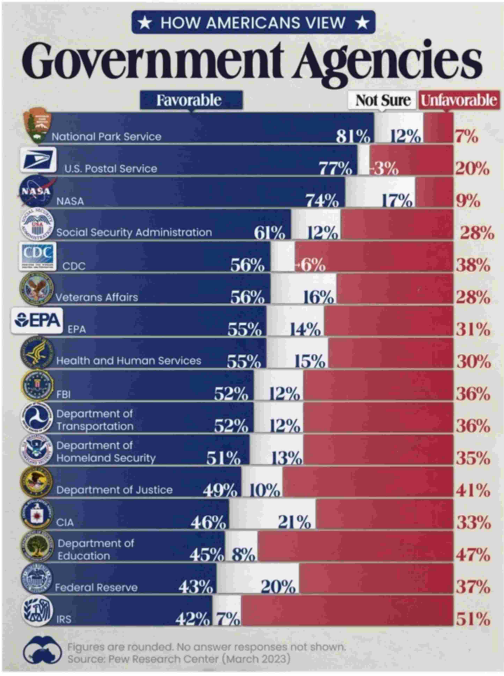
Bar–Line Hybrid Chart



行业	2024Q4 (亿元)	2023Q4 (亿元)	2022Q4 (亿元)	2024年同比 (%)
信息技术服务业	10000	5000	5000	65
金融业	50000	40000	35000	25
电热燃及水生产和供应业	95000	80000	65000	15
采矿业	20000	20000	15000	15
租赁、商业服务业与贸易业	200000	180000	150000	10
制造业	190000	170000	145000	10
建筑业	50000	45000	40000	10
批发和零售业	75000	70000	60000	10
交通运输	160000	145000	130000	10
水利、环境和公共设施管理业	90000	80000	70000	5
房地产业	95000	85000	80000	5
住宿和餐饮业	2000	2000	2000	-10
农林牧渔业	5000	5000	5000	-30

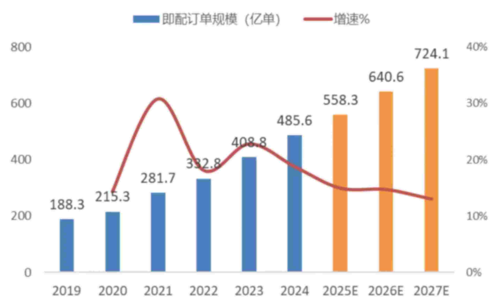
Figure A28 | The markdown output for various types of Charts.

100% Stacked Bar Chart



Agency	Favorable	Not Sure	Unfavorable
National Park Service	81%	12%	7%
U.S. Postal Service	77%	3%	20%
NASA	74%	17%	9%
Social Security Administration	61%	12%	28%
CDC	56%	6%	38%
Veterans Affairs	56%	16%	28%
EPA	55%	14%	31%
Health and Human Services	55%	15%	30%
FBI	52%	12%	36%
Department of Transportation	52%	12%	36%
Department of Homeland Security	51%	13%	35%
Department of Justice	49%	10%	41%
CIA	46%	21%	33%
Department of Education	45%	8%	47%
Federal Reserve	43%	20%	37%
IRS	42%	7%	51%

Bar-Line Hybrid Chart

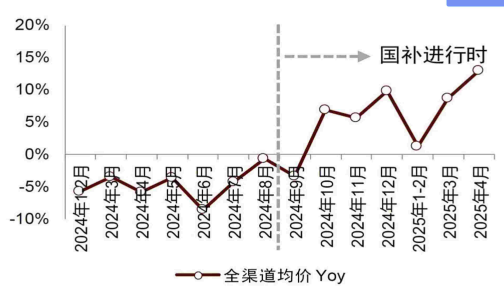


年份	即配订单规模 (亿单)	增速%
2019	188.3	
2020	215.3	15.0%
2021	281.7	31.0%
2022	332.8	18.0%
2023	408.8	22.0%
2024	485.6	18.0%
2025E	558.3	14.0%
2026E	640.6	14.0%
2027E	724.1	12.0%

Pie Chart

类别	数值	百分比
住房	3600	28.13%
食品	2400	18.75%
服装	1800	14.06%
交通	1500	11.72%
教育	1200	9.37%
医疗	900	7.03%
娱乐	800	6.25%
其他	600	4.69%

Line Chart



月份	全渠道均价 Yoy
2024年1-2月	-5%
2024年3月	-4%
2024年4月	-5%
2024年5月	-4%
2024年6月	-8%
2024年7月	-5%
2024年8月	-1%
2024年9月	-3%
2024年10月	7%
2024年11月	6%
2024年12月	10%
2025年1-2月	1%
2025年3月	9%
2025年4月	13%

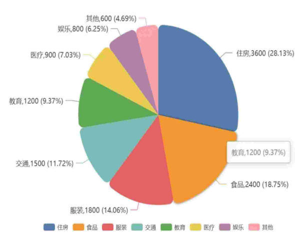


Figure A29 | The markdown output for various types of Charts.

E. Compare with Others

PaddleOCR-VL showcases superior performance in scenarios involving PDF pages with complex layout, consistently outperforming existing state-of-the-art (SOTA) models. This is evident from Figures A30 and A31, which highlight its exceptional capability in handling pages with intricate layouts and unique elements, surpassing other solutions.

Moreover, the model demonstrates exceptionally high recognition accuracy in several domains, including Multilingual Text Recognition, Handwriting Text Recognition, and Vertical Text Recognition. Figures A32- A37 illustrate how PaddleOCR-VL outperforms competitors such as MinerU2.5 [2] and MonkeyOCR [1], which tend to misidentify languages like Russian and Hindi as English, overlook some handwritten characters, and struggle with vertical text recognition.

In dealing with complex tables, PaddleOCR-VL's parsing accuracy stands out, as evidenced by Figures A38 and A39. This is a domain where other models frequently encounter difficulties.

Additionally, Figure A40 demonstrates PaddleOCR-VL's proficiency in accurately parsing complex formulas. In contrast, other SOTA models often produce incorrect or flawed outputs when faced with challenging mathematical notations.

Finally, as depicted in Figures A41 and A42, PaddleOCR-VL also excels in Chart Recognition. It outperforms multi-modal large language models like Qwen2.5VL-72B [24] and GPT-4o by accurately reconstructing the structure and content of charts.

E.1. Layout Detection

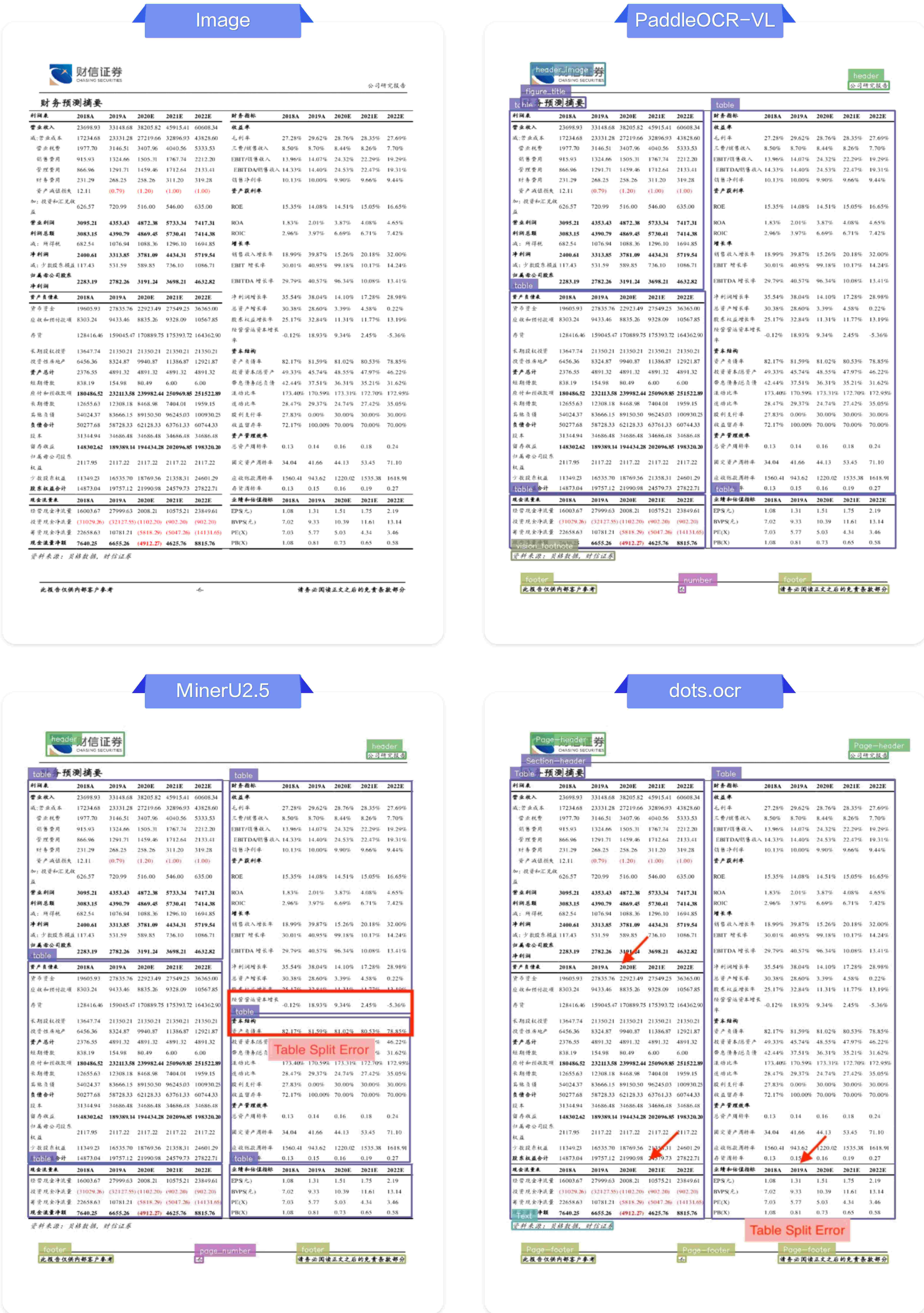


Figure A30 | Compare with others in Layout Detection.

Figure A31 | Compare with others in Layout Detection.

E.2. Text Recognition

E.2.1. Multilingual Text Recognition

Figure A32 | Compare with others in Multilingual Text Recognition.

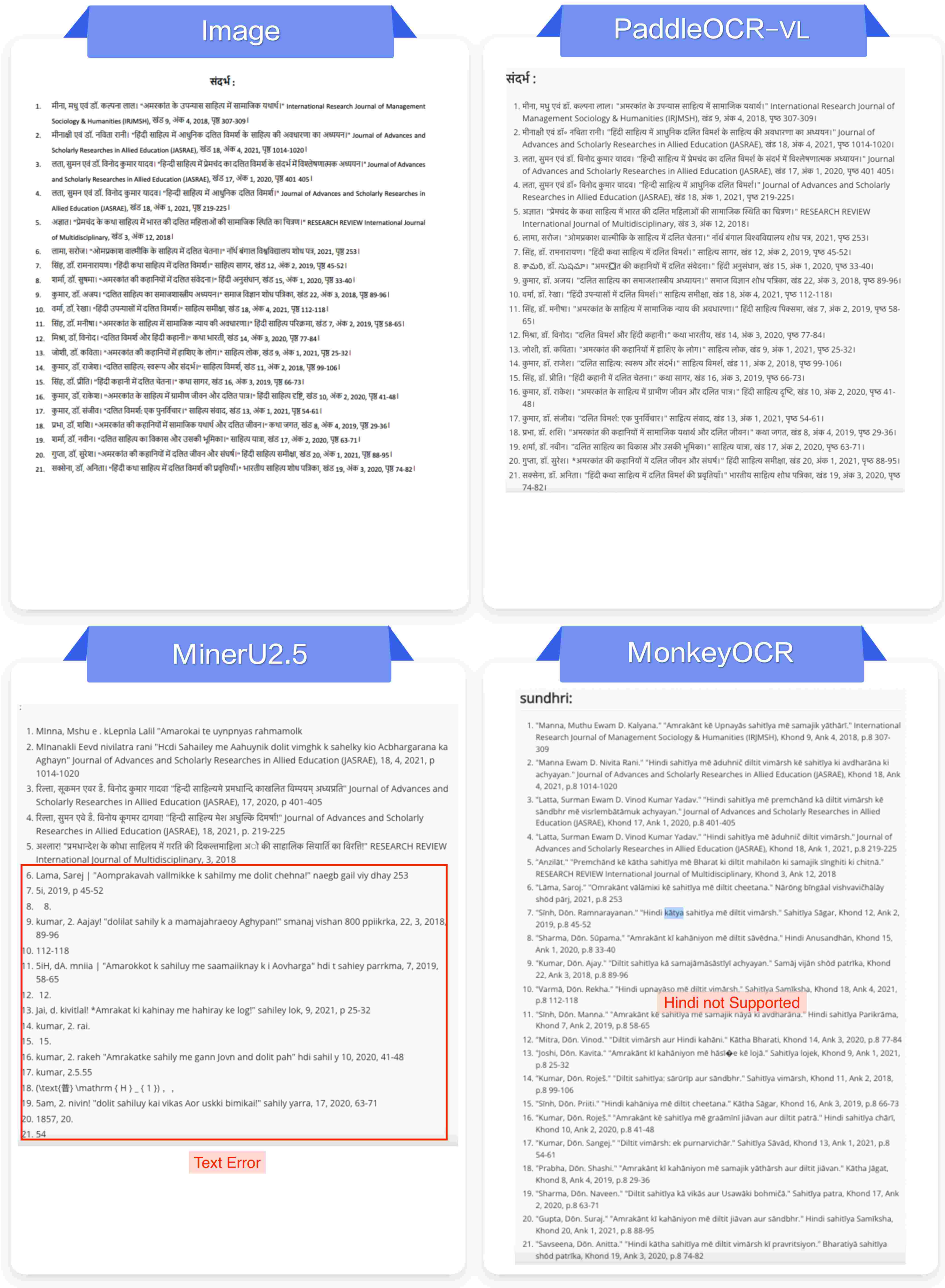


Figure A33 | Compare with others in Multilingual Text Recognition.

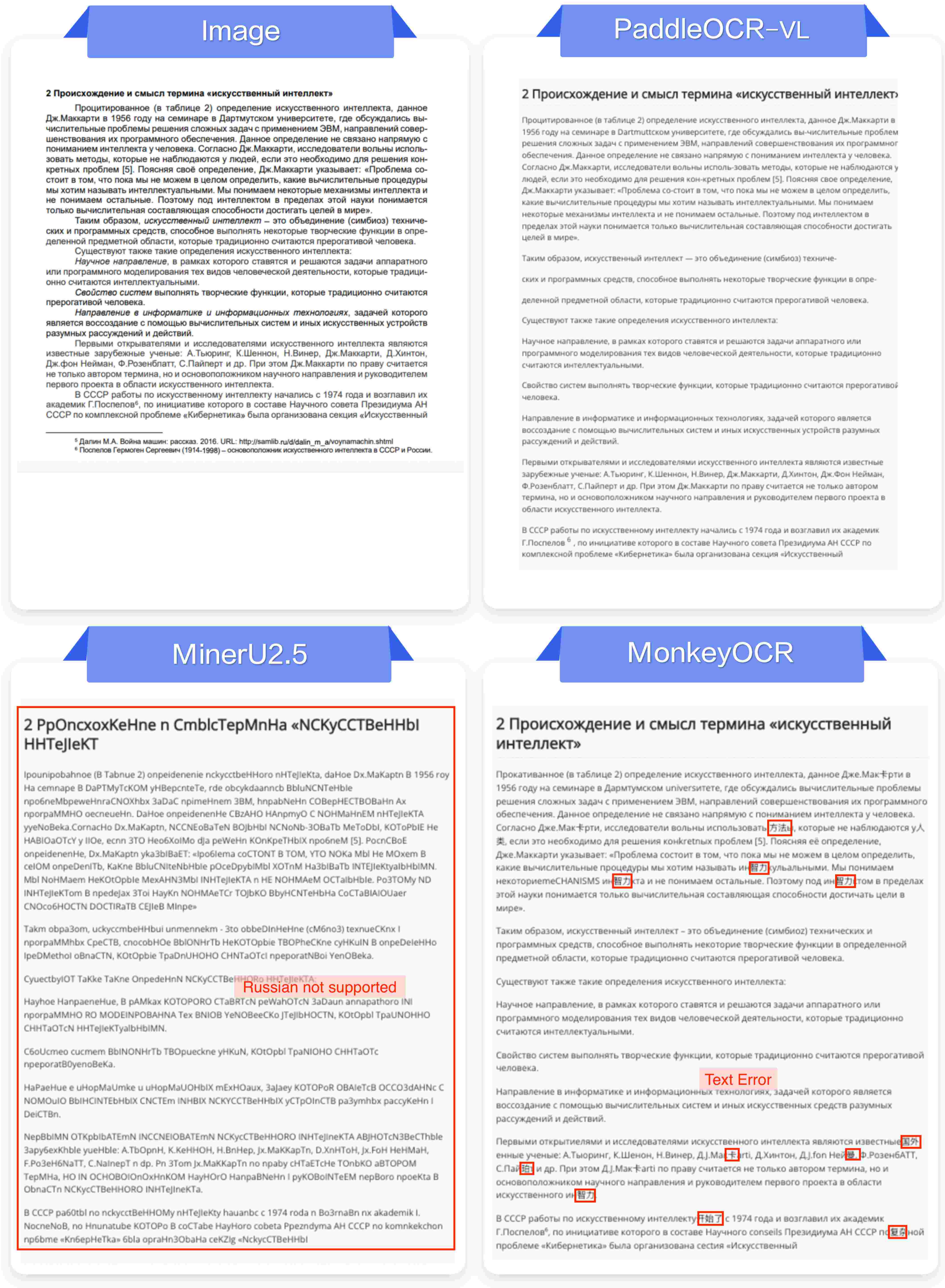


Figure A34 | Compare with others in Multilingual Text Recognition.

E.2.2. *Handwriting Text Recognition*

Figure A35 | Compare with others in Handwriting Text Recognition.

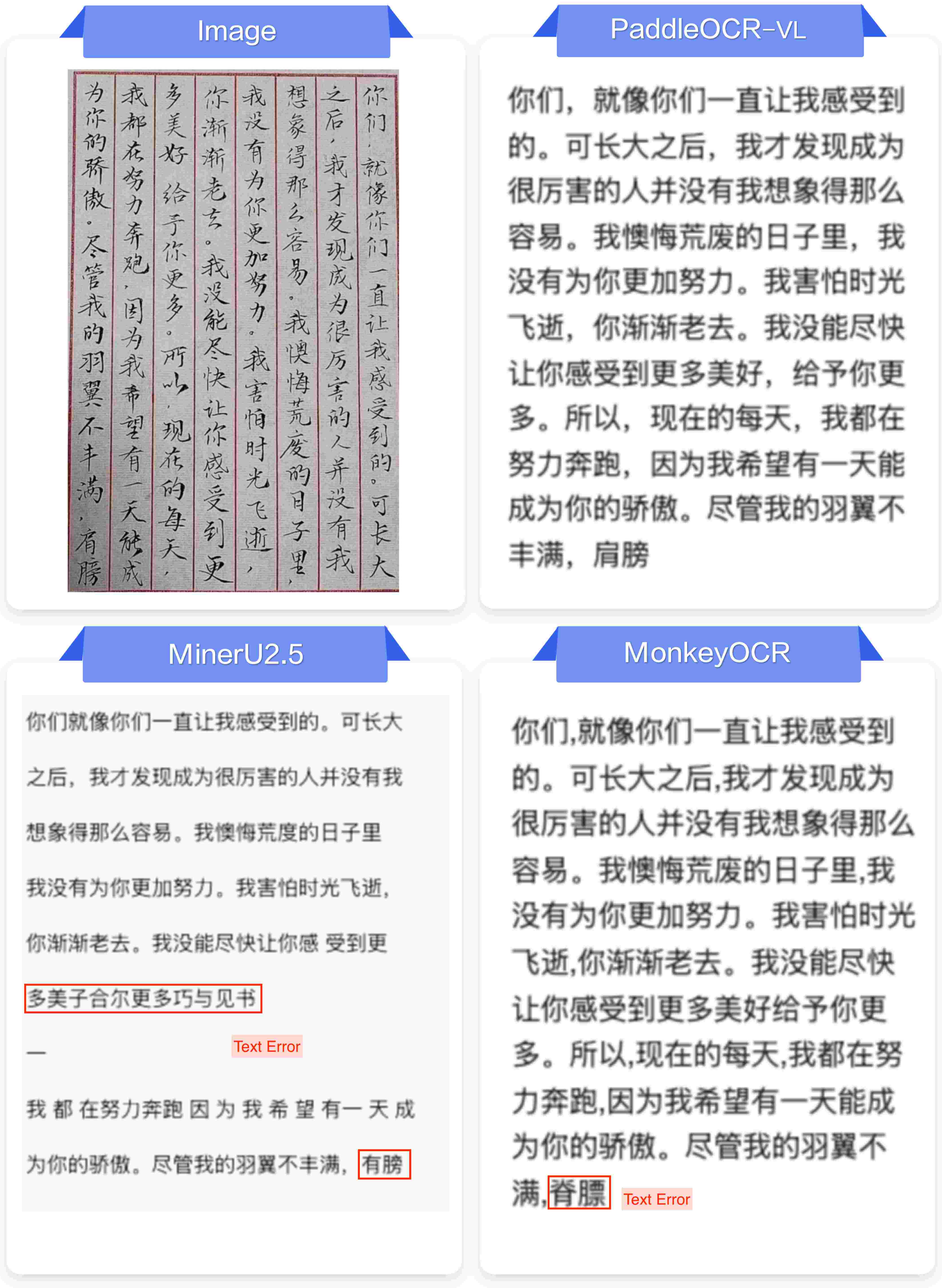
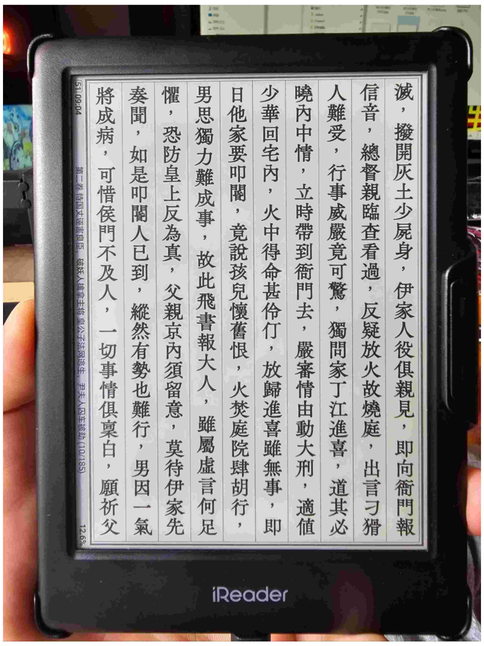


Figure A36 | Compare with others in Handwriting Text Recognition.

E.2.3. Vertical Text Recognition

Image



PaddleOCR-VL

滅，撥開灰土少屍身，伊家人役俱親見，即向衙門報信音，總督親臨查看過，反疑放火故燒庭，出言刁猾人難受，行事威嚴竟可驚，獨問家丁江進喜，道其必曉內中情，立時帶到衙門去，嚴審情由動大刑，適值少華回宅內，火中得命甚伶仃，放歸進喜雖無事，即日他家要叩閽，竟說孩兒懷舊恨，火焚庭院肆胡行，男思獨力難成事，故此飛書報大人，雖屬虛言何足懼，恐防皇上反為真，父親京內須留意，莫待伊家先奏聞，如是叩閽人已到，縱然有勢也難行，男因一氣將成病，可惜侯門不及人，一切事情俱稟白，願祈父

MinerU2.5

將成病，可惜侯門不及人，一切事情俱稟白，願祈父

奏聞，如是叩閽人已到，縱然有勢也難行，男因一氣

懷，恐防皇上反為真，父親京內須留意，莫待伊家先

Incorrect reading order

男思獨力難成事，故此飛書報大人，雖屬虛言何足

日他家要叩閼，竟說孩兒懷舊恨，火焚庭院肆胡行，

少華回宅內，火中得命甚伶仃，放歸進喜雖無事，即

曉內中情，立時帶到衙門去，嚴審情由動大刑，適值

人難受，行事威嚴竟可驚，獨問家丁江進喜，道其必

信音，總督親臨查看過，反疑放火故燒庭，出言刁猾

滅，撥開灰土少屍身，伊家人役俱親見，即向衙門報

MonkeyOCR

男思独力难成事，故此飞书报大人，虽属虚言何足恢，恐防皇上反为真，父亲京内须留意，莫待伊家先奏闻，如是叩衙人已到，纵然有势也难行，男因一气将成病，可惜侯门不及人，一切事情俱禀白，愿祈父

Incorrect reading order

滅，拨開灰土少尸身，伊家人役俱親見，即向衙門報信音，總督親臨查看過，反疑放火故烧庭，出言刁猾人難受，行事威嚴竟可惊，獨問家丁江進喜，道其必晓内中情，立時带到衙门去，嚴審情由動大刑，適值少hra回宅内，火中得命甚伶仃，放歸進喜虽無事，即日他家要叩都是非常，竟設孩儿怀古恨，火焚庭院肆胡行，

Figure A37 | Compare with others in Vertical Text Recognition.

E.3. Table Recognition

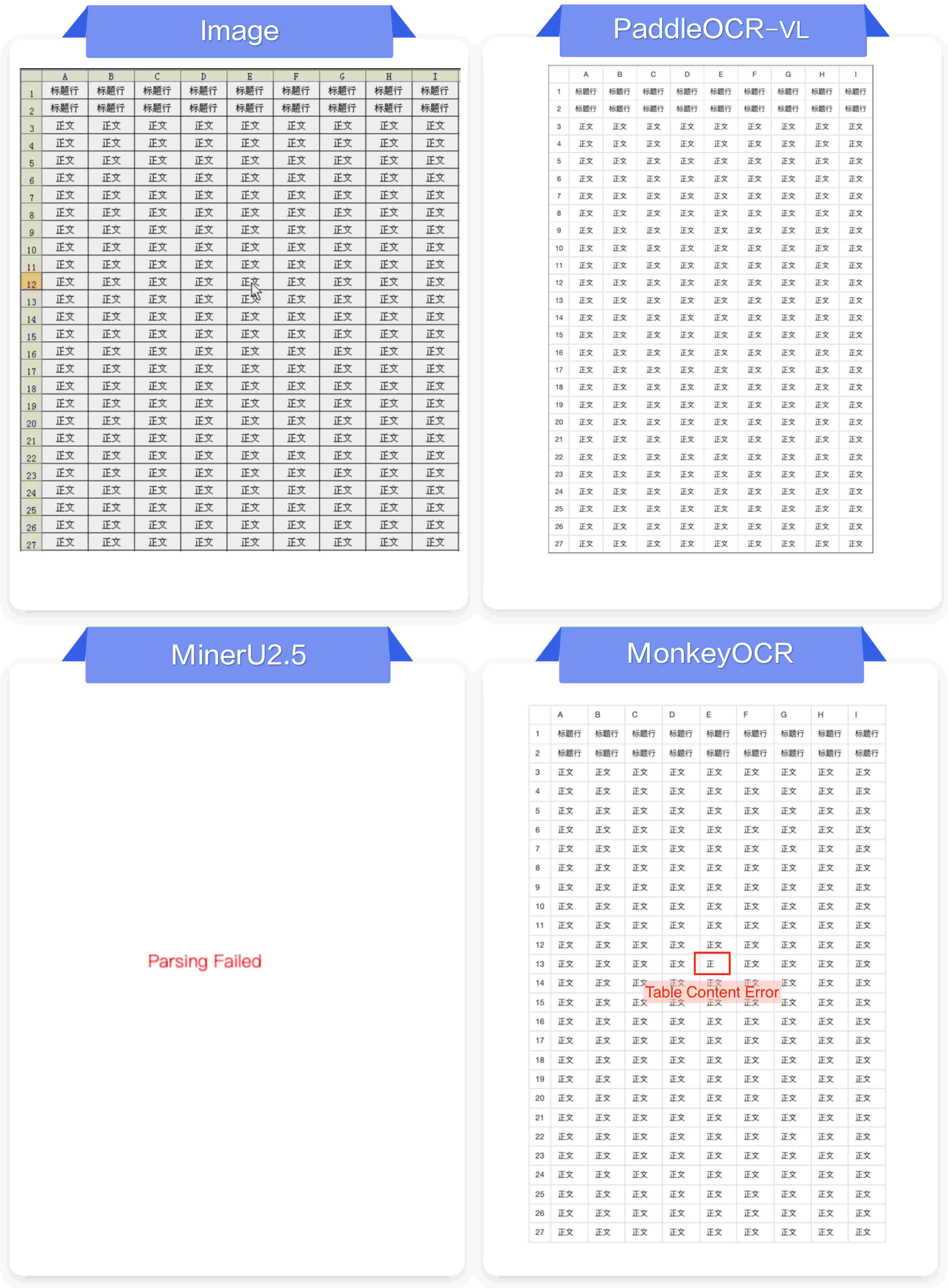


Figure A38 | Compare with others in Table Recognition.

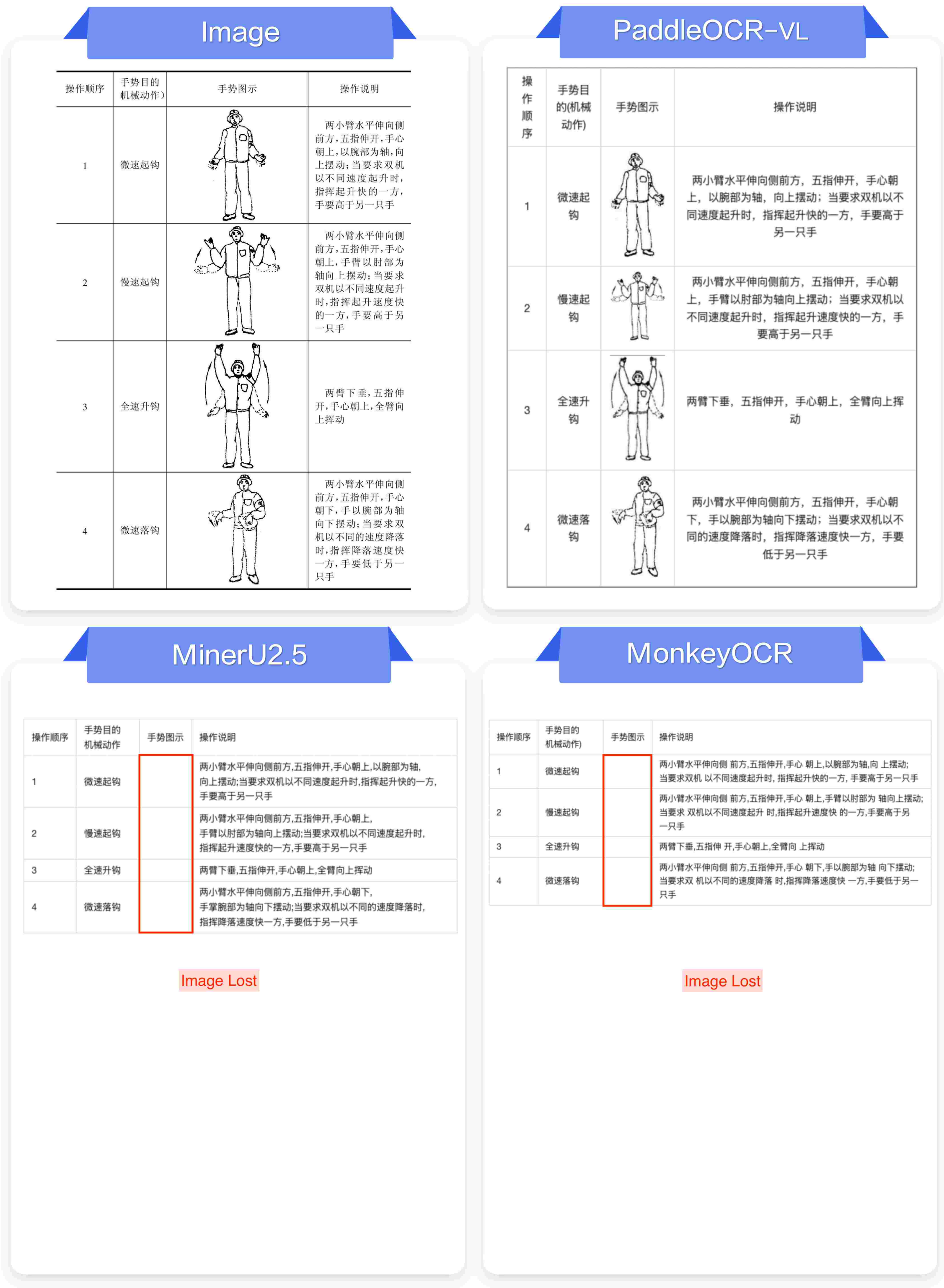


Figure A39 | Compare with others in Table Recognition.

E.4. Formula Recognition

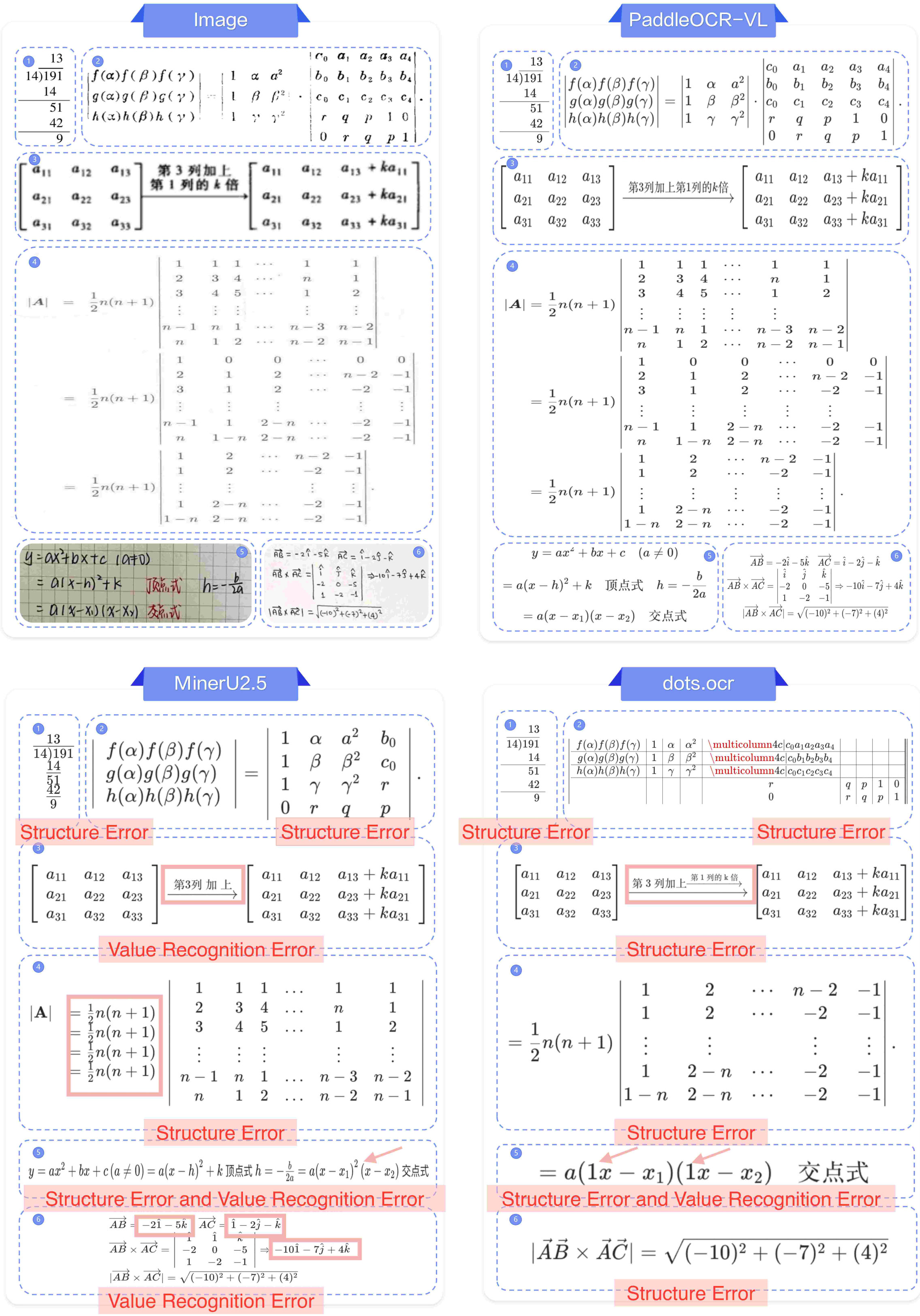
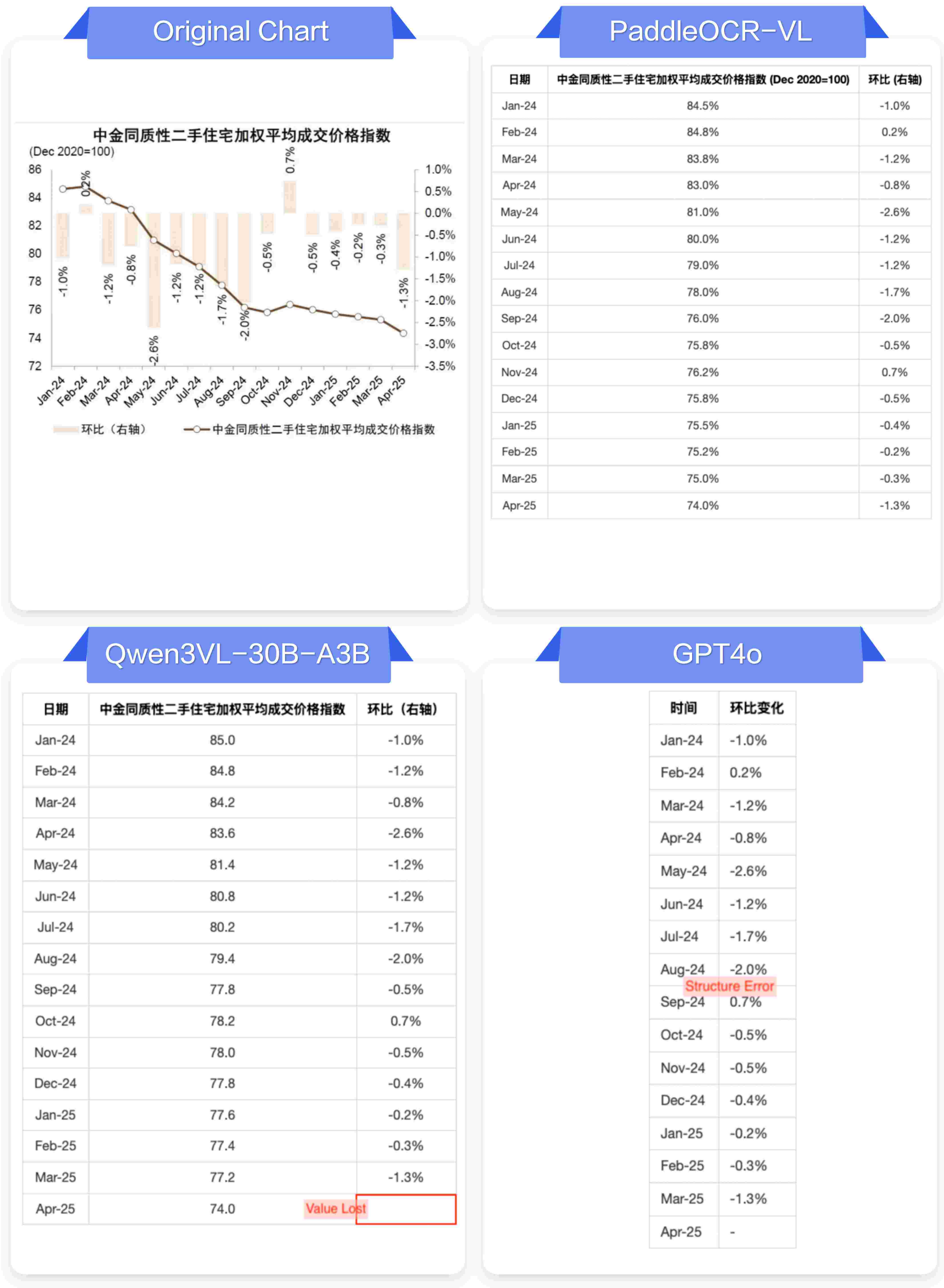


Figure A40 | Compare with others in Formula Recognition.

E.5. Chart Recognition

Figure A41 | Compare with others in Chart Recognition.



日期	中金同质性二手住宅加权平均成交价格指数 (Dec 2020=100)	环比 (右轴)
Jan-24	84.5%	-1.0%
Feb-24	84.8%	0.2%
Mar-24	83.8%	-1.2%
Apr-24	83.0%	-0.8%
May-24	81.0%	-2.6%
Jun-24	80.0%	-1.2%
Jul-24	79.0%	-1.2%
Aug-24	78.0%	-1.7%
Sep-24	76.0%	-2.0%
Oct-24	75.8%	-0.5%
Nov-24	76.2%	0.7%
Dec-24	75.8%	-0.5%
Jan-25	75.5%	-0.4%
Feb-25	75.2%	-0.2%
Mar-25	75.0%	-0.3%
Apr-25	74.0%	-1.3%

日期	中金同质性二手住宅加权平均成交价格指数	环比（右轴）
Jan-24	85.0	-1.0%
Feb-24	84.8	-1.2%
Mar-24	84.2	-0.8%
Apr-24	83.6	-2.6%
May-24	81.4	-1.2%
Jun-24	80.8	-1.2%
Jul-24	80.2	-1.7%
Aug-24	79.4	-2.0%
Sep-24	77.8	-0.5%
Oct-24	78.2	0.7%
Nov-24	78.0	-0.5%
Dec-24	77.8	-0.4%
Jan-25	77.6	-0.2%
Feb-25	77.4	-0.3%
Mar-25	77.2	-1.3%
Apr-25	74.0	

时间	环比变化
Jan-24	-1.0%
Feb-24	0.2%
Mar-24	-1.2%
Apr-24	-0.8%
May-24	-2.6%
Jun-24	-1.2%
Jul-24	-1.7%
Aug-24	-2.0%
Sep-24	0.7%
Oct-24	-0.5%
Nov-24	-0.5%
Dec-24	-0.4%
Jan-25	-0.2%
Feb-25	-0.3%
Mar-25	-1.3%
Apr-25	-

Figure A42 | Compare with others in Chart Recognition.