LongDocURL: a Comprehensive Multimodal Long Document Benchmark Integrating *Understanding*, *Reasoning*, and *Locating*

Chao Deng[1,2,3*†], **Jiale Yuan**[4*], **Pi Bu**[4], **Peijie Wang**[1,2], **Zhong-Zhi Li**[1,2], **Jian Xu**[1,2], **Xiao-Hui Li**[1,2], **Yuan Gao**[4], **Jun Song**[4‡], **Bo Zheng**[4], **Cheng-Lin Liu**[1,2,3‡]
MAIS, Institute of Automation, Chinese Academy of Sciences[1]
School of Artificial Intelligence, University of Chinese Academy of Sciences[2]
Zhongguancun Academy, Beijing[3] Taobao & Tmall Group of Alibaba[4]
dengchao2023@ia.ac.cn, liucl@nlpr.ia.ac.cn
{yuanjiale.yjl, bupi.wj, jsong.sj}@alibaba-inc.com

Abstract

Large vision language models (LVLMs) have improved the document understanding capabilities remarkably, enabling the handling of complex document elements, longer contexts, and a wider range of tasks. However, existing document understanding benchmarks have been limited to handling only a small number of pages and fail to provide a comprehensive analysis of layout elements locating. In this paper, we first define three primary task categories: **Long Document Understanding**, numerical **Reasoning**, and cross-element **Locating**, and then propose a comprehensive benchmark—**LongDocURL**—integrating above three primary tasks and comprising 20 sub-tasks categorized based on different primary tasks and answer evidences. Furthermore, we develop a semi-automated construction pipeline and collect 2,325 high-quality question-answering pairs, covering more than 33,000 pages of documents, significantly outperforming existing benchmarks. Subsequently, we conduct comprehensive evaluation experiments on both open-source and closed-source models across 26 different configurations, revealing critical performance gaps in this field. The code and data: https://github.com/dengc2023/LongDocURL.

1 Introduction

The research of document understanding has been advanced remarkably in the last decade. However, past works mostly rely on smaller, specialized models, necessitating the design of independent models for each specific task (*e.g.*, document structure parsing). This strategy not only increases the labor in model development but also limits the applicability of the models. Recently, this field has undergone transformation with the rise of large language models (LLMs) and large vision-language models (LVLMs), such as the open-source InternLM-XC2-4KHD (Dong et al., 2024b) and TextMonkey (Liu et al., 2024). These models are showcasing their capabilities in handling complex document elements like charts and images, managing longer contexts up to 128k or more, and tackling diverse tasks in addition to OCR task, such as table question answering and layout understanding.

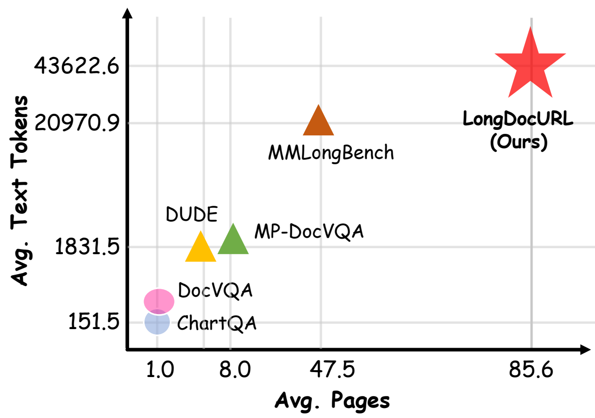


Figure 1: Comparison with other datasets in average pages and text tokens per document.

Despite the advances in model capabilities, the evaluation of complex document tasks is somewhat lacking. Take DocVQA (Mathew et al., 2021) as an example; it is one of the standard benchmarks for document understanding, but it can only assess single-page documents, and many models can easily exceed an accuracy of 95% (*e.g.*, Qwen2-VL (Alibaba, 2024)). The benchmarks detailed in Table 1 exhibit limitations in adequately addressing complex elements, longer contexts, and diverse tasks: **1) Complex elements:** Most benchmarks fail to cover all elements such as paragraphs, titles, tables, and figures, focusing instead on only some of the contents. Additionally, discussions about the interrelations among different elements are scarce. **2) Longer contexts:** Current benchmarks for multi-page document question answering, such as MP-DocVQA (Tito et al., 2022) and DUDE (Lan-

* Equal contribution.
† Work done during an internship at Alibaba Group.
‡ Corresponding authors.

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1135–1159
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

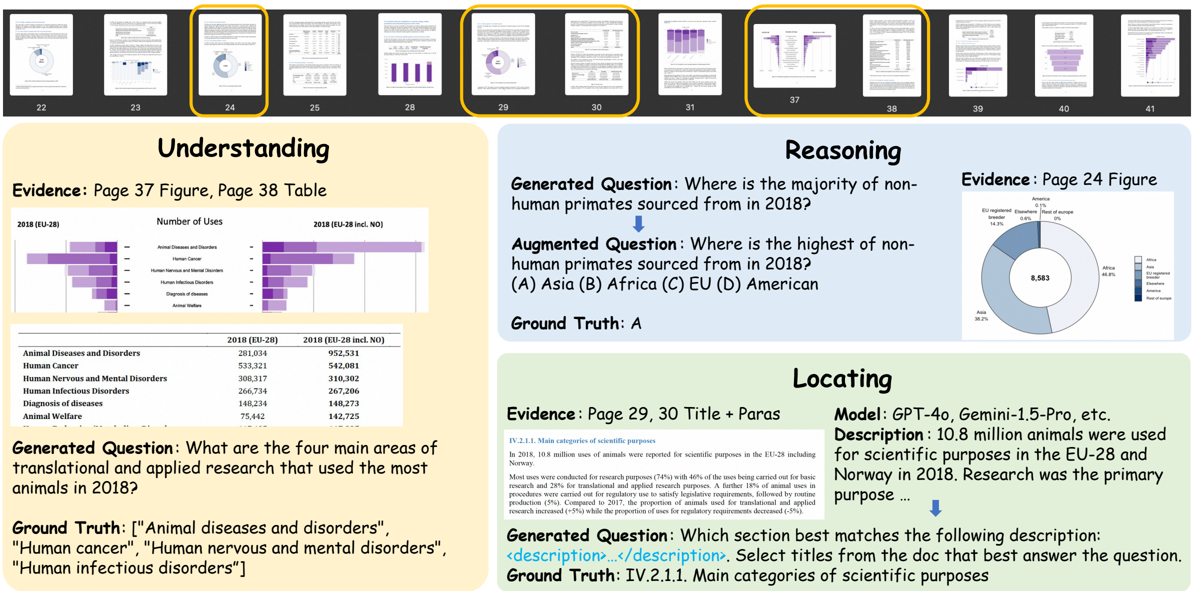


Figure 2: LongDocURL comprises 20 sub-tasks focusing on three task categories: **U**nderstanding, numerical **R**easoning, and cross-element **L**ocating. (Top) Thumbnail of a document example. Orange boxes indicate answer evidence pages. (Bottom) Data examples generated from the document and screenshots of relevant part of answer evidence pages.

deghem et al., 2023), do not assess documents exceeding 20 pages. While MMLongBench-Doc (Ma et al., 2024) collects longer documents, it offers only approximately 1k effective samples, with only about 30% of questions involving cross-page information. **3) Diverse tasks:** Existing work focuses more on OCR or easy question-answering tasks, neglecting the exploration of capabilities in other areas such as cross-element locating task (as shown in Figure 2c). The above findings indicate that existing benchmarks lag behind the advances of models, which could hinder the development of document understanding.

In this paper, we present a comprehensive document benchmark including three task categories: 1) **Understanding**: extracting information from documents by identifying keywords, parsing the structure of tables, *etc*. Answers are found directly in the document. 2) Numerical **Reasoning**: processing numerical information through counting, calculating, comparing, and summarizing, requiring both extracted information and reasoning for concluding. 3) Cross-Element **Locating**: As mentioned earlier, discussions about the interrelations among different types of elements are scarce. It is often necessary to establish a task that evaluates models' ability to analyze relations among different types of elements. For instance, in Para-Title Locating task, as shown in Figure 2c, models must summarize relevant sections to identify parts that match a given abstract and then determine the relation between the paragraph and its section titles. This task requires switching element types (*i.e.*, paragraphs to titles) during the answering process.

Our benchmark, named **LongDocURL**, comprises 20 sub-tasks according to different primary tasks and answer evidence. More details are presented in Section 3. To efficiently assemble the evaluation dataset for LongDocURL, we design a semi-automated pipeline comprising four modules. Specifically, a Extract & Filter module identifies documents of suitable length with rich layouts from diverse document sources. A QA Generation module utilizes a multi-step iterative querying process with advanced models (*e.g.*, GPT-4o) to generate QA pairs with evidence sources. Finally, the Automated Verification and Human Verification modules ensure the quality of the generated content. Through this semi-automated pipeline, we ultimately produce 2,325 QA pairs, covering more than 33,000 pages of documents. Thereafter, we conduct comprehensive evaluation experiments with 26 different configurations (varying the model and input format). These evaluation results indicate that the highest-performing closed-source model, GPT-4o, scored 64.5, leading all models, while the best score for open-source models is only 30.6. This result reveals a potential gap in document understanding and shows the need for further improvement. Our contributions are as follows:

- We introduce three primary tasks of long document and propose a comprehensive benchmark

Benchmarks	Data Size				Answer Evidence		Task Type		
	#Docs	#Avg. Pages	#Avg. Tokens	#QA	Multi-page(%)	Cross-element(%)	U	R	L
single-page									
DocVQA (Mathew et al., 2021)	-	1.0	151.5	-	✗	not specified	✓	✗	✗
ChartQA (Masry et al., 2022)	-	1.0	236.9	-	✗	not specified	✓	✗	✗
multi-page(<=20)									
MP-DocVQA (Tito et al., 2022)	-	8.3	2,026.6	-	✗	not specified	✓	✗	✗
DUDE (Landeghem et al., 2023)	-	5.7	1,831.5	-	✓(2.1%)	not specified	✓	✓	✗
multi-page(>20)									
MMLongBench-Doc (Ma et al., 2024)	135	47.5	21,214.1	1,082	✓(33.0%)	✓(22.6%)	✓	✓	✗
M-Longdoc (Chia et al., 2024)	180	210.8	-	851	-	-	✓	✓	✗
LongDocURL (Ours)	396	85.6	43,622.6	2,325	✓(52.9%)	✓(37.1%)	✓	✓	✓

Table 1: Comparison between LongDoc**URL** and previous document understanding datasets. Task types: (**U**)nderstanding, (**R**)easoning, and (**L**)ocating.

comprising 20 sub-tasks categorized based on different primary tasks and answer evidences to support more fine-grained evaluation.

- We develop a cost-efficient semi-automated construction pipeline and generate 2,325 high-quality QA pairs, covering more than 33,000 pages of documents, which significantly outperforms existing benchmarks.

- We conduct comprehensive evaluation experiments of both open-source and closed-source models under 26 different configurations.

2 Related Work

Models for Document Understanding. There are two main types of language models for document understanding: (1) OCR-dependent models, which use Optical Character Recognition (OCR) to extract text for processing, including the LayoutLM series (Xu et al., 2020, 2021; Huang et al., 2022) and text-only LLMs (Meta, 2024; QwenTeam, 2024); and (2) end-to-end models, which use a visual encoder to extract features from document images, integrating them with text for input into language model backbones. Most of document-related LVLMs fall into this category, such as GPT4o (OpenAI, 2024), Gemini-1.5 (GeminiTeam, 2024), Claude-3.5-Sonnet (Anthropic, 2024), mPLUG-DocOwl2 (Hu et al., 2024), Qwen2-VL (Alibaba, 2024).

Methods for Long Document Understanding. To address the challenges of cross-page document understanding with excessively long context lengths, early approaches employed hierarchical encoding methods (Tito et al., 2022; Kang et al., 2024; Cho et al., 2021; Dong et al., 2024a). In these approaches, an independent encoder processes the OCR text and visual modal information for each page, which is then passed to a small language model decoder for cross-page contextual learning. However, this approach is limited by the redundancy in OCR inputs, which restricts the context length and leads to the accumulation of errors (Xu et al., 2020, 2021; Appalaraju et al., 2021). Recently, with the rise of multimodal large models, methods based on multimodal Retrieval-Augmented Generation (MM-RAG) (Yu et al., 2024; Blau et al., 2024; Ding et al., 2024a; Zhang et al., 2024a,b; Cho et al., 2024) and end-to-end multi-page large models (Jiang et al., 2024; Li et al., 2024; Jia et al., 2024) have emerged. These models leverage the world knowledge of large language models to enhance understanding. End-to-end approaches mitigate error accumulation by dynamically reducing the number of visual tokens across multiple pages/images and build large scale instruction turning dataset (Jiang et al., 2024; Li et al., 2024; Jia et al., 2024; Hu et al., 2024), allowing for longer context lengths. Methods such as multi-page RAG facilitate dynamic interactions with OCR and other text information to remove redundant multimodal tokens.

Benchmarks for Long Document Understanding. Multi-page or long documents place higher demands on the model's capabilities in cross-page understanding. Current multi-page document benchmarks, such as MP-DocVQA and DUDE, do not assess documents exceeding 20 pages. MMLongBench-Doc (Ma et al., 2024) and M-Longdoc (Chia et al., 2024) have been proposed to evaluate the understanding capabilities of longer documents, which have an average of 47.5 and 210.8 pages per document, respectively. Meanwhile, MMVQA (Ding et al., 2024a) and MVQA (Ding et al., 2024b) are proposed to better evaluate the retrieval-based method. WebQuest (Wang et al., 2024) focuses on evaluating models' performance on text-rich web images.

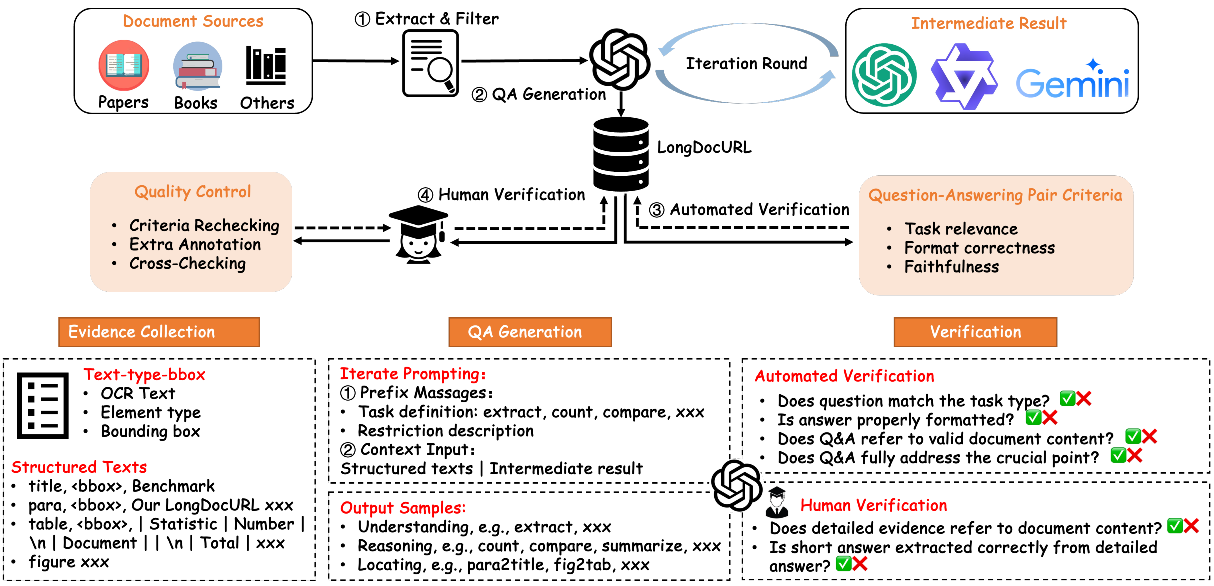


Figure 3: Overview of our semi-automated construction pipeline. The pipeline comprises four modules: (a) *Extract & Filter*; (b) *QA Generation*; (c) *Automated Verification*; (d) *Human Verification*.

3 LongDocURL

3.1 Overview

Firstly, each question-answering pair can be categorized by three primary tasks: *Understanding*, *Reasoning*, and *Locating*, as discussed in Section 1. Secondly, we define four types of answer evidences based on element type: (1) *Text*: pure texts, such as paragraph; (2) *Layout*: text elements with special layout meaning (generalized text), such as title, header, footer, table name and figure name; (3) *Figure*: including charts and general images. (4) *Table*. In addition, each question-answering pair can be classified into *single-page* or *multi-page* based on the number of answer evidence pages and *single-element* or *cross-element* based on the number of types of evidence elements. Based on different primary tasks and answer evidences, we divide our dataset into 20 sub-tasks. As shown in Figure 4, for the *Understanding* or *Reasoning* task, we divide our dataset into 8 sub-tasks according to the number of answer evidence pages. Compared to the two tasks, we pay more attention to the inter-relations among different types of elements in the *Locating* task and we build 4 sub-tasks based on the combination of different element types. Data examples are presented in Appendix F.

3.2 Q&A Construction

3.2.1 Evidence Collection

To objectively evaluate LVLM long document question-answering comprehension ability, we first crawl 200k PDF-formatted documents from CommonCrawl[1] and filter them by page length(*i.e.*, 50~150) and language(*i.e.*, English) to create a candidate set of approximately 3,000 documents. Then, we categorize these candidates by document type. Specifically, we randomly select 5~10 pages from a document, and prompt GPT-4o to classify its document type based on document content and layout. We finally retain 396 documents to construct our benchmark. These documents span eight types: *research reports & papers*, *user manuals & guides*, *books & e-books*, *theses & dissertations*, *work & project summaries*, *presentation materials*, *project proposals*, and *meeting minutes & summaries*, with an average of 85.6 pages per document.

Thereafter, we utilize both PyMuPDF[2] and Docmind[3] to parse the PDF and extract texts and layout information from the documents. For instance, the tables are converted into markdown format with Docmind. We organize the extracted results in the format of ***text-type-bbox*** triples as a symbolic representation of elements: 1) ***text***: the recognized text of the elements; 2) ***type***: the element type. 3) ***bbox***: the bounding box of the element. Notably, we build the element triples at the region level, such as paragraph, table, chart, footnote and title, instead of the line level.

[1] https://corp.digitalcorpora.org/corpora/files/CC-MAIN-2021-31-PDF-UNTRUNCATED/

[2] https://pymupdf.readthedocs.io

[3] https://www.aliyun.com/product/ai/docmind

Figure 4: Our LongDocURL comprises 20 sub-tasks. **Inner**: divided by the primary task categories (Understanding, Reasoning, and Locating). **Middle**: divided by the number of answer evidence pages (Single-Page, Multi-Page), and the number of types of evidence elements (Cross-Element). **Outer**: divided by the types of evidence elements (Text, Table, Figure, Layout).

3.2.2 Q&A Generation

Directly prompting LVLM to generate conversational question-answering pairs based on a single or multiple document images proves often ineffective, due to the inability to fully parse diverse elements present in the documents. Similar to LLaVA (Liu et al., 2023), we adopt a two-stage pipeline: Initially, we parse our PDF-formatted documents and get the "text-type-bbox" triples discussed in Section 3.2.1. Subsequently, we design prompts to query LLMs/LVLMs in a multi-step round, and finally generate question-answering pairs. Specifically, as shown in Figure 3, we present the definition of each task and description of related restriction as a part of the prompts. Details can be found in Appendix B.

3.3 Q&A Verification

3.3.1 Automated Verification

We design an automated method to verify the quality of synthesized question-answering pairs to identify and correct corresponding issues. As shown in Figure 3, a qualified question-answering pair should be verified using these criteria: (1) **Task Relevance**. (2) **Format Correctness**. (3) **Faithfulness**.

The verification results are utilized to classify samples: those that fail are marked as negative, and those that pass are marked as positive. We analyze the verification results and observe approximately 75.2% of the raw data in the Cross-Title Locating task are marked as negative samples, while the percentage is 19.6% in the Cross-Table Locating task. Additional statistics data is presented in Table 5. In the next stage (Section 3.3.2), we present both the final result and texts of verification chains to the human annotator as reference information to guide further verification.

Statistic	Number
Document	
- Total	396
- Type	8
- Avg. pages	85.6
Question & Answer	
- Total	2,325
- Avg. question tokens	35.5
- Max. question tokens	277
Task Type	
- Understanding	1,243 (53.5%)
- Reasoning	387 (16.6%)
- Locating	695 (29.9%)
Type of Evidence Element	
- Pure-text	994 (42.8%)
- Layout	779 (33.5%)
- Table	556 (23.9%)
- Figure	871 (37.5%)
Number of Evidence Pages	
- Single-page question	1,093 (47.0%)
- Multi-page questions	1,230 (52.9%)
- Unanswerable questions	2 (0.1%)
Number of Evidence Element Types	
- Single-element question	1,463(62.9%)
- Cross-element question	862(37.1%)
Answer Format	
- String	941 (40.5%)
- Integer	431 (18.5%)
- Float	185 (8.0%)
- List	757 (32.6%)
- None	11 (0.5%)

Table 2: Document and Q&A statistics based on different tasks and answer evidences.

3.3.2 Human Verification

There are two shortcomings of automated verification in Section 3.3.1: (1) It only completes the classification of positive and negative samples, but does not recycle them, which causes waste. (2) When verifying the consistency between the question-answer pair and the document, only the text information is referenced, which may lost significant visual structure information compared to the source document. In the human verification stage, we focus on tasks that are challenging for automated verification, such as recovering negative samples and checking the consistency between the question-answering pair and the visual source information. The Human Verification module comprises the following parts:

- **Criteria Rechecking.** The annotator reviews the intermediate process of machine verification to determine whether it is reasonable and whether the negative samples can be recovered by simply correcting the question-answer pair.
- **Extra Annotation.** Human annotators should oversee the question-answer pair process using the source visual document, not the parsed one. First, they need to verify if the evidence provided by the model aligns with the document's content, and then ensure that the final answer can be derived step by step from the intermediate process.
- **Cross-Checking.** After the annotators complete the first round of annotation, we require the annotators to cross-check each other's annotation results to improve the quality of annotation.

The labor resource allocation for data annotation is detailed in Appendix H.

3.4 Dataset Statistics

As shown in Table 2 and Figure 8, our LongDocURL comprises 396 documents and 2,325 question-answering samples. The average pages per document range from 50 to 150. The cross-element locating task we designed contributes 37.1% of the data, providing support for evaluating the model's cross-element interaction capabilities in long contexts. The dataset includes five types of answers, ensuring compatibility with automated evaluation while maintaining completeness and accuracy. In addition, we compare the characteristics of question set in our LongDocURL with that in MMLongBench-Doc. The results are presented in Appendix C.

4 Experiments

4.1 Evaluation Protocols

Allowing models to think freely can enhance performance but often results in lengthy responses. Current rule-based scorers are only effective with specific short formats, making it difficult to evaluate longer responses. Therefore, we need an *answer extractor* (GPT4o) to convert detailed responses into concise ones. Following MATHVISTA (Lu et al., 2024) and MMLongBench-Doc (Ma et al., 2024), we implement a three-stage protocol: *response generation*, *answer extraction*, and *score calculation*. In the *score calculation* stage, we divide scores into five categories (*Integer*, *Float*, *String*, *List*, and *None*). Different from MMLongBench-Doc, we adopt a softer and more reasonable scoring standard. See Appendix D for more details.

4.2 Experimental Setup

Models We divide the experiments into two categories: **text-input** and **image-input**. For text-input configuration, document texts parsed by OCR engines are input into LLMs/LVLMs. We conduct our experiments on both open-source and closed-source models across 26 different configurations.

Input Paradigm The documents in our LongDocURL have a max of 150 pages, and most of models are unable to fully process the context to obtain the answer due to GPU memory or interface limitations. Therefore, similar to the **merge** method used in previous benchmarks (Ma et al., 2024), we design the **cut-off** paradigm for the evaluation of current models.

For LVLMs, we cut 30 continuous pages around the answer evidences from the original PDF-formatted document, and feed the converted images into the models. Details of selection rules are in Appendix D.2. As for LLMs, we input the texts parsed by OCR engines, including PyMuPDF and Docmind.

Other Configurations We assess the proprietary models using API resources, while the evaluation of the open-source models is conducted on H20 machines with 96G memory. To reduce variance, we set the temperature coefficient to 0.0 when generating free-form response and extracting short answer.

4.3 Main Results

As shown in Table 3 and Table 7, we calculate generalized accuracy scores to assess model capabilities. Regarding LVLMs, we draw the following conclusions: (1) **Highest Scoring Model**: Only GPT-4o meets the passing standard and scores highest at 64.5, indicating that our LongDocURL presents significant challenges for current models. (2) **Comparison of Open-source and Closed-source models**: Proprietary models demonstrate better overall performance compared to open-source models. Among open-source models, only Qwen2-VL (score 30.6) and LLaVA-OneVision (scores 22.0 and 25.0) exceed a score of 20, while

Model	Size	Understanding										Reasoning										Locating					Total
		single-page					multi-page					single-page					multi-page					cross-element					
		TXT	LAY	FIG	TAB	all	TXT	LAY	FIG	TAB	all	TXT	LAY	FIG	TAB	all	TXT	LAY	FIG	TAB	all	CTi	CTa	PTi	FTa	all	
		259	91	94	263	612	443	172	174	115	631	40	12	28	98	158	115	40	85	69	229	201	126	137	231	695	2325
OCR (PyMuPDF[4]) + Large Language Models (LLMs)																											
Open-source Models																											
LLaVA-Next-Interleave	7B	23.0	23.8	11.8	16.7	19.5	36.2	32.3	25.0	29.2	33.1	9.4	14.5	3.6	8.2	8.7	27.3	15.0	20.8	23.3	23.0	6.4	4.9	7.8	2.6	5.8	18.7
LLaVA-OneVision	7B	27.0	21.8	18.7	17.8	22.7	43.1	37.6	33.6	33.4	39.1	14.4	39.8	9.4	8.2	10.4	31.9	25.0	29.0	20.5	27.0	13.8	1.8	12.2	14.6	11.2	23.3
LLaVA-OneVision-Chat	7B	28.6	22.8	25.4	19.4	24.6	43.0	36.3	35.4	31.0	38.6	14.4	39.6	14.3	10.0	12.4	31.2	20.0	28.0	22.4	26.7	16.3	2.7	16.7	16.4	14.0	24.6
Qwen2-VL	7B	29.4	24.8	20.2	19.8	24.5	42.3	36.6	33.6	33.3	38.1	16.9	31.4	17.9	8.2	12.5	32.0	22.5	33.8	23.7	29.4	17.5	3.0	17.9	17.0	14.9	25.0
Qwen2.5-Instruct	7B	27.1	29.4	21.4	21.9	24.7	36.5	32.9	33.8	28.0	34.3	12.4	24.5	28.6	17.3	18.3	32.0	32.5	29.2	24.2	27.4	31.3	7.7	13.5	28.2	20.5	25.9
Qwen2.5-Instruct	14B	29.8	22.8	23.1	24.3	26.4	38.4	34.6	37.4	33.5	36.1	21.9	39.8	32.1	19.4	21.4	29.3	35.0	22.1	22.7	25.6	34.9	14.6	16.8	29.7	24.2	27.9
Qwen2.5-Instruct	32B	28.8	23.9	24.8	24.7	26.6	34.9	32.6	31.3	36.0	33.4	22.5	33.3	35.7	15.3	20.3	31.1	**40.0**	32.7	25.3	29.4	33.5	23.1	28.5	31.2	29.5	29.2
Qwen2.5-Instruct	72B	31.9	30.3	23.3	25.3	28.3	42.3	39.7	35.1	34.6	39.4	22.5	25.0	35.7	15.3	20.2	**38.0**	39.9	32.6	26.7	32.0	36.2	21.8	38.6	35.3	34.2	32.9
Proprietary Models																											
Qwen-Max	-	31.4	31.1	24.0	22.5	27.1	40.0	35.3	33.5	34.8	37.0	22.5	25.0	32.1	14.3	19.0	31.9	37.5	31.5	22.9	28.3	37.9	25.6	38.3	29.1	34.0	31.4
Gemini-1.5-Pro	-	29.4	31.8	25.2	25.8	27.8	41.3	40.3	35.5	32.0	38.6	21.9	23.1	28.6	20.4	21.4	31.9	**40.0**	**36.2**	24.3	30.4	39.2	21.3	32.2	35.1	32.8	32.0
Qwen-VL-Max	-	**37.2**	**37.0**	29.1	26.3	**32.0**	**45.4**	44.1	**39.9**	**44.5**	**43.3**	**28.7**	**66.2**	**45.8**	20.4	26.8	37.1	37.5	33.8	27.1	32.1	25.5	13.9	34.4	30.4	27.3	33.3
GPT-4o	-	33.5	29.6	27.4	27.0	30.4	42.6	42.7	38.0	33.5	40.2	27.5	25.0	42.9	21.4	25.9	33.7	37.5	31.5	25.6	30.0	**42.5**	24.1	40.8	35.6	37.2	34.7
O1-preview	-	33.5	31.3	**29.9**	**29.6**	31.1	41.7	**45.0**	38.5	39.2	40.1	22.5	33.3	**45.8**	**24.5**	**27.1**	34.9	37.5	29.2	**38.6**	**34.1**	39.3	**26.6**	**44.9**	**37.8**	**38.6**	**35.8**
Large Vision Language Models (LVLMs)																											
Open-source Models																											
InternLM-XC2.5	7B	2.9	3.0	2.7	2.1	2.7	5.5	5.3	3.2	2.7	4.5	2.5	8.3	0.0	0.0	0.6	3.5	2.5	1.2	1.3	2.6	0.6	0.4	0.5	1.3	0.7	2.4
mPLUG-DocOwl2	7B	6.2	6.9	4.6	3.5	5.4	11.6	10.7	10.6	8.8	9.9	2.5	8.3	0.0	1.0	1.3	7.4	7.5	4.7	4.2	5.5	2.4	0.0	1.8	2.4	1.8	5.3
Pixtral	12B	10.1	8.2	6.7	4.6	7.5	10.2	8.6	5.5	7.2	8.4	2.7	8.9	0.0	0.0	0.7	10.0	5.0	4.6	2.5	5.8	3.7	1.3	1.9	2.2	2.4	5.6
Llama-3.2	11B	11.9	9.9	13.1	8.1	10.3	15.5	16.0	12.4	13.0	15.3	2.5	8.3	10.7	6.1	7.0	13.0	15.0	9.3	6.4	11.0	3.0	1.4	3.8	1.6	2.7	9.2
LLaVA-Next-Interleave	7B	19.7	17.1	8.6	8.1	14.0	30.3	25.4	19.7	16.6	26.3	7.5	16.7	10.1	5.1	6.9	21.4	12.5	19.9	8.5	17.2	6.6	1.1	3.7	2.4	3.8	14.1
LLaVA-OneVision	7B	28.6	32.4	19.2	14.6	22.3	36.7	37.1	28.1	23.5	33.7	10.7	22.9	6.5	6.0	7.6	26.4	20.0	24.0	12.2	22.6	17.7	9.2	6.8	25.5	14.1	22.0
LLaVA-OneVision-Chat	7B	29.5	34.1	26.0	17.9	24.8	36.9	38.4	34.5	26.9	36.0	6.9	14.5	6.5	8.2	8.0	31.8	22.5	29.3	16.5	26.6	22.5	9.9	13.6	29.7	18.7	25.0
Qwen2-VL	7B	37.9	37.2	30.0	29.8	33.4	43.8	38.0	37.8	33.5	40.2	19.0	54.9	21.4	17.2	19.3	33.2	22.5	34.5	16.5	28.6	25.7	22.1	17.6	27.0	22.6	30.6
Proprietary Models																											
Qwen-VL-Max	-	61.1	54.5	58.2	53.7	57.7	63.0	57.2	56.1	53.3	60.0	44.1	**66.6**	57.1	38.6	44.0	45.8	42.5	38.5	41.0	43.8	25.4	16.7	47.5	49.8	36.0	49.5
Gemini-1.5-Pro	-	54.8	57.2	60.1	48.8	52.8	61.4	56.9	54.2	51.8	58.4	37.7	57.9	46.4	31.3	35.9	53.4	52.5	47.8	40.4	48.6	48.4	18.9	46.7	68.1	46.4	50.9
GPT-4o	-	**71.1**	**67.5**	**68.5**	**66.4**	**68.6**	**70.1**	**67.4**	**69.1**	**63.8**	**68.6**	**56.7**	**66.6**	**70.8**	**49.5**	**55.7**	**67.3**	**55.0**	**61.7**	**61.1**	**62.8**	50.7	**40.1**	**66.8**	**78.5**	**59.6**	**64.5**
Human Baseline																											
Human Experts	-	-	-	-	-	-	-	-	-	-	87.2	-	-	-	-	-	-	-	-	-	81.3	-	-	-	-	81.6	84.8

Table 3: Generalized accuracy scores(0~1) on LongDocURL. There are 3 types of tasks: (U)nderstanding, (R)easoning, and (L)ocating. There are 4 types of evidence elements: pure text(TXT), layout(LAY), chart & image(FIG), and table(TAB). There are 3 types of evidence pages/elements: single-page(SP), multi-page(MP), and cross-element(CE). Cross-element means at least two element types in the evidence(*e.g.*, chart and table). CTi: Cross-Title, CTa: Cross-Table, PTi: Para-Title, FTa: Figure-Table. The highest scores among models in each section are highlighted in **green** . *Human baselines* are obtained from 102 uniformly distributed samples.

other models with fewer than 13B parameters fall below this threshold.

To compare the performance of models using text input versus image input, we included the O1-preview and Qwen2.5 series. The experimental results show that the overall scores of LLMs significantly lower than LVLMs, with the top LLM score trailing the top LVLM score by about 30 points. This gap is mainly because important document information is lost when parsed into plain text using PyMuPDF. Our dataset features numerous table-related and chart-related Q&A pairs, and the loss of structural information hinders LLMs' ability to extract critical evidence. These results highlights our LongDocURL as a benchmark for assessing the document structure parsing capability of LVLMs.

4.3.1 Fine-Grained Analysis

We include more fine-grained results in Table 7 and Figure 5, based on document sources, task categories, document elements, and evidence pages.

Document Sources As shown in Figure 5, the models perform better on *books*, *reports*, *manuals*, and *project proposals*, likely due to their simpler layout, which facilitates key information extraction. Conversely, the models struggle with less common documents like *meeting minutes* and *work summary*, which have limited data.

Task Type Observations reveal: (1) Proprietary LVLMs perform comparably on reasoning and locating tasks, but image-to-text conversion impacts reasoning capabilities more severely. For instance, switching to text input, GPT-4o's reasoning scores drop by 31.6 points versus 22.4 points for locating. (2) Strong models are balanced in reasoning and locating, whereas weaker models perform poorly on locating, suggesting a training focus on capabilities of understanding and reasoning over spatial and logical relationships in locating tasks.

Document Elements Models score highest on *Text* questions and lowest on *Table* ones, highlighting deficiencies in document structure parsing. *Figure* and *Layout* question types yield similar scores. Scores for cross-element tasks fall between single-page and multi-page QA, closely aligning with the overall assessment.

Single-page vs Multi-page Single-page QA accuracy is lower than multi-page QA. This reveals that answers for some questions can be gathered from multiple pages, thereby reducing the difficulty. However, models like GPT-4o and Qwen-VL-Max

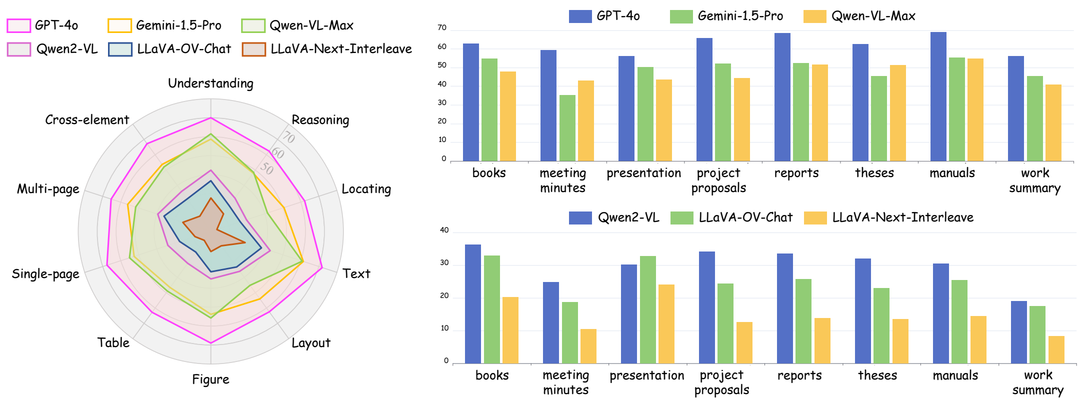


Figure 5: Fine-grained Results. We choose 3 proprietary and 3 open-source models to conduct further analysis based on (left) **task types**, **document elements**, **evidence pages**, and (right) **document sources**.

show lower accuracy on multi-page QA, revealing a contradiction where their scores on locating tasks in multi-page QA are lower, thus skewing overall performance.

Model	Size	Image-input		Text-input	
		cut-off	merge	pymupdf	docmind
Open-source Models					
Qwen2-VL	7B	30.7	7.6[4]	24.4	45.3
LLaVA-Next-Interleave	7B	12.6	10.9	16.9	29.7
LLaVA-OneVision-Chat	7B	24.1	14.0	23.0	39.1
Proprietary Models					
Gemini-1.5-Pro	-	48.1	-[5]	34.0	64.8
GPT-4o	-	64.4	44.9	36.5	66.2
O1-preview	-	-	-	38.0	63.4

Table 4: Comparison among different input paradigms on a subset of 20% data.

4.4 Ablation of Input Paradigms

To explore the optimal input format in long document question-answering, we conduct ablation experiments across two image-input and two text-input paradigms. The image-input paradigms include: (1) **cut-off**, following the configuration detailed in Section 4.2, and (2) **merge**, where document images are combined from raw document lengths (50~150) into 20~30 new images. Further details can be found in Appendix D.1.

We note that the table structure information significantly degrades when parsed by PyMuPDF, while the markdown-format table texts parsed by Docmind retain greater structural integrity. To assess the impact of structural information loss on model performance, we conducted experiments with two input types: (1) **text-input-docmind**, utilizing texts parsed by Docmind, and (2) **text-input-pymupdf**, utilizing texts parsed by PyMuPDF. The analysis of the results presented in Table 4 led us to the following conclusions:

Text-input vs. Image-input: The scores in the *cut-off* paradigm are higher than that in the *text-input-pymupdf* paradigm, but lower than that in the *text-input-docmind* paradigm, indicating that this method can effectively extract table structure information, but it can be improved further.

Cut-off vs. Merge: The *merge* method preserves a greater number of context tokens by concatenating multiple images, while the *cut-off* method succeeds in acquiring prior information by shortening the context window. Experimental results suggest that the cut-off may yield better problem-solving capabilities than merging, providing insights for the future construction of multimodal Retrieval-Augmented Generation (RAG) systems.

Impact of Structural Information: For proprietary models, performance utilizing Docmind is at least 25 points higher than that with PyMuPDF, while the disparity is 15 points for open-source models. The absence of table structure information significantly hampers the performance of both open-source and proprietary models.

5 Error Analysis and Model Insights

Error Analysis We randomly sample 97 erroneous records and conduct a error analysis of the best-performing model, GPT-4o. We classify the errors into 7 types (See Appendix G for detailed descriptions and qualitative examples) and show their distribution in Figure 6.

Perceptual errors constitute the most prevalent category (32.7%), arising from inaccuracies in recognizing, parsing, or counting document elements

[4] 363 out of 465 Q&A pairs meet OOM problems with Qwen2-VL. We calculate normalized score on the size of 465.

[5] Not completed due to resource limitations.

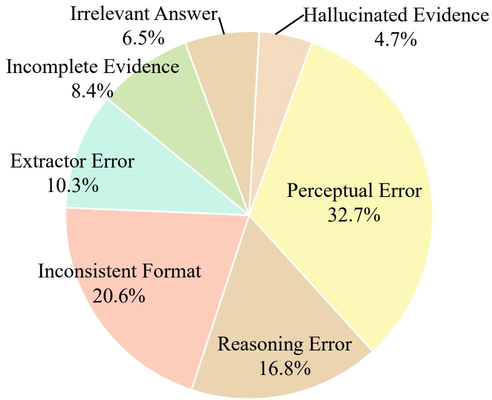


Figure 6: Error distribution over 97 annotated GPT-4o errors.

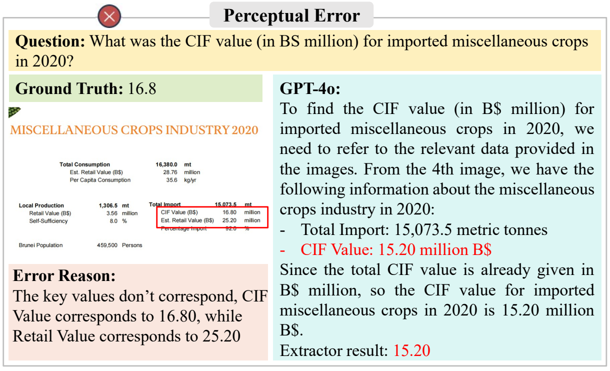


Figure 7: A basic perception error, with the error highlighted in red.

such as tables, figures, and formulas. Typical manifestations include incorrect heading hierarchies and mismatched figure-text correspondences, indicating that complex document structures and OCR performance remain primary bottlenecks for LVLMs. An example of perceptual errors in LongDocURL is shown in Figure 7. *Reasoning errors* represent the third-largest category (16.8%), occurring during calculation, comparison, or summarization even when relevant evidence is correctly identified. A significant proportion of errors (20.6%) involves *inconsistent formats*, highlighting the inflexibility of rule-based evaluation. For instance, a response of "50212000" for "$50.2 million" was incorrectly scored as entirely wrong. The remaining error categories (*Hallucinated Evidence*, *Irrelevant Answer*, *Incomplete Evidence*, and *Extractor Error*) collectively accounted for 29.9% of cases.

Model Insights We analyze open-source models from two key aspects: *dataset adaptation* and *training strategy*. First, Qwen2-VL (LVLM) demonstrates superior performance over its LLM variant in Table 3, attributable to its extensive multimodal training that strengthens both comprehension and generation capabilities across modalities. Second, LLaVA-OneVision-Chat outperforms its base model (LLaVA-OneVision), as it employ Direct Preference Optimization (DPO) and human feedback to boost generalization and reasoning in long-document tasks. Additionally, we observe frequent failures caused by poor layout analysis and table/chart parsing. Incorporating layout parsing into continual pretraining or fine-tuning could mitigate these errors. Alternatively, a multi-stage response framework—where models first parse document structure before reasoning—may improve performance.

6 Conclusion

In this study, we address the limitations of existing document benchmarks. We propose LongDocURL, which includes 20 capabilities across 3 tasks, 3 evidence modes, and 4 document elements. A semi-automated pipeline generated over 2,300 high-quality question-answer pairs, covering more than 33,000 pages of documents. Subsequently, we conducted a comprehensive evaluation of 26 different parameter amounts of both open-source and closed-source models, revealing potential gaps in document understanding.

Limitations

From the perspective of dataset source, the types of documents discussed in this paper are still limited. A broader range of data sources would provide richer document layouts and element information, which would be more beneficial for evaluation. Moreover, the dataset can be further expanded by the automated construction pipeline. On the other hand, designing better model structures and training processes to improve performance on LongDocURLwill be more important. However, this may have gone beyond the scope of this paper.

Acknowledgments

This work has been supported by the National Natural Science Foundation of China (NSFC) grant U23B2029 and Zhongguancun Academy Project No.20240102.

References

Alibaba. 2024. Qwen2-VL: Enhancing vision-language model's perception of the world at any resolution. *Preprint*, arXiv:2409.12191.

Anthropic. 2024. Claude 3.5 sonnet.

Srikar Appalaraju, Bhavan Jasani, Bhargava Urala Kota, Yusheng Xie, and R. Manmatha. 2021. Docformer: End-to-end transformer for document understanding. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021, Montreal, QC, Canada, October 10-17, 2021*, pages 973–983. IEEE.

Tsachi Blau, Sharon Fogel, Roi Ronen, Alona Golts, Shahar Tsiper, Elad Ben-Avraham, Aviad Aberdam, Roy Ganz, and Ron Litman. 2024. GRAM: global reasoning for multi-page VQA. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2024, Seattle, WA, USA, June 16-22, 2024*, pages 15598–15607. IEEE.

Yew Ken Chia, Liying Cheng, Hou Pong Chan, Chaoqun Liu, Maojia Song, Sharifah Mahani Aljunied, Soujanya Poria, and Lidong Bing. 2024. M-longdoc: A benchmark for multimodal super-long document understanding and a retrieval-aware tuning framework. *Preprint*, arXiv:2411.06176.

Jaemin Cho, Jie Lei, Hao Tan, and Mohit Bansal. 2021. Unifying vision-and-language tasks via text generation. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event*, volume 139 of *Proceedings of Machine Learning Research*, pages 1931–1942. PMLR.

Jaemin Cho, Debanjan Mahata, Ozan Irsoy, Yujie He, and Mohit Bansal. 2024. M3docrag: Multi-modal retrieval is what you need for multi-page multi-document understanding. *Preprint*, arXiv:2411.04952.

Yihao Ding, Kaixuan Ren, Jiabin Huang, Siwen Luo, and Soyeon Caren Han. 2024a. MMVQA: A comprehensive dataset for investigating multipage multimodal information retrieval in pdf-based visual question answering. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 6243–6251. ijcai.org.

Yihao Ding, Kaixuan Ren, Jiabin Huang, Siwen Luo, and Soyeon Caren Han. 2024b. PDF-MVQA: A dataset for multimodal information retrieval in pdf-based visual question answering. *CoRR*, abs/2404.12720.

Qi Dong, Lei Kang, and Dimosthenis Karatzas. 2024a. Multi-page document VQA with recurrent memory transformer. In *Document Analysis Systems - 16th IAPR International Workshop, DAS 2024, Athens, Greece, August 30-31, 2024, Proceedings*, volume 14994 of *Lecture Notes in Computer Science*, pages 57–70. Springer.

Xiaoyi Dong, Pan Zhang, Yuhang Zang, Yuhang Cao, Bin Wang, Linke Ouyang, Songyang Zhang, Haodong Duan, Wenwei Zhang, Yining Li, et al. 2024b. Internlm-Xcomposer2-4KHD: A pioneering large vision-language model handling resolutions from 336 pixels to 4k hd. *ArXiv preprint*, abs/2404.06512.

GeminiTeam. 2024. Gemini 1.5: Unlocking multimodal understanding across millions of tokens of context. *Preprint*, arXiv:2403.05530.

Anwen Hu, Haiyang Xu, Liang Zhang, Jiabo Ye, Ming Yan, Ji Zhang, Qin Jin, Fei Huang, and Jingren Zhou. 2024. mPLUG-DocOwl2: High-resolution compressing for ocr-free multi-page document understanding. *CoRR*, abs/2409.03420.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document AI with unified text and image masking. In *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, pages 4083–4091. ACM.

Mengzhao Jia, Wenhao Yu, Kaixin Ma, Tianqing Fang, Zhihan Zhang, Siru Ouyang, Hongming Zhang, Meng Jiang, and Dong Yu. 2024. Leopard: A vision language model for text-rich multi-image tasks. *CoRR*, abs/2410.01744.

Dongfu Jiang, Xuan He, Huaye Zeng, Cong Wei, Max Ku, Qian Liu, and Wenhu Chen. 2024. MANTIS: interleaved multi-image instruction tuning. *CoRR*, abs/2405.01483.

Lei Kang, Rubèn Tito, Ernest Valveny, and Dimosthenis Karatzas. 2024. Multi-page document visual question answering using self-attention scoring mechanism. In *Document Analysis and Recognition - ICDAR 2024 - 18th International Conference, Athens, Greece, August 30 - September 4, 2024, Proceedings, Part VI*, volume 14809 of *Lecture Notes in Computer Science*, pages 219–232. Springer.

Jordy Van Landeghem, Rafal Powalski, Rubèn Tito, Dawid Jurkiewicz, Matthew B. Blaschko, Lukasz Borchmann, Mickaël Coustaty, Sien Moens, Michal Pietruszka, Bertrand Anckaert, Tomasz Stanislawek, Pawel Józiak, and Ernest Valveny. 2023. Document understanding dataset and evaluation (DUDE). In *IEEE/CVF International Conference on Computer Vision, ICCV 2023, Paris, France, October 1-6, 2023*, pages 19471–19483. IEEE.

Feng Li, Renrui Zhang, Hao Zhang, Yuanhan Zhang, Bo Li, Wei Li, Zejun Ma, and Chunyuan Li. 2024. Llava-next-interleave: Tackling multi-image, video, and 3d in large multimodal models. *CoRR*, abs/2407.07895.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. 2023. Visual instruction tuning. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

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

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. 2024. Mathvista: Evaluating mathematical reasoning of foundation models in visual contexts. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. 2024. Mmlongbench-doc: Benchmarking long-context document understanding with visualizations. *CoRR*, abs/2407.01523.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq R. Joty, and Enamul Hoque. 2022. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 2263–2279. Association for Computational Linguistics.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. 2021. Docvqa: A dataset for VQA on document images. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2021, Waikoloa, HI, USA, January 3-8, 2021*, pages 2199–2208. IEEE.

Meta. 2024. Introducing meta llama 3: The most capable openly available llm to date.

OpenAI. 2024. Hello gpt-4o.

QwenTeam. 2024. Qwen2.

Rubèn Tito, Dimosthenis Karatzas, and Ernest Valveny. 2022. Hierarchical multimodal transformers for multi-page docvqa. *CoRR*, abs/2212.05935.

Maria Wang, Srinivas Sunkara, Gilles Baechler, Jason Lin, Yun Zhu, Fedir Zubach, Lei Shu, and Jindong Chen. 2024. Webquest: A benchmark for multimodal QA on web page sequences. *CoRR*, abs/2409.13711.

Yang Xu, Yiheng Xu, Tengchao Lv, Lei Cui, Furu Wei, Guoxin Wang, Yijuan Lu, Dinei A. F. Florêncio, Cha Zhang, Wanxiang Che, Min Zhang, and Lidong Zhou. 2021. LayoutLMv2: Multi-modal pre-training for visually-rich document understanding. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, pages 2579–2591. Association for Computational Linguistics.

Yiheng Xu, Minghao Li, Lei Cui, Shaohan Huang, Furu Wei, and Ming Zhou. 2020. LayoutLM: Pre-training of text and layout for document image understanding. In *KDD '20: The 26th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, CA, USA, August 23-27, 2020*, pages 1192–1200. ACM.

Shi Yu, Chaoyue Tang, Bokai Xu, Junbo Cui, Junhao Ran, Yukun Yan, Zhenghao Liu, Shuo Wang, Xu Han, Zhiyuan Liu, and Maosong Sun. 2024. Visrag: Vision-based retrieval-augmented generation on multi-modality documents. *CoRR*, abs/2410.10594.

Jinxu Zhang, Yongqi Yu, and Yu Zhang. 2024a. Cfret-dvqa: Coarse-to-fine retrieval and efficient tuning for document visual question answering. *CoRR*, abs/2403.00816.

Jinxu Zhang, Yongqi Yu, and Yu Zhang. 2024b. CREAM: coarse-to-fine retrieval and multi-modal efficient tuning for document VQA. In *Proceedings of the 32nd ACM International Conference on Multimedia, MM 2024, Melbourne, VIC, Australia, 28 October 2024 - 1 November 2024*, pages 925–934. ACM.

A More Statistics of LongDocURL

Figure 8 illustrate our dataset distribution characteristics across document pages, answer evidence page, evidence pages length, document sources, and evidence element types.

B Method Details

B.1 QA Construction

B.1.1 Prompt Template for QA Generation

[System]
You are an expert in document question-answering dialogue synthesis. Please complete the following instructions based on the given text. The response must be true and accurate, and no additional content should be output.

[Task Description]
Your task is <detailed_task_description>

[Restriction]
Ensure questions and answers are suitable and correct.
Only include questions that have definite answers, that is:

- one can see the content in the image that the question asks about and can answer confidently;
- one can determine confidently from the image that it is not in the image, don't ask any question that cannot be answered confidently.

Provide detailed evidence description first and then give final short answers. Use examples or reasoning steps to support your content. You can include multiple paragraphs if necessary.
<other_restriction_description>

[Response Examples]
<few_qa_examples>

[Context Input]
<structured_text> | <previous_response>

Notes: <structured_text> refers to **text-type-bbox** triple processed by Docmind engine, which is discussed in detail in Section 3.2.1. <previous_response> refers to possible intermediate result(*e.g.*, summary sentences in our Para-Title Locating task. Specifically, we prompt LVLMs to generate summary sentences of one or multiple paragraphs under certain section titles first, and then utilize these summaries and title names to construct question-answering data.)

B.1.2 Prompt Template for QA Verification

[System]
You are an expert in document question-answering verification. Please complete the following instructions based on the given text. The response must be true and accurate, and no additional content should be output.

[Task description & Verification criteria]
Your task is to ensure the quality of question-answer pairs in the context provided. you need to follow the steps outlined below to systematically evaluate each pair's effectiveness and accuracy. Implement this process diligently to maintain high standards across the batch of QA pairs.

1. Question type check

Does the question match the task description: <task_description>
Make sure the question meets the required task context.

2. Formatting and Presentation

Is answer properly formatted?
Ensure the answer uses a list format to store the title content.

3. Relevance Check
Does the question relate directly to document content rely on the context provided, the answer accurately reflect the information in the document?
Ensure the question is formulated based on information explicitly stated or implied in the document. The question should not introduce concepts unrelated to the document's content.
Validate that the answer references specific data or statements from the document. Avoid including extraneous information not supported by the document.

4. Clarity and Precision
Is the question clear and unambiguous? And is the answer concise and precise? Ensure the language is straightforward and easily understandable, avoid complex phrasing that may confuse the reader.
The intention of the question and answer pair must be clear and direct, avoiding verbosity and unnecessary detail.
Ensure the answer fully addresses the question without omitting crucial information.

5. Consistency and Coherence
Check for logical flow and coherence, ensuring the question aligns seamlessly with the document's narrative or arguments. Verify that the answer does not contradict or misrepresent other sections of the document.
By practicing this process, you can confirm whether the quality of the question-answer pairs meets the requirements. If all the above conditions are met, please output yes, otherwise output no.

Generated QA: <qa>

Input Context: <evidence_context>

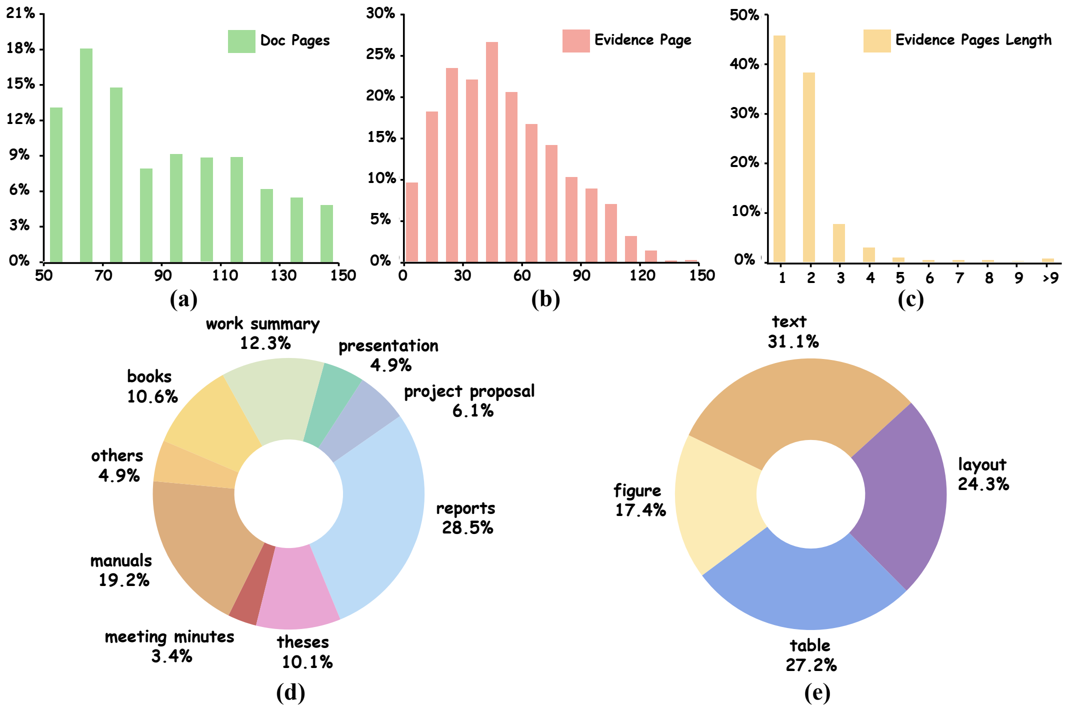


Figure 8: The statistical analysis of our dataset about the distribution characteristics across (a) **document pages**, (b) **answer evidence page**, (c) **evidence pages length**, (d) **document sources**, and (e) **evidence element types**.

B.1.3 Statistics of QA Verification

Verify	U + R	L
Before	2857	1520
After	1630	695
	57.1%↓	45.7%↓

Table 5: Statistics of changes in the amount of data before and after verification.

C Comparison with MMLongBench-Doc

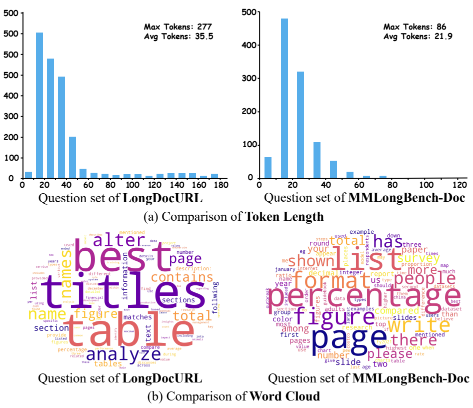


Figure 9: The dataset attributes comparison between our LongDocURL and MMLongBench-Doc.

D Experimental Details

D.1 Merging Rules of Images Input

As discussed in Section 4.4, we design a **merge** paradigm for the evaluation of current LLMs/LVLMs.

`#Columns_merged = 2 if total_pages/30 <= 4 else 3`

Total_pages	#Columns_merged	#Images_merged
50<x<=60	2	x/2
60<x<=90	2	x/3
90<x<=120	2	x/4
120<x<=150	3	x/5

Table 6: Merging rules in the ablation experiment of input paradigms. **#Columns_merged**: the number of columns in the sub-image array in the merged image; **#Images_merged**: the number of new images after merged.

D.2 Selection Rules of Images Input

As discussed in Section 4.2, we design a **cut-off** paradigm for the evaluation of current LLMs/LVLMs. We provide pseudo-code below to express the selection rules.

```
# All page ids mentioned are based on the order in the raw document.
# context_start_end: page start/end id of context in qa generation
# num_of_images_input: the number of input images, 30 images in cut-off paradigm
# total_pages: total pages in the raw document
# img_start, img_end: input page start/end id
raw_start_page, raw_end_page =
context_start_end
raw_pages_len = raw_end_page - raw_start_page
img_start = max(0, raw_start_page -
    (num_of_images_input - raw_pages_len)//2
img_end = img_start + num_of_images_input
if img_end >= total_pages:
    img_end = total_pages
    img_start = max(0,
    img_end - num_of_images_input)
return img_start, img_end
```

D.3 Prompt for Response Generation

<document_images>

You are an expert in visual document question-answering, please answer our questions based on the given doc images.

Following is our question:
<question_start><question></question_end>

D.4 Prompt for Answer Extraction

Prompt for answer extraction is displayed in Figure 10. Based on the template given in MMLongBench-Doc (Ma et al., 2024), we make some modifications, which are marked in blue.

D.5 Scoring Rules

Following MATHVISTA (Lu et al., 2024), we evaluate the model's responses by scoring the extracted answers against the reference answers. Following MMLongBench-Doc (Ma et al., 2024), our scorer is rule-based and employs different strategies according to the format of the reference answer. We detail its rules as below:

String: We firstly use a series of regular expressions to determine whether the answers require exact matching (*e.g.*, telephone numbers, email addresses, website addresses, file names, times, dates, *etc.*) If an exact match is needed, we perform a straightforward string comparison and score the answer either 0 or 1. Otherwise, we calculate the ANLS (Average Normalized Levenshtein Similarity) with a pre-defined threshold ($\tau = 0.5$).

Integer: We perform an exact match comparison and score the answer either 0 or 1.

Float: We view the prediction and reference answers as equal if they fall within a 1% relative tolerance.

List: Compared with MMLongBench-Doc, we adopt a relatively soft rule for scoring answers in list format: (1) If the prediction does not have the same number of elements as the reference, it incurs a length-dependent penalty instead of receiving a score of 0, which we think more reasonable. (2) The score of models on single element of the reference list is the highest one among the scores which are calculated and combined between the element and each one of the prediction list. Compared with MMLongBench-Doc, we assume that the sorting positions of the two lists are not always one-to-one corresponding, allowing more errors, and our rules are gentler and more tolerant. We use pseudo-code below to express the scoring rules in MMLongBench-Doc and our LongDocURL, respectively. The element-wise scoring strategies is determined by the formats of elements (*i.e.*, string, integer or float).

```
# MMLongBench-Doc
pred_list, ref_list = sorted(pred_list), sorted(ref_list)
Score(pred_list, ref_list) = min([score(pred, ref) for
pred, ref in zip(pred_list, ref_list)])

# LongDocURL
pred_list, ref_list = sorted(pred_list), sorted(ref_list)
greedy_scores_list = [
max([score(pred, ref) for pred in pred_list]) for ref in
ref_list
]
Score(ref_list, pred_list) =
sum(greedy_scores_list) / len(ref_list) * min(1,
len(ref_list) / len(pred_list)) ** 0.5
```

The part of the rule description that differs from MMLongBench-Doc is mainly in the **List** section.

D.6 Fine-Grained Evaluation Results

Detailed results are presented in Table 7. Related analysis is in Section 4.3.1.

E Case Study

Figure 11 and Figure 12 display the response of different models, and we give a short error analysis for each case.

F Data Examples

Figure 13, Figure 14 and Figure 15 provide samples for three primary tasks.

The answer extracting prompt used to make long response concise during evaluation

[Task Description]
Given the question and analysis, you are tasked to extract answers with required formats from the free-form analysis.

Your extracted answers should be one of the following formats: (1) Integer, (2) Float, (3) String and (4) List. If you find the analysis the question can not be answered from the given documents, type "Not answerable". Exception: If the analysis only tells you that it can not read/understand the images or documents, type "Fail to answer".

Please use <concise_answer> and <concise_answer> tokens at the start and end of the extracted answer. For example, if the extracted answer is number 3, the format is <concise_answer>3<concise_answer>.

Please use <answer_format> and <answer_format> tokens at the start and end of the answer format. For example, if the answer format is List, the format is <answer_format>List<answer_format>.

Please make your response as concise as possible. Also note that your response should be formatted as below:

Extracted answer: <concise_answer>[answer]<concise_answer>
Answer format: <answer_format>[answer format]<answer_format>

[Few Shots]
Please read the following example, then extract the answer from the model response and type it at the end of the prompt.

— Example 1:
Question: List the primary questions asked about the services in this report.
Analysis: The primary questions asked about the services in the report for The Limes Residential Home are:
1. Is the service safe?
2. Is the service effective?
3. Is the service caring?
4. Is the service responsive?
5. Is the service well-led?
Extracted answer: <concise_answer>['Is the servife safe?', 'Is the service effective', 'Is the serve caring?', 'Is the service responsive?', 'Is the service well-led?']<concise_answer>
Answer format: <answer_format>List<answer_format>

— Example 2:
Question: How many regulations of the HSCA 2008 are breached in all according to this report?
Analysis: According to the report, the provider breached 10 Health and Social Care Act 2008 (Regulated Activities) Regulations in total. Here are the specifics:
1. Regulation 13: Safeguarding service users from abuse and improper treatment
2. Regulation 12: Safe care and treatment
3. Regulation 18: Staffing
4. Regulation 11: Need for consent
5. Regulation 10: Dignity and respect
6. Regulation 9: Person-centred care
7. Regulation 17: Good governance
8. Regulation 18 (CQC Registration Regulations 2009): Notification of other incidents
9. Regulation 18: Failure to maintain an accurate and up-to-date care plan
10. Regulation 11: Failure to implement the Mental Capacity Act 2005 code of practice effectively
These breaches involve issues concerning staffing, safeguarding, medicines management, dignity and respect, consent, care planning, governance, and failure to notify the CQC of incidents.
Extracted answer: <concise_answer>10<concise_answer>
Answer format: <answer_format>Integer<answer_format>

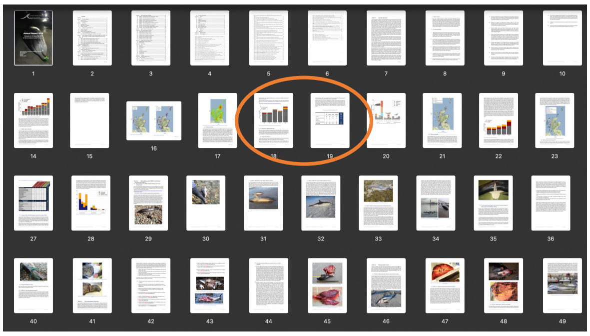
— Example 3:
Question: According to the survey that is the percentage of Chinese who are paying more or about the same attention to politics after Trump's election?
Analysis: The survey provided does not specify the percentage of Chinese individuals specifically who are paying more or about the same attention to politics after Trump's election. The report focuses primarily on American demographics and does not include specific details about the Chinese population in relation to this question. If you need information about a different demographic or a summary of the findings from the American demographic, I can certainly help with that!
Extracted answer: <concise_answer>Not answerable<concise_answer>
Answer format: <answer_format>None<answer_format>

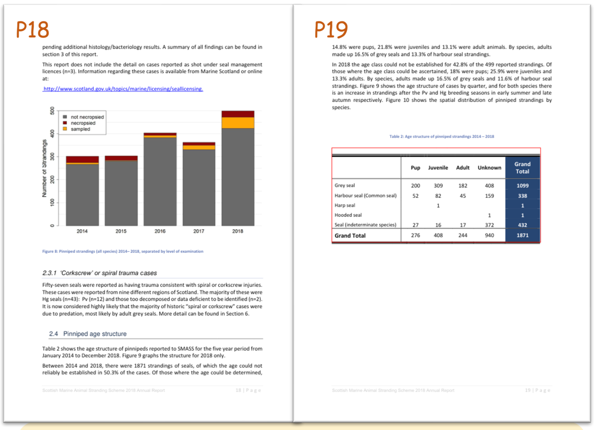
— Example 4:
Question: How many quotations from male respondent over 50 years old are included in this report?
Analysis: The image you've provided appears to be a screenshot of a document with multiple charts. However, the text is too small and blurry to read accurately. If you can provide a clearer image or more context, I might be able to help you with your question.
Extracted answer: <concise_answer>Fail to answer<concise_answer>
Answer format: <answer_format>None<answer_format>

Figure 10: Prompt for answer extraction during evaluation.

Model	Size	Task Type			Evidence Element				Page/Element			Total
		U	R	L	TXT	LAY	FIG	TAB	SP	MP	CE	
		1243	387	695	994	779	556	871	1093	1230	862	2325
OCR (PyMuPDF[4]) + Large Language Models (LLMs)												
Open-source Models												
LLaVA-Next-Interleave	7B	26.4	17.2	5.8	26.0	13.8	15.4	14.0	14.4	22.6	16.0	18.7
LLaVA-Next-Interleave-DPO	7B	28.7	16.4	6.4	27.4	15.7	17.7	14.2	15.1	24.3	17.5	20.0
LLaVA-OneVision	7B	31.0	20.2	11.2	32.6	19.2	22.1	15.1	17.5	28.5	21.0	23.3
LLaVA-OneVision-Chat	7B	31.7	20.9	14.0	33.1	19.8	25.3	16.6	20.0	28.8	22.2	24.6
Qwen2-VL	7B	31.4	22.5	14.9	33.2	20.6	25.0	17.2	20.7	28.7	22.6	25.0
Qwen2.5-Instruct	7B	29.6	23.7	20.5	31.4	27.0	24.3	18.2	21.5	29.7	24.2	25.9
Qwen2.5-Instruct	14B	31.3	23.9	24.2	33.3	29.3	24.8	21.6	23.5	31.8	26.5	27.9
Qwen2.5-Instruct	32B	30.0	25.7	29.5	31.9	30.4	28.9	25.8	26.4	31.6	29.6	29.2
Qwen2.5-Instruct	72B	33.9	27.2	34.2	37.4	33.8	34.0	28.6	29.3	36.0	35.3	32.9
Proprietary Models												
Qwen-Max	-	32.1	24.5	34.0	34.6	32.8	32.6	27.7	28.3	34.1	33.2	31.4
Gemini-1.5-Pro	-	33.3	26.7	32.8	35.5	34.7	32.4	26.9	28.3	35.4	33.1	32.0
Qwen-VL-Max	-	**37.7**	30.0	27.3	**39.6**	31.2	35.1	27.9	31.1	35.2	34.2	33.3
GPT-4o	-	35.4	**28.3**	37.2	37.6	36.3	36.3	30.2	31.9	37.2	35.8	34.7
O1-preview	-	35.6	31.2	**38.6**	37.4	**37.1**	**37.3**	**34.6**	**33.5**	**37.8**	**38.6**	**35.8**
Large Vision Language Models (LVLMs)												
Open-source Models												
InternLM-XC2.5	7B	3.6	1.8	0.7	3.9	2.2	1.9	1.2	1.9	2.9	2.3	2.4
mPLUG-DocOwl2	7B	7.7	3.8	1.8	8.1	4.7	5.5	3.1	3.7	6.7	6.1	5.3
Pixtral	12B	7.9	3.7	2.4	8.8	4.8	4.3	3.1	4.8	6.2	5.3	5.6
Llama-3.2	11B	12.9	9.4	2.7	11.8	6.9	8.7	6.3	7.9	10.3	6.8	9.2
LLaVA-Next-Interleave	7B	20.2	13.0	3.8	21.8	10.8	12.1	6.7	10.0	17.8	11.3	14.1
LLaVA-Next-Interleave-DPO	7B	21.6	13.9	7.6	22.5	13.9	15.4	8.7	12.1	19.8	13.5	16.2
LLaVA-OneVision	7B	28.1	16.5	14.1	30.8	23.9	17.9	11.6	16.5	26.8	20.8	22.0
LLaVA-OneVision-Chat	7B	30.5	19.0	18.7	32.2	26.5	24.4	15.4	19.8	29.7	24.2	25.0
Qwen2-VL	7B	36.9	24.8	22.6	37.7	29.7	28.6	23.7	27.2	33.6	29.9	30.6
Proprietary Models												
Qwen-VL-Max	-	58.8	43.9	36.0	58.0	40.2	52.3	44.6	51.6	47.6	48.0	49.5
Gemini-1.5-Pro	-	55.7	43.4	46.4	58.7	50.4	50.0	41.8	48.7	52.8	49.9	50.9
GPT-4o	-	**68.6**	**59.9**	**59.6**	**70.7**	**60.0**	**67.4**	**60.3**	**65.8**	**63.2**	**65.4**	**64.5**
Human Baseline												
Human Experts	-	-	-	-	-	-	-	-	-	-	-	84.8

Table 7: Generalized accuracy scores(0~1) on LongDocURL. There are 3 types of tasks: (U)nderstanding, (R)easoning, and (L)ocalization. There are 4 types of evidence elements: pure text(TXT), layout(LAY), chart & image(FIG), and table(TAB). There are 3 types of evidence pages/elements: single-page(SP), multi-page(MP), and cross-element(CE). Cross-element means at least two element types in the evidence(*e.g.*, chart and table). The highest scores among models in each section are highlighted in **green** .





Question: What is the total number of pinniped strandings reported between 2014 and 2018?
Ground Truth: 1871

GPT-4o ✗
Response: The total number of pinniped strandings reported between 2014 and 2018 can be found in the bar graph in the 9th image. The numbers for each year are as follows:
- 2014: Approximately 150
- 2015: Approximately 150
- 2016: Approximately 200
- 2017: Approximately 200
- 2018: Approximately 500

Adding these together gives a total of approximately 1200 pinniped strandings reported between 2014 and 2018.
Final Answer: 1200

Gemini-1.5-Pro ✓
Response: Between 2014 and 2018, a total of 1871 pinniped strandings were reported. This is shown in Table 2 on page 19.
Final Answer: 1871

Qwen2-VL ✗
Response: The total number of pinniped strandings reported between 2014 and 2018 is 690.
Final Answer: 690

LLaVA-OneVision-Chat ✗
Response: The total number of pinniped strandings reported between 2014 and 2018 is 1,386.
Final Answer: 1386

Qwen-VL-Max ✗
Response: To determine the total number of pinniped strandings reported between 2014 and 2018, we need to sum the number of strandings for each year within this range.

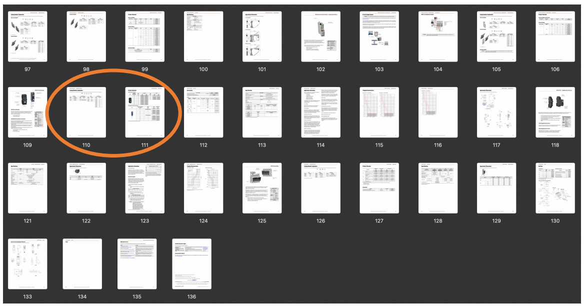
From the bar chart in the image, we can see the following data for pinniped strandings:
- 2014: Approximately 300 strandings
- 2015: Approximately 350 strandings
- 2016: Approximately 400 strandings
- 2017: Approximately 450 strandings
- 2018: Approximately 500 strandings

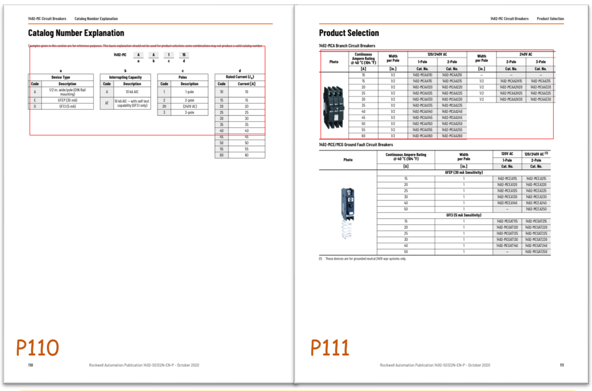
Adding these numbers together gives us the total number of pinniped strandings reported between 2014 and 2018:
300 (2014) + 350 (2015) + 400 (2016) + 450 (2017) + 500 (2018) = 1900 strandings
Therefore, the total number of pinniped strandings reported between 2014 and 2018 is 1900.
Final Answer: 1900

LLaVA-Next-Interleave ✗
Response: The total number of pinniped strandings reported between 2014 and 2018 is 1,372.
Final Answer: 1372

Error Analysis: Wrong Evidence Sources.
Details: Relevant evidence can be found in both the table in page 19 and the bar chart in page 18, but the total num can be extracted directly from the table in page 19. Only Gemini-1.5-Pro get the right answer. GPT-4o and Qwen-VL-Max struggle to estimate values from the bar chart. Another three open-source models response with wrong short answer.

Figure 11: Case 1. Evidence source: ["Table"]. Evidence pages: [26, 27]. The correct extracted information and reasoning are colored in green, and the wrong ones are colored in red.





Question: How many different ratings of 1492-MCA branch circuit breakers are listed in Table?
Ground Truth: 11

GPT-4o ✅
Response: The table on the 13th image lists the 1492-MCA branch circuit breakers with the following ratings:
1. 10 A
2. 15 A
3. 20 A
4. 25 A
5. 30 A
6. 35 A
7. 40 A
8. 45 A
9. 50 A
10. 55 A
11. 60 A
There are 11 different ratings of 1492-MCA branch circuit breakers listed in the table.
Final Answer: 11

Qwen2-VL ❌
Response: The number of different ratings of 1492-MCA branch circuit breakers listed in the table is 10.
Final Answer: 10

Qwen-VL-Max ❌
Response: To determine the number of different ratings of 1492-MCA branch circuit breakers listed in Table, we need to count the unique values in the "Rated Current" column of the table.
Final Answer: Not answerable

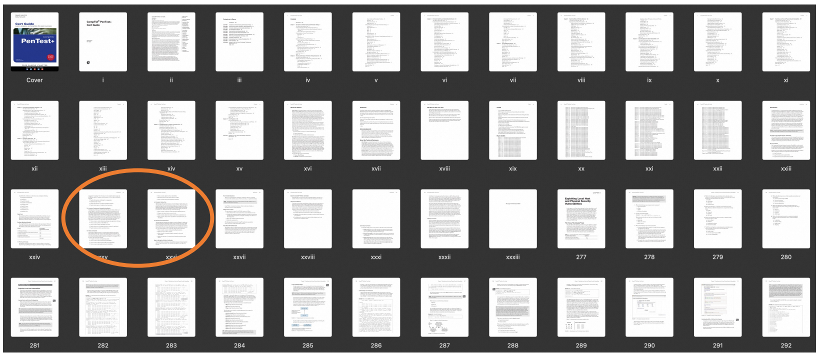
LLaVA-Next-Interleave ❌
Response: There are 14 different ratings of 1492-MCA branch circuit breakers listed in Table.
Final Answer: 14

Gemini-1.5-Pro ❌
Response: The table on page 111 lists six different ratings for 1492-MCA branch circuit breakers: 15 A, 20 A, 25 A, 30 A, 35 A, 40 A, 45 A, 50 A, 55 A, and 60 A.
Final Answer: 10

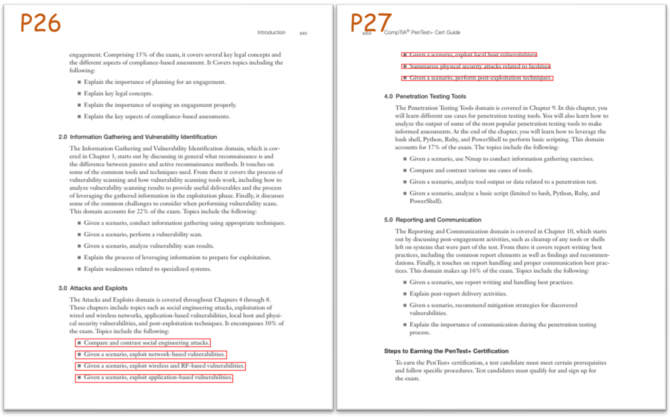
LLaVA-OneVision-Chat ❌
Response: There are 12 different ratings of 1492-MCA branch circuit breakers listed in Table.
Final Answer: 12

Error Analysis: Missing Key Evidence.
Details: Relevant evidence can be found in the tables in page 110 and page 111. This question requires models to count elements in a table, testing models' fine-grained table structure perception and parsing capabilities. Only GPT-4o answer correctly.

Figure 12: Case 2. Evidence source: ["Table"]. Evidence pages: [110, 111]. The correct extracted information and reasoning are colored in green, and the wrong ones are colored in red.



Document Thumbnail

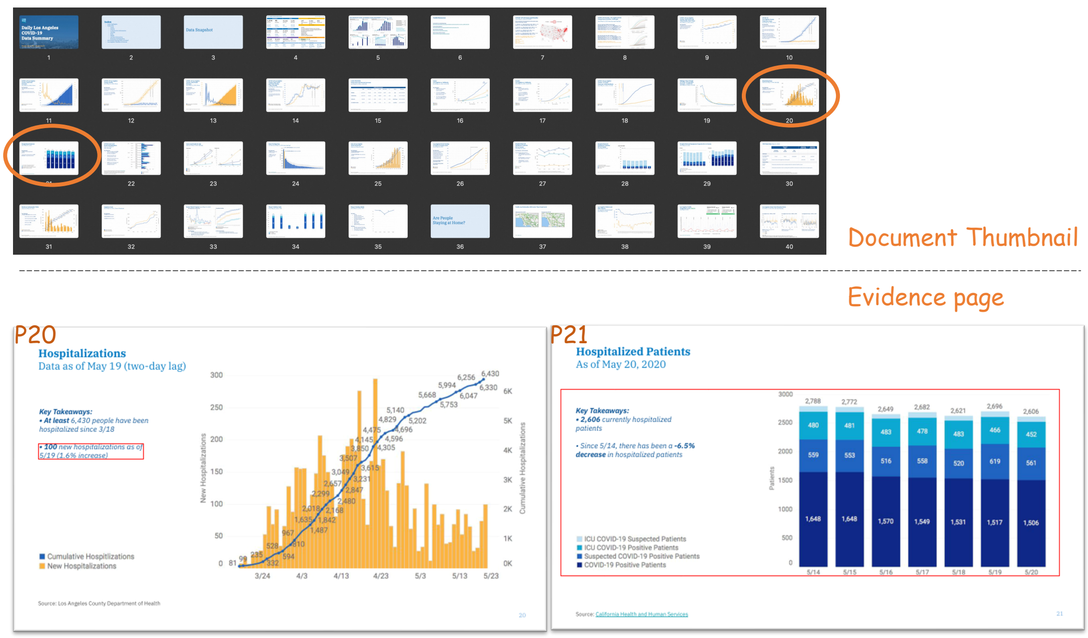


Evidence page

Question: What topics are covered under the "3.0 Attacks and Exploits" domain?
Detailed Evidence Description:
The topics covered under the "3.0 Attacks and Exploits" domain are:
1. Compare and contrast social engineering attacks (<box>(0.21, 0.83, 0.63, 0.85)</box>, page 26)
2. Given a scenario, exploit network-based vulnerabilities (<box>(0.21, 0.86, 0.68, 0.88)</box>, page 26)
3. Given a scenario, exploit wireless and RF-based vulnerabilities (<box>(0.21, 0.89, 0.74, 0.91)</box>, page 26)
4. Given a scenario, exploit application-based vulnerabilities (<box>(0.21, 0.92, 0.7, 0.94)</box>, page 26)
5. Given a scenario, exploit local host vulnerabilities (<box>(0.19, 0.11, 0.61, 0.12)</box>, page 27)
6. Summarize physical security attacks related to facilities (<box>(0.19, 0.14, 0.65, 0.15)</box>, page 27)
7. Given a scenario, perform post-exploitation techniques (<box>(0.19, 0.17, 0.66, 0.18)</box>, page 27)
These topics are distributed across pages 26 and 27.

Ground Truth: ["Compare and contrast social engineering attacks",
"Given a scenario, exploit network-based vulnerabilities",
"Given a scenario, exploit wireless and RF-based vulnerabilities",
"Given a scenario, exploit application-based vulnerabilities",
"Given a scenario, exploit local host vulnerabilities",
"Summarize physical security attacks related to facilities",
"Given a scenario, perform post-exploitation techniques"]
Task: Understanding(extract)
Evidence Pages: [26, 27]
Evidence Sources: ["Layout (Title)", "Text"]
Answer Format: List

Figure 13: Data Example of Understanding QA.



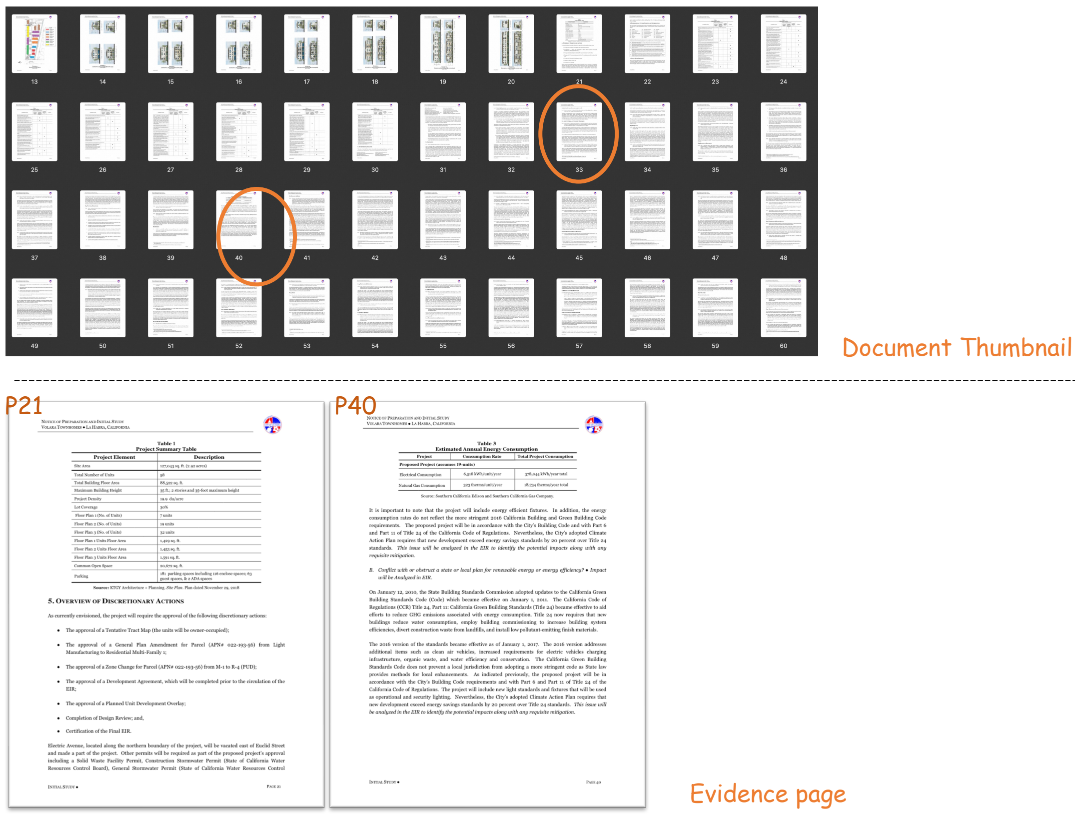
Question: What is the total number of new hospitalizations and current hospitalizations as of May 19, 2020?

Detailed Evidence Description:

As of May 19, 2020, there were 100 new hospitalizations and 2,606 current hospitalizations. The information about new hospitalizations is found on page 20 with the text "100 new hospitalizations as of 5/19 (1.6% increase)" (<box>(0.04, 0.38, 0.24, 0.43)</box>, page 20). The information about current hospitalizations can be found on page 21 in the figure referenced by the text "2,606 currently hospitalized" (<box>(0.01, 0.2, 0.96, 0.83)</box>, page 21).

Ground Truth: 2706
Task: Reasoning
Evidence Pages: [20, 21]
Evidence Sources: ["Text", “Figure”]
Answer Format: Integer

Figure 14: Data Example of Reasoning QA.



Question: Which tables provide detailed information on the project's design and specifications?
Select table names from the doc that best answer the question, do not alter or analyze the table names themselves.

Detailed Evidence Description:

1. **Project Specifications and Design (Table 1)**: This table provides a detailed overview of the project elements, including the site area, total number of units, building floor area, maximum building height, project density, lot coverage, floor plans, common open space, and parking. xxx
2. **Initial Study Checklist (Table 2)**: These tables provide a comprehensive checklist for evaluating the potential environmental impacts of the project. Each section addresses specific issues such as aesthetics xxx
3. **Energy Consumption (Table 3)**: This table outlines the projected energy consumption for the proposed project, assuming 19 units. It details the electrical xxx

…

Ground Truth: ["Table 1 Project Summary Table",
"Table 3 Estimated Annual Energy Consumption"]
Task: Locating(cross-table)
Evidence Pages: [21, 40]
Evidence Sources: ["Layout(table name)", "Table"]
Answer Format: List

Figure 15: Data Example of Locating QA.

G Error Analysis

We conduct a detailed error analysis of the best-performing model, GPT-4o. We randomly sample 97 errors and examine their distribution in detail. Based on this analysis, we classify the errors into five types, with the specific categories and their distribution shown in Figure 6. Except for the *Extractor Error* caused by our automatic evaluation pipeline, we present detailed definitions and descriptions of another six categories below:

Perceptual Error: This category remains prevalent in complex document understanding tasks, where models frequently fail to accurately recognize, parse, or enumerate visual elements like tables, figures, and formulas in document screenshots. See Figure 7 and Figure 16 for examples.

Reasoning Error: Ranking as the third most prevalent error type, these failures occur during critical cognitive operations—including numerical calculations, comparative analyses, and content summarization—despite correct evidence retrieval. See Figure 17 for examples.

Inconsistent Format: While our ground truth annotations strictly preserve original document content, GPT-4o sometimes generates outputs with formatting discrepancies. These variations principally involve: (1) abbreviation/full-form inconsistencies, (2) inclusion/omission of numerical units, and (3) mismatches in decimal precision. See Figure 18 for examples.

Incomplete Evidence: The LongDocURL benchmark reveals significant model limitations in processing extensive multimodal documents, where essential query-relevant information is sometimes omitted from responses. See Figure 19 for examples.

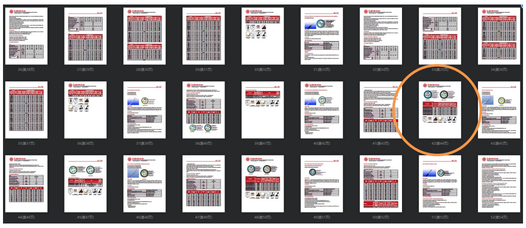
Irrelevant Answer: Fundamental misinterpretations of query intent lead to responses that fail to address the actual information needs. See Figure 20 for examples.

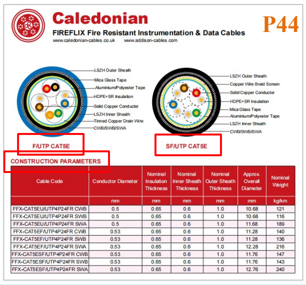
Hallucinated Evidence: The most severe error type involves model-generated fabrications, where non-existent document content is created to substantiate incorrect responses. See Figure 21 for examples.

H Description of labeling labor

At the dataset verification stage, we have 21 full-time data annotators responsible for the labeling work in the human verifying process, while 6 professional annotators with postgraduate degrees or above perform the final data quality verification and cross-checking work.

Perceptual Error





Question: List names of the figures at the page which contains a table whose name is "CONSTRUCTION PARAMETERS".

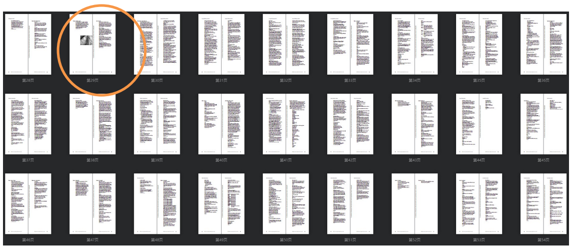
Ground Truth: ["F/UTP CAT5E", "SF/UTP CAT5E"]

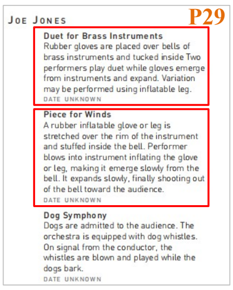
Error Reason:
Inaccurate recognizing of figure and figure names.

GPT-4o:
The page that contains the table named "CONSTRUCTION PARAMETERS" is the 2nd image. The names of the figures on that page are:
1. 300V
2. 500V
Extractor result: ["300V", "500V"]

Figure 16: An example of *Perceptual Error*.

Reasoning Error





GPT-4o:
The performance "Duet for Brass Instruments" involves placing rubber gloves over the bells of brass instruments and having two performers play a duet while the gloves emerge from the instruments and expand. A variation may be performed using inflatable legs.
The performance "Piece for Winds" involves stretching a rubber inflatable glove or leg over the rim of the instrument and stuffing it inside the bell. The performer then blows into the instrument, inflating the glove or leg, making it emerge slowly from the bell and eventually shooting out of the bell toward the audience.
Both performances involve the use of rubber gloves or inflatable legs, but "Duet for Brass Instruments" includes a variation with inflatable legs, which adds an additional element to the performance. Therefore, "Duet for Brass Instruments" has a slightly more elaborate usage due to the inclusion of this variation.
Extractor result: "Duet for Brass Instruments"

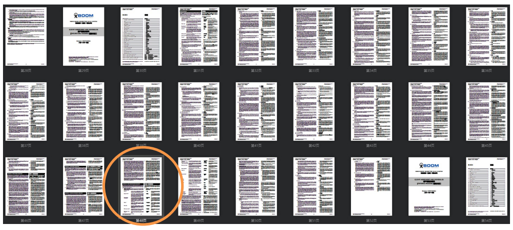
Question: Which performance's usage is more elaborate between 'Duet for Brass Instruments' and 'Piece for Winds'?

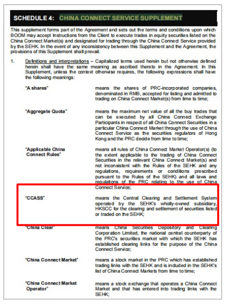
Ground Truth: "Piece for Winds"

Error Reason:
Correct analysis but wrong final summarization.

Figure 17: An example of *Reasoning Error*.

Inconsistent Format





Question: What is the name of the wholly-owned subsidiary of SEHK responsible for clearing and settlement of securities?

Ground Truth: "HKSCC"

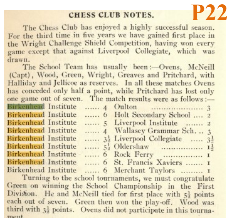
Error Reason:
Abbreviation/full-form inconsistency.

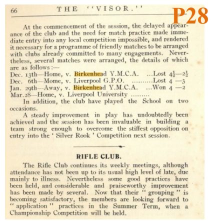
GPT-4o:
The name of the wholly-owned subsidiary of SEHK responsible for clearing and settlement of securities is "Hong Kong Securities Clearing Company Limited" (HKSCC).

Extractor result: "Hong Kong Securities Clearing Company Limited"

Figure 18: An example of *Inconsistent Format*.

Incomplete Evidence





Question: How many matches did the Birkenhead Institute Chess Club play during the season?

Ground Truth: 11

Error Reason:
Ignore evidences in the page 28.
9 (page 22) + 2 (page 28) = 11.

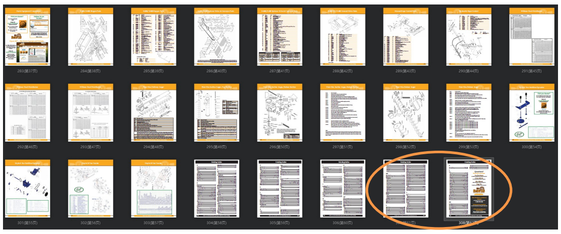
GPT-4o:
The Birkenhead Institute Chess Club played a total of 10 matches during the season. The match results are listed on the 12th image:

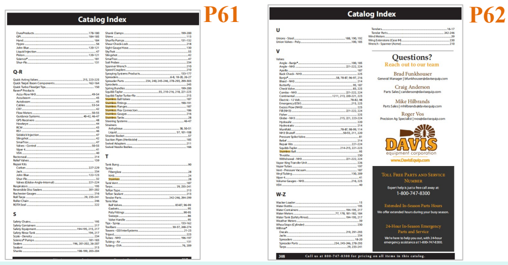
1. Birkenhead Institute vs. Oulton: 4 – 0
2. Birkenhead Institute vs. Holt Secondary School: 6 – 2
3. Birkenhead Institute vs. Wallasey Grammar School: 3 – 3
4. Birkenhead Institute vs. Liverpool Institute: 5 – 1
5. Birkenhead Institute vs. Liverpool Collegiate: 3.5 – 3.5
6. Birkenhead Institute vs. Oldershaw: 5 – 1
7. Birkenhead Institute vs. Rock Ferry: 6 – 0
8. Birkenhead Institute vs. St. Francis Xaviers: 6 – 1
9. Birkenhead Institute vs. Merchant Taylors: 6 – 0

Extractor result: 9

Figure 19: An example of *Incomplete Evidence*.

Irrelevant Answer





Question: How many occurrences of "Stainless" categories are spread across different pages?

Ground Truth: 7

Error Reason:
Query "category", not simple "term".
6 (page 61) + 1 (page 62) = 7.

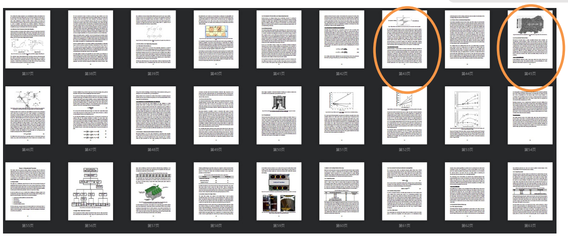
GPT-4o:
Based on the provided images, the term "Stainless" appears in the following instances:
1. Image 1 (Page 279): - Item 26: Carriage Bolt, 5/16 "UNC x 1" Lg, SS - Item 30: Rivet, 5/32 …
2. Image 7 (Page 285): - Item 23: Carriage Bolt, 5/16"

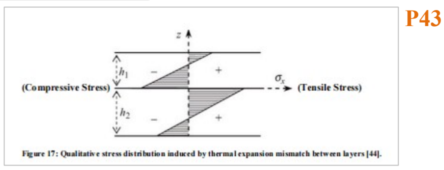
…

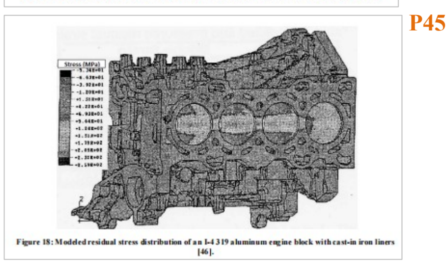
In total, there are 36 occurrences of the "Stainless" category spread across different pages.
Extractor result: 36

Figure 20: An example of *Irrelevant Answer*.

Hallucinated Evidence







Question: Figure 18 illustrates stresses due to thermal expansion mismatch between layers, yes or no?

Ground Truth: no

Error Reason:
When GPT-4o cannot obtain valid evidence from the image, it tends to cater to the description of the query.
In fact, Figure 17 corresponds thermal-mismatch stress while Figure 18 describes the stress in an aluminum engine block.

GPT-4o:
Yes, Figure 18 illustrates stresses due to thermal expansion mismatch between layers.
Extractor result: yes

Figure 21: An example of *Hallucinated Evidence*.