A. The Annotation Results

A.1. Poster and Chart Annotations

As shown in Fig. 6, we present a comparison between our annotations and the original annotations.

In the original poster annotations, some text image layers are damaged, and some bounding boxes are inaccurate, which hinders precise text recognition and localization. In the annotations obtained using our Re-rendering Strategy, we reconstruct the text layers and achieve accurate bounding box annotations. Additionally, we include global content information from the original dataset, such as text format, title and keywords to enhance global awareness during instruction-tuning data generation. We use the values of "text_with_box" as the annotations for poster's full-page parsing data.

In original chart annotations from ChartQA, some bounding boxes are associated with bars/lines rather than text value. This is inconsistent with our text and bounding box correspondence objectives. Additionally, the bounding boxes in the original annotations are not accurate. In our re-rendered annotations, we align the bounding boxes with the text value, and the bounding boxes are accurate. Furthermore, we randomly erase some of the values to ensure that the model can infer the missing values based on other visual information. We use the entire chart JSON dict as the annotations for the full-page parsing of the chart. Due to space constraints, we omit some of the content using '...'.

A.2. PDF Document Annotations

As shown in Fig. 7, we present a comparison between the ordered annotations from MinerU and the unordered but comprehensive annotations from PyMuPDF, along with the combined annotations using our Merge Strategy. We utilize green arrows to indicate the ordered annotations and gray arrows to indicate the naive scanning order. By combining the two annotation methods, we achieve full-page parsing annotations that are both comprehensive and as ordered as possible. Moreover, our method can become more effective as the performance of the ordered annotation tools improves. Due to space constraints, we omit the content in the middle of this passage.

B. Prompts and Instructions

B.1. Prompt Details

In Fig. 8, we show three different prompts for poster, chart and PDF document:

For poster, we deploy plain text input as prompt. Besides format information and rule information, we also provide GPT-4o with some of the overall content and style information from the original Crello dataset. This helps in achieving a better global understanding, thereby improving the generation of instruction-tuning data.

For chart, since the content of the charts only contains some numbers and lacks an introduction to the meaning of the content being statistically represented, we add a question answering data from the original ChartQA to help GPT-4o better understand the meaning conveyed by the chart content. Additionally, when the model generates output, we output the masked values in the grounded format "<ocr>text</ocr><bbox>null</bbox>" as well. After obtaining the output, we perform format filtering to degrade this part of the content into plain text and remove the degraded plain text item in "necessary bbox".

For PDF document, we directly send the images and simple output format rules to GPT-4o, obtaining the output with the original text wrapped in "<ocr></ocr>". Then, we use PyMuPDF to query these contents, find the corresponding coordinates, and normalize them. After that, we wrap the coordinates in "<bbox></bbox>" and append them to the original wrapped text.

After obtaining the output from GPT-4o, we perform format filtering to remove samples that do not meet the format requirements. And we also perform grounded data checking to correct or remove sampls that have incorrect grounding content. We change the grounded blocks "<ocr></ocr><bbox></bbox>" in the questions to "<bbox></bbox>". Finally, we combine various answers and reasoning to obtain different types of tasks introduced in Sec. 4.

B.2. Instruction Details

As shown in Fig. 9, we introduce the instruction utilized in Multi-granular Parsing tasks and the response format prompts which are followed by the questions for instruction-tuning data. For Bounding Box & Text Localization, we introduce 4 instructions for each task, the answer is directly wrapped with "<ocr></ocr>" or "<bbox></bbox>". For Full-page Parsing, we introduce 3 instructions for poster data, 1 for chart, and 1 for PDF document data. The responses are in the format shown in Fig. 6 and Fig. 7.

We add different response format prompts to different questions based on the format of the responses to help the model output corresponding results when users interact with it. For generated question and answer pairs, we combine different question and answer pairs with various response format prompts to obtain diverse grounded data. For Grounded Answering Data, we have 7 response format prompts to be added after the question, and the answer should be directly a simple text wrapped with "<ocr></ocr>" and followed by the coordinates wrapped with "<bbox></bbox>". For Grounded Reasoning Data, we combine two random-chose prompts to-

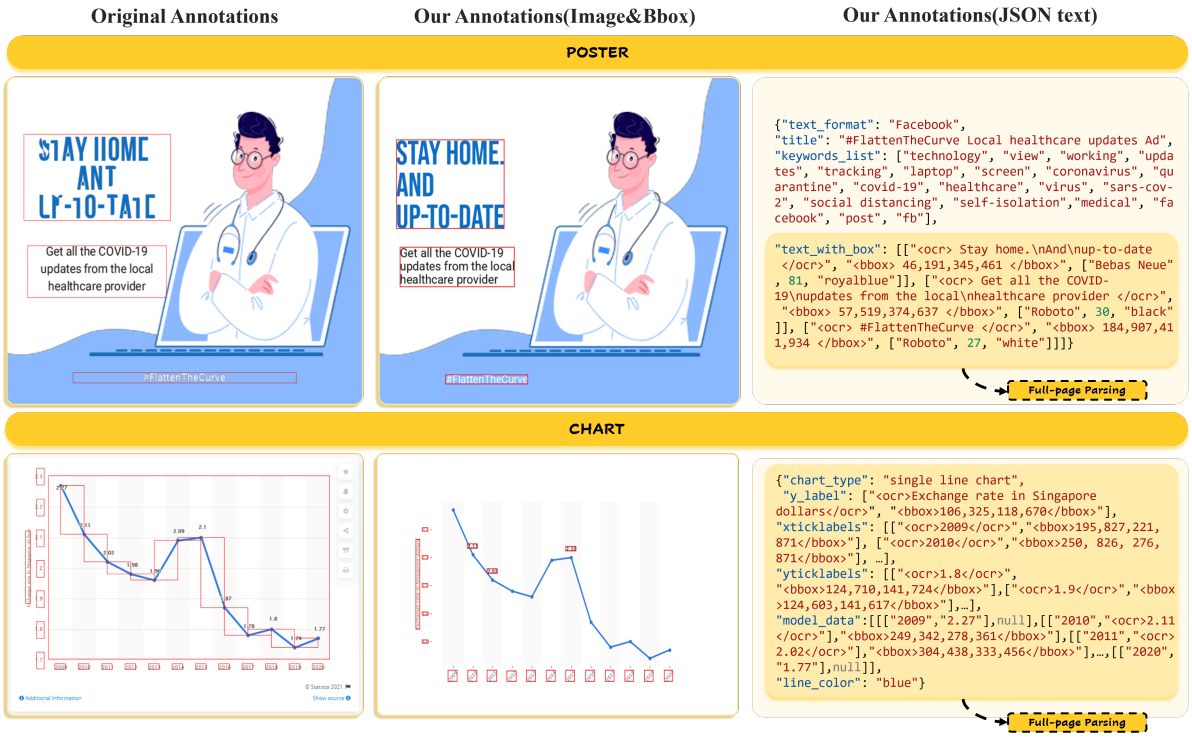
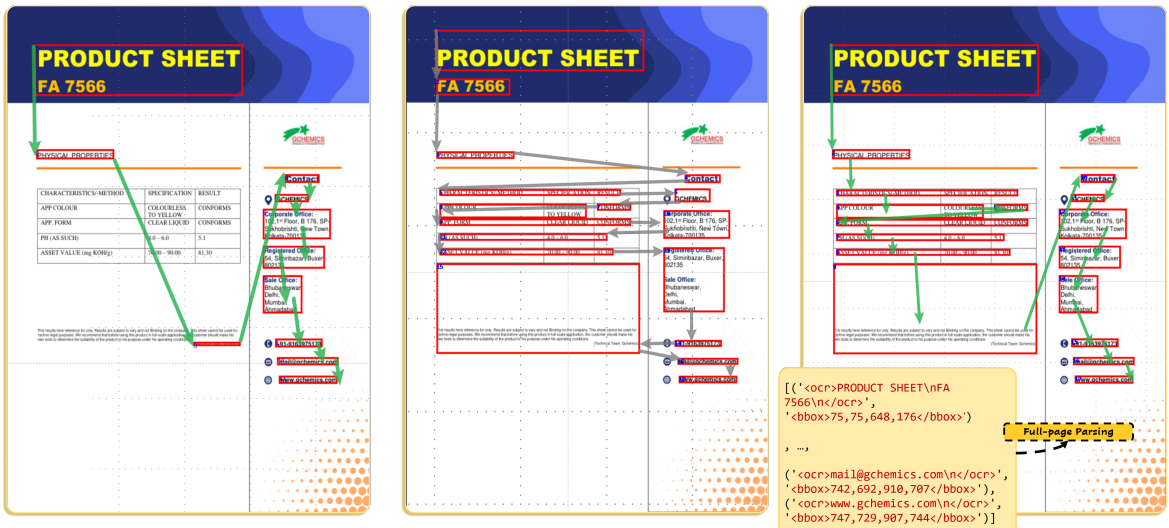


Figure 6. Comparison of origin annotation and our new constructed annotation.



(a) Ordered Annotation from MinerU (b) Unordered Annotation from PyMuPDF (c) Ours using Merge Strategy

Figure 7. PDF Document Parsing results comparison of ordered annotation from MinerU, unordered annotation from PyMuPDF, and our annotation using the Merge Strategy. **Green** arrow indicates the ordered annotations and **Gray** arrow indicates the naive scanning order.

gether to form the response format prompt, and the response should be such a sentence structure involves a segment of grounded reasoning followed by "Answer: " and a concise answer. For Grounded Open-ended Answering, we simply use the first part of the response format prompts for Grounded Reasoning, and the response should be a grounded reasoning sentence. For Plain text Answering, we add no response format prompt to keep consistent with the original question answering.

Poster Q&A Generation Prompts for GPT-4o

System prompt:
You are a poster expert. I'll give you the data about a poster, including some implicit content and a Text box list. Each list item correspond to one box in the poster and represents a segment of text in the poster. The format of each item is [<ocr>text</ocr>, <bbox>bounding box coordinates</bbox>, (font name, font size, font color)].The list order is somewhat disorganized. You need to reorder these text boxes to make them logically smooth. Please generate 3 most valuable question & answer & explanation & The necessary bbox list to obtain the answer for the content in text box list in json format:
[{'question':'','answer':'','explanation':'',' necessary bbox':['<ocr></ocr><bbox></bbox>']},...].
The question types can include factual judgments, factual inquiries, multi-step reasoning, math reasoning, extended understanding, summarize, etc. Note that the implicit content is only for your comprehension for better questions, so don't ask for that content. Note that the response must be related at least one item in the list. In question, you can substitute the text in '<ocr></ocr>' with corresponding bbox in '<bbox></bbox>', but don't ask too simple questions, like 'What is the text mentioned in <bbox>A,B,C,D</bbox>?'.Don't ask about detailed font name or size. When citing a text with corresponding bbox in answer / explanation, enclose it in '<ocr></ocr>' followed by its bbox in '<bbox></bbox>', formatted as follows: '<ocr> apple </ocr><bbox>A,B,C,D</bbox>', where 'A,B,C,D' represents the bbox for 'apple'. The necessary bbox must in your explanation. Assume you directly obtain the poster data based on the poster image, so avoid phrases like 'based on data'. Your answer should be one word or one span, and your explanation should be simple but reasonable.
User prompt:
Here is a poster with the theme [title] in [text_format] style. Here are some related keywords:[keywords_list]. Text box list:[Full-page parsing JSON data]

Chart Q&A Generation Prompts for GPT-4o

System prompt:
You are a chart expert. Here is the data about a chart, including its components and corresponding boun ding boxes (bboxes). Additionally, I will give you a question and answer pair to let you know what the chart data represents for. Please generate 3 most valuable question & answer & explanation &The necessary bbox list. Output in JSON format:
[{'question':'','answer':'',’explanation':'',' necessary bbox':['<ocr></ocr><bbox></bbox>']},...].
The type can include multi step reasoning, math reasoning, extended understanding, color-related resoning, summarize etc.In question, you can generate pure text question, or question with bbox in '<bbox></bbox>' to subsitute text in '<ocr></ocr>'.
When citing a text or data with corresponding bbox in your answer, enclose it in '<ocr></ocr>' followed by its bbox in '<bbox></bbox>', formatted as follows: '<ocr> 50% </ocr><bbox>A,B,C,D</bbox>' or '<ocr> 50% </ocr><bbox>null</bbox>', where ‘A,B,C,D’ or ‘null’ represents the bbox for ‘50%’. The items in “necessary bbox” is the necessary bbox(es) to refer to get the answer, and they must appear in explanation. The items in “necessary bbox” must also be formatted as follows: '<ocr> 50% </ocr><bbox>A,B,C,D</bbox>' or '<ocr> 50% </ocr><bbox>null</bbox>’, where ‘A,B,C,D’ or ‘null ’ represents the bbox for ‘50%’. Assume you directly obtain the data based on the chart image, so avoid phrases like 'based on data', 'according to the dict' or specific color codes (e.g., '#ff3522'). Your answer should be simple.

User prompt: [Full-page parsing JSON data] + [a Q&A sample from ChartQA]

PDF Document Q&A Generation Prompts for GPT-4o

PDF images

+

System prompt:
You are an expert in document reading. Express the original text from the document as much as possible in your reply and use '<ocr></ocr>'to enclose the unique origin key words or spans in document.

User Prompt:
Generate a summary of the document and 3 most valuable question & answer & explanation. Make sure your answer is a word or a span, it is correct and explanation is reasonable. The type can include multi-step reasoning, math reasoning, etc. The explanation should be simple but clear.
Note that the output should be in JSON format, like {'summary':'', 'QAs ':[{'question':'','answer':'','explanation':''},{'question':'','answer':'','expla nation':''}}

Figure 8. Prompts utilized as input to gpt-4o for poster, chart and PDF document.

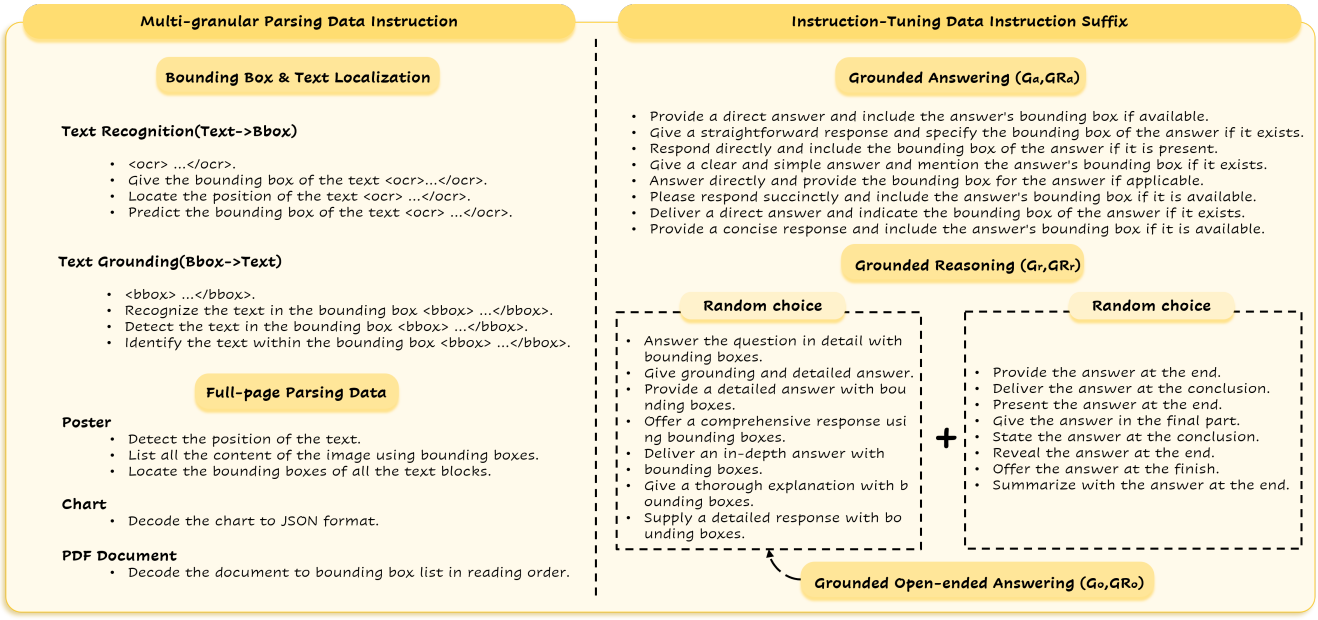


Figure 9. The instruction utilized in Multi-granular Parsing tasks and the response format prompts for instruction-tuning data.

C. Qualitative Results

C.1. Analysis about Fig. 1

We present the inference results of DOGR for grounding, grounding-and-referring, and referring tasks. The specific coordinates of the annotations are omitted, and the grounded text is highlighted with colored boxes. The colors of the boxes within the document image correspond to the colors of the text boxes. For the grounding task, we present four examples. The first sample demonstrates that DOGR can perform fine-grained question answering on general document images with rich content and complex layouts, successfully grounding the corresponding information. The second sample illustrates the model's excellent recognition and localization capabilities for diverse and small text in poster-type images. Additionally, we showcase two samples of grounding in chart-type images, which indicate that DOGR possesses a certain level of mathematical ability, enabling it to provide grounded reasoning during calculation, as well as the capability to estimate values in charts that lack textual annotations based on the axes. For the grounding-and-referring and referring tasks, DOGR is able to recognize the content of user-selected regions and provide reasonable grounded or plain textual reasoning and responses. DOGR exhibits robust fine-grained grounding and referring capabilities, allowing for reliable grounded reasoning and accommodating diverse user interactions, significantly enhancing the overall user experience.

C.2. More Qualitative Results

C.2.1. DOGR-Bench Examples

As shown in Fig. 10, we present three inference examples of DOGR-Bench, demonstrating unlabeled value reading capability, grounding summarization capability, and referring summarization capability. The first figure is an example of reading unlabeled values. We add auxiliary mark boxes in red to the coordinate axes to facilitate reading the values, each box's height represents 5. It can be observed that the model's output values are accurate. The figure on the lower left is an example of grounding summarization for an entire document. DOGR is able to perceive the content of the entire document and perform grounding summarization effectively. The figure on the right is an example of referring summarization for a specific area. The model can also accomplish this task, thereby helping users improve reading efficiency.

C.2.2. Other Examples

It is noteworthy that our model demonstrates strong generalization capabilities beyond some training domains and tasks, as illustrated in Fig. 11.

Strong generalization. As shown in the first row of examples, we feed DOGR with screenshots of the paper content and the specially shaped fan chart that the model does not process before. DOGR is able to correctly respond, demonstrating its strong generalization and usability in actual document reading scenarios.

Handwriting ground-and-refer ability. The middle sample showcases the model's ability to recognize and ground-

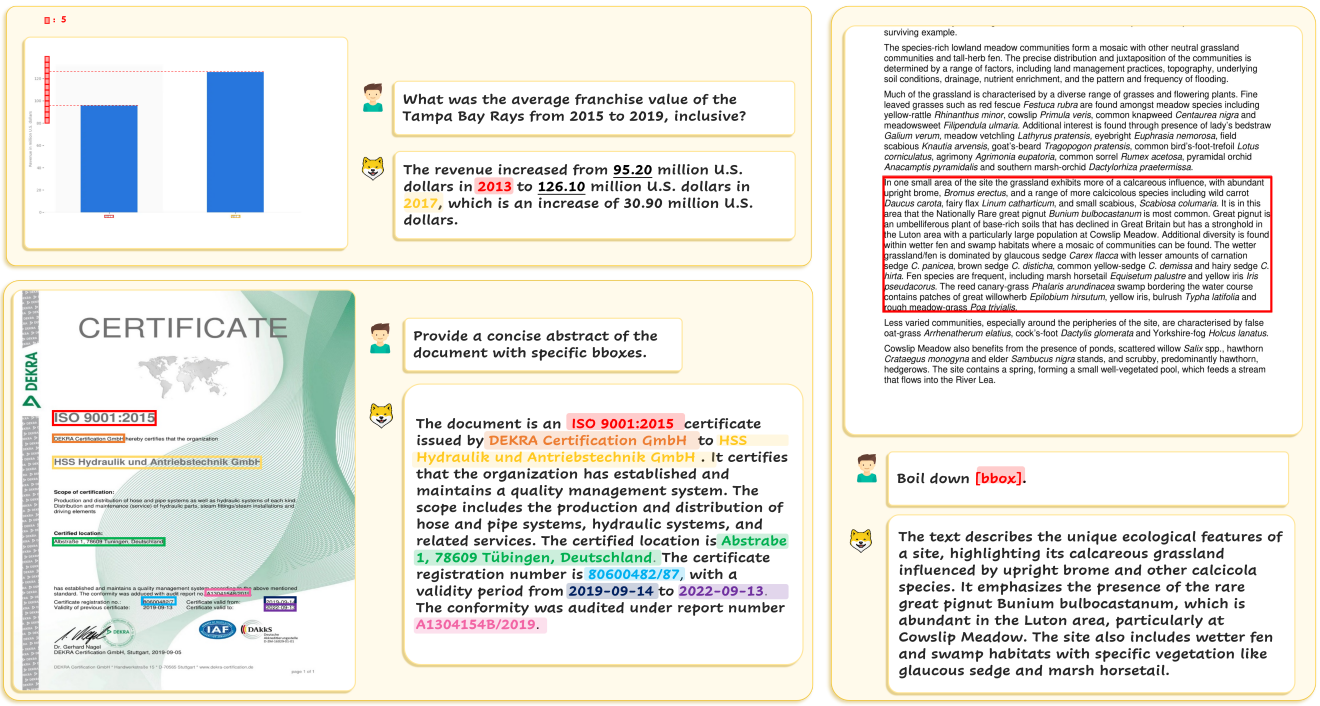


Figure 10. DOGR's inference results on DOGR-Bench.

ing handwritten content. DOGR can fully understand this casual handwriting and provide grounded output. It is worth mentioning that our training data does not contain the grounding or referring tasks on such handwritten images.

Other capabilities. The bottom sample demonstrates an additional untrained capability of our model, specifically referring translation. DOGR is able to provide translations for the asked region. It is important to note that our training data do not include tasks similar to referring translation, even rarely includes languages other than English. Therefore, we believe that DOGR can effectively handle the relationship between regions and corresponding text, and seamlessly integrate the capabilities of large models with grounding and referring abilities.

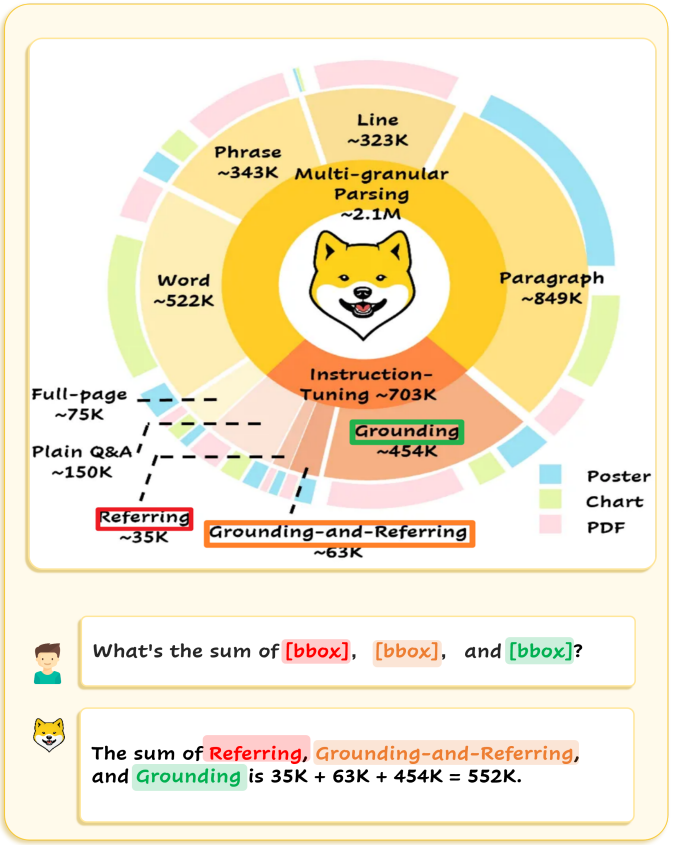
C.2.3. Failure Cases

Although DOGR demonstrates strong grounding and referring capabilities, there are still some shortcomings, as shown in Fig. 12.

Referring. The upper left figure illustrates an example of incorrect referring. It mistakenly associates the bounding box that should correspond to the text "INITIATIVE" in the question with the text "LOYALTY" below it. Additionally, its width is re-estimated according to the size corresponding to "LOYALTY," resulting in an incorrect answer.

Grounding. When encountering some unfamiliar text content such as tables in the upper right image, DOGR can understand the content, effectively identified the line breaks between "Doc" and "VQA" and merged them together, and provide correct answers, but the grounding boxes are inaccurate. There is also a issue of incomplete grounding content, such as the bottom sample, which often occurs when the content requiring grounding is interrupted or wrapped to the next line. Although this does not affect understanding, the text and bounding boxes provided by the model do not completely match.

Comprehension. When facing with some unfamiliar structural document, such as the chart shown in the middle right, the model gives incorrect answers. However, due to the intuitive expressiveness of grounding and referring, readers can quickly determine that the model's answer is incorrect. These samples can also provide evidence of the model's deficiencies, laying the foundation for further improvements.



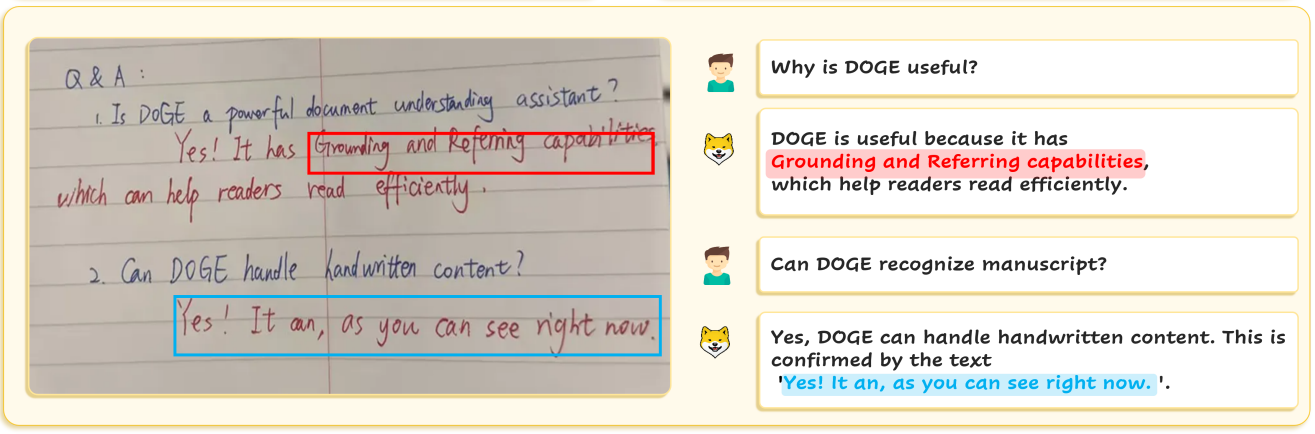
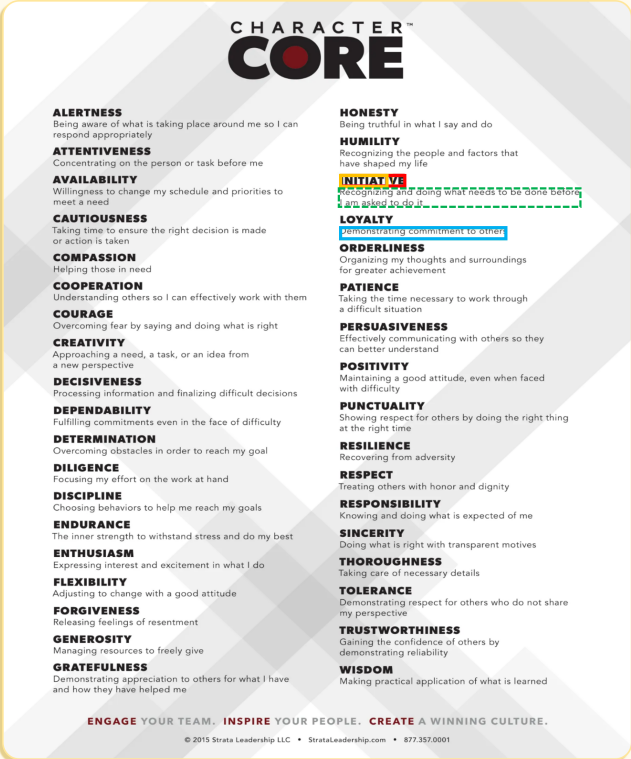


Figure 11. Other DOGR's inference samples.

Incorrect Referring



How is [bbox] described in the document?

LOYALTY is defined as the commitment to others, emphasizing the importance of being dedicated and supportive in relationships.
Answer:
Demonstrating commitment to others

Correct Answer:
Recognizing and doing what needs to be done before I am asked to do it.

Inaccurate Grounding

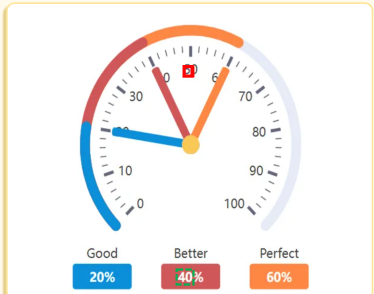
Table 3. Performance comparison on 10 general document benchmarks.

Model	Size	Doc VQA	Info VQA	Deep Form	KLC	WTQ	Tab Fact	Chart QA	Text VQA	Text Caps	Visual MRC
Donut[13]	1B	67.5	11.6	61.6	30.0	18.8	54.6	41.8	43.5	74.4	93.91
DocOwl[9]	7B	62.2	38.2	42.6	30.3	26.9	60.2	57.4	52.6	111.9	188.8
UReader[43]	7B	65.4	42.2	49.5	32.8	29.4	67.6	59.3	57.6	118.4	221.7
TextMonkey[16]	9B	73.0	28.6	59.7	37.8	31.9	-	66.9	65.9	-	-
Vary[39]	7B	76.3	-	-	-	-	-	66.1	-	-	-
TokenPacker[40]	13B	70.0	-	-	-	-	-	-	-	-	-
DocPeida[8]	7B	47.1	15.2	-	-	-	-	46.9	60.2	-	-
QwenVL[1]	9B	65.1	35.4	-	-	-	-	65.7	63.8	-	-
IXC 2.5[50]	7B	90.9	70.0	**71.2**	-	53.6	**85.2**	82.2	**78.2**	-	307.5
DocOwl-1.5-Chat[10]	8B	82.2	50.7	68.8	38.7	40.6	80.2	70.2	68.6	131.6	246.4
DocOwl-2[11]	8B	80.7	46.4	66.8	37.5	36.5	78.2	70.0	66.7	131.8	217.4
InternVL2[6]	8B	91.6	**74.8**	-	-	-	-	83.3	77.4	-	-
DOGE	8B	**91.7**	70.7	70.8	**40.4**	**58.8**	84.5	**83.6**	76.6	**145.9**	**332.5**

Which mehod has the best performance on DocVQA?

The method with the best performance on DocVQA is DOGE, achieving a score of 91.7.

Incorrect comprehension

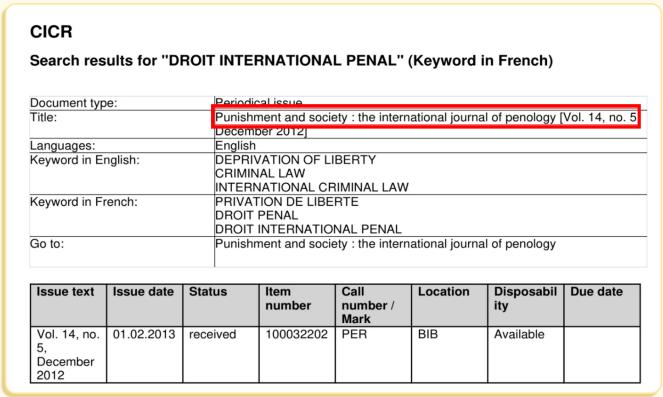


What's the value of the red pointer?

The red pointer is at 50.

Correct Answer:
The red pointer is at 40.

Incomplete Grounding



CICR

Search results for "DROIT INTERNATIONAL PENAL" (Keyword in French)

Document type:	Periodical issue
Title:	Punishment and society : the international journal of penology [Vol. 14, no. 5 December 2012]
Languages:	English
Keyword in English:	DEPRIVATION OF LIBERTY CRIMINAL LAW INTERNATIONAL CRIMINAL LAW
Keyword in French:	PRIVATION DE LIBERTE DROIT PENAL DROIT INTERNATIONAL PENAL
Go to:	Punishment and society : the international journal of penology

Issue text	Issue date	Status	Item number	Call number / Mark	Location	Disposability	Due date
Vol. 14, no. 5, December 2012	01.02.2013	received	100032202	PER	BIB	Available	

What is the title of the periodical issue related to 'DROIT INTERNATIONAL PENAL'?

Punishment and society : the international journal of penology [Vol. 14, no. 5, December 2012]

Text in its bbox:
Punishment and society : the international journal of penology [Vol. 14, no. 5,

Figure 12. Failure cases of DOGR.

Table 6. The detailed composition of the DOGR dataset's Multi-granular parsing data and Instruction-Tuning data.

Type	Number	SubType	Number	SubSubType	Number
Multi-granular Parsing	2,114,414	Full Page	75,391	Poster	20,867
				Chart	31,716
				PDF	22,808
		Word	522,682	Poster	65,446
				Chart	354,731
				PDF	102,505
		Span	343,596	Poster	56,006
				Chart	58,212
				PDF	229,378
		Line	323,211	Poster	5,577
				Chart	6,268
				PDF	311,366
		Paragraph	849,534	Poster	511,998
				Chart	229,378
				PDF	108,158
Instruction-Tuning	703,724	Grounding	454,404	Poster	96,718
				Chart	62,636
				PDF	295,050
		Grounding + Referring	63,243	Poster	36,663
				Chart	849
				PDF	25,731
		Referring	35,197	Poster	19,207
				Chart	618
				PDF	15,372
		Plain Text Q&A	150,880	Poster	35,831
				Chart	45,948
				PDF	69,101

D. Dataset Details

D.1. Dataset Statistic

As shown in Tab. 6, we present the detailed composition of the DOGR dataset's Multi-granular Parsing data and Instruction Tuning data. The Multi-granular Parsing data includes five granularities: word, phrase, line, paragraph, and full-page parsing. The Instruction-Tuning data comprises three types of grounded data: grounding referring, grounding-and-referring, and plain text Q&A. We provide the detailed data volume for each type.

Fig. 13 shows the distribution of block counts for poster, chart, and PDF document. Poster has fewer grounded blocks within each image, larger areas, diverse font styles, and larger font sizes. The average text length per block for poster is 3.25 words. Chart, on the other hand, has a higher number of blocks with small areas and small font sizes. Each block in a chart corresponds to a small component value within the chart, with an average text length of 1.34 words per block. PDF document has a moderate distribution of block counts, larger areas, and small font sizes. Each block in a PDF contains relatively longer text, with an average length of 22.55 words per block.

D.2. Construction Cost Details

In terms of multi-granular parsing data construction, DOGR-Engine can achieve boundary box annotation and content extraction on any document dataset with a similar rendering process. It can also scale up to a larger volume of PDF source files without additional manual costs. When constructing instruction fine-tuning data, our main cost lies in the API calls to GPT-4o. We use the version gpt-4o-2024-08-06. For poster and chart, our input is long text containing full page parsing data, while for PDF, the input is document images plus short prompt texts. During construction, we can generate multiple question and answer pairs for a single image simultaneously and use batch API requests to save costs. Ultimately, we can construct over 1,000 grounded question and answer pairs for an average cost of $1, which is far more efficient and cost-effective than manual construction.

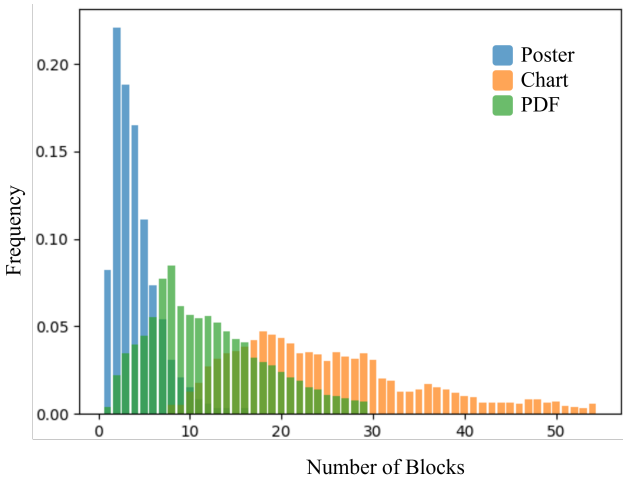


Figure 13. Block count distribution of poster, chart, and PDF.

D.3. Hallucinations Avoidance

Regarding hallucinations, GPT-4o is quite capable of handling our tasks. By manual checking, we observe that hallucinations occur **very rarely**. We also filter out samples where the text and bounding boxes do not correspond correctly. Additionally, for DOGR-Bench, we manually filter out the samples that contain hallucinations or other errors.

E. Training Data Details

For the pre-training data, we utilize DocStruct4M [14] along with our 2.1M multi-granular parsing data to enhance DOGR's foundational grounding and referring capabilities.

For the fine-tuning data, we compose our 703k Instruction-Tuning data with 575k DocDownStream-1.0 data from [14] and 716k other document-related data from various datasets, resulting in a final fine-tuning dataset of 2M. In Tab. 7, we show the detailed data source and sampled number of 716k other document-related data.

Table 7. The detailed statistics of 716k other document-related data.

Dataset	# Samples	Dataset	# Samples
IIIT5K [36]	1,990	RoBUT WTQ[65]	38,241
TextOCR-GPT4V [3]	25,104	AI2D (InternVL [6])	12,403
FigureQA [18]	1,000	Infographic VQA [35]	8,489
Diagram Image2Text	295	LRV Chart [26]	1,776
K12 Printing	20,000	SROIE	33,616
AI2D (GPT4V Detailed Caption) [19]	4,864	MultiHiertt	7,614
VisText [46]	9,964	RoBUT WikiSQL	74,984
ChartQA [32]	18,260	VisualMRC[45]	3,022
DVQA [17]	20,000	TextCaps [41]	21,942
Magpie Pro [56] (L3 ST)	50,000	Chart2Text [37]	26,956
HiTab [9]	2,495	HME100K [61]	74,492
RoBUT SQA	8,509	Magpie Pro (L3 MT)	50,000
ChromeWriting [55]	8,825	Magpie Pro (Qwen2 ST)	50,000
Screen2Words [51]	15,725	Rendered Text [55]	9,995
IAM [31]	5,658	TQA [20]	27,302
AI2D (Original)	2,429	SynthDog-EN [21]	40,000
MMC bInstruction Arxiv[27]	20,000	MMC bInstruction NON-Arxiv	20,000

F. Detailed Evaluation of MLLMs on DOGR-Bench.

F.1. Evaluation Prompt

Since models other than InternVL2.5 and QwenVL2.5 do not have a clearly defined format for grounding or referring content, we directly interpret the input using the prompt shown in Fig. 15(Right) and restrict the output format for evaluation. For InternVL2.5, the use of `<ref></ref>` is required to trigger grounding capabilities. Therefore, when outputs require answers and corresponding coordinates, we additionally include `<ref></ref>` to obtain a more standardized output for evaluation. For QwenVL2.5, since the model outputs pixel coordinates of the input image rather than the coordinates normalized according to the respective width and height, we first convert the coordinates in our input to facilitate the model's accurate understanding of location information. We also restrict the output content to closely mimic the existing JSON format for evaluation purposes. For example, in reasoning tasks, the model first outputs a JSON list, then generates an explanation, and finally provides the answer. After obtaining the output, we normalize the output coordinates based on the input size for evaluation. The prompt we use for Qwen2.5 is shown in Fig. 15(Left).

F.2. Limited Grounding Ability

Although both InternVL2.5[7] and QwenVL2.5[1] can perform general image grounding, such as providing bounding boxes for a target based on a description, as shown in Fig. 16, when handling document grounding, we use a grounding-triggering prompt to obtain the output coordinates corresponding to the given content. InternVL2.5-72B consistently outputs inaccurate and unstable bounding boxes, while QwenVL2.5 also produces a large number of inaccurate bounding boxes. These simple examples illustrate the models' shortcomings in document grounding capabilities.

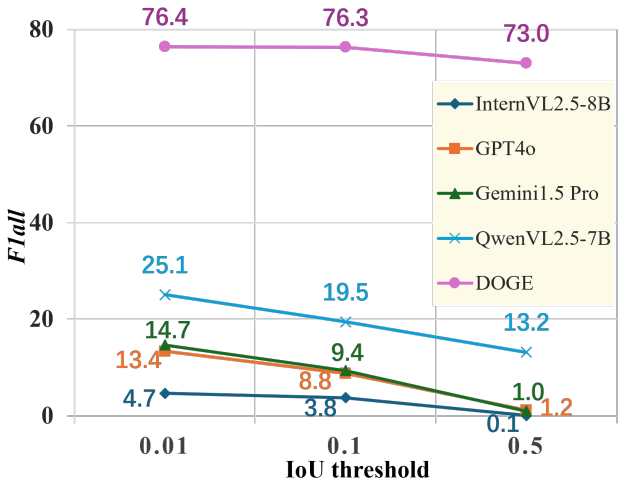


Figure 14. MLLM's G_a $F1_{all}$ comparison on different IoU threshold.

F.3. IoU Threshold Sensitivity

We also conduct a statistical analysis of the G_a $F1_{all}$ scores of different MLLMs at various IoU thresholds. As shown in Fig. 14, among the MLLMs analyzed, apart from DOGR, Qwen2.5VL-7B exhibits the best grounding capability. DOGR is not sensitive to changes in the IoU threshold, indicating that the limitation on DOGR's G_a $F1_{all}$ is due to the model's reasoning capabilities. DOGR has already overcome the performance degradation caused by inaccurate grounding. In contrast, the insufficient basic grounding capabilities of other MLLMs make them very sensitive to changes in the IoU threshold, with their G_a $F1_{all}$ scores dropping to nearly 0 when the IoU threshold is set to 0.5.

G. More Experiments

Effectiveness of our pretraining data. To better assess base performance of pretrained model using our multi-granular parsing data, we further introduce our text&bbox localization test set **DOGR-Local15k** comprising four granularities (word, phrase, line, paragraph) across three categories of data: poster, chart, and general document, similar to DocLocal4k[14]. DOGR$^-$ is pre-trained without using our constructed multi-granular parsing data. For the chart data, we claim that the chart type annotations in DocStruct4M for during DOGR$^-$'s pre-training are inconsistent with our grounding training objectives, which aims to match the text and the bbox. In the original data, the text corresponds to bounding boxes of bar/line charts. Therefore, these value is not referenceable, and we mark them in gray. Moreover, in the case where the granularity of data categories for charts is relatively singular, we directly re-

Table 8. Grounding and recognition performance comparison of the same model pretrained w. and w.o. our pretraining data.

	Poster					PDF Document					Chart
Text Grounding											
Model	word	phrase	line	paragraph	ALL	word	phrase	line	paragraph	ALL	ALL
DOGR$^-$	58.0	63.12	58.96	76.43	63.19	33.88	47.62	51.5	62.88	48.97	24.03
DOGR	**91.88**	**94.62**	**96.28**	**91.04**	**93.53**	**70.25**	**82.25**	**87.5**	**83.62**	**80.91**	**86.12**
Text Recognition											
DOGR$^-$	83.34	60.96	41.67	36.95	55.73	49.66	58.62	50.34	47.27	51.47	68.86
DOGR	**92.94**	**94.05**	**86.52**	**86.08**	**89.9**	**73.57**	**86.48**	**76.85**	**71.8**	**77.17**	**95.24**

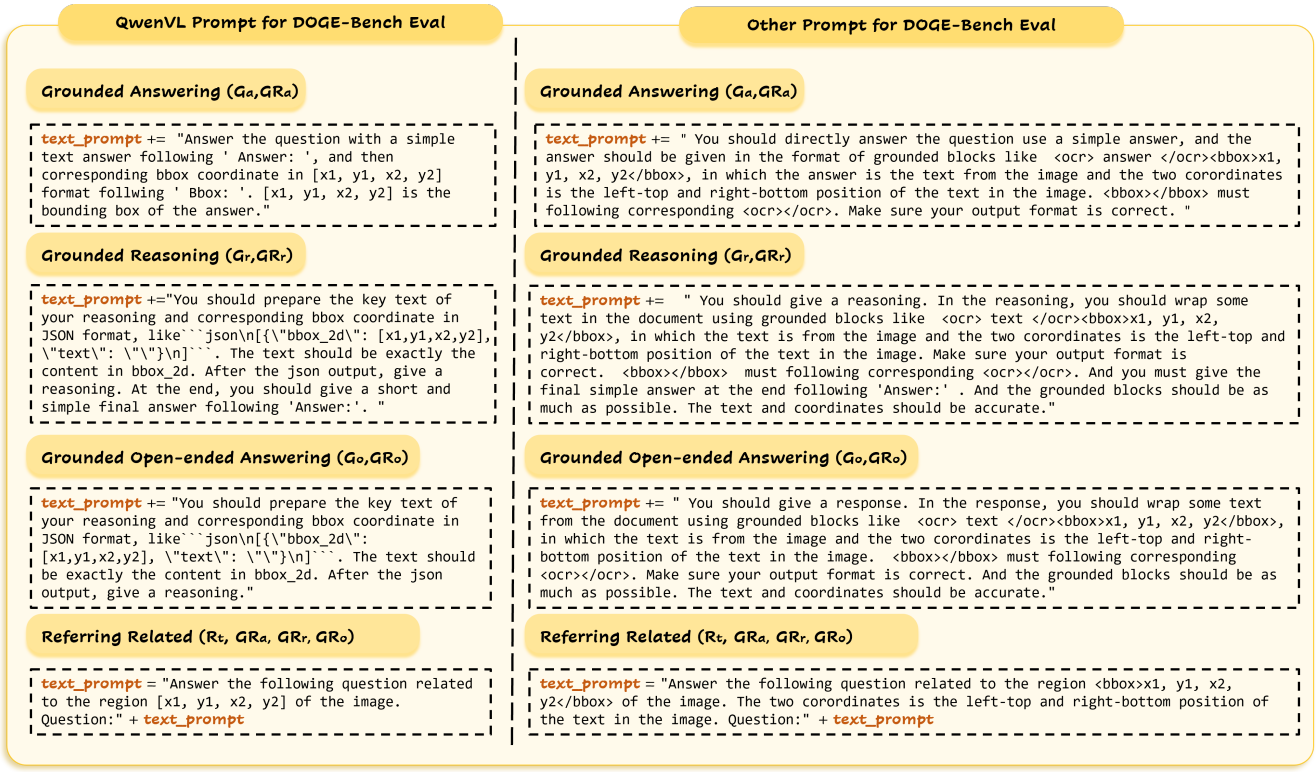


Figure 15. Prompts using in evaluation MLLMs on DOGR-Bench.

ported the results for "ALL".

As shown in Tab. 8, after incorporating our multi-granular parsing data for pre-training, the model exhibits enhancements in text recognition and grounding tasks. Our improvements on DOGR-Local15k are significant, thereby laying a solid foundation for accurate document grounding interactions.

The performance boost of data generated by GPT-4o. We utilize a weaker LLM, *Gemini 1.5 flash*, to generate 25k chart grounding QA data (Gemini25k). As shown in Tab.9, both Gemini25k and GPT-4o25k lead to improvement on ChartQA, which is mainly because constructed QA data enriches the original data. However, GPT-4o's stronger instruction-following capability results in a larger amount of grounded texts in the constructed QA, making it better than gemini-1.5-flash on G_r task.

Importance of accurate bounding box annotation. To validate importance of accurate bounding box annotation, we conduct experiments on chart data. We perform random offset and scaling of bboxes in GPT4o data within 30% to get 25k G_r inaccurate training data GPT4o25k$^-$ with errors for chart. We compare the performance gain of 25k

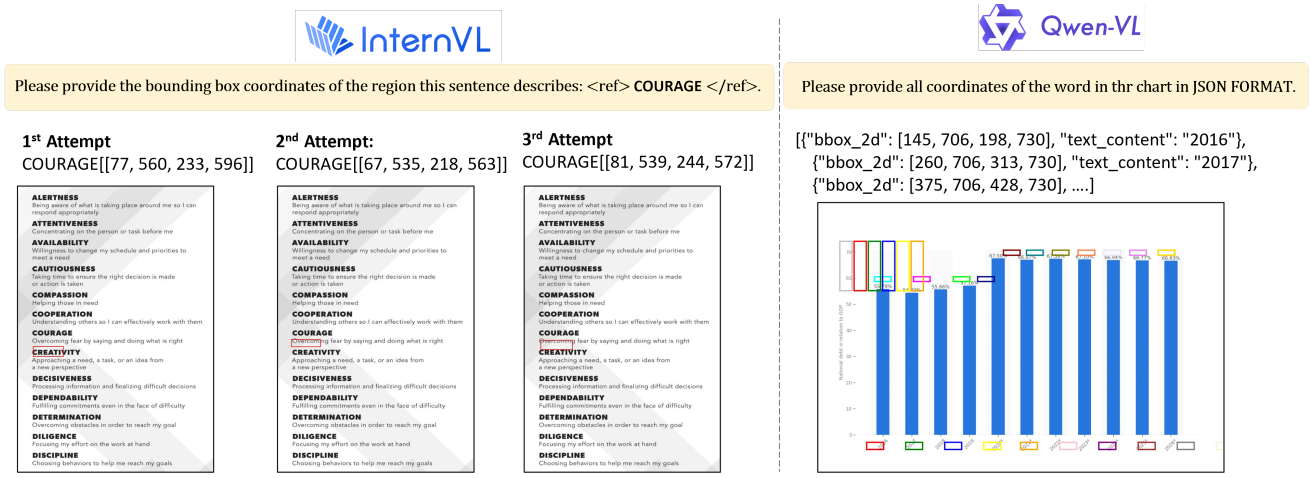


Figure 16. QwenVL2.5-72B and InternVL2.5-72B are all struggle with document grounding.

rerendered data with accurate bboxes and 25k grounded data with inaccurate bboxes on ChartQA and G_r of chart on DO-Bench. Both text are from GPT-4o. As shown in Tab.10, adding GPT4o25k$^-$, the performance on ChartQA is improved and the model possesses grounding capabilities. Adding accurate annotation data GPT4o25k$^-$ leads to more precise bboxes, thus largely improving the grounding performance(F1 of G_r).

Table 9. The performance boost of data generated by GPT-4o and weaker Gemini 1.5 flash.

Training Data	ChartQA-Acc	Gr-Acc	Gr-F1
baseline(ChartQA)	80.92	-	-
+GPT4o25k	81.88	67.0	32.91
+Gemini25k	81.98	50.12	1.10

Table 10. Accurate bbox is helpful for grounding.

Training Data	ChartQA-Acc	Gr-Acc	Gr-F1
baseline(ChartQA)	80.92	-	-
+GPT4o25k$^-$	81.42	66.1	20.2
+GPT4o25k	81.88	67.0	32.91