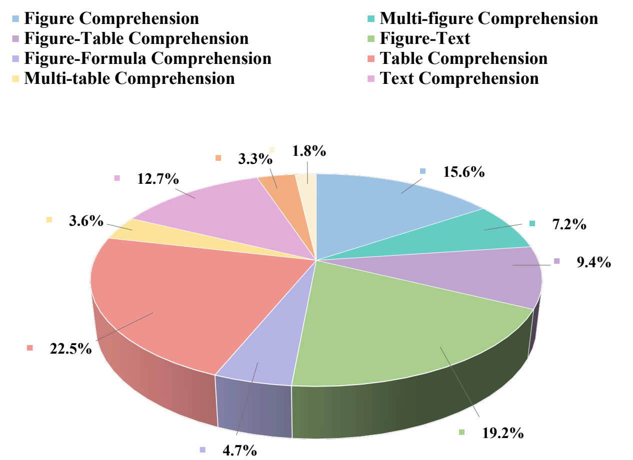
Supplementary Materials of MMCR: Benchmarking Cross-Source Reasoning in Scientific Papers

A. Benchmark Details

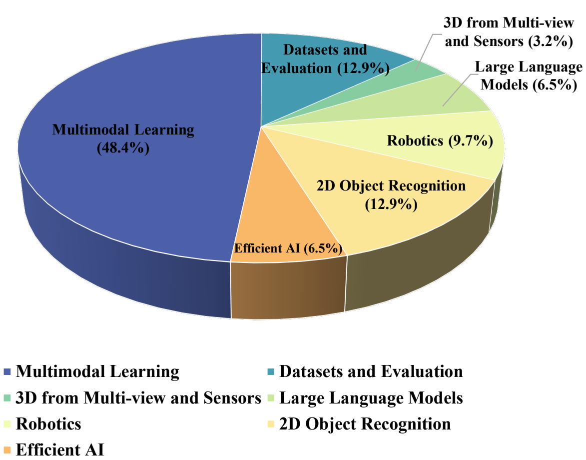
We appreciate the reviewer's thoughtful comment on MMCR's classification as a reasoning benchmark. We respectfully maintain that MMCR tests reasoning capabilities as it aligns with the reviewer's cited definition of reasoning as "multi-step/multi-hop question answering."Our definition of "cross-source reasoning" in MMCR refers to questions that require synthesizing information from multiple sources within scientific papers to derive answers that cannot be obtained from any single source alone.

Taking the question in Figure S.7 as an example, the reasoning process in MMCR directly parallels HotpotQA's definition of "inferring the bridge entity to complete the 2nd-hop question." As illustrated in Figure S.7, answering MMCR questions typically requires first inferring which specific figure or table is being referenced through textual descriptions (e.g., "the t-SNE visualization of CLIP encoding features"). This constitutes the first reasoning hop and establishes the critical bridge entity. Only after successfully identifying this bridge entity can the model proceed to the second hop, extracting relevant information from the identified sources and synthesizing it to derive the answer.

The reasoning complexity is further amplified by scientific papers' high information density (19 pages average), input as pure images rather than OCR-processed text, and questions requiring numerical reasoning and calculations. The benchmark comprises scientific papers across seven academic subjects of artificial intelligence, with questions systematically categorized into ten distinct types based on their required evidence sources. The distribution of these categories is illustrated in Fig. S.1. Representative examples demonstrating each evidence type are presented in Figs. S.5 to S.14.



(a)



(b)

Figure S.1. Distribution of Questions by Evidence Types and Research Domains. (a) Percentage distribution across ten evidence source types. (b) Distribution across seven AI research subjects.

B. Evaluation Details

B.1. Evaluation Prompt

Figure S.2 presents the prompts with and without the use of Chain-of-Thought (CoT). For InternVL2.5, we employed the official CoT prompt released by the developers. For the remaining benchmark models—MiniCpm-o 2.6, Qwen2.5-VL, and Idefics3—we implemented a unified CoT prompt to ensure methodological consistency across experiments.

Evaluation Prompts: CoT Prompt

CoT Prompt for InternVL-2.5:
"Your task is to answer the question below. Give step by step reasoning before you answer, and when you're ready to answer, please use the format: '\Final answer: ... \'
Question: {question}"

CoT Prompt for MiniCpm-o 2.6, Qwen2.5-VL, Idefics3:
"Carefully read the following multichoice question, solve it step by step and finally pick the option associated with the correct answer in the format of 'Answer: Selected option."

w/o CoT Prompt:
"Please select the correct answer from the options above."

Figure S.2. Evaluation Prompt

B.2. Answer Option Inference Details for LLM Responses

B.2.1. Implementation Details

Unlike existing benchmarks that employ LLM-based methods for open-ended response extraction, our benchmark utilizes a heuristic rule-based approach for multiple-choice answer inference.

The rule-based approach for multiple-choice answer inference comprises two stages: primary option-based inference, followed by text-based inference as a fallback strategy. Specifically, the option-based inference method counts the occurrence of option identifiers (A, B, C, etc.) in the response. A valid inference is made when exactly one option identifier is detected. When option-based inference fails, the text-based inference serves as a fallback mechanism. It converts both the model response and choice contents to lowercase before searching for exact matches of choice content within the response. This method succeeds only when precisely one choice content is found in the processed response.

This two-stage approach ensures robust answer extraction while maintaining high precision through strict matching criteria. When both methods fail to identify a unique answer, false will be returned to indicate inference failure.

B.2.2. Existing Problems

The rule-based approach for multiple-choice answer inference offers efficiency by eliminating additional LLM calls. However, it occasionally fails to accurately extract responses despite correct model reasoning. We categorize such cases as *Matching Errors*. As shown in Tab. S.1. The error distribution analysis demonstrates that extraction failures represent a negligible proportion of the total errors, with only two instances identified across all cases. Figure S.21 illustrates one representative example of such a Matching Error.

Matching errors	Total error cases	Error Rate
2	109	1.8%

Table S.1. Example Distribution of Matching Errors in Response Extraction from GPT-4o.The table shows the proportion of matching errors among all error cases, demonstrating that extraction failures constitute only 1.8% of total errors identified in our evaluation.

C. Extended Analysis

C.1. Analysis of Error Cases

We conducted systematic error analysis of GPT-4o's performance on our benchmark to investigate its limitations in cross-source reasoning within scientific papers. Through manual examination of 109 incorrect responses, we identified seven distinct error categories. A comprehensive analysis of all error categories, accompanied by representative examples, is presented in (Figs. S.15 to S.21).

C.2. Performance Across Evidence Types

We analyze model performance across different evidence source types, with detailed results presented in Fig. S.3. The radar chart visualization demonstrates GPT-4o's consistent superiority across most categories compared to the other five VLMs. Particularly in text comprehension tasks, both QwenVL-2.5-72B and GPT-4o achieve notable accuracy (68.57 and 65.71 respectively), likely benefiting from their extensive pretraining corpora.

However, substantial performance degradation is observed in cross-source integration tasks featured in MMCR, particularly in Figure-Text Comprehension, Figure-Table Comprehension, and Multi-Figure Comprehension, where the majority of VLMs achieve accuracy scores below 50. The pronounced disparity between single-source and cross-source task performance reveals a fundamental limitation: while MLLMs exhibit proficiency in individual modality processing, they demonstrate reduced effectiveness in tasks requiring synthesis of information from heterogeneous sources.

C.3. Annotation requirements

Before initiating the formal annotation process, a systematic taxonomy of task types and subject domains was established. This methodological framework ensures annotation

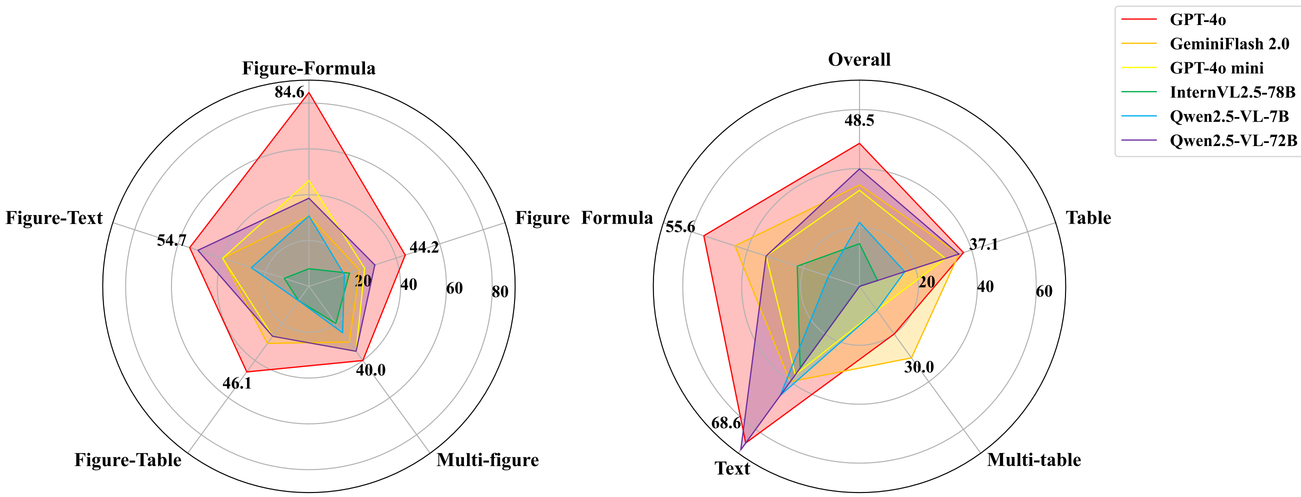


Figure S.3. Fine-grained results on various evidence source types.

consistency and maintains rigorous quality standards across the dataset construction process. Rigorous quality control protocols were implemented throughout the annotation process to establish a robust benchmark for evaluating the comprehensive capabilities of VLMs. Question formulation followed a structured protocol that integrates document-specific content with domain knowledge requirements, establishing a rigorous framework for in-depth assessment of scientific paper comprehension. The questions in MMCR are designed to evaluate comprehensive document understanding, specifically focusing on cross-source reasoning capabilities. The detailed evaluation requirements are illustrated in Fig. S.4.

C.4. Annotation process

The annotation process was conducted by expert annotators, who underwent comprehensive training to ensure annotation consistency and quality standards. The standardized training protocol comprised several systematic phases: 1) **Initial standardization:** The project leader provided annotated sample papers to the annotators, which were subject to multiple rounds of verification. This iterative process ensured that the annotators fully understood the expectations and standards required for the annotation. 2) **Domain-specific allocation:** Annotators were assigned to subject domains aligned with their primary research expertise, selecting one to two domains from predefined categories. Within each domain, five representative publications were systematically identified for annotation. This domain-specific allocation ensures optimal alignment between expert knowledge and content analysis, maintaining annotation quality and disciplinary rigor. 3) **Quality assurance:** Completed annotations underwent systematic review by the project coordinator to ensure adherence to established protocols. When deviations from annotation standards were identified, annotators received structured feedback and supplementary training for remediation. This iterative quality assurance process continued until all annotations achieved compliance with predetermined quality benchmarks. 4)**Supplementary annotation:** In the final phase, expert annotators conducted supplementary annotation rounds in accordance with established protocols to expand the dataset while maintaining consistency standards.

Examples for Annotation Requirements

Requirement I:
"The question is specifically designed to examine the cross-source reasoning ability of the model in scientific papers, and it must be correctly derived exclusively from the designated information source, not from any additional information source. Annotators must strictly adhere to this requirement."

Requirement II:
"In order to mitigate the risk of the model relying exclusively on prior knowledge to generate responses, the annotated questions and answers were meticulously structured to ensure that accurate responses could only be derived by synthesizing information distributed across multiple pages of the document. This approach prevents the model from bypassing the intended reasoning process and reduces the potential for information leakage or unintended biases that might arise from relying on external knowledge."

Requirement III:
"To further ensure that the model genuinely comprehends the content of each paper, at least eight questions were annotated for every paper. This requirement was set to ensure a robust and comprehensive evaluation of the model's understanding across a variety of aspects within each paper."

Figure S.4. Annotation requirements

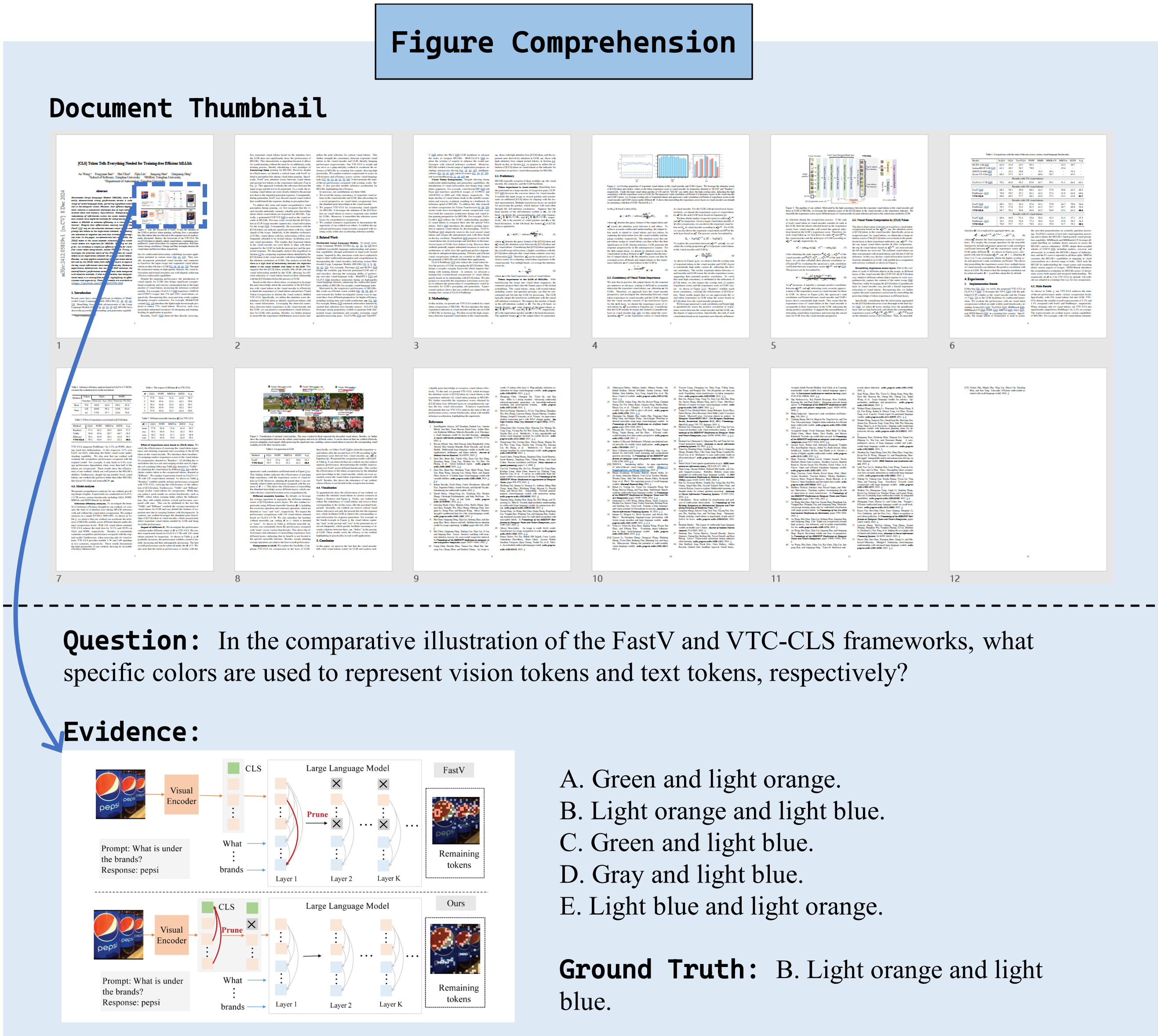


Figure S.5. **The demo of figure comprehension**.

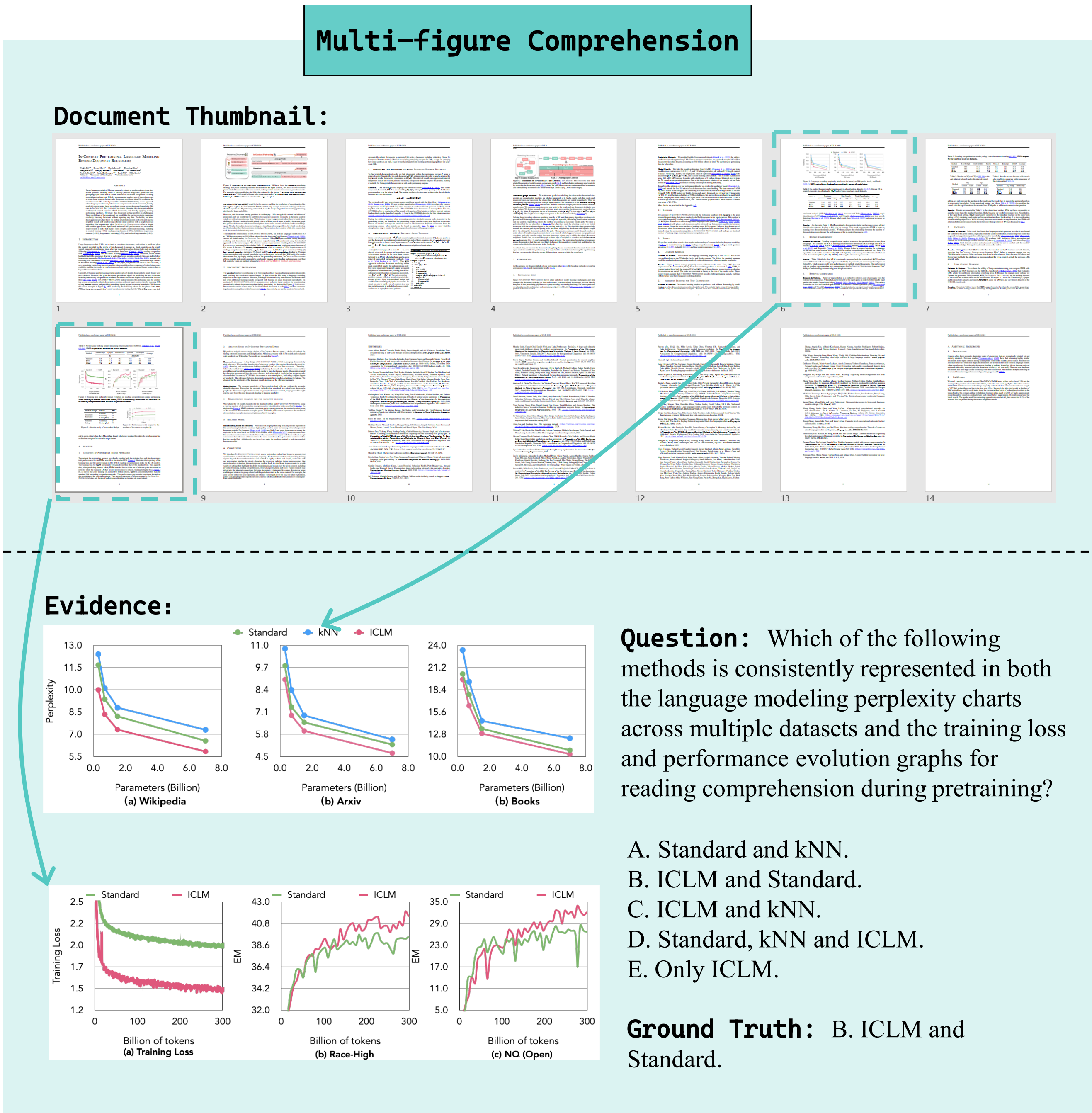
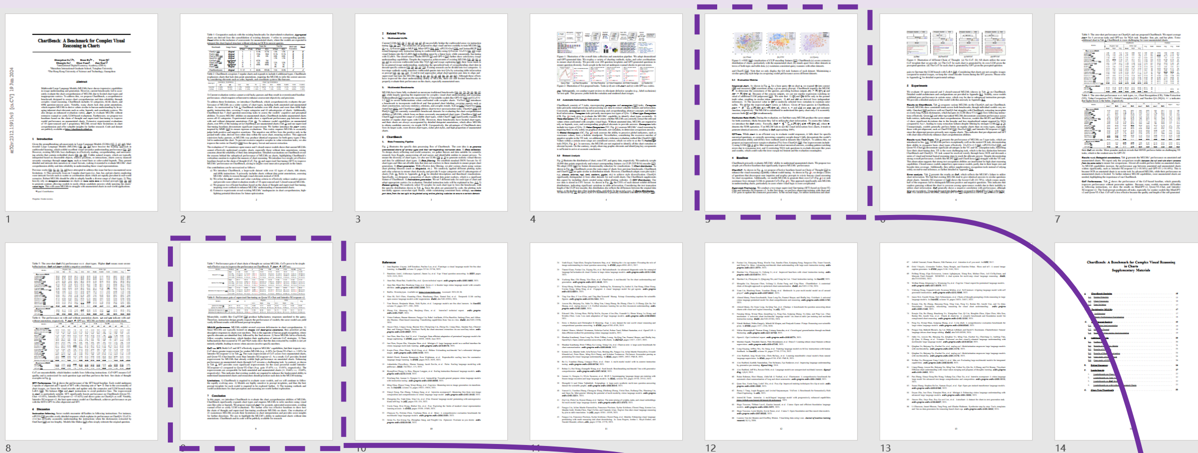


Figure S.6. **The demo of multi-figure comprehension**.

Figure-Table Comprehension

Document Thumbnail:

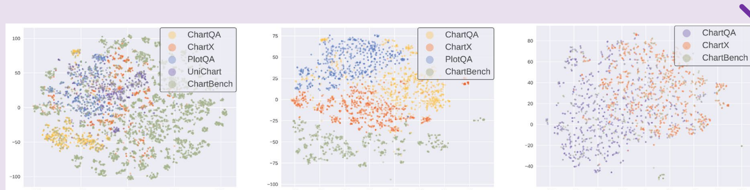


Question: In the t-SNE visualization of CLIP encoding features for Chart type data, what color represents the dataset that achieves the highest average score(75.76) among open-source MLLMs in the zero-shot performance comparison table?

A. Light green.
B. Yellow.
C. Orange.
D. Light blue.
E. Purple..

Evidence:

Models	ChartBench								ChartQA			
	Regular Type			Extra Type			Avg.	Rank	Human	Aug.	Avg.	Rank
	Acc+	NQA	Avg.	*Acc+*	NQA	Avg.						
Open source MLLMs												
VisualGLM [20]	3.46	1.83	3.13	4.22	4.84	4.35	3.68	#18	18.96	6.80	12.88	#12
Shikra [13]	8.59	2.35	7.34	7.50	9.05	7.81	7.55	#17	16.24	7.28	11.76	#15
OneChart [10]	12.34	2.26	10.33	8.75	3.37	7.68	9.12	#16	**85.30**	49.10	67.20	#5
InstructBLIP [17]	17.96	0.87	14.55	5.50	5.37	5.47	10.43	#15	15.92	7.92	11.92	#14
ChartVLM [74]	8.02	**43.74**	15.24	5.92	18.21	8.37	12.06	#14	42.08	82.48	62.28	#6
Internlm-XComposer [82]	19.70	1.22	16.01	10.11	5.79	9.25	12.94	#13	13.20	7.84	10.52	#16
CogVLM-Chat [70]	14.41	12.96	14.12	11.89	13.68	12.25	13.26	#12	34.24	28.56	31.40	#9
SPHINX [41]	17.87	6.17	15.54	17.92	12.74	16.89	16.13	#11	21.44	11.20	16.32	#11
BLIP2 [38]	21.65	0.96	17.53	18.44	4.84	15.74	16.70	#10	13.52	6.00	9.76	#17
CogAgent [27]	20.39	26.61	21.63	14.36	25.79	16.64	19.35	#9	54.08	80.56	67.32	#4
MiniGPT-v2 [12]	22.37	2.43	18.40	25.06	5.26	21.11	19.61	#8	15.60	8.48	12.04	#13
ChartLlama [26]	22.02	16.87	21.00	22.56	18.32	21.71	21.30	#7	58.40	**93.12**	**75.76**	#1
mPLUG-Owl-bloomz [78]	27.80	2.35	22.73	25.47	6.21	21.64	22.21	#6	7.84	4.88	6.36	#18
LLaVA-v1.5 [46]	25.61	8.09	22.12	27.39	15.26	24.97	23.39	#5	22.64	13.04	17.84	#10
Qwen-VL-Chat [4]	29.46	23.57	28.28	26.56	21.05	25.46	26.98	#4	42.48	75.20	58.84	#7
DocOwl-v1.5 [29]	35.27	37.30	35.67	26.86	29.47	27.38	31.89	#3	48.24	86.72	67.48	#3
Mini-Gemini [40]	39.57	25.57	36.78	31.81	25.79	30.61	33.96	#2	44.32	57.04	50.68	#8
Internlm-XComposer-v2 [19]	**57.89**	40.96	**54.52**	**41.75**	**31.58**	**39.73**	**47.78**	#1	63.12	81.92	72.64	#2
Closed source MLLMs												
ERNIE [5]	47.39	25.74	43.08	46.39	33.37	43.82	43.37	#3	-	-	-	
GPT-4V [54]	53.26	33.04	49.23	55.83	40.00	52.68	50.74	#2	-	-	78.50	#2
GPT-4O [54]	**65.00**	**40.00**	**60.02**	**63.33**	**41.05**	**58.89**	**59.45**	#1	-	-	**85.70**	#1



(a) Chart (b) CSV (c) Query

Ground Truth: B. Yellow.

Figure S.7. **The demo of figure-table comprehension.**

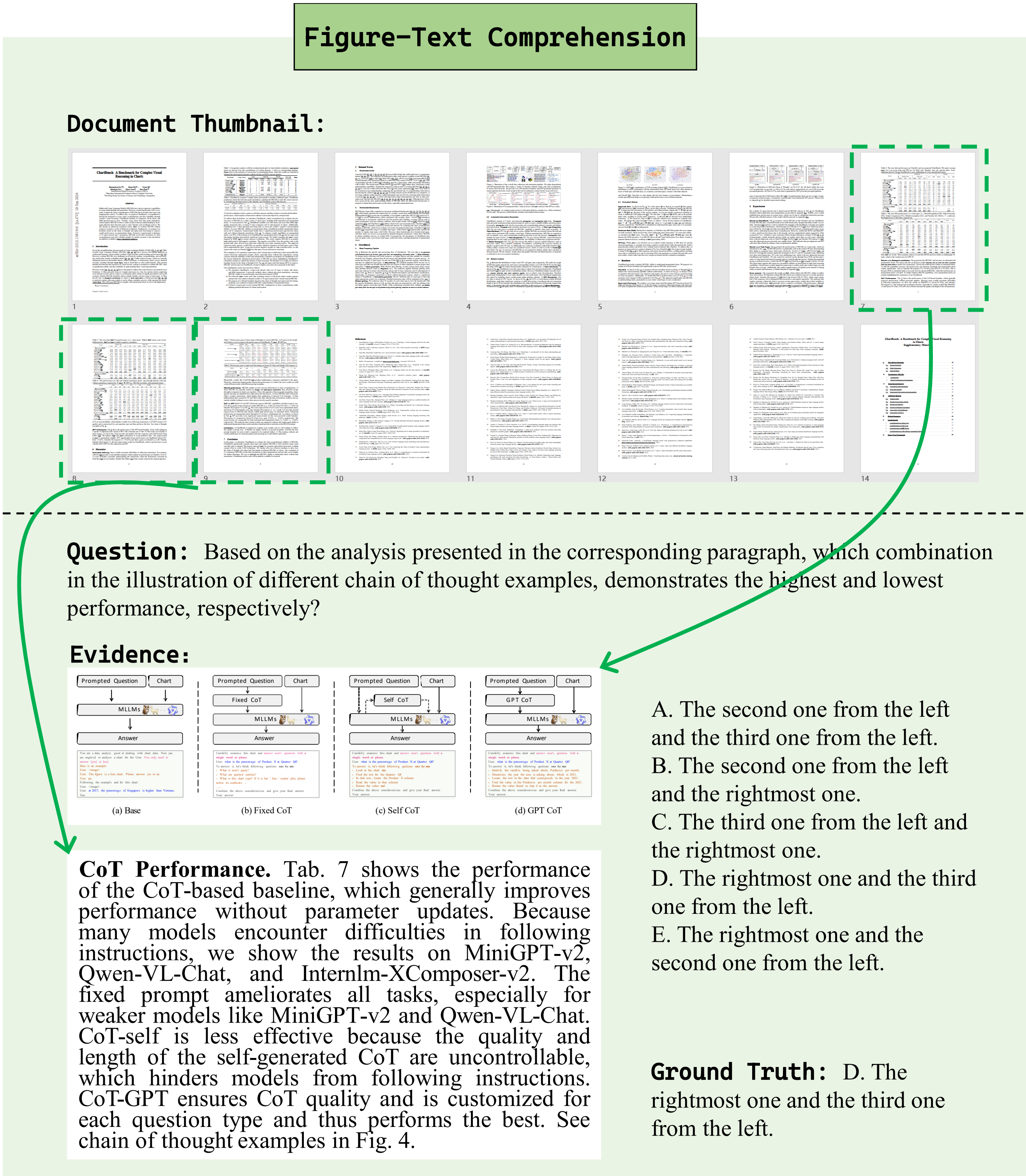
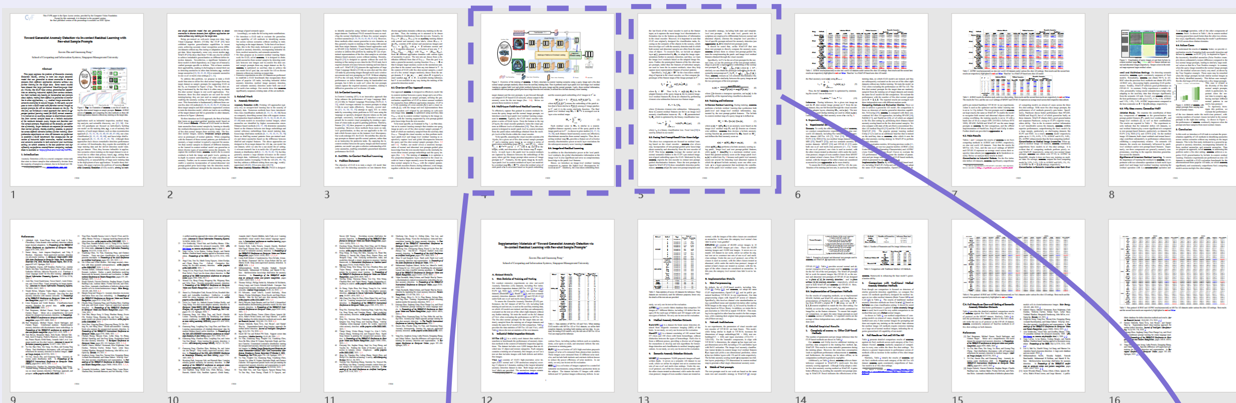


Figure S.8. **The demo of figure-text comprehension**.

Figure-Formula Comprehension

Document Thumbnail:

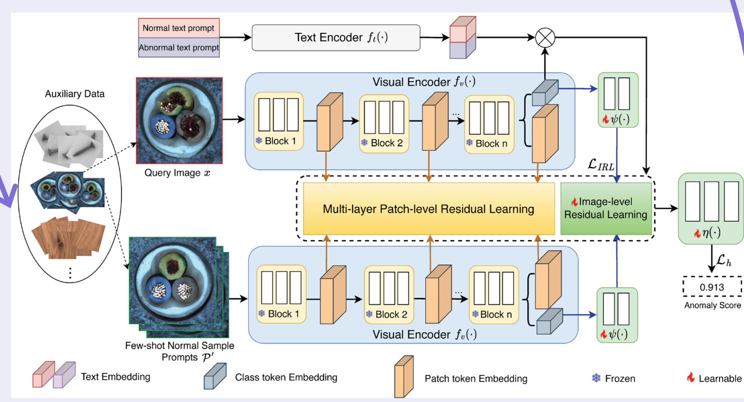


Question: Which colored rectangular volume in the InCTRL architecture are utilized to compute s_a(x)(i.e. the probability of the input x being classified as abnormal)?

A. Green and light orange.
B. Pale blue and light orange.
C. Light orange and yellow.
D. Pale blue and green.
E. Pale red and light purple.

Ground Truth: E. Pale red and light purple.

Evidence:

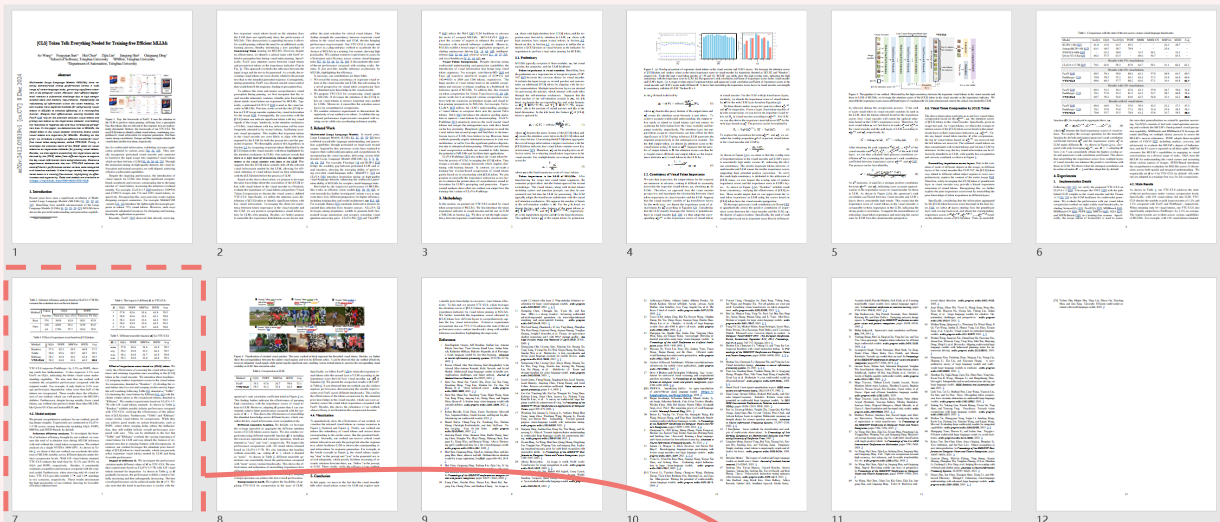


$$s_a(x) = \frac{\exp(F_a^T f_v(x))}{\exp\left(F_n^T f_v(x)\right) + \exp(F_a^T f_v(x))}$$

Figure S.9. **The demo of figure-formula comprehension**.

Table Comprehension

Document Thumbnail:



Question: Among the various ensemble functions adapted in the VTC-CLS method, which one exhibits the best performance on the GQA benchmark?

A. "Median."
B. "Max."
C. "Min."
D. "None."
E. "Avg."

Evidence:

Table 5. Different ensemble function $E(\cdot)$ in VTC-CLS.

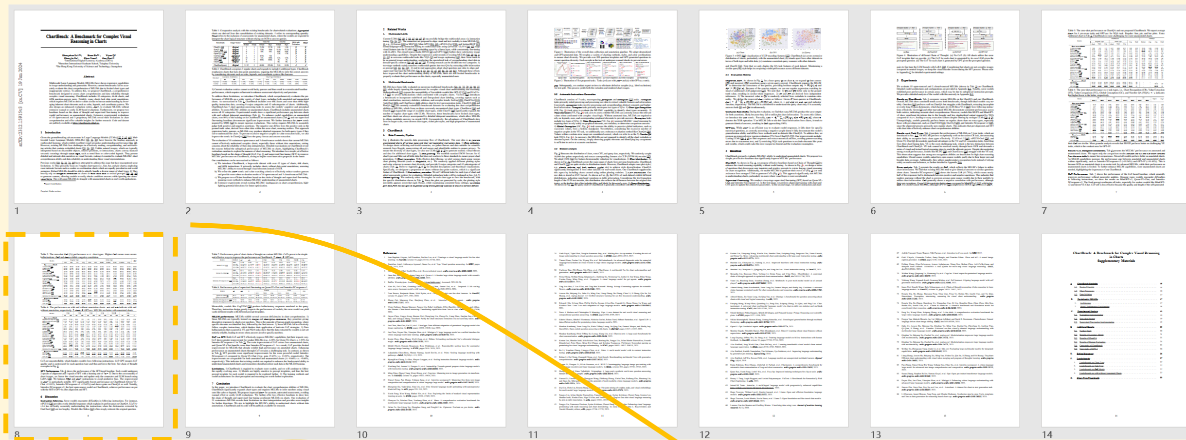
$E(\cdot)$	GQA	POPE	MMVet	SEED	Avg
none	57.9	82.6	31.6	61.8	58.5
avg	58.2	84.0	33.4	62.2	59.5
max	58.2	83.8	31.7	61.8	58.9
min	58.3	83.8	33.0	62.0	59.3

Ground Truth: C. "Min."

Figure S.10. **The demo of table comprehension**.

Multi-table Comprehension

Document Thumbnail:



Question: In Table 3, what is the average score rank on ChartBench for the open-source MLLM which demonstrates the lowest NQA score among its peers in Table 4?

A. #18.
B. #16.
C. #10.
D. #12.
E. #17.

Evidence:

Table 3: The zero-shot performance on ChartQA and our proposed ChartBench. We report average *Acc+* for 4 yes-or-no tasks and GPT-acc for NQA task. Regular: line, pie, and bar plots. Extra: additional chart in Tab. 2. ChartBench is more challenging for more unannotated charts.

Models	ChartBench								ChartQA			
	Regular Type			Extra Type			Avg.	Rank	Human	Aug.	Avg.	Rank
	Acc+	NQA	Avg.	*Acc+*	NQA	Avg.						
Open source MLLMs												
VisualGLM [20]	3.46	1.83	3.13	4.22	4.84	4.35	3.68	#18	18.96	6.80	12.88	#12
Shikra [13]	8.59	2.35	7.34	7.50	9.05	7.81	7.55	#17	16.24	7.28	11.76	#15
OneChart [10]	12.34	2.26	10.33	8.75	3.37	7.68	9.12	#16	**85.30**	49.10	67.20	#5
InstructBLIP [17]	17.96	0.87	14.55	5.50	5.37	5.47	10.43	#15	15.92	7.92	11.92	#14
ChartVLM [74]	8.02	**43.74**	15.24	5.92	18.21	8.37	12.06	#14	42.08	82.48	62.28	#6
Internlm-XComposer [82]	19.70	1.22	16.01	10.11	5.79	9.25	12.94	#13	13.20	7.84	10.52	#16
CogVLM-Chat [70]	14.41	12.96	14.12	11.89	13.68	12.25	13.26	#12	34.24	28.56	31.40	#9
SPHINX [41]	17.87	6.17	15.54	17.92	12.74	16.89	16.13	#11	21.44	11.20	16.32	#11
BLIP2 [38]	21.65	0.96	17.53	18.44	4.84	15.74	16.70	#10	13.52	6.00	9.76	#17
CogAgent [27]	20.39	26.61	21.63	14.36	25.79	16.64	19.35	#9	54.08	80.56	67.32	#4
MiniGPT-v2 [12]	22.37	2.43	18.40	25.06	5.26	21.11	19.61	#8	15.60	8.48	12.04	#13
ChartLlama [26]	22.02	16.87	21.00	22.56	18.32	21.71	21.30	#7	58.40	**93.12**	**75.76**	#1
mPLUG-Owl-bloomz [78]	27.80	2.35	22.73	25.47	6.21	21.64	22.21	#6	7.84	4.88	6.36	#18
LLaVA-v1.5 [46]	25.61	8.09	22.12	27.39	15.26	24.97	23.39	#5	22.64	13.04	17.84	#10
Qwen-VL-Chat [4]	29.46	23.57	28.28	26.56	21.05	25.46	26.98	#4	42.48	75.20	58.84	#7
DocOwl-v1.5 [29]	35.27	37.30	35.67	26.86	29.47	27.38	31.89	#3	48.24	86.72	67.48	#3
Mini-Gemini [40]	39.57	25.57	36.78	31.81	25.79	30.61	33.96	#2	44.32	57.04	50.68	#8
Internlm-XComposer-v2 [19]	**57.89**	40.96	**54.52**	**41.75**	**31.58**	**39.73**	**47.78**	#1	63.12	81.92	72.64	#2
Closed source MLLMs												
ERNIE [5]	47.39	25.74	43.08	46.39	33.37	43.82	43.37	#3	-	-	-	
GPT-4V [54]	53.26	33.04	49.23	55.83	40.00	52.68	50.74	#2	-	-	78.50	#2
GPT-4O [54]	**65.00**	**40.00**	**60.02**	**63.33**	**41.05**	**58.89**	**59.45**	#1	-	-	**85.70**	#1

Table 4: The zero-shot performance w.r.t. task types, i.e., Chart Recognition (CR), Value Extraction (VE), Value Comparison (VC), Global Conception (GC), and Number QA (NQA). ↑ / ↓ indicates that higher/lower is the better, respectively.

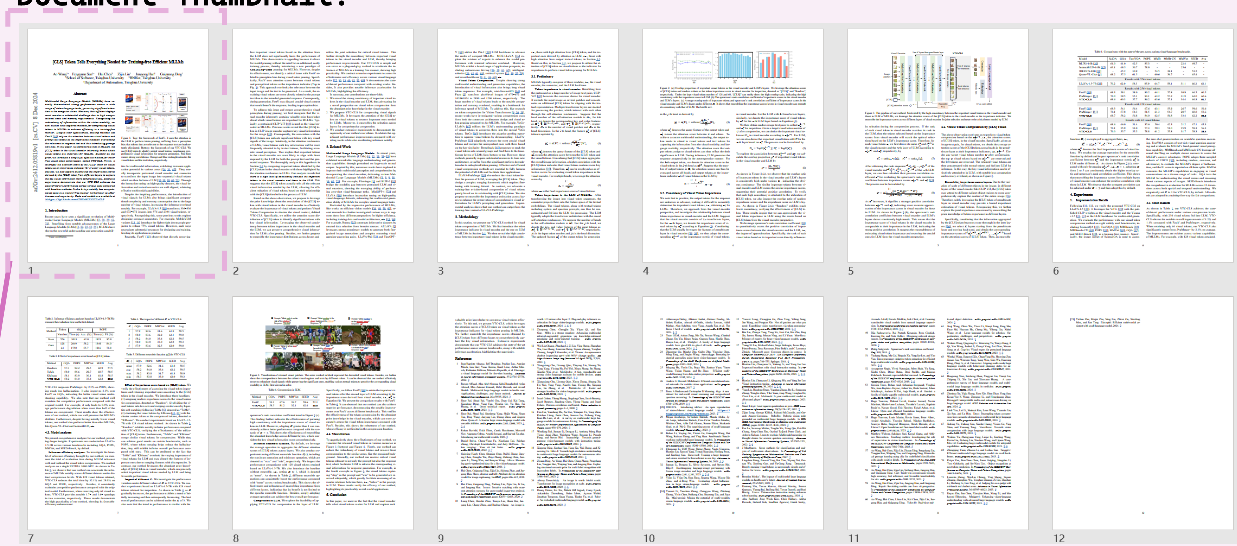
Models	CR		VE		VC		GC		NQA↑	Avg.↑
	Acc+ ↑	*CoR*↓	*Acc+*↑	*CoR*↓	*Acc+*↑	*CoR*↓	*Acc+*↑	*CoR*↓		
Open source MLLMs										
VisualGLM [20]	16.29	79.19	0.00	99.67	0.00	99.81	0.00	99.71	3.19	3.68
Shikra [13]	2.10	93.57	11.90	80.71	10.62	87.71	7.86	82.71	5.38	7.55
OneChart [10]	3.71	94.33	15.48	82.14	17.57	73.71	11.38	85.67	2.76	9.12
InstructBLIP [17]	49.57	36.67	0.00	100.00	0.05	99.81	0.00	99.90	2.90	10.43
ChartVLM [74]	0.00	100.00	9.05	85.48	10.05	83.81	8.52	86.19	32.19	12.06
Internlm-XComposer [82]	42.29	56.95	6.86	85.14	2.48	96.57	9.67	78.48	3.29	12.94
CogVLM-Chat [70]	29.14	69.33	2.81	94.29	14.19	80.71	7.33	90.14	13.29	13.26
SPHINX [41]	38.48	51.38	10.38	80.67	14.33	77.38	9.62	80.90	9.14	16.13
BLIP2 [38]	60.05	37.05	4.24	89.29	14.05	78.86	3.86	90.00	2.71	16.70
MiniGPT-v2 [12]	29.05	49.24	22.00	**55.14**	24.29	53.33	18.10	61.76	3.71	19.35
CogAgent [27]	62.57	37.10	1.19	94.90	7.33	88.24	1.19	94.76	26.24	19.61
ChartLlama [26]	49.86	44.19	8.38	84.14	20.43	69.48	10.67	83.81	17.52	21.30
mPLUG-Owl-bloomz [78]	32.33	51.24	23.14	76.76	25.33	69.29	26.48	71.00	4.10	22.21
LLaVA-v1.5 [46]	47.86	36.24	15.81	66.24	26.05	56.48	16.52	66.57	11.33	23.39
Qwen-VL-Chat [4]	51.67	42.71	11.14	84.57	27.29	63.14	21.71	74.86	22.43	26.98
DocOwl-v1.5 [29]	30.43	65.05	34.48	58.24	31.10	55.19	30.48	63.19	33.76	31.89
Mini-Gemini [40]	**80.52**	**17.86**	17.62	70.43	26.00	59.38	22.00	71.10	25.67	33.96
Internlm-XComposer-v2 [19]	68.29	30.24	**36.63**	57.71	**54.63**	**27.71**	**45.80**	**51.46**	**36.71**	**47.78**
Closed source MLLMs										
ERNIE [5]	65.24	19.52	**44.76**	**44.76**	32.86	41.43	47.14	47.62	29.24	43.37
GPT-4V [54]	96.19	2.86	30.95	63.33	48.57	34.76	46.19	47.62	36.19	50.74
GPT-4O [54]	**97.62**	**1.43**	43.33	**44.76**	**66.19**	**16.19**	**53.33**	**41.43**	**40.48**	**59.45**

Ground Truth: C. #10.

Figure S.11. **The demo of multi-table comprehension**.

Text Comprehension

Document Thumbnail:



Evidence:

Despite the inspiring performance, the introduction of visual signals for LLMs also brings significant computational complexity and memory consumption due to the large number of visual tokens, increasing the inference overhead notably. For example, LLaVA-1.5 [38] transforms 336×336 and 672×672 images into 576 and 2304 visual tokens, respectively. Recognizing this, some previous works explore designing compact connectors. For example, MobileVLM variants [13, 14] introduce the lightweight downsample projector to reduce 75% visual tokens. However, such ways necessitate substantial resources for designing and training, limiting its application in practice.

Question: Based on the discussion in the paper, which of the following accurately represents the limitations of MobileVLM variants?

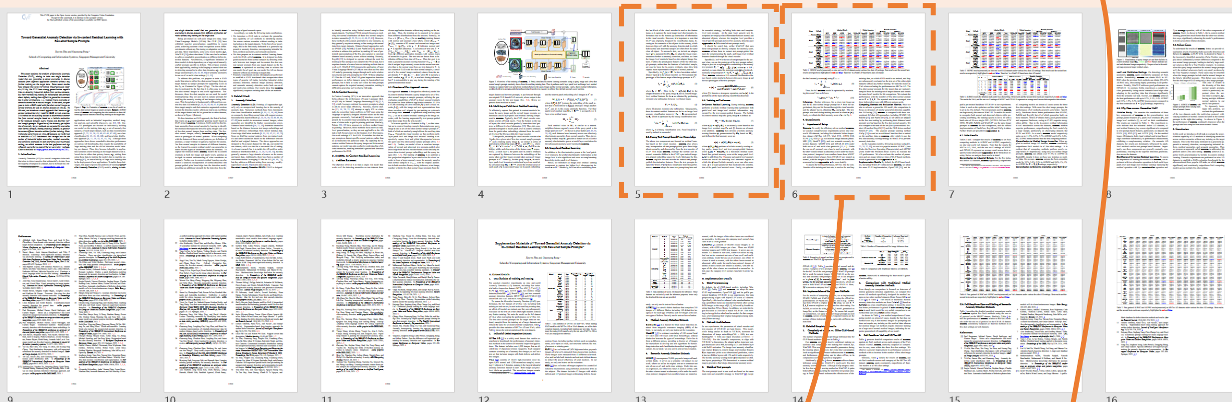
A. Requiring substantial resources for designing and training.
B. Directly removing less important visual tokens based on the attention from the LLM.
C. Overlooking the relevance between the input image and the text to be generated.
D. Discarding crucial visual context that would benefit the response.
E. Demonstrating limited adaptability to diverse visual input formats and resolutions.

Ground Truth: A. Requiring substantial resources for designing and training.

Figure S.12. **The demo of text comprehension**.

Formula Comprehension

Document Thumbnail:



Question: In the inference process, which numbered equation describes how the final anomaly score is computed for a given test image x_t?

A. Equation (10).
B. Equation (9).
C. Equation (8).
D. Equation (7).
E. Equation (6).

Evidence:

the final anomaly score $s(x)$ using X_{train}:

$$\mathcal{L}_h = \frac{1}{N} \sum_{x \in X_{train}} \mathcal{L}_b(s(x), y_x). \qquad (9)$$

Thus, the full InCTRL model is optimized by minimizing the overall loss as follows:

$$\mathcal{L}_{InCTRL} = \mathcal{L}_{IRL} + \mathcal{L}_h. \qquad (10)$$

Inference. During inference, for a given test image x_t and the K-shot normal image prompt set $\mathcal{P}$ from the target dataset, they are fed forward through the visual encoder and the adapter layers, obtaining $\mathbf{M}_{x_t}$ and $s_i(x_t)$. The text prompt sets used during training are used to obtain $s_a(x_t)$. Lastly, we obtain the final anomaly score of x_t via Eq. 8.

$$s_a(x) = \frac{\exp(\mathbf{F}_a^\intercal f_v(x))}{\exp(\mathbf{F}_n^\intercal f_v(x)) + \exp(\mathbf{F}_a^\intercal f_v(x))}, \qquad (6)$$

where $[\cdot]^\intercal$ denotes a transpose operation, and $s_a(x)$ is the probability of the input x being classified as abnormal.

3.6. Training and Inference

In-Context Residual Learning. During training, InCTRL performs a holistic residual learning that synthesizes both patch-level and image-level residual information, augmented by the text prompt-guided features. The holistic in-context residual map of a query image x is defined as:

$$\mathbf{M}_x^+ = \mathbf{M}_x \oplus s_i(x) \oplus s_a(x), \qquad (7)$$

where $s_i(x) = \eta(\mathbf{F}_x; \Theta_\eta)$ is an anomaly score based on the image-level residual map $\mathbf{F}_x$ and $\oplus$ denotes an element-wise addition. InCTRL then devises a holistic anomaly scoring function ϕ, parameterized by Θ_ϕ, based on $\mathbf{M}_x^+$, and defines the final anomaly score as:

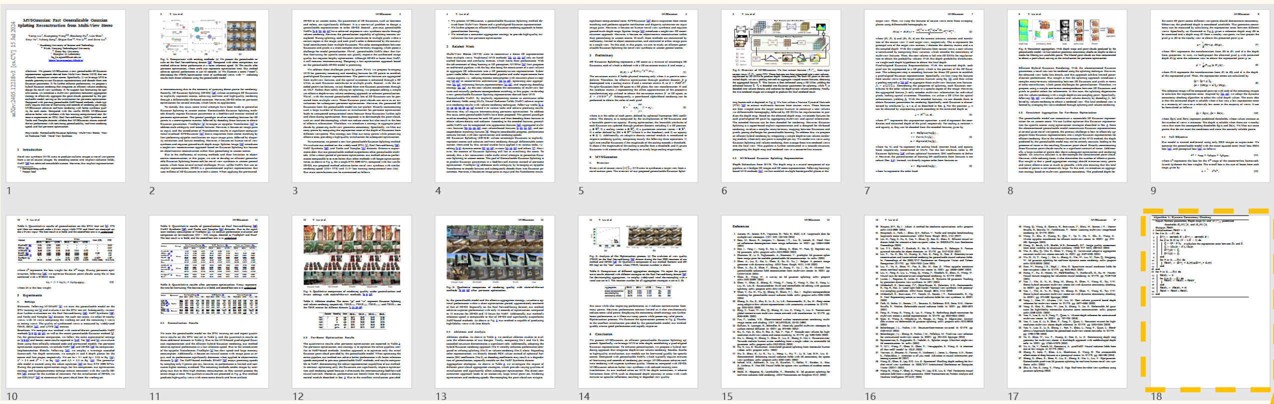
$$s(x) = \phi(\mathbf{M}_x^+; \Theta_\phi) + \alpha s_p(x), \qquad (8)$$

Ground Truth: C. Equation (8).

Figure S.13. **The demo of formula comprehension**.

Pseudocode Comprehension

Document Thumbnail:



Question: How many steps are there in the Dynamic Consistency Checking algorithm?

A. 21.
B. 18.
C. 24.
D. 16.
E. 20.

Ground Truth: A. 21.

Evidence:

Algorithm 1: Dynamic Consistency Checking

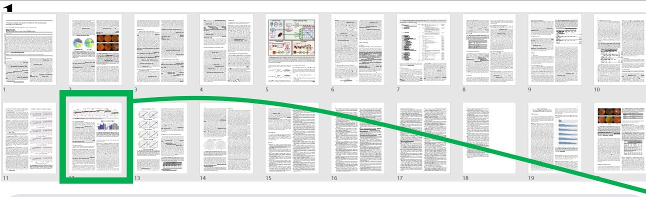
Input: Camera parameters, Depth maps D_0 and $\{D_i\}_{i=1}^{N}$, predefined thresholds $\{\theta_p(n)\}_{n=1}^{N_\theta}$ and $\{\theta_d(n)\}_{n=1}^{N_\theta}$

Output: $Mask$

```
1  Initialization: Mask ← 0
2  for i in (1, ..., N) do
3      Err_p^i ← zeros(H, W), Err_d^i ← zeros(H, W)
4      for p in (0, 0) to (H − 1, W − 1) do
5          ξ_p^i ← ||p − p'||_2,   ▷ calculate the reprojetcion error between D_0 and D_i
6          ξ_d^i ← ||D_0(p) − d'||_1 / D_0(p)
7          Err_p^i(p) ← ξ_p^i
8          Err_d^i(p) ← ξ_d^i
9      end
10     for n in (1, ..., N_θ) do
11         Mask_n^i ← (Err_p^i < θ_p(n)) & (Err_d^i < θ_d(n))
12     end
13 end
14 for n in (1, ..., N_θ) do
15     Mask_n ← 0
16     for i in (1, ..., N) do
17         Mask_n ← Mask_n + Mask_n^i
18     end
19     Mask_n ← (Mask_n > n)
20     Mask ← Mask ∪ Mask_n
21 end
```

Figure S.14. **The demo of pseudocode comprehension.**

I. Hallucinated Evidence: Case



12

Fig. 5: **Vision-language few-shot Adapters.** The results of different Adapters in the few-shot setting. The metric presented is the average accuracy, averaged across 5 cross-validation folds. ZS: zero-shot (*i.e.*, prompt-based classification with domain-knowledge prompts).

5.2.2. Vision-language Adapters

Recent emergent literature in computer vision has investigated strategies, often referred to as Adapters, to fine-tune vision-language models in low-data (few-shot) regimes for the target tasks, e.g., Clip-Adapter (Gao et al., 2023) and Tip-Adapter (Zhang et al., 2022a). These strategies typically integrate the knowledge driven from the pre-trained language encoder along with the vision features and use additional layers in the networks. Still, the utility of these adapters remains largely unexplored in the medical domain. Figure 5 depicts the results obtained by different vision-language Adapters using our pre-trained FLAIR foundation model and expert-knowledge prompts, across the different tasks. The results point to the powerful capabilities of zero-shot classification in different scenarios. In most of the cases, zero-shot inference, enhanced with domain-expert knowledge prompts, outperforms adaptation using $k \leq 5$ shots (see Figure 5 MESSIDOR, FIVES, REFUGE, 20x3). As for the vision-language Adapters (Zhang et al., 2022a; Gao et al., 2023), these do not seem to provide consistent improvements, neither over zero-shot classification (when $k \leq 5$) nor over basic Linear Probing (when $k = 10$).

5.2.3. Ablation experiments

In this section, we present ablation experiments that motivate different decisions in the design of the proposed framework.

What features to use for knowledge transfer. Vision-language pre-training models align the image-encoder features, $\vartheta_f(\cdot)$, to the text representations via a projection, $\vartheta_p(\cdot)$, along with a mapping to the unit hyper-sphere using an l2 normalization. Regarding the transferability of the pre-trained visual features to downstream domains and tasks via linear probing (LP), the standard feature-representation choice in prior literature is often based on both projection and normalization (Radford et al., 2021; Gao et al., 2023; Zhang et al., 2022a). In the following ablation experiment, we evaluate the feature transferability for the different evaluation datasets using the following three options: vision, projected, and projected-and-normalized features. We evaluated the three options under both the low and large-data regimes, using $k = 10$ and 80% of the dataset for training.

Figure 6 depicts the results, which show performance improvements across most of the tasks when using visual representation $\vartheta_f(\cdot)$ for transferability, in comparison to using projected features $\vartheta_p(\cdot)$ or projected-and-normalized features $\vartheta_p(\cdot)$ + norm. Motivated by these observations, we selected

Fig. 6: **Study of the transferred features for adaptation.** Evaluation of the performance of the linear-probe transferability of the features extracted from the vision encoder, $\vartheta_f(\cdot)$, the inter-modality projection head, $\vartheta_p(\cdot)$, and its hypersphere normalization, $\vartheta_p(\cdot)$ + norm. The metric presented is the average accuracy, averaged across 5 cross-validation folds. The results are presented for the low-data (10 shots) and large-data (80% of the whole dataset) regimes.

original feature representation $\vartheta_f(\cdot)$ for the transferability experiments in this work.

Generalization of linear-probe adaptation under domain shifts. The *pre-train-and-adapt* strategy using image-language models and computationally efficient linear-probe adaptation has shown promising performances on downstream computer-vision tasks. In the following, we aim at conducting a more comprehensive evaluation of this linear-probe strategy, in order to assess the capacity of the adaption stage in response to new changes in a target domain (*i.e.*, there are domain shifts after adaptation). To do so, we employ the supplementary evaluation subsets; see Table S1. In particular, we evaluate the performance of the linear probe, which has been fine-tuned on a source domain, in a novel target domain. More concretely, the adaptation is performed as follows using two datasets A and B: The model is fine-tuned on A and tested on B, and vice-versa. Again, two feature representations are evaluated for transferability: features extracted from the vision encoder, $\vartheta_f(\cdot)$, and features based on the inter-modality projection head, $\vartheta_p(\cdot)$. We juxtapose the performance of the linear probe to fine-tuning all the model trainable parameters on the source data (*i.e.*, using a standard supervised-learning setting, but with parameter initialization using either FLAIR or Imagenet model), as well as to the zero-shot performance. The experiments are carried out in the large-data regime, to evaluate the best-case scenario, in which the available data is not a limiting factor.

Figure 7 depicts the results from these experiments, which point to the following takeaways: The Fine-tuned, dataset-

Question: What observation can be made from the vision-language few-shot Adapters comparison across the five datasets presented?

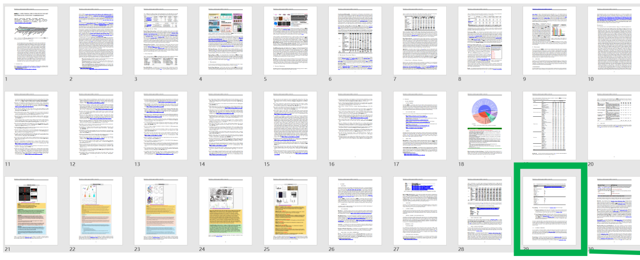
A. Vision-language adapters consistently outperform both basic linear probing and zero-shot classification across all five datasets.
B. Vision-language adapters outperform basic linear probing in some, but not all, of the five datasets, while consistently improving upon zero-shot classification.
C. Vision-language adapters do not demonstrate consistent improvements over either zero-shot classification or basic linear probing across the datasets.
D. Vision-language adapters show inconsistent improvements over zero-shot classification, but generally outperform basic linear probing across the datasets.
E. Vision-language adapters show consistent improvements over zero-shot classification, but only marginally outperform basic linear probing across the datasets.
Ground Truth: C.

Response: The correct answer is: B. Vision-language adapters outperform basic linear probing in some, but not all, of the five datasets, while consistently improving upon zero-shot classification. ✗
This observation aligns with the detailed analysis of the results presented in the study, where vision-language adapters provide benefits over zero-shot classification across the datasets but do not always show better performance than basic linear probing in every dataset.

Error Type: Hallucinated Evidence
Analysis: The model fabricates unsupported conclusions without proper evidence from the source materials. The model incorrectly synthesizes performance claims about vision-language adapters that directly contradict the actual evidence presented in both the text and figures. The source materials explicitly state that vision-language adapters show no consistent improvements over either zero-shot classification (k≤5) or basic Linear Probing (k=10). Instead of accurately interpreting the available data, the model **generated fictional performance patterns**, demonstrating a clear case of evidence hallucination.

Figure S.15. Illustration of a Hallucinated Evidence Error Case. The figure demonstrates how the model fabricates unsupported conclusions about vision-language adapter performance, contradicting the evidence highlighted in green from the source materials.

II. Incomplete Evidence: Case 1



Question: What observation can be made from the vision-language few-shot Adapters comparison across the five datasets presented?

A. 2; 36 hours..
B. 3; 40 hours.
C. 2; 40 hours.
D. 3; 36 hours.
E. 3; 45 hours
Ground Truth: C.

Response: The correct answer is: A. 2; 36 hours. ✗

Error Type: Incomplete Evidence
Analysis: The model's response demonstrates **selective attention to the available evidence**. While accurately identifying the two-stage nature of the visual pre-training process from the text "The visual pre-training process involves two stages," in the paper, it **only captured partial evidence information for the second question**. Specifically, the model focused on "the second stage took around 36 hours" **while overlooking the crucial detail that "the first stage took approximately 4 hours."** A complete consideration of both time components would yield a total of 40 hours (4 + 36), making C the correct answer. This partial evidence consideration led to an inaccurate conclusion.

Published as a conference paper at AI4MAT-Vienna 2024

Table 11: Hyperparameters for visual language pre-training on interleaved text and image data.

Hyperparameter	Values
base model	https://huggingface.co/meta-llama/Llama-2-7b-hfb
vision encoder	https://huggingface.co/openai/clip-vit-large-patch14-336
projector	2-layer MLP
Stage 1: Projector Initialization	
epochs	1
global batch size	256
learning rate	0.001
learning rate scheduler	cosine
weight decay	0.0
warmup ratio	0.03
max length	4096
tune LLM	✗
tune vision encoder	✗
tune projector	✓
Stage 2: Visual Language Pre-training	
epochs	1
global batch size	128
learning rate	0.00005
learning rate scheduler	cosine
weight decay	0.0
warmup ratio	0.03
max length	4096
tune LLM	✓
tune vision encoder	✗
tune projector	✓

Training Stages The visual pre-training process (Lin et al., 2023) involves two stages:

1. **Projection initialization**: In this stage, the LLM and the visual encoder are both pre-trained and remain fixed. The projector, however, is randomly initialized. Only the projector is fine-tuned during this stage, using image-caption pairs from (Liu et al., 2024).

2. **Visual language pre-training**: During this stage, both the LLM and the projector are fine-tuned on the interleaved image and text data. This includes data from general domains provided by MMC4 (Zhu et al., 2024), as well as scientific articles and figures from our dataset MMSci. Previous research (Lin et al., 2023) has shown that tuning both the LLM and the projector yields better results than tuning only one of them. Throughout this stage, the visual encoder remains fixed.

We did not conduct the further visual instruction-tuning for this model, as our primary objective was to infuse scientific knowledge into the LLM for the consecutive text-only material generation task. The two stages were conducted on a computing cluster equipped with eight NVIDIA A100 GPUs, each with 40GB of memory. The first stage took approximately 4 hours, and the second stage took around 36 hours.

A.3.5 Materials Generation

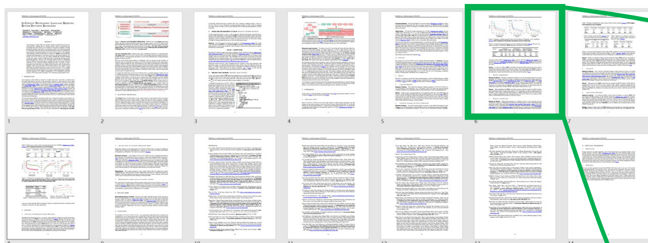
As a case study to investigate whether scientific knowledge has been effectively infused into the LLM (LLaMA2-7B in our experiments) and whether it can enhance performance on material science-related tasks, we follow the methodology from Gruver et al. (2024) to explore the material generation task. The primary objective is to format material crystal structures into text strings and fine-tuning the LLM to generate stable materials.

Prompt design We adhere to the prompt design described in (Gruver et al., 2024). There are two types of prompts in the training data: the generation prompt with one or multiple conditions and infilling prompts, where partial crystal structure strings are masked and the model generates the masked parts. The specific prompt templates are shown below, adapted from (Gruver et al., 2024).

29

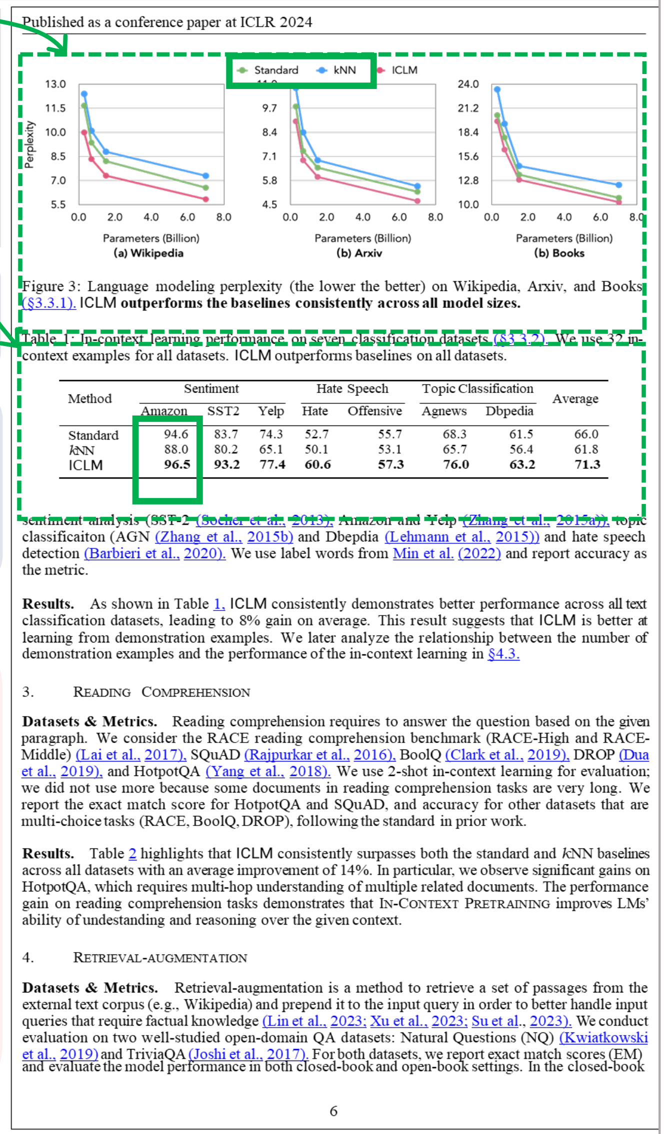
Figure S.16. Illustration of an Incomplete Evidence Error Case. The model correctly identifies the two-stage nature of visual pre-training but overlooks crucial timing information, considering only the second stage duration (36 hours) while missing the first stage duration (4 hours).

III. Perceptual Error: Case 1



Question: Referring to the color scheme in Figure 3's legend, what are the respective Sentiment-Amazon scores in Table 1 for the methods represented by blue, green and pink?

A. 80.2, 83.7, 93.2.
B. 94.6, 88.0, 96.5.
C. 61.8, 66.0, 71.3.
D. 88.0, 94.6, 96.5.
E. 96.5, 94.6, 88.0.
Ground Truth: D.



Published as a conference paper at ICLR 2024

Figure 3: Language modeling perplexity (the lower the better) on Wikipedia, Arxiv, and Books (§3.3.1). ICLM **outperforms the baselines consistently across all model sizes.**

Table 1: In-context learning performance on seven classification datasets (§3.3.2). We use 32 in-context examples for all datasets. ICLM outperforms baselines on all datasets.

Method	Sentiment			Hate Speech		Topic Classification		Average
	Amazon	SST2	Yelp	Hate	Offensive	Agnews	Dbpedia	
Standard	94.6	83.7	74.3	52.7	55.7	68.3	61.5	66.0
*k*NN	88.0	80.2	65.1	50.1	53.1	65.7	56.4	61.8
ICLM	**96.5**	**93.2**	**77.4**	**60.6**	**57.3**	**76.0**	**63.2**	**71.3**

sentiment analysis (SST-2 (Socher et al., 2013), Amazon and Yelp (Zhang et al., 2015a)), topic classificaiton (AGN (Zhang et al., 2015b) and Dbepdia (Lehmann et al., 2015)) and hate speech detection (Barbieri et al., 2020). We use label words from Min et al. (2022) and report accuracy as the metric.

Results. As shown in Table 1, ICLM consistently demonstrates better performance across all text classification datasets, leading to 8% gain on average. This result suggests that ICLM is better at learning from demonstration examples. We later analyze the relationship between the number of demonstration examples and the performance of the in-context learning in §4.3.

3. Reading Comprehension

Datasets & Metrics. Reading comprehension requires to answer the question based on the given paragraph. We consider the RACE reading comprehension benchmark (RACE-High and RACE-Middle) (Lai et al., 2017), SQuAD (Rajpurkar et al., 2016), BoolQ (Clark et al., 2019), DROP (Dua et al., 2019), and HotpotQA (Yang et al., 2018). We use 2-shot in-context learning for evaluation; we did not use more because some documents in reading comprehension tasks are very long. We report the exact match score for HotpotQA and SQuAD, and accuracy for other datasets that are multi-choice tasks (RACE, BoolQ, DROP), following the standard in prior work.

Results. Table 2 highlights that ICLM consistently surpasses both the standard and *k*NN baselines across all datasets with an average improvement of 14%. In particular, we observe significant gains on HotpotQA, which requires multi-hop understanding of multiple related documents. The performance gain on reading comprehension tasks demonstrates that In-Context Pretraining improves LMs' ability of undestanding and reasoning over the given context.

4. Retrieval-augmentation

Datasets & Metrics. Retrieval-augmentation is a method to retrieve a set of passages from the external text corpus (e.g., Wikipedia) and prepend it to the input query in order to better handle input queries that require factual knowledge (Lin et al., 2023; Xu et al., 2023; Su et al., 2023). We conduct evaluation on two well-studied open-domain QA datasets: Natural Questions (NQ) (Kwiatkowski et al., 2019) and TriviaQA (Joshi et al., 2017). For both datasets, we report exact match scores (EM) and evaluate the model performance in both closed-book and open-book settings. In the closed-book

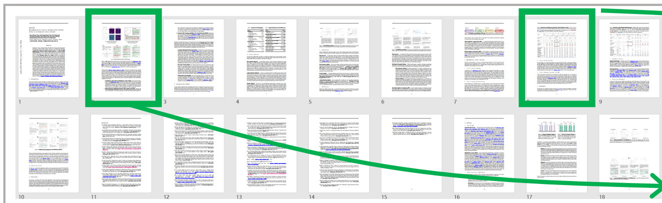
6

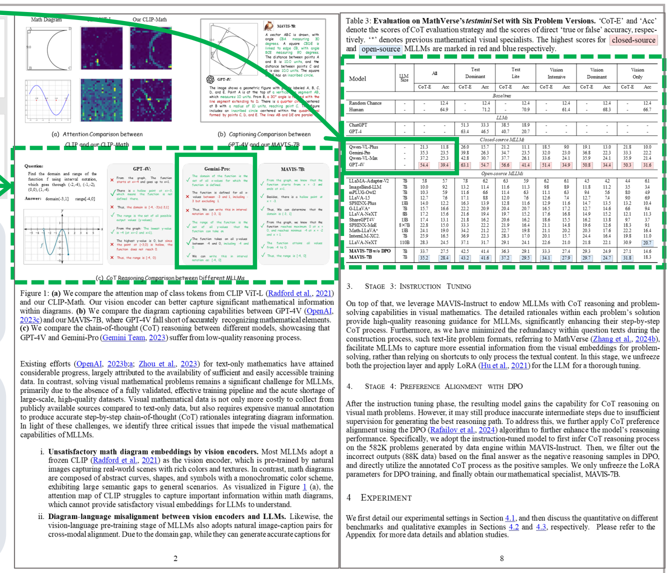
Response: To answer this question, we align the methods represented by their respective colors in Figure 3's legend with the Sentiment-Amazon scores in Table 1. From Figure 3:
- Blue: Standard method
- Green: kNN method
- Pink: ICLM method
From Table 1 (Sentiment-Amazon column):
- Standard method = 94.6
- kNN method = 88.0
- ICLM method = 96.5
Thus, the respective scores for blue, green, and pink are **94.6, 88.0, and 96.5**.
The correct answer is: B. 94.6, 88.0, 96.5. ✗

Error Type: Perceptual Error
Analysis: This response demonstrates a critical perceptual error in color-method association. While the model correctly identified the numerical scores from Table 1 for each method, it **incorrectly mapped the colors to their corresponding methods in Figure 3's legend**. Specifically, it **reversed the color assignments for the Standard method (actually shown in green, not blue) and the KNN method (actually shown in blue, not green).** This error highlights a fundamental limitation of current LVLMs in visual-textual alignment tasks, particularly in color-coding interpretation.

Figure S.17. Illustration of a Perceptual Error Case. The figure demonstrates how the model correctly identifies numerical scores but incorrectly maps color codes to methods, mismatching the blue and green representations in the legend despite clear visual evidence.

IV. Extractor Error: Case 1





Question: What is the accuracy score under the 'All' setting for the MLLM that occupies the middle position in the CoT reasoning comparison between different MLLMs in Figure 1, as reported in Table 3?

A. 39.4.
B. 23.5.
C. 35.3.
D. 54,4.
E. 28.4.
Ground Truth: B.

Response: The accuracy score under the 'All' setting for the MLLM that occupies the middle position in the CoT reasoning comparison in Figure 1 is:
C. 35.3. ✗

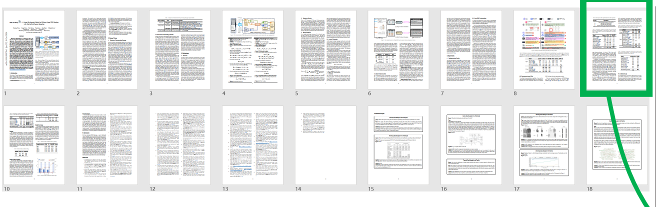
This corresponds to Gemini-Pro as reported in Table 3.

Error Type: **Extractor Error**

Analysis: The model's interpretation of Table 3 reveals a critical confusion between different metric columns. While correctly identifying the target model using spatial information from Figure 1, the model incorrectly extracted the CoT evaluation score (35.3) instead of the required accuracy score (23.5) under the 'All' setting. This error demonstrates the model's inability to distinguish between different evaluation metrics and accurately align the requested information from tabular data.

Figure S.18. Illustration of an Extractor Error Case. The figure demonstrates how the model confuses different metric columns in Table 3, extracting the CoT evaluation score (35.3) instead of the correct accuracy score (23.5) despite accurately identifying the target model from spatial information.

V. Reasoning Error: Case 1



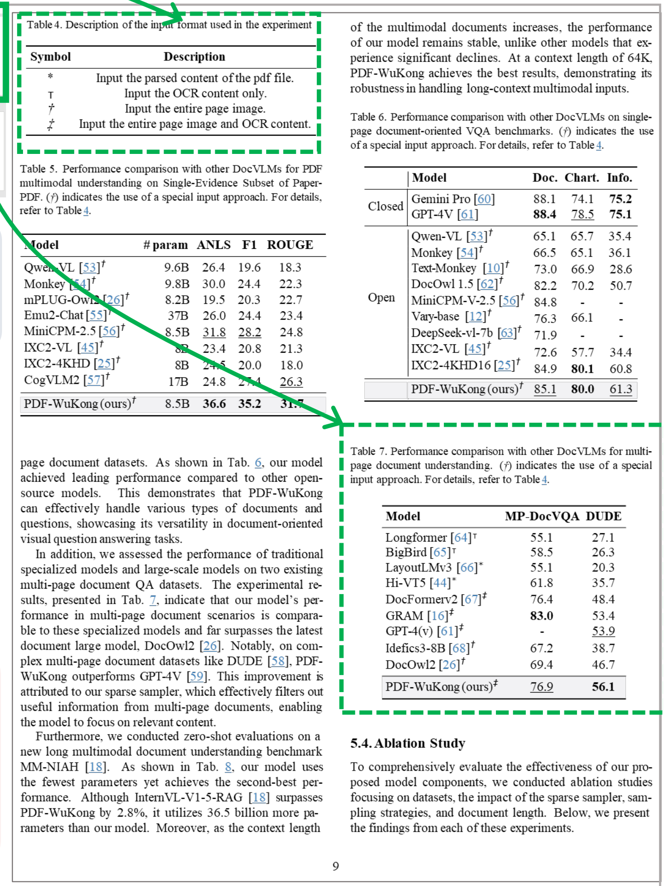


Table 4. Description of the input format used in the experiment

Symbol	Description
*	Input the parsed content of the pdf file.
$\top$	Input the OCR content only.
$\dagger$	Input the entire page image.
$\ddagger$	Input the entire page image and OCR content.

Table 5. Performance comparison with other DocVLMs for PDF multimodal understanding on Single-Evidence Subset of Paper-PDF. ($\dagger$) indicates the use of a special input approach. For details, refer to Table 4.

Model	# param	ANLS	F1	ROUGE
Qwen-VL [53]†	9.6B	26.4	19.6	18.3
Monkey [54]†	9.8B	30.0	24.4	22.3
mPLUG-Owl2 [26]†	8.2B	19.5	20.3	22.7
Emu2-Chat [55]†	37B	26.0	24.4	23.4
MiniCPM-2.5 [56]†	8.5B	31.8	28.2	24.8
IXC2-VL [45]†	8B	23.4	20.8	21.3
IXC2-4KHD [25]†	8B	24.5	20.0	18.0
CogVLM2 [57]†	17B	24.8	27.4	26.3
PDF-WuKong (ours)†	8.5B	**36.6**	**35.2**	**31.7**

page document datasets. As shown in Tab. 6, our model achieved leading performance compared to other open-source models. This demonstrates that PDF-WuKong can effectively handle various types of documents and questions, showcasing its versatility in document-oriented visual question answering tasks.

In addition, we assessed the performance of traditional specialized models and large-scale models on two existing multi-page document QA datasets. The experimental results, presented in Tab. 7, indicate that our model's performance in multi-page document scenarios is comparable to these specialized models and far surpasses the latest document large model, DocOwl2 [26]. Notably, on complex multi-page document datasets like DUDE [58], PDF-WuKong outperforms GPT-4V [59]. This improvement is attributed to our sparse sampler, which effectively filters out useful information from multi-page documents, enabling the model to focus on relevant content.

Furthermore, we conducted zero-shot evaluations on a new long multimodal document understanding benchmark MM-NIAH [18]. As shown in Tab. 8, our model uses the fewest parameters yet achieves the second-best performance. Although InternVL-V1-5-RAG [18] surpasses PDF-WuKong by 2.8%, it utilizes 36.5 billion more parameters than our model. Moreover, as the context length of the multimodal documents increases, the performance of our model remains stable, unlike other models that experience significant declines. At a context length of 64K, PDF-WuKong achieves the best results, demonstrating its robustness in handling long-context multimodal inputs.

Table 6. Performance comparison with other DocVLMs on single-page document-oriented VQA benchmarks. ($\dagger$) indicates the use of a special input approach. For details, refer to Table 4.

	Model	Doc.	Chart.	Info.
Closed	Gemini Pro [60]	88.1	74.1	**75.2**
	GPT-4V [61]	**88.4**	78.5	**75.1**
Open	Qwen-VL [53]†	65.1	65.7	35.4
	Monkey [54]†	66.5	65.1	36.1
	Text-Monkey [10]†	73.0	66.9	28.6
	DocOwl 1.5 [62]†	82.2	70.2	50.7
	MiniCPM-V-2.5 [56]†	84.8	-	-
	Vary-base [12]†	76.3	66.1	-
	DeepSeek-vl-7b [63]†	71.9	-	-
	IXC2-VL [45]†	72.6	57.7	34.4
	IXC2-4KHD16 [25]†	84.9	**80.1**	60.8
	PDF-WuKong (ours)†	85.1	**80.0**	61.3

Table 7. Performance comparison with other DocVLMs for multi-page document understanding. ($\dagger$) indicates the use of a special input approach. For details, refer to Table 4.

Model	MP-DocVQA	DUDE
Longformer [64]$^{\top}$	55.1	27.1
BigBird [65]$^{\top}$	58.5	26.3
LayoutLMv3 [66]*	55.1	20.3
Hi-VT5 [44]*	61.8	35.7
DocFormerv2 [67]‡	76.4	48.4
GRAM [16]‡	**83.0**	53.4
GPT-4(v) [61]‡	-	53.9
Idefics3-8B [68]†	67.2	38.7
DocOwl2 [26]†	69.4	46.7
PDF-WuKong (ours)‡	76.9	**56.1**

5.4. Ablation Study

To comprehensively evaluate the effectiveness of our proposed model components, we conducted ablation studies focusing on datasets, the impact of the sparse sampler, sampling strategies, and document length. Below, we present the findings from each of these experiments.

9

Question: Which model has an input approach different from the model proposed in this paper in the multi-page document comprehension setting?

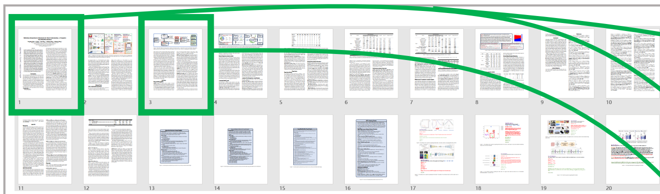
A. GRAM.
B. DocFormerv2.
C. PDF-WuKong.
D. Idefics3-8B.
E. GPT-4(v).
Ground Truth: D.

Response: The correct answer is: E. GPT-4(v). ❌
Reason: GPT-4(v) employs a vision-language model approach, which is inherently different in its input approach compared to PDF-WuKong's end-to-end sparse sampling framework for interleaved text and images in long multi-page document comprehension.

Error Type: Reasoning Error
Analysis: The superscripts of each model in Table7 show the input approach of the model proposed and the models in the options in the multi-page document comprehension setting. Analyzed together with the explanation of input approach in Table4, the input approach of option "D" is Input the entire page image, while the input approach of PDF-Wukong and the rest of the options is Input the entire page image and OCR content. The input approach of option "D" is Input the entire page image, while the input approach of PDF-Wukong and the rest of the options is Input the entire page image and OCR content. The model fails to correctly combine the context and diagrams in the text to comprehend the problem, and make the correct reasoning with the options.

Figure S.19. Illustration of a Reasoning Error Case. The figure demonstrates how the model fails to correctly interpret input approach differences between models despite clear evidence from Table 1 and Table 7, misidentifying GPT-4(v) as having a distinct approach from other models.

VI. Irrelevant Answer: Case 1



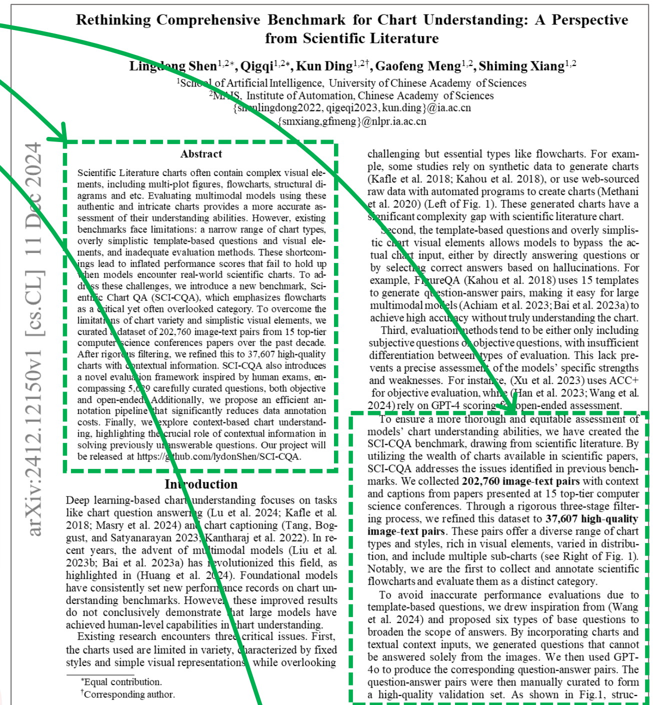
Question: What is the number of datasets that pass the refinement filter as a percentage of the total number of datasets collected?

A. 28.17%.
B. 16.25%.
C. 20.51%.
D. 18.54%.
E. 13.77%.
Ground Truth: D.

Response: To calculate the percent of the number of datasets that pass the refinement filter as a percentage of the total number of datasets collected:
The total number of datasets collected = 75,506
The number of datasets filtered = 54,557
Calculated, 54,557 divided by 75,506 equals 72.26%
Approximately 72.26 of the total data remained after refinement. Hence, when checking the options, none of them give the correct answer as listed values do not match refine computation exactly.

?

Error Type: Irrelevant Answer
Analysis: The model is required to answer the number of datasets that pass the refinement filter as a percentage of the total number of datasets collected. But the model finds irrelevant information, the number of papers that pass the data processing pipeline as a percentage of the percentage of the total number of articles collected initially, and did not calculate the number of required high-quality image-text pairs (37,607) as a percentage of the total image-text pairs (202,760). Misunderstood the meaning of dataset and found irrelevant information.



arXiv:2412.12150v1 [cs.CL] 11 Dec 2024

Rethinking Comprehensive Benchmark for Chart Understanding: A Perspective from Scientific Literature

Lingdong Shen[1,2*], Qigqi[1,2*], Kun Ding[1,2†], Gaofeng Meng[1,2], Shiming Xiang[1,2]
[1]School of Artificial Intelligence, University of Chinese Academy of Sciences
[2]MAIS, Institute of Automation, Chinese Academy of Sciences
{shenlingdong2022, qigeqi2023, kun.ding}@ia.ac.cn
{smxiang,gfmeng}@nlpr.ia.ac.cn

Abstract

Scientific Literature charts often contain complex visual elements, including multi-plot figures, flowcharts, structural diagrams and etc. Evaluating multimodal models using these authentic and intricate charts provides a more accurate assessment of their understanding abilities. However, existing benchmarks face limitations: a narrow range of chart types, overly simplistic template-based questions and visual elements, and inadequate evaluation methods. These shortcomings lead to inflated performance scores that fail to hold up when models encounter real-world scientific charts. To address these challenges, we introduce a new benchmark, Scientific Chart QA (SCI-CQA), which emphasizes flowcharts as a critical yet often overlooked category. To overcome the limitations of chart variety and simplistic visual elements, we curated a dataset of 202,760 image-text pairs from 15 top-tier computer science conferences papers over the past decade. After rigorous filtering, we refined this to 37,607 high-quality charts with contextual information. SCI-CQA also introduces a novel evaluation framework inspired by human exams, encompassing 5,629 carefully curated questions, both objective and open-ended. Additionally, we propose an efficient annotation pipeline that significantly reduces data annotation costs. Finally, we explore context-based chart understanding, highlighting the crucial role of contextual information in solving previously unanswerable questions. Our project will be released at https://github.com/lydonShen/SCI-CQA.

Introduction

Deep learning-based chart understanding focuses on tasks like chart question answering (Lu et al. 2024; Kafle et al. 2018; Masry et al. 2024) and chart captioning (Tang, Boggust, and Satyanarayan 2023; Kantharaj et al. 2022). In recent years, the advent of multimodal models (Liu et al. 2023b; Bai et al. 2023a) has revolutionized this field, as highlighted in (Huang et al. 2024). Foundational models have consistently set new performance records on chart understanding benchmarks. However, these improved results do not conclusively demonstrate that large models have achieved human-level capabilities in chart understanding.

Existing research encounters three critical issues. First, the charts used are limited in variety, characterized by fixed styles and simple visual representations, while overlooking challenging but essential types like flowcharts. For example, some studies rely on synthetic data to generate charts (Kafle et al. 2018; Kahou et al. 2018), or use web-sourced raw data with automated programs to create charts (Methani et al. 2020) (Left of Fig. 1). These generated charts have a significant complexity gap with scientific literature chart.

Second, the template-based questions and overly simplistic chart visual elements allows models to bypass the actual chart input, either by directly answering questions or by selecting correct answers based on hallucinations. For example, FigureQA (Kahou et al. 2018) uses 15 templates to generate question-answer pairs, making it easy for large multimodal models (Achiam et al. 2023; Bai et al. 2023a) to achieve high accuracy without truly understanding the chart.

Third, evaluation methods tend to be either only including subjective questions or objective questions, with insufficient differentiation between types of evaluation. This lack prevents a precise assessment of the models' specific strengths and weaknesses. For instance, (Xu et al. 2023) uses ACC+ for objective evaluation, while (Han et al. 2023; Wang et al. 2024) rely on GPT-4 scoring for open-ended assessment.

To ensure a more thorough and equitable assessment of models' chart understanding abilities, we have created the SCI-CQA benchmark, drawing from scientific literature. By utilizing the wealth of charts available in scientific papers, SCI-CQA addresses the issues identified in previous benchmarks. We collected **202,760 image-text pairs** with context and captions from papers presented at 15 top-tier computer science conferences. Through a rigorous three-stage filtering process, we refined this dataset to **37,607 high-quality image-text pairs**. These pairs offer a diverse range of chart types and styles, rich in visual elements, varied in distribution, and include multiple sub-charts (see Right of Fig. 1). Notably, we are the first to collect and annotate scientific flowcharts and evaluate them as a distinct category.

To avoid inaccurate performance evaluations due to template-based questions, we drew inspiration from (Wang et al. 2024) and proposed six types of base questions to broaden the scope of answers. By incorporating charts and textual context inputs, we generated questions that cannot be answered solely from the images. We then used GPT-4o to produce the corresponding question-answer pairs. The question-answer pairs were then manually curated to form a high-quality validation set. As shown in Fig.1, struc-

*Equal contribution.
†Corresponding author.

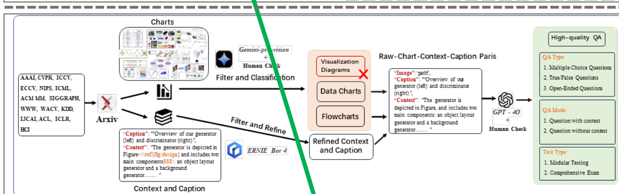


Figure 2: SCI-CQA data processing pipeline, multiple proprietary models is employed to monitor and enhance data quality, supplemented by rigorous manual verification at key stages, whichs led to the creation of a high-quality evaluation dataset.

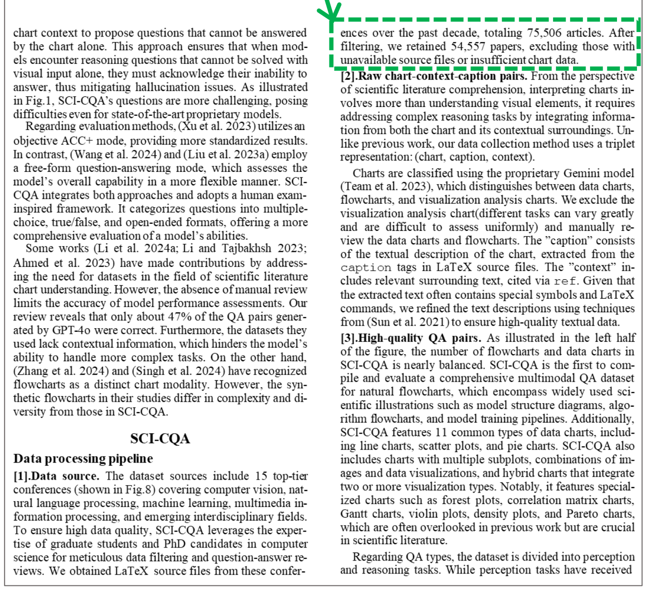


chart context to propose questions that cannot be answered by the chart alone. This approach ensures that when models encounter reasoning questions that cannot be solved with visual input alone, they must acknowledge their inability to answer, thus mitigating hallucination issues. As illustrated in Fig.1, SCI-CQA's questions are more challenging, posing difficulties even for state-of-the-art proprietary models.

Regarding evaluation methods, (Xu et al. 2023) utilizes an objective ACC+ mode, providing more standardized results. In contrast, (Wang et al. 2024) and (Liu et al. 2023a) employ a free-form question-answering mode, which assesses the model's overall capability in a more flexible manner. SCI-CQA integrates both approaches and adopts a human exam-inspired framework. It categorizes questions into multiple-choice, true/false, and open-ended formats, offering a more comprehensive evaluation of a model's abilities.

Some works (Li et al. 2024a; Li and Tajbakhsh 2023; Ahmed et al. 2023) have made contributions by addressing the need for datasets in the field of scientific literature chart understanding. However, the absence of manual review limits the accuracy of model performance assessments. Our review reveals that only about 47% of the QA pairs generated by GPT-4o were correct. Furthermore, the datasets they used lack contextual information, which hinders the model's ability to handle more complex tasks. On the other hand, (Zhang et al. 2024) and (Singh et al. 2024) have recognized flowcharts as a distinct chart modality. However, the synthetic flowcharts in their studies differ in complexity and diversity from those in SCI-CQA.

SCI-CQA

Data processing pipeline

[1].Data source. The dataset sources include 15 top-tier conferences (shown in Fig.8) covering computer vision, natural language processing, machine learning, multimedia information processing, and emerging interdisciplinary fields. To ensure high data quality, SCI-CQA leverages the expertise of graduate students and PhD candidates in computer science for meticulous data filtering and question-answer reviews. We obtained LaTeX source files from these conferences over the past decade, totaling 75,506 articles. After filtering, we retained 54,557 papers, excluding those with unavailable source files or insufficient chart data.

[2].Raw chart-context-caption pairs. From the perspective of scientific literature comprehension, interpreting charts involves more than understanding visual elements, it requires addressing complex reasoning tasks by integrating information from both the chart and its contextual surroundings. Unlike previous work, our data collection method uses a triplet representation: (chart, caption, context).

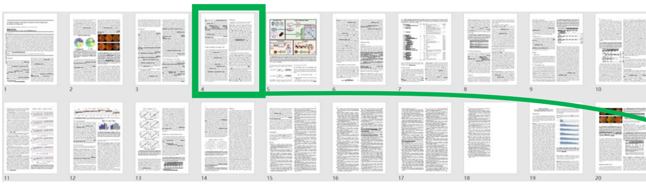
Charts are classified using the proprietary Gemini model (Team et al. 2023), which distinguishes between data charts, flowcharts, and visualization analysis charts. We exclude the visualization analysis chart(different tasks can vary greatly and are difficult to assess uniformly) and manually review the data charts and flowcharts. The "caption" consists of the textual description of the chart, extracted from the `caption` tags in LaTeX source files. The "context" includes relevant surrounding text, cited via `ref`. Given that the extracted text often contains special symbols and LaTeX commands, we refined the text descriptions using techniques from (Sun et al. 2021) to ensure high-quality textual data.

[3].High-quality QA pairs. As illustrated in the left half of the figure, the number of flowcharts and data charts in SCI-CQA is nearly balanced. SCI-CQA is the first to compile and evaluate a comprehensive multimodal QA dataset for natural flowcharts, which encompass widely used scientific illustrations such as model structure diagrams, algorithm flowcharts, and model training pipelines. Additionally, SCI-CQA features 11 common types of data charts, including line charts, scatter plots, and pie charts. SCI-CQA also includes charts with multiple subplots, combinations of images and data visualizations, and hybrid charts that integrate two or more visualization types. Notably, it features specialized charts such as forest plots, correlation matrix charts, Gantt charts, violin plots, density plots, and Pareto charts, which are often overlooked in previous work but are crucial in scientific literature.

Regarding QA types, the dataset is divided into perception and reasoning tasks. While perception tasks have received

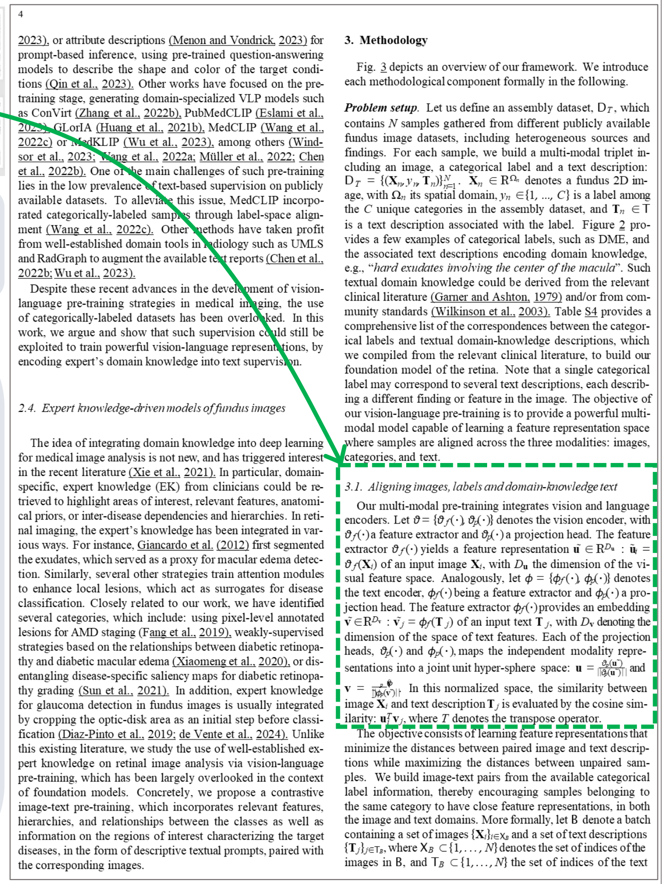
Figure S.20. Illustration of an Irrelevant Answer Case. The figure demonstrates how the model misinterprets the question target, calculating the percentage of papers passing the pipeline (54,557/75,506) instead of the required percentage of high-quality image-text pairs (37,607/202,760), revealing a fundamental misunderstanding of the dataset definition

VII. Matching Error: Case 1



Question: According to the paper, which of the following statements about the text description defined before section '3.1' is True?

A. The text descriptions are designed to provide a one-to-one correspondence between each categorical label and a unique textual representation.
B. A single categorical label may correspond to several text descriptions, each describing a different feature of the image.
C. A single text description may correspond to multiple categorical labels, as various diseases can exhibit similar symptoms..
D. The text descriptions are generated by a Large Language Model that has been fine-tuned on extensive medical imaging data..
E. The text descriptions are annotated by a several invited expert radiologists to ensure accurate representations.
Ground Truth: B.



4

2023), or attribute descriptions (Menon and Vondrick, 2023) for prompt-based inference, using pre-trained question-answering models to describe the shape and color of the target conditions (Qin et al., 2023). Other works have focused on the pre-training stage, generating domain-specialized VLP models such as ConVirt (Zhang et al., 2022b), PubMedCLIP (Eslami et al., 2023), GLorIA (Huang et al., 2021b), MedCLIP (Wang et al., 2022c) or MedKLIP (Wu et al., 2023), among others (Windsor et al., 2023; Wang et al., 2022a; Müller et al., 2022; Chen et al., 2022b). One of the main challenges of such pre-training lies in the low prevalence of text-based supervision on publicly available datasets. To alleviate this issue, MedCLIP incorporated categorically-labeled samples through label-space alignment (Wang et al., 2022c). Other methods have taken profit from well-established domain tools in radiology such as UMLS and RadGraph to augment the available text reports (Chen et al., 2022b; Wu et al., 2023).

Despite these recent advances in the development of vision-language pre-training strategies in medical imaging, the use of categorically-labeled datasets has been overlooked. In this work, we argue and show that such supervision could still be exploited to train powerful vision-language representations, by encoding expert's domain knowledge into text supervision.

2.4. Expert knowledge-driven models of fundus images

The idea of integrating domain knowledge into deep learning for medical image analysis is not new, and has triggered interest in the recent literature (Xie et al., 2021). In particular, domain-specific, expert knowledge (EK) from clinicians could be retrieved to highlight areas of interest, relevant features, anatomical priors, or inter-disease dependencies and hierarchies. In retinal imaging, the expert's knowledge has been integrated in various ways. For instance, Giancardo et al. (2012) first segmented the exudates, which served as a proxy for macular edema detection. Similarly, several other strategies train attention modules to enhance local lesions, which act as surrogates for disease classification. Closely related to our work, we have identified several categories, which include: using pixel-level annotated lesions for AMD staging (Fang et al., 2019), weakly-supervised strategies based on the relationships between diabetic retinopathy and diabetic macular edema (Xiaomeng et al., 2020), or disentangling disease-specific saliency maps for diabetic retinopathy grading (Sun et al., 2021). In addition, expert knowledge for glaucoma detection in fundus images is usually integrated by cropping the optic-disk area as an initial step before classification (Diaz-Pinto et al., 2019; de Vente et al., 2024). Unlike this existing literature, we study the use of well-established expert knowledge on retinal image analysis via vision-language pre-training, which has been largely overlooked in the context of foundation models. Concretely, we propose a contrastive image-text pre-training, which incorporates relevant features, hierarchies, and relationships between the classes as well as information on the regions of interest characterizing the target diseases, in the form of descriptive textual prompts, paired with the corresponding images.

3. Methodology

Fig. 3 depicts an overview of our framework. We introduce each methodological component formally in the following.

Problem setup. Let us define an assembly dataset, $\mathrm{D}_{\mathcal{T}}$, which contains N samples gathered from different publicly available fundus image datasets, including heterogeneous sources and findings. For each sample, we build a multi-modal triplet including an image, a categorical label and a text description: $\mathrm{D}_{\mathcal{T}} = \{(\mathbf{X}_n, y_n, \mathbf{T}_n)\}_{n=1}^{N}$. $\mathbf{X}_n \in \mathbb{R}^{\Omega_n}$ denotes a fundus 2D image, with Ω_n its spatial domain, $y_n \in \{1, ..., C\}$ is a label among the C unique categories in the assembly dataset, and $\mathbf{T}_n \in \mathrm{T}$ is a text description associated with the label. Figure 2 provides a few examples of categorical labels, such as DME, and the associated text descriptions encoding domain knowledge, e.g., "*hard exudates involving the center of the macula*". Such textual domain knowledge could be derived from the relevant clinical literature (Garner and Ashton, 1979) and/or from community standards (Wilkinson et al., 2003). Table S4 provides a comprehensive list of the correspondences between the categorical labels and textual domain-knowledge descriptions, which we compiled from the relevant clinical literature, to build our foundation model of the retina. Note that a single categorical label may correspond to several text descriptions, each describing a different finding or feature in the image. The objective of our vision-language pre-training is to provide a powerful multi-modal model capable of learning a feature representation space where samples are aligned across the three modalities: images, categories, and text.

3.1. Aligning images, labels and domain-knowledge text

Our multi-modal pre-training integrates vision and language encoders. Let $\vartheta = \{\vartheta_f(\cdot), \vartheta_p(\cdot)\}$ denotes the vision encoder, with $\vartheta_f(\cdot)$ a feature extractor and $\vartheta_p(\cdot)$ a projection head. The feature extractor $\vartheta_f(\cdot)$ yields a feature representation $\tilde{\mathbf{u}} \in \mathbb{R}^{D_\mathbf{u}}$: $\tilde{\mathbf{u}}_i = \vartheta_f(\mathbf{X}_i)$ of an input image $\mathbf{X}_i$, with $D_\mathbf{u}$ the dimension of the visual feature space. Analogously, let $\phi = \{\phi_f(\cdot), \phi_p(\cdot)\}$ denotes the text encoder, $\phi_f(\cdot)$ being a feature extractor and $\phi_p(\cdot)$ a projection head. The feature extractor $\phi_f(\cdot)$ provides an embedding $\tilde{\mathbf{v}} \in \mathbb{R}^{D_\mathbf{v}}$: $\tilde{\mathbf{v}}_j = \phi_f(\mathbf{T}_j)$ of an input text $\mathbf{T}_j$, with $D_\mathbf{v}$ denoting the dimension of the space of text features. Each of the projection heads, $\vartheta_p(\cdot)$ and $\phi_p(\cdot)$, maps the independent modality representations into a joint unit hyper-sphere space: $\mathbf{u} = \frac{\vartheta_p(\tilde{\mathbf{u}})}{\|\vartheta_p(\tilde{\mathbf{u}})\|}$ and $\mathbf{v} = \frac{\phi_p(\tilde{\mathbf{v}})}{\|\phi_p(\tilde{\mathbf{v}})\|}$. In this normalized space, the similarity between image $\mathbf{X}_i$ and text description $\mathbf{T}_j$ is evaluated by the cosine similarity: $\mathbf{u}_i^T \mathbf{v}_j$, where T denotes the transpose operator.

The objective consists of learning feature representations that minimize the distances between paired image and text descriptions while maximizing the distances between unpaired samples. We build image-text pairs from the available categorical label information, thereby encouraging samples belonging to the same category to have close feature representations, in both the image and text domains. More formally, let B denote a batch containing a set of images $\{\mathbf{X}_i\}_{i \in \mathrm{X}_\mathcal{B}}$ and a set of text descriptions $\{\mathbf{T}_j\}_{j \in \mathrm{T}_\mathcal{B}}$, where $\mathrm{X}_\mathcal{B} \subset \{1, \ldots, N\}$ denotes the set of indices of the images in B, and $\mathrm{T}_\mathcal{B} \subset \{1, \ldots, N\}$ the set of indices of the text

Response: The correct answer is: B. A single categorical label may correspond to several text descriptions, each describing a different feature of the image. ×?
Explanation: According to the text, the authors define text descriptions that encode domain knowledge for each disease. A single categorical label can correspond to multiple text descriptions, as these descriptions aim to capture various features, hierarchies, and dependencies of the related pathology.

Error Type: Matching Error
Analysis: The model **correctly made choice "B", but due to a matching error in the Rule-based methods**, the correct choice was not successfully extracted from the model's answer, resulting in a matching error.

Figure S.21. Illustration of a Matching Error Case.