FineRich: TOWARDS HOLISTIC FINE-GRAINED TEXT-RICH IMAGE UNDERSTANDING VIA VERSATILE HTML-BASED SYNTHESIS

Anonymous authors
Paper under double-blind review

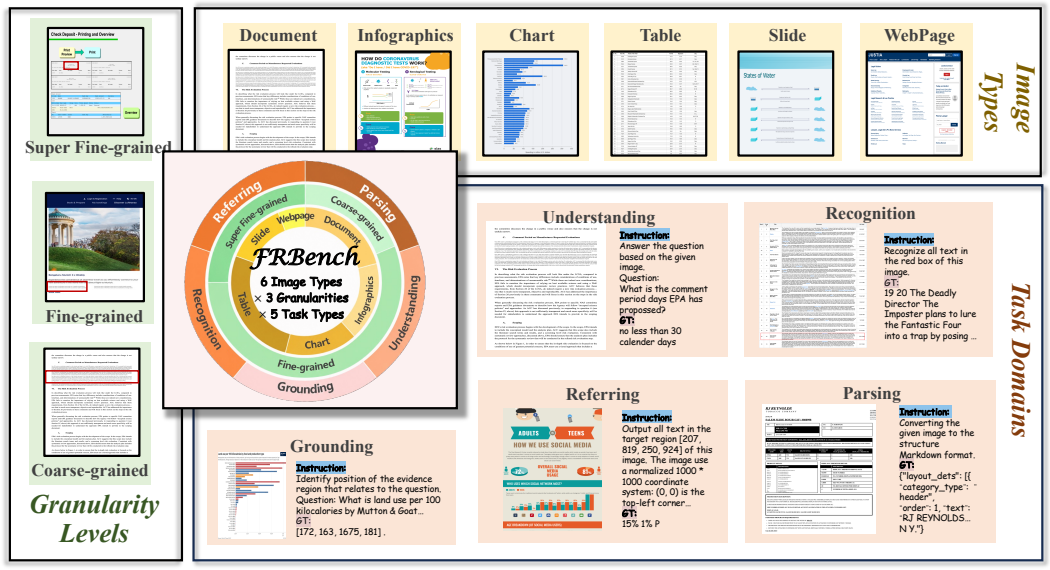


Figure 1: Overview of our FRBench, which is a holistic fine-grained text-rich image understanding benchmark across six text-rich image types, three granularity levels, and five task domains for fine-grained regions, covering all 90 <image, granularity, task> combinations.

ABSTRACT

Fine-grained Text-rich Image Understanding (Fine-grained TIU) is challenging for multimodal large language models (MLLMs) to accurately perceive and understand textual details from their long visual contexts. However, existing benchmarks remain limited in scope and cannot comprehensively evaluate these capabilities. In this paper, we introduce FRBench, a holistic benchmark that systematically covers all 90 combinations of six image types, three granularity levels, and five task domains, enabling a thorough evaluation for fine-grained TIU. To construct high-quality data, we develop a versatile HTML-based synthesis pipeline that renders diverse text-rich images and derives source-traceable TIU annotations. Specifically, we convert source images into HTML and edit evidence region related to the query to increase local information density. Using this pipeline, we further automatically build a 34K training set and propose FineRich, a novel two-stage training method for fine-grained TIU, which includes: 1) a region-to-image OPSD stage that first strengthens basic fine-grained text recognition; and 2) a multi-task GRPO training stage further specializes the model across diverse TIU tasks. Experiments show that FineRich achieves leading performance on FRBench while improving performance on other fine-grained TIU and general OCR benchmarks, demonstrating strong fine-grained capabilities and remarkable generalization.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

1 INTRODUCTION

Though MLLMs have attained strong performance on general OCR tasks Wang et al. (2025); Team (2025); Yu et al. (2025); Team (2026c), they still struggle with fine-grained text-rich image understanding, which requires perceiving and understanding texts in small regions of a larger image Bai et al. (2025b;a). Redundant information in the long visual contexts can obscure fine-grained text details, causing MLLMs to overlook or misinterpret key information.

Recent works have sought to evaluate MLLMs on fine-grained TIU tasks Zhu et al. (2026); Fu et al. (2025); Liu et al. (2024b); Zhou et al. (2025); Liu et al. (2024a). Despite their progress, existing fine-grained TIU benchmarks still fall short in: 1) limited image diversity: current benchmarks primarily focus on a few image types, such as documents, whereas text-rich images combine text with various visual elements and exhibit diverse layouts and visual structures across a broader range of image types; 2) incomplete task coverage: existing works tend to emphasize full-image analysis, whereas evaluating fine-grained TIU requires tasks that target local details, such as grounding and recognition within small regions; and 3) no granularity differentiation: it is necessary to distinguish different granularity levels to reveal how model performance changes from coarse regions to fine details.

To comprehensively evaluate fine-grained TIU capabilities, we introduce FRBench, a holistic benchmark spanning six text-rich image types and five task domains focused on both global layouts and local details. It distinguishes three granularity levels based on the area ratio of the target region to the full image, which are coarse-grained, fine-grained, and super fine—grained. FRBench not only includes all these image, task, and granularity types but also covers all 90 combinations of them, enabling joint analysis across multiple dimensions, such as examining how each task difficulty varies with region sizes. Moreover, as Table 1 shows, FRBench involves the lowest region area ratio among all compared benchmarks, allowing evaluation on finer text details.

Inspired by the diversity and flexibility of HTML, we propose a versatile HTML-based pipeline for synthesizing fine-grained text-rich images and annotations. We first convert images from diverse sources into HTML Mathew et al. (2021; 2022); Masry et al. (2022); Tanaka et al. (2023); Wang et al. (2024a); Liu et al. (2024b); Kim et al. (2024) and use their original question–answer pairs to locate the evidence DOMs. Then we generate structurally similar but distracting new DOMs and insert them alongside the evidence DOM within its original container in the source HTML. By increasing local information density, we ultimately obtain fine-grained samples. Since HTML rendering can directly expose the text contents and region coordinates of DOMs, we can obtain accurate annotations for various region-level tasks, such as local text recognition.

Using this pipeline, we automatically generate 2,500 samples, of which 2,190 pass manual check and constitute FRBench. It yields a pass rate of 87.6%, demonstrating the robustness of this synthesis pipeline. We further synthesize 34K samples as training set and propose FineRich, a two-stage training method tailored to fine-grained TIU, which includes: 1) a region-to-image OPSD stage that strengthens basic fine-grained text recognition. In this stage, the student model recognizes text in target region from the full image, while the teacher model's input is the crop of target region, providing more precise supervision on text details; Different to the crop-to-full-image distillation framework of Vision-OPD (Yuan et al., 2026), FineRich introduces teacher-guided prefix correction and sparse token selection with reliability weighting. and 2) a multi-task GRPO stage jointly optimizes multiple fine-grained TIU tasks with various task-specific rewards. It develops expertise for each task,

Benchmark	Image Type						Task Type					Multi Granularity	All Combination	Region Area Ratio	Region Annotation
	Document	Infographics	Chart	Table	Slide	WebPage	Understanding	Grounding	Recognition	Referring	Parsing				
OmniDocBench Ouyang et al. (2025)	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	-	Manual + Auto
PureDocBench Li et al. (2026)	✓	✗	✗	✓	✓	✗	✗	✗	✗	✗	✓	✗	✗	-	Auto
TIU-Bench Zhang et al. (2025)	✓	✗	✓	✓	✗	✓	✓	✗	✗	✓	✗	✗	✗	-	Manual + Auto
Fox Liu et al. (2024a)	✓	✗	✗	✗	✗	✗	✓	✗	✓	✗	✗	✗	✗	7.17%	Auto
OCRBench v2 Fu et al. (2025)	✓	✓	✓	✓	✓	✓	✓	✓	✗	✓	✗	✗	✗	4.46%	Manual
DOGR Zhou et al. (2025)	✓	✗	✓	✗	✗	✗	✓	✓	✗	✓	✗	✗	✗	1.52%	Auto
VisualWebBench Liu et al. (2024b)	✗	✗	✗	✗	✗	✓	✓	✓	✓	✗	✗	✗	✗	1.43%	Manual + Auto
MMDocBench Zhu et al. (2026)	✓	✓	✓	✓	✗	✗	✓	✓	✓	✗	✗	✗	✗	1.78%	Manual + Auto
FRBench	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	**1.15%**	Auto

Table 1: Comparison between FRBench and other TIU benchmarks. Here, **Multi-Granularity** refers to distinguish different ranges of region sizes, **All Combination** means coverage of all <image, task, granularity> combinations, **Region Area Ratio** is calculated by the average ratio of involved region areas to the full image area of each sample, **Region Annotation** specifically refers to the position of each region in image.

advancing the model from recognizing local text to understanding its semantic contents. To mitigate noise from the student model in the first OPSD stage, we additionally introduce a prefix guide. It first compares the first K output tokens (prefix) of the student and teacher. A mismatch indicates that the student has perceived incorrect region, where FineRich replaces its prefix with the teacher prefix to guide student to focus on correct regions and gradually optimize its grounding. When their prefixes match, FineRich will retain the full student rollout to boost its fine-grained recognition.

We evaluate both open-source and closed-source MLLMs on FRBench across all task, image, granularity types and their combinations, providing a comprehensive assessment of existing MLLMs on fine-grained TIU. Experiments also show the superiority of FineRich for fine-grained TIU on our FRBench. It is worth noting that, FineRich also achieves gains on other fine-grained TIU benchmarks and general OCR benchmarks, suggesting its generalization and verifies that improvements in fine-grained capabilities can further benefit general text recognition.

2 Related Work

General OCR Benchmarks. Existing OCR benchmarks cover diverse image types and tasks. DocVQA Mathew et al. (2021) and InfographicVQA Mathew et al. (2022) evaluate document and infographic QA, while ChartQA Masry et al. (2022), CharXiv Wang et al. (2024b), MMTab Zheng et al. (2024), and TableVQA-Bench Kim et al. (2024) target charts and tables. SlideVQA Tanaka et al. (2023), MMLongBench-Doc Ma et al. (2024), WebSRC Chen et al. (2021), WebQuest Wang et al. (2024a), and SeedBench2Plus Li et al. (2024) extend coverage to slides, multi-page documents, webpages, and maps. Beyond QA, OCRBench Liu et al. (2024c) covers recognition and information extraction, while OmniDocBench Ouyang et al. (2025) and PureDocBench Li et al. (2026) assess document parsing. These benchmarks primarily emphasize overall recognition and understanding, whereas FRBench targets local textual details across image and task types.

Fine-grained Text-rich Image Understanding. Recent benchmarks increasingly target local details. Fox Liu et al. (2024a) evaluates region-level recognition, DOGR Zhou et al. (2025) examines document grounding and referring, and VisualWebBench Liu et al. (2024b) assesses webpage elements. MMDocBench Zhu et al. (2026) combines grounding, recognition, and QA with supporting-region annotations, while OCRBench v2 Fu et al. (2025) includes text localization and handwritten content extraction. TIU-Bench Zhang et al. (2025) broadens coverage through text reading and structured parsing. However, image and task coverage remains uneven, with limited granularity differentiation. FRBench instead covers all combinations of six image types, five tasks, and three granularity levels.

MLLMs for Fine-grained Text-rich Image Understanding. Recent MLLMs emphasize text-rich image understanding through broad general OCR evaluations Wang et al. (2025); Yu et al. (2025); Team (2025); Team et al. (2025); Team (2026c); Qwen Team (2026a); Team (2026d); OpenAI (2026); Team (2026a;b), but dedicated fine-grained TIU evaluation remains limited. Some MLLMs include benchmarks such as OCRBench v2 Fu et al. (2025), yet their reported results reveal persistent fine-grained challenges despite strong general OCR capabilities Bai et al. (2025a;b); Qwen Team (2026b). We systematically evaluate open-source and closed-source MLLMs of varying sizes across fine-grained dimensions. We also introduce FineRich, a tailored two-stage training method that improves fine-grained perception and understanding, outperforming the evaluated MLLM baselines.

3 FRBench

3.1 Fine-grained Text-rich Image Synthesis

Inspired by diversity and flexibility of HTML, which can simulate various types of text-rich images and allow modification of their visual appearance through DOM editing, we propose a versatile HTML-based pipeline to synthesize annotations for fine-grained TIU.

Convert Image to HTML. As shown in Figure 2, we first collect diverse types of text-rich images from existing text recognition and understanding test sets Mathew et al. (2021; 2022); Masry et al. (2022); Tanaka et al. (2023); Kim et al. (2024); Liu et al. (2024b); Wang et al. (2024a). We then

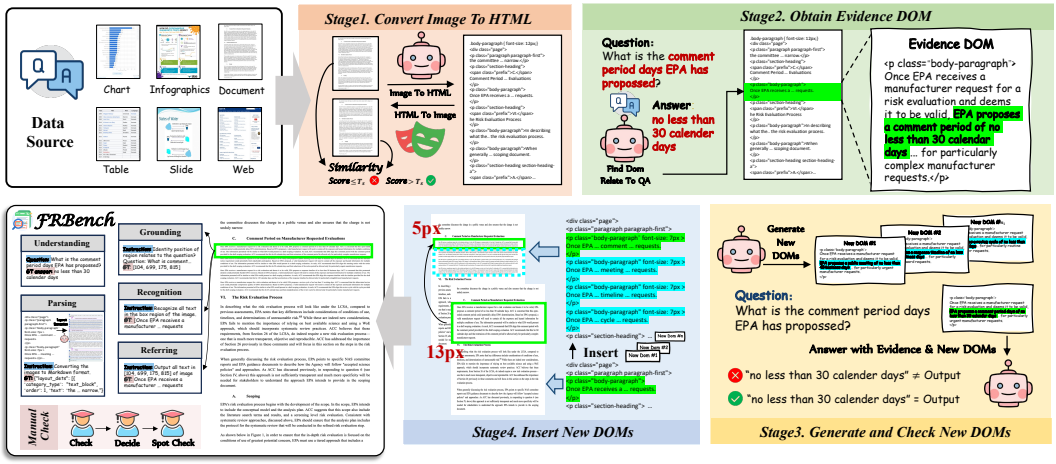


Figure 2: Overview of our HTML-based data synthesis pipeline. We convert source images into HTMLs and then generate and insert new HTML DOMs to increase information density of local regions. Via HTML rendering, we can obtain accurate DOM contents and coordinates as annotations for various TIU tasks.

use an MLLM M_{I2H} to convert these images into HTML. Since this conversion may introduce information loss or structural distortions, we render the resulting HTML into images using Playwright Microsoft (2020) and compare them with the source images. The similarity score s_i between source image p_i^o and HTML-rendered image p_i^h of the i-th sample is calculated as the average of three metrics: CLIP visual similarity, SSIM structure similarity and BLEU similarity between image texts extracted by EasyOCR JaidedAI (2020). Only samples with similarity scores above T_s are retained.

Obtain Evidence DOM and Generate New DOMs. Then we leverage MLLM M_L to locate the evidence DOM in the generated HTML that support the question–answer pair associated with the source image. Based on the evidence DOM, we use M_L to generate n_i new DOMs with similar structures but different contents. To prevent generating too many new doms to cause excessive information density, where final texts cannot be read, we dynamically determine n_i based on the original density of the evidence DOM container c_i. In particular, if c_i contains sibling DOMs, it is considered moderately content-dense, and n_i will be set to n_s, where n_s denotes the number of existing DOMs in c_i. Otherwise, n_i is set to N_n to ensure sufficient DOM expansion. To prevent generated new doms from invalidating the source question–answer annotation, we additionally ask M_L to answer the source question based on evidence DOM and all generated new DOMs. Only when the resulting answer matches the source answer will we retain the new DOMs.

Insert New DOMs. Finally, we gather all existing DOMs in evidence DOM container together with generated new DOMs and calculate the scaling factor required to fit them within the container. We apply CSS styles to keep container size fixed and scale all these DOMs during HTML rendering, thereby increasing local information density. After converting this edited HTML into image via Playwright, we finally obtain fine-grained text-rich image.

3.2 Task Construction

We reuse the source question–answer pair for the synthesized image p_f to construct the understanding task. From the rendered HTML, we obtain the evidence DOM text t_r and coordinates b_r for three region-centric tasks (Figure 2). Grounding requires predicting b_r given the question. Recognition requires reading t_r from region-labeled image p'_f, where the bounding box b_r is drawn on p_f. Referring instead provides b_r as textual coordinates and asks for t_r, assessing visual perception and text–visual alignment. We use M_L to generate OmniDocBench-format annotations Ouyang et al. (2025) from the HTML of p_f, requiring structured text recognition outputs. For parsing, we further design fine-grained metrics to assess local detail preservation during full-image parsing.

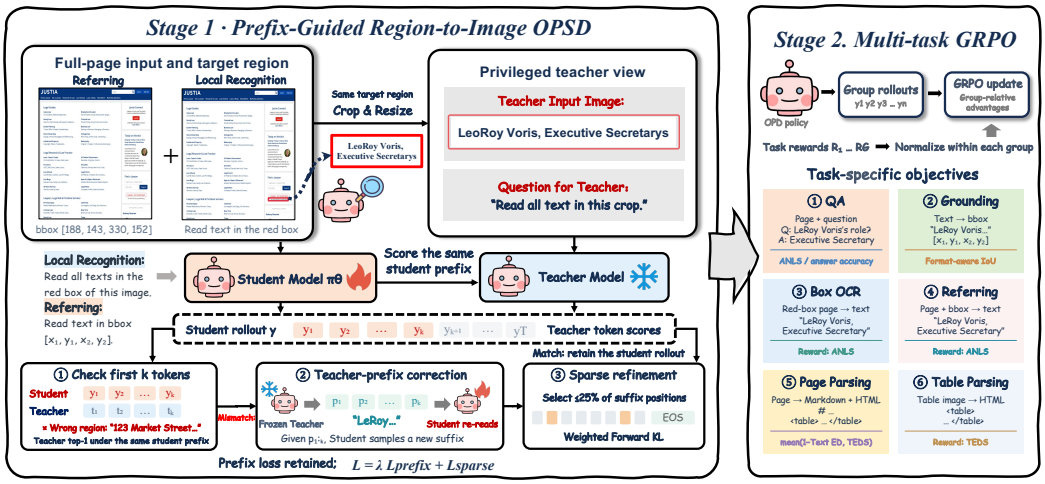


Figure 3: Overview of FineRich. We first transfer crop-level recognition through prefix-guided region-to-image OPSD, then apply multi-task GRPO to refine recognition, referring, grounding, understanding, and parsing.

3.3 Data Statistics

FRBench. We synthesize 2,500 samples and perform multi-round manual verification: each sample is checked by at least one annotator, uncertain cases are reviewed by another, and a final random spot check covers 10% of samples. We retain 2,190 samples, yielding an 87.6% acceptance rate. FRBench comprises 876 images across six text-rich image types and five task domains, with distributions detailed in Appendix A.1. Based on the evidence region-to-image area ratio r, we define three granularity levels: super fine-grained ($0 < r \leq 0.2\%$), fine-grained ($0.2\% < r \leq 2\%$), and coarse-grained ($2\% < r \leq 100\%$). FRBench covers all 90 <image, task, granularity> combinations for holistic fine-grained TIU evaluation. Its average r is 1.15%, a 59.79% reduction from the source HTML-rendered images and lower than those of the compared benchmarks (Table 1), enabling finer-grained evaluation.

Training Data. Using the same pipeline and images from existing training sets Mathew et al. (2021; 2022); Masry et al. (2022), we synthesize 14K samples across all five task domains for multi-task GRPO and 20K recognition samples for OPSD. Only Image-to-HTML conversion requires a powerful MLLM and subsequent steps process limited-length HTML text. We therefore replace M_L with a lightweight model M'_L to reduce synthesis costs. A manual spot check of 200 training samples yields an 83.5% acceptance rate, supporting cost-effective data synthesis.

4 Method

4.1 Prefix-Guided Region-to-Image OPSD

Crop-guided transfer. Fine-grained text regions are often small and difficult to locate and read amid dense surrounding content. Given image x, instruction q, and evidence box b, the student reads the full image, while a frozen copy of its initial checkpoint serves as the teacher and reads the target crop. The crop isolates local text, providing clearer supervision for fine-grained recognition. Both score a response z under the same preceding text but different visual inputs:

$$P_{S,t} = \pi_\theta(\cdot \mid x, q, z_{<t}), \qquad P_{T,t} = \pi_T(\cdot \mid \mathrm{crop}(x, b), q_b, z_{<t}), \tag{1}$$

where q_b requests transcription of the cropped text. Building on this formulation, FineRich incorporates teacher-guided prefix detection and correction, sparse token selection with history-based reliability weighting, and subsequent multi-task GRPO refinement.

Prefix guidance and correction. An unreliable beginning can lead the student to continue reading the wrong content. For recognition and referring, we check the first K tokens of a student rollout y against the teacher's top-k candidate sets $\mathcal{V}_{T,t}^{(k)}$, evaluated under the same student history $y_{<t}$. If any

token falls outside its candidate set, the teacher generates a replacement prefix $p_{1:K}$ from the crop using its own history $p_{<t}$. The student then regenerates the continuation:

$$z = \begin{cases} p_{1:K} \oplus \hat{y}_{>K}, & \text{if } \exists\, t \le K :\ y_t \notin \mathcal{V}_{T,t}^{(k)}, \\ y, & \text{otherwise,} \end{cases} \qquad \hat{y}_{>K} \sim \pi_\theta(\cdot \mid x, q, p_{1:K}), \tag{2}$$

where $\oplus$ denotes concatenation. The teacher prefix guides the student toward the target text while leaving the continuation student-generated. The teacher re-scores z, and a prefix loss trains the student to reproduce the supplied beginning.

Sparse refinement. We focus KL supervision on uncertain or disputed tokens, while reducing supervision after unreliable generated text. Let $\mathcal{C}$ contain response positions excluding supplied prefixes. We rank them by $s_t = [1 - (1 - h_t)(1 - d_t)]w_t$, where h_t is student entropy min–max normalized over $\mathcal{C}$ with denominator floor 10^{-6}, and $d_t = \mathrm{JSD}(\widetilde{P}_{T,t}, \widetilde{P}_{S,t})/\log 2$. The distributions $\widetilde{P}$ use the teacher's top-100 tokens plus residual probability mass.

Reliability is $w_t = \eta_t \exp[\mathrm{clip}(\frac{1}{4}\sum_{i\in\mathcal{C}, i<t} v_i, -4, 0)]$, where $v_i = \mathrm{clip}(\log P_{T,i}(z_i) - \log P_{S,i}(z_i), -5, 5)$ on teacher-supported tokens and zero otherwise. We set $\eta_t = 0$ if any preceding candidate token falls outside the teacher's top-100 support, and one otherwise. Selecting the top $\lceil 0.25|\mathcal{C}|\rceil$ positions and removing zero weights gives $\mathcal{S}$. We minimize

$$\mathcal{L}_{\mathrm{OPSD}} = \mathbb{E}\left[\frac{1}{N}\sum_{t\in\mathcal{S}} w_t\, \mathrm{KL}(\widetilde{P}_{T,t} \| \widetilde{P}_{S,t}) + \lambda \mathcal{L}_{\mathrm{prefix}}\right], \tag{3}$$

where $N = \max(1, |\mathcal{C}|)$, $\lambda = 0.1$, and $\mathcal{L}_{\mathrm{prefix}} = -K^{-1}\sum_{t=1}^{K} \log \pi_\theta(p_t \mid x, q, p_{<t})$ on corrected responses and zero otherwise. Losses are averaged over responses; teacher targets and selection weights are detached. At inference, only the student is needed.

4.2 Multi-Task GRPO Refinement

Starting from OPSD, we optimize five task domains, with Parsing split into document and table configurations. Understanding, Recognition, and Referring use ANLS-style similarity; Grounding uses IoU. Understanding and Grounding scores are multiplied by $0.9 + 0.1f$, where f indicates valid output format. Document parsing averages normalized text edit similarity and table TEDS over available element types; table parsing uses TEDS alone.

For each input, the rollout policy samples G responses with rewards R_i. GRPO (Shao et al., 2024) uses group-relative advantages and a token-averaged objective:

$$A_i = \frac{R_i - \bar{R}}{\sigma_R + 10^{-6}}, \qquad \mathcal{L}_{\mathrm{GRPO}} = -\frac{\sum_{i,t}[\ell_{\mathrm{clip}}(r_{i,t}, A_i) - \beta \widehat{D}_{i,t}]}{\sum_i |y_i|}. \tag{4}$$

Here $\bar{R}$ and σ_R are the group reward mean and sample standard deviation, and $r_{i,t}$ is the current-to-rollout token probability ratio. The surrogate is $\min\{rA, \mathrm{clip}(r, 0.8, 1.2)A\}$, floored at $3A$ when $A < 0$. We use $\beta = 0.01$ and the sampled reference penalty $\widehat{D} = e^{\delta} - \delta - 1$, where δ is the reference-to-current log-probability ratio. The reference remains frozen at the OPSD checkpoint.

5 Experiments

5.1 Benchmarks and Experimental Settings

Benchmarks. For fine-grained TIU, except for our FRBench, we also evaluate FineRich on Fox Liu et al. (2024a), DOGR Zhou et al. (2025), OCRBench v2 Fu et al. (2025), MMDocBench Zhu et al. (2026), and VisualWebBench Liu et al. (2024b). For general text recognition, we evaluate FineRich on OCRBench Liu et al. (2024c), CC-OCR Yang et al. (2025), SeedBench2Plus Li et al. (2024), SlideVQA Tanaka et al. (2023) and OmniDocBench Ouyang et al. (2025).

Metrics. For FRBench, we report **ANLS**↑ and **EM**↑ for understanding, recognition, and referring, and **IOU**↑ for grounding. For parsing, we adopt **TextEdit**↓, **Table TEDS**↑, and **ReadOrderEdit**↓

Table 2: Comparison results across five fine-grained TIU tasks. Here, the best results are in **bold** and the second-best results are underlined, **TxE**, **TDS**, **CDS**, **ROE** refer to TextEdit, Table TEDS, Chart TEDS and ReadOrderEdit, respectively.

Models	Params	Understanding		Grounding	Recognition		Referring		Parsing							Overall↑
		ANLS↑	EM↑	IoU↑	ANLS↑	EM↑	ANLS↑	EM↑	TxE↓	TDS↑	CDS↑	ROE↓	RTM↑	RROD↓	AVG↑	
Closed-Source Models																
GPT5.5	-	80.04	63.93	24.51	**95.56**	**84.70**	57.09	43.15	0.50	50.40	64.48	0.51	54.03	0.54	54.85	59.57
Gemini3.6	-	81.05	67.58	35.33	92.10	78.54	**87.82**	**75.11**	0.28	**82.56**	61.38	0.33	**70.22**	**0.38**	**71.42**	**72.75**
Open-Source Models																
Qwen3-VL Series																
Qwen3-VL-2B	2B	58.25	42.24	11.78	39.32	25.11	14.67	10.50	0.30	66.87	55.08	0.36	65.03	0.42	64.29	47.65
Qwen3-VL-4B	4B	74.73	58.45	15.95	73.85	51.83	35.44	21.00	0.22	67.77	56.63	**0.31**	64.67	0.40	66.83	58.41
Qwen3-VL-8B	8B	78.55	62.10	18.00	73.96	53.42	37.74	21.69	0.25	70.69	60.68	0.34	68.25	0.41	68.55	60.31
Qwen3-VL-235B	235B/22B	81.76	**68.49**	17.81	84.98	67.12	68.39	49.09	0.25	69.09	62.84	0.33	66.70	0.40	68.37	65.80
Qwen3.5 Series																
Qwen3.5-2B	2B	65.01	49.77	6.88	25.29	9.59	9.42	3.88	0.43	49.69	51.55	0.44	50.67	0.55	52.18	39.41
Qwen3.5-4B	4B	70.28	54.57	9.84	70.21	45.43	14.42	6.85	0.39	61.51	34.29	0.38	53.42	0.48	52.51	46.85
Qwen3.5-9B	9B	80.19	66.89	13.47	78.65	55.48	24.95	12.79	0.36	65.78	58.33	0.40	58.82	0.49	61.78	55.55
Qwen3.5-27B	27B	81.91	65.98	19.56	88.06	69.18	51.96	30.37	0.30	67.00	56.46	0.34	62.37	0.42	63.86	62.11
Qwen3.5-122B	122B/10B	81.93	66.44	17.46	84.12	60.27	30.84	15.30	0.30	70.70	54.48	0.35	63.03	0.42	64.56	59.08
Qwen3.5-397B	397B/17B	**82.19**	67.58	20.79	87.63	71.92	39.08	22.15	0.35	57.02	54.27	0.38	52.72	0.53	57.20	57.31
Qwen3.8 Series																
Qwen3.8-27B	27B	81.06	64.61	25.12	84.84	61.19	55.94	32.65	0.26	73.21	**69.49**	0.32	62.78	**0.38**	69.77	65.76
Qwen3.8-Flash	125B/6B	80.95	65.07	17.66	92.23	71.00	53.24	33.33	0.33	68.12	51.88	0.37	56.24	0.48	60.78	60.90
Others																
MiMo-VL-7B-RL	7B	17.62	13.70	5.06	65.08	38.81	7.01	3.88	0.34	39.21	0.00	0.51	54.08	0.43	39.86	31.78
InternVL3.5-8B	8B	58.82	38.13	7.96	40.42	16.89	17.67	4.57	0.34	63.98	58.25	0.36	55.64	0.52	61.09	46.15
MiniCPM-V-4.5	9B	56.85	43.15	9.81	53.12	26.94	19.98	6.16	0.27	61.95	56.40	0.34	56.52	0.48	62.04	48.49
Qwen3.6-27B	27B	81.76	67.12	16.92	90.13	67.81	52.92	35.16	0.31	69.45	53.86	0.35	57.34	0.46	62.40	61.42
Qwen2.5-VL-72B	72B	65.61	49.54	10.58	81.92	59.59	51.95	30.82	0.31	69.86	59.23	0.38	60.34	0.50	64.54	58.53
GLM-4.6V	109B	68.77	54.79	26.05	67.28	41.55	60.11	38.13	0.34	65.75	28.74	0.37	61.81	0.49	55.59	55.57
GLM-5.3-Flash	320B/18B	79.18	64.38	9.20	90.34	69.18	16.50	5.71	0.35	69.20	48.51	0.39	59.84	0.50	60.74	54.77
Kimi-K2.6	1T/32B	80.71	65.75	27.61	73.41	39.95	40.66	19.41	0.30	77.78	46.87	0.35	64.22	0.45	64.80	60.20
Kimi-K3	2.8T/104B	77.74	63.47	13.67	89.09	68.72	20.68	10.96	0.32	80.58	57.50	0.35	63.31	0.49	67.34	58.82
Our Models																
FineRich-4B (Ours)	4B	77.43	62.79	**42.22**	86.26	68.26	86.87	70.09	**0.18**	66.72	54.50	**0.31**	67.57	0.40	67.59	70.39
FineRich-8B (Ours)	8B	80.29	66.89	36.31	83.95	65.53	80.96	63.70	0.21	67.35	62.12	**0.31**	70.14	**0.38**	69.55	69.96

from OmniDocBench Ouyang et al. (2025), and **Chart TEDS**↑ from OCRBench v2 Fu et al. (2025). We further introduce **Region Text Match (RTM**↑**)** and **Region Reading Order Distance (RROD**↓**)** to assess local detail preservation during full-image parsing (Appendix A.1). Parsing **AVG**↑ averages four metrics: $1 - \mathrm{TextEdit}$, Table TEDS, Chart TEDS, and RTM. Following the metric-level aggregation protocol of OCRBench v2, **Overall** averages eight normalized metrics: these four parsing metrics, the three task-specific ANLS scores, and grounding IOU. Here, since parsing comprises four distinct and important subtasks: text recovery, table parsing, chart parsing, and local evidence preservation, we retain their corresponding metrics when computing Overall.

Implementation Details. For data synthesis, we leverage Kimi-K2.6 Team (2026b) as M_{I2H} to convert images into HTML. For the following synthesis steps, we leverage Kimi-K2.6 Team (2026b) and Qwen3.8-27B as M_L and M'_L for test and training data construction, respectively. The image similarity score threshold T_s is 0.6 and the RTM threshold T_{RTM} during evaluation is 0.3. The number of new DOMs to generate N_n is 3.

We initialize FineRich-4B and FineRich-8B from Qwen3-VL-4B and Qwen3-VL-8B, respectively.For OPSD, we use a frozen teacher, a prefix length of $K = 4$, teacher top-100 token distributions, and a sparse token selection ratio of 25%. The prefix loss weight is set to $\lambda = 0.1$. GRPO starts from the OPSD checkpoint, with a clipping range of $[0.8, 1.2]$ and a KL coefficient of $\beta = 0.01$.

Models. We conduct evaluation experiments with 21 open-source and 2 closed-source MLLMs on the proposed FRBench. We select those open-source models with strong text-rich image understanding capabilities, including Qwen2.5-VL Bai et al. (2025b), Qwen3-VL Bai et al. (2025a), Qwen3.5 Team (2026d), Qwen3.6 Qwen Team (2026a) and Qwen3.8 Qwen Team (2026b) series at different scales, MiMo-VL-7B-RL Team (2025), MiniCPM-V-4.5 Yu et al. (2025), InternVL3.5-8B Wang et al. (2025), GLM-4.6V Team et al. (2025), Kimi-K2.6 Team (2026b) and Kimi-K3 Team (2026c). For closed-source models, we evaluate recent GPT5.5 OpenAI (2026) and Gemini3.6 Team (2026a).

5.2 Comparison Results

Performance on Different Tasks. Table 2 highlights several challenges in fine-grained TIU. Grounding remains challenging, as MLLMs struggle to precisely locate evidence regions. Lower performance on referring than recognition suggests limitations in text-visual alignment, particularly

Table 3: Comparison results across three granularity levels. Here we report task Overall.

Models	Super Fine-grained	Fine-grained	Coarse-grained
Closed-Source Models			
GPT5.5	55.51	58.18	65.46
Gemini3.6	**69.80**	**73.42**	73.71
Open-Source Models			
Qwen3-VL Series			
Qwen3-VL-2B	42.01	47.52	53.74
Qwen3-VL-4B	53.93	59.18	60.41
Qwen3-VL-8B	57.21	61.02	60.32
Qwen3-VL-235B	62.62	65.92	67.79
Qwen3.5 Series			
Qwen3.5-2B	35.89	40.04	42.17
Qwen3.5-4B	43.15	47.53	48.84
Qwen3.5-9B	51.33	56.21	58.14
Qwen3.5-27B	58.10	63.15	63.77
Qwen3.5-122B	56.31	59.83	59.90
Qwen3.5-397B	54.25	59.56	55.38
Qwen3.8 Series			
Qwen3.8-27B	65.13	66.58	61.19
Qwen3.8-Flash	57.26	62.33	60.82
Others			
MiMo-VL-7B-RL	30.44	33.21	30.02
InternVL3.5-8B	38.82	47.36	50.82
MiniCPM-V-4.5	44.11	48.79	53.23
Qwen3.6-27B	60.92	61.61	60.50
Qwen2.5-VL-72B	54.60	58.77	62.13
GLM-4.6V	49.73	56.18	59.88
GLM-5.3-Flash	50.33	55.17	58.55
Kimi-K2.6	56.73	60.61	63.16
Kimi-K3	54.44	59.10	62.57
Our Models			
FineRich-4B (Ours)	62.57	71.44	**76.01**
FineRich-8B (Ours)	65.76	70.43	71.52

Table 4: Comparison results across six image types. Here, we report the overall metric of all tasks for each image type and the best results are **bold** and second-bests are underlined.

Models	Document	Infographics	Chart	Table	Slide	WebPage
Closed-Source Models						
GPT5.5	56.41	57.52	58.95	69.77	59.65	53.43
Gemini3.6	**76.58**	72.66	57.43	**79.92**	69.82	**71.86**
Open-Source Models						
Qwen3-VL Series						
Qwen3-VL-2B	46.01	30.07	36.91	53.01	45.12	41.99
Qwen3-VL-4B	62.10	55.98	43.07	59.55	60.62	58.90
Qwen3-VL-8B	63.77	57.59	44.18	60.82	62.99	63.68
Qwen3-VL-235B	64.69	69.10	57.18	68.27	66.24	67.73
Qwen3.5 Series						
Qwen3.5-2B	41.34	32.86	32.53	42.41	38.22	33.94
Qwen3.5-4B	53.14	37.87	30.98	52.62	40.70	45.45
Qwen3.5-9B	58.91	47.09	43.24	56.70	59.09	54.60
Qwen3.5-27B	66.80	63.73	48.84	63.53	61.52	66.72
Qwen3.5-122B	62.75	59.27	47.09	61.37	60.60	56.26
Qwen3.5-397B	64.12	60.33	48.94	58.05	52.75	61.10
Qwen3.8 Series						
Qwen3.8-27B	70.44	60.82	53.01	58.64	68.54	69.19
Qwen3.8-Flash	69.07	59.20	49.36	61.44	62.62	63.94
Others						
MiMo-VL-7B-RL	38.90	37.29	24.42	28.64	43.10	41.00
InternVL3.5-8B	44.02	33.44	36.35	51.07	49.30	32.17
MiniCPM-V-4.5	47.99	37.57	36.62	53.92	41.23	40.57
Qwen3.6-27B	69.24	61.24	48.21	62.77	65.16	61.33
Qwen2.5-VL-72B	60.02	54.68	47.03	63.15	58.64	52.97
GLM-4.6V	61.74	51.99	43.98	60.89	63.82	59.49
GLM-5.3-Flash	57.54	47.17	45.75	58.19	50.97	49.75
Kimi-K2.6	66.63	56.99	47.05	65.07	61.52	58.64
Kimi-K3	59.48	52.41	49.49	66.46	50.90	47.42
Our Models						
FineRich-4B (Ours)	71.66	59.43	**64.15**	76.05	71.49	71.44
FineRich-8B (Ours)	69.16	**73.43**	64.09	72.19	**72.41**	68.48

in mapping textual coordinates to image regions. For parsing, models still struggle to preserve local textual details when processing the full image. Larger models generally achieve better overall performance, but dense models often outperform sparse models even with fewer parameters.

Performance on Different Granularity Levels. In Table 3, most models perform worse at finer granularities, highlighting the difficulty of perceiving and understanding text within small regions. However, some results show a decline from fine-grained to coarse-grained regions. This is because the effect of granularity varies across tasks, and we provide a further analysis in Appendix A.2.

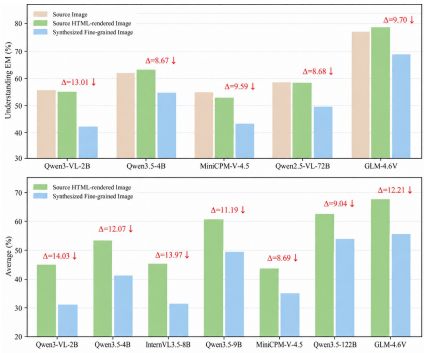


Figure 4: Comparison results with different MLLMs on Source, Source HTML-rendered and Synthesized Fine-grained images using multiple TIU tasks. Here, Average refers to the average score of Understanding ANLS, Grounding IOU, Recognition ANLS and Referring ANLS.

Performance on Different Image Types. As shown in Table 4, performance varies considerably across image types, with charts generally posing greater challenges than other types. We find that Chart understanding often requires jointly interpreting axes, legends, and values, demanding stronger capabilities to perceive and understand these elements together. Infographics also remain difficult for many MLLMs, while performance on slides and webpages varies across models. These differences suggest that fine-grained TIU capabilities do not transfer uniformly across image types. We also report task-specific performance across different image types in Appendix A.2 to further examine their distinct characteristics in fine-grained TIU.

Fine-grained Generalization to Other Benchmarks. Table 5 shows that FineRich improves fine-grained text-rich image understanding beyond FRBench. The average score increases from 56.32 to 60.67 at 4B and from 58.61 to 60.57 at 8B. These results support improved transfer of localization and fine-grained understanding capabilities.

Table 5 shows that the fine-grained gains are accompanied by improvements on all five evaluated general benchmarks at both model scales. The average score rises from 78.70 to 79.75 for the

Table 5: Comparison results between baseline on fine-grained TIU and general OCR benchmarks.

Model	Fine-grained TIU						General OCR					
	OCRBench v2	DOGR Relaxed Acc	MMDocBench F1/IOU	VisualWeb Bench	Fox Overall	**Avg**	OCRBench	CC-OCR	SeedBench 2Plus	SlideVQA	OmniDocBench	**Avg**
Qwen3-VL-4B	61.36	57.23	70.95/17.52	77.03	41.74	56.32	86.90	75.49	81.55	62.83	86.71	78.70
Qwen3-VL-8B	63.45	61.50	72.86/17.80	79.48	43.30	58.61	90.30	79.41	76.39	64.58	86.29	79.39
FineRich-4B	62.54	59.96	71.75/25.77	77.34	54.76	60.67	87.50	77.01	82.34	63.09	88.81	79.75
FineRich-8B	63.38	63.73	73.49/26.32	79.51	46.33	60.57	90.80	80.21	77.08	66.28	86.78	80.23

4B model and from 79.39 to 80.23 for the 8B model. At 4B, the largest gains are observed on OmniDocBench (+2.10 points) and CC-OCR (+1.52 points). At 8B, SlideVQA improves by 1.70 points, with additional gains on OCRBench, CC-OCR, SeedBench2Plus, and OmniDocBench. Together, these results indicate that the evaluated FineRich models improve fine-grained capabilities without sacrificing performance on the tested general text-rich benchmarks.

5.3 Ablation Study

Table 6: Training-stage ablation on FRBench across five task domains. Here, **Und** is understanding ANLS, **Ground** is Grounding IOU, **Recog** is Recognition ANLS, **Refer** is Referring ANLS, **Parse** is Parsing AVG metric, **Overall** refers to Overall metric of all tasks and best results are in **bold**.

Model	Und	Ground	Recog	Refer	Parse	Overall
Qwen3-VL-4B	74.73	15.95	73.85	35.44	66.83	58.41
Qwen3-VL-8B	78.55	18.00	73.96	37.74	68.55	60.31
Qwen3-VL-4B-OPSD	68.94	26.32	83.94	90.89	62.69	65.15
Qwen3-VL-8B-OPSD	72.89	23.57	85.19	81.46	62.34	64.06
Qwen3-VL-4B-Prefix-OPSD	74.23	19.40	79.00	80.56	66.11	64.71
Qwen3-VL-8B-Prefix-OPSD	77.90	26.15	83.42	78.29	68.52	67.48
Qwen3-VL-4B-GRPO	75.41	36.93	80.53	76.52	65.37	66.36
Qwen3-VL-8B-GRPO	79.69	25.37	84.15	73.59	68.52	67.11
Qwen3-VL-4B-(OPSD + GRPO)	74.35	30.92	93.71	75.64	64.57	66.61
FineRich-4B (Prefix-OPSD + GRPO)	77.43	42.22	86.26	86.87	67.59	**70.39**
FineRich-8B (Prefix-OPSD + GRPO)	80.29	36.31	83.95	80.96	69.55	**69.96**

Increased Difficulty of Synthesized Fine-Grained Images As shown in the upper part of Figure 4, we evaluate understanding task on source images, source HTML-rendered images, and synthesized fine-grained images using their shared question–answer pairs. MLLMs achieve similar performance on source images and their HTML-rendered counterparts, as the latter are rendered from HTML converted from the source images and contain essentially the same content. Performance drops substantially on the synthesized fine-grained images, indicating that smaller evidence regions make understanding more challenging. Using the shared evidence DOMs in the source and fine-grained synthesized HTML, we further evaluate grounding, recognition, and referring tasks on both sets of rendered images. The lower part of Figure 4 shows a more significant performance drop across these tasks, which are proved to be more sensitive to granularity levels.

Ablation of FineRich Training Configurations. Table 6 reveals a clear trade-off across tasks under OPSD. Compared with the original backbones, OPSD reduces QA scores by 5.79/5.66 points and Parsing scores by 4.14/6.21 points at 4B/8B, despite substantial gains in Referring and improvements in Recognition and Grounding. This pattern suggests over-specialization toward local text perception and region-level tasks, at the expense of question answering and structured parsing. With prefix guidance, QA recovers by 5.29/5.01 points and Parsing by 3.42/6.18 points relative to vanilla OPSD, bringing these scores closer to their original levels. However, these recoveries are accompanied by lower Recognition and Referring scores. The benefit of prefix guidance is therefore better understood as mitigating task imbalance rather than uniformly improving every task or guaranteeing a higher aggregate score at both model scales.

6 Conclusion

In this work, we introduce FRBench, a holistic fine-grained TIU benchmark covering six image types, five task domains, three granularity levels, and all 90 <image, task, granularity> combinations. To construct comprehensive fine-grained TIU data, we develop a versatile HTML-based synthesis pipeline that generates diverse fine-grained text-rich images and associated TIU tasks. Based on the training data synthesized by this pipeline, we further propose FineRich, a two-stage training method tailored to fine-grained TIU and substantially improves performance on FRBench. Notably, FineRich also yields improvements on other fine-grained TIU benchmarks and general OCR benchmarks, demonstrating remarkable generalization.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, et al. Qwen3-vl technical report. In *arXiv preprint arXiv:2511.21631*, 2025a.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. In *arXiv preprint arXiv:2502.13923*, 2025b.

Xingyu Chen, Zihan Zhao, Lu Chen, JiaBao Ji, Danyang Zhang, Ao Luo, Yuxuan Xiong, and Kai Yu. Websrc: A dataset for web-based structural reading comprehension. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pp. 4173–4185, 2021.

Ling Fu, Zhebin Kuang, Jiajun Song, Mingxin Huang, Biao Yang, Yuzhe Li, Linghao Zhu, Qidi Luo, Xinyu Wang, Hao Lu, Zhang Li, Guozhi Tang, Bin Shan, Chunhui Lin, Qi Liu, Binghong Wu, Hao Feng, Hao Liu, Can Huang, Jingqun Tang, Wei Chen, Lianwen Jin, Yuliang Liu, and Xiang Bai. OCRBench v2: An improved benchmark for evaluating large multimodal models on visual text localization and reasoning. In *Advances in Neural Information Processing Systems*, volume 38, pp. 107930–107981, 2025.

JaidedAI. Easyocr. 2020.

Yoonsik Kim, Moonbin Yim, and Ka Yeon Song. TableVQA-Bench: A visual question answering benchmark on multiple table domains. In *arXiv preprint arXiv:2404.19205*, 2024.

Bohao Li, Yuying Ge, Yi Chen, Yixiao Ge, Ruimao Zhang, and Ying Shan. Seed-bench-2-plus: Benchmarking multimodal large language models with text-rich visual comprehension. In *arXiv preprint arXiv:2404.16790*, 2024.

Zhiheng Li, Zongyang Ma, Jiaxian Chen, Jianing Zhang, Zhaolong Su, Yutong Zhang, Zhiyin Yu, Ruiqi Liu, Xiaolei Lv, Bo Li, Jun Gao, Ziqi Zhang, Chunfeng Yuan, Bing Li, and Weiming Hu. How far is document parsing from solved? PureDocBench: A source-traceable benchmark across clean, degraded, and real-world settings. In *arXiv preprint arXiv:2605.07492*, 2026.

Chenglong Liu, Haoran Wei, Jinyue Chen, Lingyu Kong, Zheng Ge, Zining Zhu, Liang Zhao, Jianjian Sun, Chunrui Han, and Xiangyu Zhang. Focus anywhere for fine-grained multi-page document understanding. In *arXiv preprint arXiv:2405.14295*, 2024a.

Junpeng Liu, Yifan Song, Bill Yuchen Lin, Wai Lam, Graham Neubig, Yuanzhi Li, and Xiang Yue. VisualWebBench: How far have multimodal LLMs evolved in web page understanding and grounding? In *First Conference on Language Modeling*, 2024b.

Yuliang Liu, Zhang Li, Mingxin Huang, Biao Yang, Wenwen Yu, Chunyuan Li, Xu-Cheng Yin, Cheng-Lin Liu, Lianwen Jin, and Xiang Bai. OCRBench: On the hidden mystery of OCR in large multimodal models. *Science China Information Sciences*, 67(12):220102, 2024c. doi: 10.1007/s11432-024-4235-6.

Yubo Ma, Yuhang Zang, Liangyu Chen, Meiqi Chen, Yizhu Jiao, Xinze Li, Xinyuan Lu, Ziyu Liu, Yan Ma, Xiaoyi Dong, Pan Zhang, Liangming Pan, Yu-Gang Jiang, Jiaqi Wang, Yixin Cao, and Aixin Sun. MMLongBench-Doc: Benchmarking long-context document understanding with visualizations. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95963–96010, 2024.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, pp. 2263–2279, 2022.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 2200–2209, 2021.

Minesh Mathew, Viraj Bagal, Rubèn Tito, Dimosthenis Karatzas, Ernest Valveny, and C. V. Jawahar. InfographicVQA. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1697–1706, 2022.

Microsoft. Playwright enables reliable web automation for testing, scripting, and ai agents. 2020.

OpenAI. Introducing gpt-5.5. 2026.

Linke Ouyang, Yuan Qu, Hongbin Zhou, Jiawei Zhu, Rui Zhang, Qunshu Lin, Bin Wang, Zhiyuan Zhao, Man Jiang, Xiaomeng Zhao, Jin Shi, Fan Wu, Pei Chu, Minghao Liu, Zhenxiang Li, Chao Xu, Bo Zhang, Botian Shi, Zhongying Tu, and Conghui He. OmniDocBench: Benchmarking diverse PDF document parsing with comprehensive annotations. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24838–24848, 2025.

Qwen Team. Qwen3.6-27B: Flagship-level coding in a 27B dense model, 2026a.

Qwen Team. Qwen3.8-Max: A new bar for coding and cowork, 2026b.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. In *arXiv preprint arXiv:2402.03300*, 2024.

Ryota Tanaka, Kyosuke Nishida, Kosuke Nishida, Taku Hasegawa, Itsumi Saito, and Kuniko Saito. SlideVQA: A dataset for document visual question answering on multiple images. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 13636–13645, 2023.

Gemini Team. Introducing gemini 3.6 flash, 3.5 flash-lite, and 3.5 flash cyber. 2026a.

Kimi Team. Kimi-2.6 swarm workstation: The industrial-grade ai hive. 2026b.

Kimi Team. Kimi k3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*, 2026c.

Qwen Team. Qwen3.5: Towards native multimodal agents. 2026d.

V Team, Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, Shuaiqi Duan, Weihan Wang, Yan Wang, Yean Cheng, Zehai He, Zhe Su, Zhen Yang, Ziyang Pan, Aohan Zeng, Baoxu Wang, Bin Chen, Boyan Shi, Changyu Pang, Chenhui Zhang, Da Yin, Fan Yang, Guoqing Chen, Haochen Li, Jiale Zhu, Jiali Chen, Jiaxing Xu, Jiazheng Xu, Jing Chen, Jinghao Lin, Jinhao Chen, Jinjiang Wang, Junjie Chen, Leqi Lei, Letian Gong, Leyi Pan, Mingdao Liu, Mingde Xu, Mingzhi Zhang, Qinkai Zheng, Ruiliang Lyu, Shangqin Tu, Sheng Yang, Shengbiao Meng, Shi Zhong, Shiyu Huang, Shuyuan Zhao, Siyan Xue, Tianshu Zhang, Tianwei Luo, Tianxiang Hao, Tianyu Tong, Wei Jia, Wenkai Li, Xiao Liu, Xiaohan Zhang, Xin Lyu, Xinyu Zhang, Xinyue Fan, Xuancheng Huang, Yadong Xue, Yanfeng Wang, Yanling Wang, Yanzi Wang, Yifan An, Yifan Du, Yiheng Huang, Yilin Niu, Yiming Shi, Yu Wang, Yuan Wang, Yuanchang Yue, Yuchen Li, Yusen Liu, Yutao Zhang, Yuting Wang, Yuxuan Zhang, Zhao Xue, Zhengxiao Du, Zhenyu Hou, Zihan Wang, Peng Zhang, Debing Liu, Bin Xu, Juanzi Li, Minlie Huang, Yuxiao Dong, and Jie Tang. Glm-4.5v and glm-4.1v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. In *arXiv preprint arXiv:2507.01006*, 2025.

Xiaomi LLM-Core Team. Mimo-vl technical report. In *arXiv preprint arXiv:2506.03569*, 2025.

Maria Wang, Srinivas Sunkara, Gilles Baechler, Jason Lin, Yun Zhu, Fedir Zubach, Lei Shu, and Jindong Chen. WebQuest: A benchmark for multimodal QA on web page sequences. In *arXiv preprint arXiv:2409.13711*, 2024a.

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

Zirui Wang, Mengzhou Xia, Luxi He, Howard Chen, Yitao Liu, Richard Zhu, Kaiqu Liang, Xindi Wu, Haotian Liu, Sadhika Malladi, Alexis Chevalier, Sanjeev Arora, and Danqi Chen. CharXiv: Charting gaps in realistic chart understanding in multimodal LLMs. In *Advances in Neural Information Processing Systems*, volume 37, pp. 113569–113697, 2024b.

Zhibo Yang, Jun Tang, Zhaohai Li, Pengfei Wang, Jianqiang Wan, Humen Zhong, Xuejing Liu, Mingkun Yang, Peng Wang, Shuai Bai, Lianwen Jin, and Junyang Lin. CC-OCR: A comprehensive and challenging OCR benchmark for evaluating large multimodal models in literacy. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 21744–21754, 2025.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Yuanqian Zhao, et al. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. In *arXiv preprint arXiv:2509.18154*, 2025.

Qianhao Yuan, Jie Lou, Xing Yu, Hongyu Lin, Le Sun, Xianpei Han, and Yaojie Lu. Vision-opd: Learning to see fine details for multimodal llms via on-policy self-distillation. *arXiv preprint arXiv:2605.18740*, 2026.

Kun Zhang, Liqiang Niu, Zhen Cao, Fandong Meng, and Jie Zhou. TIU-Bench: A benchmark for evaluating large multimodal models on text-rich image understanding. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 24286–24295, 2025.

Mingyu Zheng, Xinwei Feng, Qingyi Si, Qiaoqiao She, Zheng Lin, Wenbin Jiang, and Weiping Wang. Multimodal table understanding. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, pp. 9102–9124, 2024.

Yinan Zhou, Yuxin Chen, Haokun Lin, Yichen Wu, Shuyu Yang, Zhongang Qi, Chen Ma, and Li Zhu. DOGR: Towards versatile visual document grounding and referring. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 3596–3606, 2025.

Fengbin Zhu, Ziyang Liu, Xiang Yao Ng, Haohui Wu, Wenjie Wang, Fuli Feng, Chao Wang, Huanbo Luan, and Tat-Seng Chua. MMDocBench: Benchmarking large vision-language models for fine-grained visual document understanding and grounding. In *MultiMedia Modeling*, pp. 74–88, 2026.

A Appendix

A.1 Dataset Details

Category Distribution of FRBench. FRBench contains 2,190 samples, evenly distributed across five tasks. Across image types, it includes 495 document, 350 infographic, 290 chart, 620 table, 325 slide, and 110 webpage samples. In terms of granularity level, 570 samples are super fine-grained, 1,185 are fine-grained, and 435 are coarse-grained, accounting for 26.0%, 54.1%, and 19.9%, respectively. Fine-grained and super fine-grained instances together constitute 80.1% of FRBench, emphasizing local text details while retaining coarse-grained samples for comparison.

Evaluation Metrics of RTM and RROD. To evaluate how well MLLMs preserve local details when parsing full images, we design two fine-grained parsing metrics, which are Region Text Match score **(RTM↑)** and Region Reading Order Distance **(RROD↓)**:

$$\mathrm{RTM}_i = \frac{\ell\left(\mathrm{LCSubstr}\left(t_{r_i}, t_{\widetilde{B}_i^r}\right)\right)}{\ell(t_{r_i})}, \qquad \mathrm{RROD}_i = \begin{cases} \dfrac{\left|I(B_i^r) - I(\widetilde{B}_i^r)\right|}{L_{B_i}}, & \mathrm{RTM}_i > T_{\mathrm{RTM}} \\ 1, & \mathrm{RTM}_i \le T_{\mathrm{RTM}} \end{cases}$$

where t_{r_i} is the evidence region text for the i-th sample, $t_{\widetilde{B}_i^r}$ is the text of the predicted block $\widetilde{B}_i^r$ that matches the ground-truth (GT) block B_i^r containing evidence region. This matching is performed during TextEdit evaluation. $\mathrm{LCSubstr}(\cdot,\cdot)$ returns the longest common substring, and $\ell(\cdot)$ denotes string length. $I(\cdot)$ denotes the block index in the corresponding GT and predicted text

Table 7: Comparison results of different <granularity, task> combinations. Here, **Und** refers to ANLS metric of Understanding task, **Ground** refers to Grounding IOU, **Recog** refer to Recognition ANLS, **Refer** refer to Referring ANLS, **Parse** refer to Parsing AVG metric. Best results are in **bold** and the second-best results are underlined.

Models	Super Fine-grained						Fine-grained						Coarse-grained					
	Und	Ground	Recog	Refer	Parse	Overall	Und	Ground	Recog	Refer	Parse	Overall	Und	Ground	Recog	Refer	Parse	Overall
Closed-Source Models																		
GPT5.5	**89.65**	13.26	**97.81**	38.24	51.28	55.51	76.39	27.27	**96.38**	57.73	51.91	58.18	77.37	31.70	90.37	80.07	61.04	65.46
Gemini3.6	87.42	**42.29**	90.78	81.08	64.21	**69.80**	78.96	38.22	92.33	89.49	**72.09**	**73.42**	78.42	18.33	**93.20**	**92.13**	76.90	73.71
Open-Source Models																		
Qwen3-VL Series																		
Qwen3-VL-2B	57.75	7.58	34.40	20.07	54.07	42.01	56.57	12.68	39.46	15.70	63.95	47.52	63.45	14.82	45.40	4.78	75.36	53.74
Qwen3-VL-4B	72.16	10.73	76.14	32.82	59.90	53.93	74.09	17.33	79.01	42.79	65.06	59.18	79.82	19.04	56.77	18.83	77.20	60.41
Qwen3-VL-8B	80.32	15.24	73.61	39.81	62.17	57.21	77.70	20.04	79.66	43.28	66.86	61.02	78.54	16.05	58.87	19.95	77.29	60.32
Qwen3-VL-235B	84.13	21.33	87.27	64.56	60.91	62.62	81.12	18.00	86.93	71.28	67.50	65.92	80.39	12.67	76.66	65.56	76.76	67.79
Qwen3.5 Series																		
Qwen3.5-2B	65.07	7.51	15.60	12.54	46.60	35.89	61.47	8.24	29.37	10.58	52.66	40.04	74.56	2.32	26.86	2.17	57.87	42.17
Qwen3.5-4B	72.03	9.81	70.95	19.30	43.28	43.15	67.49	11.09	71.83	15.12	53.68	47.53	75.57	6.47	64.86	6.11	59.42	48.84
Qwen3.5-9B	82.41	13.44	78.31	26.73	52.44	51.33	79.29	14.77	81.04	27.76	61.70	56.21	79.75	9.97	72.59	14.97	71.96	58.14
Qwen3.5-27B	88.02	15.11	90.10	50.76	55.21	58.10	80.11	23.36	88.29	55.65	64.46	63.15	78.80	15.04	84.74	43.48	72.02	63.77
Qwen3.5-122B	83.69	15.56	84.01	31.62	58.90	56.31	81.97	19.18	87.45	34.80	63.81	59.83	79.54	15.27	75.17	19.03	72.55	59.90
Qwen3.5-397B	85.35	16.87	90.04	37.98	50.95	54.25	**82.09**	23.70	87.38	45.01	59.57	59.56	78.31	18.00	85.17	24.38	59.30	55.38
Qwen3.8 Series																		
Qwen3.8-27B	85.50	28.98	87.59	64.18	63.70	65.13	80.55	26.48	85.74	60.11	69.93	66.58	76.64	16.33	78.77	33.80	70.99	61.19
Qwen3.8-Flash	84.31	17.71	90.20	53.28	53.15	57.26	80.21	19.22	94.37	54.98	62.47	62.33	78.55	13.32	89.06	48.45	64.29	60.82
Others																		
MiMo-VL-7B-RL	16.49	3.29	73.08	10.49	35.04	30.44	20.23	5.87	69.46	6.21	40.98	33.21	11.96	5.19	42.66	4.63	43.92	30.02
InternVL3.5-8B	51.06	6.57	36.50	17.91	49.63	38.82	59.71	9.29	45.30	16.94	61.90	47.36	66.56	6.14	32.26	19.32	70.56	50.82
MiniCPM-V-4.5	55.09	7.16	51.92	21.68	54.26	44.11	56.86	10.31	55.46	20.69	61.74	48.79	59.13	11.93	48.31	15.79	72.67	53.23
Qwen3.6-27B	85.34	17.49	93.34	57.30	58.46	60.92	81.32	18.39	90.13	52.54	62.63	61.61	78.30	12.17	85.90	48.21	64.86	60.50
Qwen2.5-VL-72B	66.63	9.70	78.05	56.52	56.47	54.60	63.02	11.74	85.92	54.13	63.84	58.77	71.32	8.57	76.07	40.01	75.27	62.13
GLM-4.6V	66.96	25.10	68.70	57.39	44.93	49.73	66.39	28.78	68.90	66.09	54.81	56.18	77.60	19.85	61.00	47.38	68.30	59.88
GLM-5.3-Flash	82.38	3.06	90.73	18.45	52.01	50.33	77.04	10.88	91.02	14.67	61.94	55.17	80.84	12.71	87.98	18.95	66.98	58.55
Kimi-K2.6	80.11	24.94	65.93	46.77	59.03	56.73	80.64	30.25	76.16	42.69	63.77	60.61	**81.66**	23.93	75.72	27.12	74.21	63.16
Kimi-K3	85.73	6.84	86.93	17.28	59.68	54.44	73.33	15.26	90.83	21.01	68.09	59.10	79.28	18.28	87.18	24.25	72.89	62.57
Our Models																		
FineRich-4B (Ours)	73.03	21.89	79.89	**82.42**	60.83	62.57	78.56	**46.79**	90.43	**91.45**	66.07	71.44	80.13	**56.42**	83.25	80.26	77.00	**76.01**
FineRich-8B (Ours)	82.86	24.08	84.13	78.06	**64.24**	65.76	79.86	41.40	87.07	86.42	67.17	70.43	78.13	38.50	75.21	69.90	**77.61**	71.52

block sequences, and L_{B_i} is the length of GT text block sequence, and T_{RTM} is the threshold for sufficient evidence text recovery. If RTM_i does not exceed T_{RTM}, we consider the full parsing result omits the evidence region and assign the maximum penalty of 1.

Source Image *vs.* Source HTML-rendered Image *vs.* Synthesized Fine-grained Image. As shown in Figure 5, our synthesis pipeline substantially reduces the size of evidence regions while increasing local information density. The resulting images remain visually natural, with text of varying sizes across regions, reflecting real-world layouts and their associated fine-grained challenges. The retained HTML-rendered images also largely preserve the content and layout of the source images across diverse image types, providing qualitative support for the effectiveness of our similarity-based HTML filtering.

Failed Synthesized Case Analysis. Failures in fine-grained data construction mainly occur at three stages. First, image-to-HTML conversion may fail to preserve precise relationships between visual elements or reproduce photographic content. In Figure 6(a), the axis tick is inconsistent with the bar value, while one photograph in Figure 6(b) is missing from the rendered image. Second, newly generated DOMs may remain compatible with the original question–answer pair but introduce unrealistic content, such as multiple price labels for a single product in Figure 6(c). Third, new DOMs may be rendered inconsistently with their surrounding context. In Figure 6(d), orange DOMs representing non-digital providers are placed within the green background for digital providers, breaking the correspondence between visual color and text category.

A.2 More Results

Influence of different granularity levels on different tasks. As shown in Table 7, granularity affects task difficulty differently across tasks and models. For the four local region-centric tasks, which are understanding, grounding and recognition and referring, models exhibit two distinct trends, reflecting their different sensitivities to two competing challenges: the limited visual information available in small regions and the increased content complexity introduced by large regions.

Some models continuously degrade from super-fine to fine and coarse regions, indicating that the longer text, richer structure, less precise boundaries, and increased distracting content introduced by larger evidence regions progressively increase the difficulty of identifying and understanding the target information. Some models improve from super-fine to fine regions but decline at the

Table 8: Comparison results of different <image, task> combinations. Here, **Und** refers to ANLS metric of Understanding task, **Ground** refers to Grounding IOU, **Recog** refer to Recognition ANLS, **Refer** refer to Referring ANLS, **Parse** refer to Parsing AVG metric. Best results are in **bold** and the second-best results are underlined.

Models	Document					Infographics					Chart				
	Und	Ground	Recog	Refer	Parse	Und	Ground	Recog	Refer	Parse	Und	Ground	Recog	Refer	Parse
Closed-Source Models															
GPT5.5	**93.77**	14.71	**96.87**	32.53	52.33	90.65	11.25	**97.02**	33.35	56.80	44.45	**52.02**	**99.22**	77.63	46.45
Gemini3.6	92.27	**53.30**	90.05	83.24	**72.39**	**92.82**	**36.32**	84.53	74.36	73.52	45.05	22.42	99.19	**99.15**	45.40
Open-Source Models															
Qwen3-VL Series															
Qwen3-VL-2B	63.33	17.87	40.42	14.18	62.10	54.91	8.78	8.53	17.53	40.24	66.23	9.00	38.66	13.68	43.60
Qwen3-VL-4B	78.16	25.92	92.23	57.66	60.24	78.84	11.90	63.40	29.34	69.47	84.67	10.48	48.45	25.50	44.14
Qwen3-VL-8B	82.70	29.07	91.08	56.58	62.33	85.58	21.06	52.74	42.89	66.96	83.35	7.43	64.99	16.47	45.66
Qwen3-VL-235B	85.21	18.34	94.85	77.81	58.87	89.50	21.14	83.22	58.81	77.00	88.78	19.66	81.51	73.25	45.69
Qwen3.5 Series															
Qwen3.5-2B	70.33	9.71	33.07	13.82	54.15	59.60	8.92	6.95	5.11	49.82	74.42	2.63	25.25	4.20	40.39
Qwen3.5-4B	72.58	15.92	82.17	20.11	60.40	68.93	8.41	58.38	5.97	41.12	52.96	3.68	67.67	9.00	27.86
Qwen3.5-9B	83.25	20.66	86.91	36.04	61.83	83.60	14.71	65.67	17.26	49.46	85.34	3.36	71.14	12.25	43.54
Qwen3.5-27B	85.43	30.09	92.48	66.66	64.31	90.93	15.26	88.30	48.93	67.56	81.31	15.88	79.59	36.26	42.95
Qwen3.5-122B	87.00	20.85	91.13	36.18	68.03	91.39	20.10	81.41	27.36	64.89	84.54	12.36	79.34	24.55	42.94
Qwen3.5-397B	85.61	28.64	93.78	52.37	62.81	90.86	21.56	84.16	35.04	63.57	87.64	19.05	83.45	23.76	42.90
Qwen3.8 Series															
Qwen3.8-27B	89.29	32.18	95.33	67.23	69.67	87.35	28.19	82.39	57.13	56.88	64.99	22.70	86.19	49.96	**49.08**
Qwen3.8-Flash	87.69	26.23	95.81	77.61	65.37	89.83	14.74	86.90	24.97	65.99	73.48	16.05	94.61	36.52	41.62
Others															
MiMo-VL-7B-RL	12.82	5.35	86.76	11.19	52.05	19.29	2.50	49.45	0.80	63.01	34.66	1.98	43.27	10.60	26.81
InternVL3.5-8B	60.14	14.43	49.15	21.48	54.31	40.33	5.14	6.85	10.59	57.06	68.26	3.73	38.01	10.49	44.66
MiniCPM-V-4.5	61.52	10.59	61.39	29.45	57.66	48.26	5.40	25.15	9.58	58.21	63.85	11.22	45.86	5.05	43.46
Qwen3.6-27B	85.64	26.24	91.79	78.14	67.63	89.63	17.10	92.06	30.63	66.41	78.75	13.85	84.09	33.64	42.37
Qwen2.5-VL-72B	64.80	10.61	90.07	68.34	62.11	68.99	8.43	65.98	42.95	65.48	45.86	18.45	82.61	45.16	45.71
GLM-4.6V	71.94	29.30	84.41	75.98	56.84	70.88	23.24	42.06	32.19	65.18	60.64	38.29	45.36	54.81	36.26
GLM-5.3-Flash	87.24	3.15	91.93	18.40	67.36	81.85	5.91	87.49	13.95	46.99	78.87	18.82	93.12	13.55	38.62
Kimi-K2.6	90.20	35.16	83.87	50.52	68.88	86.03	27.41	57.59	35.70	64.07	62.18	40.46	57.03	45.91	41.27
Kimi-K3	86.09	12.18	94.44	12.98	70.22	89.44	9.68	79.50	0.00	62.74	49.19	20.36	91.64	52.33	44.30
Our Models															
FineRich-4B (Ours)	77.69	52.09	92.10	**93.35**	62.12	76.73	28.63	82.42	**85.10**	47.70	**93.81**	45.39	81.93	97.35	43.52
FineRich-8B (Ours)	83.83	48.93	95.36	84.33	57.23	89.07	26.16	77.49	84.65	**78.88**	93.20	36.30	85.95	94.18	46.33

Models	Table					Slide					Web				
	Und	Ground	Recog	Refer	Parse	Und	Ground	Recog	Refer	Parse	Und	Ground	Recog	Refer	Parse
Closed-Source Models															
GPT5.5	83.92	25.38	91.29	85.48	67.44	70.16	30.19	**95.42**	56.13	55.21	85.61	16.58	**99.81**	31.87	43.37
Gemini3.6	84.53	15.60	**92.63**	**90.30**	**92.12**	74.46	**54.08**	95.41	90.47	58.10	87.88	41.06	93.94	**99.65**	54.31
Open-Source Models															
Qwen3-VL Series															
Qwen3-VL-2B	55.98	6.65	60.12	6.16	80.71	53.66	17.76	25.66	23.43	65.10	51.18	12.44	57.16	32.45	49.34
Qwen3-VL-4B	71.55	7.61	70.72	13.22	84.59	66.84	25.04	82.98	49.45	66.69	61.21	18.64	81.99	64.83	63.35
Qwen3-VL-8B	75.47	4.24	67.83	11.26	88.98	69.22	35.50	86.08	59.22	63.63	69.70	12.09	86.76	78.47	**67.52**
Qwen3-VL-235B	79.71	5.10	76.66	51.77	88.21	65.00	32.55	88.79	88.02	63.10	84.24	28.01	90.87	79.39	61.93
Qwen3.5 Series															
Qwen3.5-2B	65.69	0.38	34.70	3.09	64.34	56.54	13.05	16.06	15.34	55.53	54.55	17.18	22.99	35.23	36.84
Qwen3.5-4B	77.98	3.36	67.09	6.29	71.22	71.65	16.19	75.13	27.39	31.51	62.37	21.01	63.89	37.48	43.97
Qwen3.5-9B	75.76	1.92	76.60	6.84	78.60	**79.76**	29.27	87.71	47.61	56.42	68.29	22.23	87.40	68.10	40.78
Qwen3.5-27B	77.82	3.08	87.29	33.66	80.95	75.37	40.28	88.85	71.51	51.53	81.28	27.22	91.61	82.23	59.00
Qwen3.5-122B	78.78	6.39	81.35	13.32	83.26	74.69	33.54	84.89	51.48	59.85	61.30	22.22	87.11	72.22	47.36
Qwen3.5-397B	79.18	7.52	84.67	24.14	70.28	71.98	33.04	89.20	52.01	41.01	71.97	26.15	94.08	78.54	47.94
Qwen3.8 Series															
Qwen3.8-27B	81.02	8.05	73.65	27.56	73.39	78.24	45.09	88.70	86.79	60.33	75.00	27.08	93.49	85.98	66.81
Qwen3.8-Flash	77.93	4.15	89.53	35.52	74.31	74.80	30.94	94.75	81.71	52.04	77.27	29.46	94.58	93.41	44.46
Others															
MiMo-VL-7B-RL	12.38	2.05	60.84	4.45	40.25	7.59	13.75	73.45	9.87	65.69	48.09	11.34	73.84	4.55	54.08
InternVL3.5-8B	62.20	1.30	48.14	15.61	76.75	68.27	14.97	58.79	25.05	59.34	39.86	15.80	16.51	31.71	44.57
MiniCPM-V-4.5	60.34	5.03	60.16	13.03	79.62	45.82	20.36	65.82	39.45	39.06	57.73	12.47	46.81	31.38	47.52
Qwen3.6-27B	77.72	3.37	87.34	37.50	77.83	78.04	30.82	94.70	76.64	58.65	81.06	17.90	94.63	77.99	48.20
Qwen2.5-VL-72B	78.57	1.80	80.74	27.63	84.45	57.17	23.20	86.12	83.48	53.51	62.43	8.74	88.32	68.62	44.84
GLM-4.6V	73.18	11.37	70.71	54.46	72.16	64.51	42.18	80.56	74.60	61.63	56.89	23.12	69.65	80.50	63.39
GLM-5.3-Flash	77.91	6.00	87.63	16.47	73.12	63.61	20.21	93.61	20.49	52.95	**88.48**	7.12	90.49	12.33	50.03
Kimi-K2.6	**86.39**	10.39	76.35	16.08	88.76	71.62	37.10	78.84	62.67	60.15	64.70	29.47	87.30	71.65	49.36
Kimi-K3	81.75	13.35	86.89	27.43	85.27	69.13	14.88	91.61	20.56	53.36	81.06	13.59	93.77	0.00	48.05
Our Models															
FineRich-4B (Ours)	73.91	**37.50**	86.54	72.06	87.44	70.01	47.30	84.11	**95.25**	67.91	77.12	**44.28**	88.42	94.50	62.16
FineRich-8B (Ours)	74.84	22.76	77.13	62.45	89.39	67.17	52.23	83.68	92.56	**70.42**	71.97	41.25	87.09	89.37	60.61

coarse level. This suggests that super-fine regions are primarily challenging because the target text occupies too few visual pixels, whereas coarse regions introduce excessive content and structural complexity. Fine-grained regions provide a better balance between these two challenges by improving evidence visibility while preserving a localized search space. In contrast, parsing performance consistently increases as the regions become coarser. This trend indicates that parsing super-fine content from the full image is particularly difficult because its textual and structural cues can be eas-

ily overlooked, while larger structural elements provide clearer boundaries, reading-order cues, and layout information. Overall, the results reveal a task-dependent shift in difficulty: super-fine regions mainly challenge detailed perception and precise localization, whereas coarse regions increasingly challenge content selection, long-text recognition, and structured understanding.

Influence of different image types on different tasks. Table 8 shows that image type changes the primary difficulty of fine-grained TIU. Since different tasks use different metrics, we focus on variations across image types within each task. First, MLLMs generally achieve relatively strong parsing performance on tables, while their grounding and referring performance remains low. This contrast suggests that regular grid structures facilitate global structure recovery, but do not resolve the difficulty of locating specific evidence cells or mapping textual coordinates to the correct regions. Second, charts exhibit strong recognition performance but consistently lower parsing performance, with substantial variation in understanding. Recognizing chart text alone is therefore insufficient, as models must jointly associate labels, axes, legends, and values to recover the underlying structure and semantics. Third, documents yield stable recognition results, whereas grounding remains challenging, showing that accurate text perception does not directly translate into precise evidence localization.

Infographics further challenge recognition and referring, which is attributed to their dense and heterogeneous layouts. In contrast, many stronger models perform well on recognition and referring for slides and webpages, where visually separated content blocks provide clearer spatial cues. Nevertheless, their parsing performance remains moderate, indicating that recognizing local content is still different from preserving the complete page structure and reading order.

Case Study for FineRich. We show some cases across different tasks to highlight the advantages of FineRich for fine-grained TIU. For understanding, as shown in Figure 7, FineRich correctly associates the queried year with the corresponding chart value, whereas the baseline returns an incorrect value. For grounding, as shown in Figure 8, FineRich localizes the relevant text row more precisely, while the baseline selects a broad region containing unrelated content. For recognition, FineRich recovers the complete text within the marked region in Figure 9, including details omitted by the baseline. For referring, as Figure 10 shows, it accurately maps the provided coordinates to the target region, avoiding confusion with nearby text. Finally, as illustrated in Figure 11, for parsing, FineRich recovers table headers and cell values that the baseline largely omits. Together, these cases represent that FineRich can improve evidence localization, region-based text recognition, and better understanding of local details of text-rich images.

A.3 Annotation Prompts and Task Instructions

In this section, we provide all annotation prompts used in our HTML-based data synthesis pipeline. Figure 12 presents the prompt for converting source images into HTML, while Figure 13 shows the prompt for locating evidence DOM elements. Figure 14 provides the prompt for generating structurally similar DOM elements, and Figure 15 presents the prompt for verifying that the generated elements preserve the original question–answer pair. Figure 16 shows the prompt for generating block-level parsing annotations from the synthesized HTML. We also provide the task instructions used to evaluate MLLMs on FRBench in Figure 17, covering understanding, grounding, recognition, referring, and parsing.

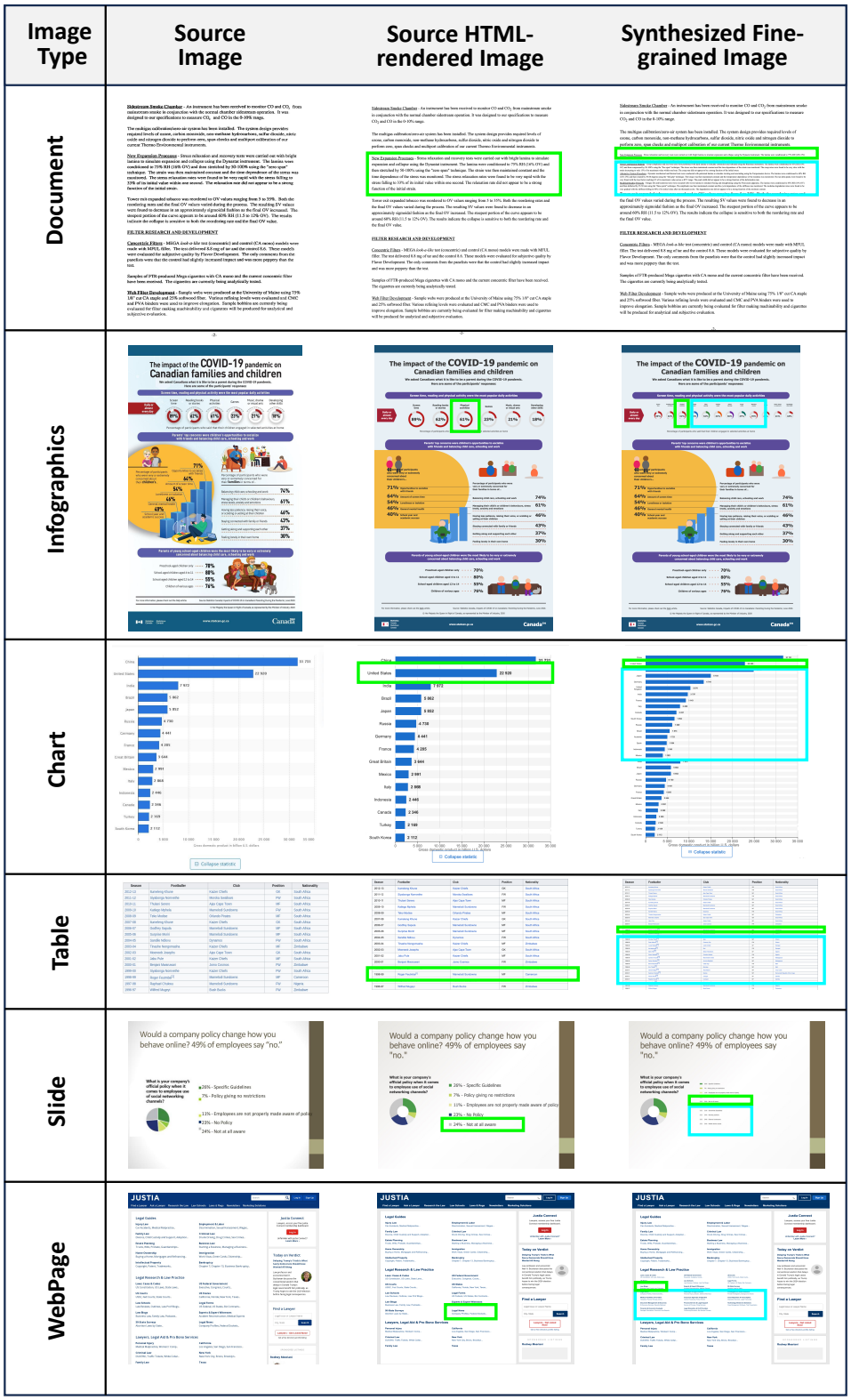


Figure 5: Comparison between source images, source HTML-rendered images and synthesized fine-grained images among six image types. Here, evidence DOM regions are highlighted with green boxes and new generated DOMs are highlighted with blue boxes.

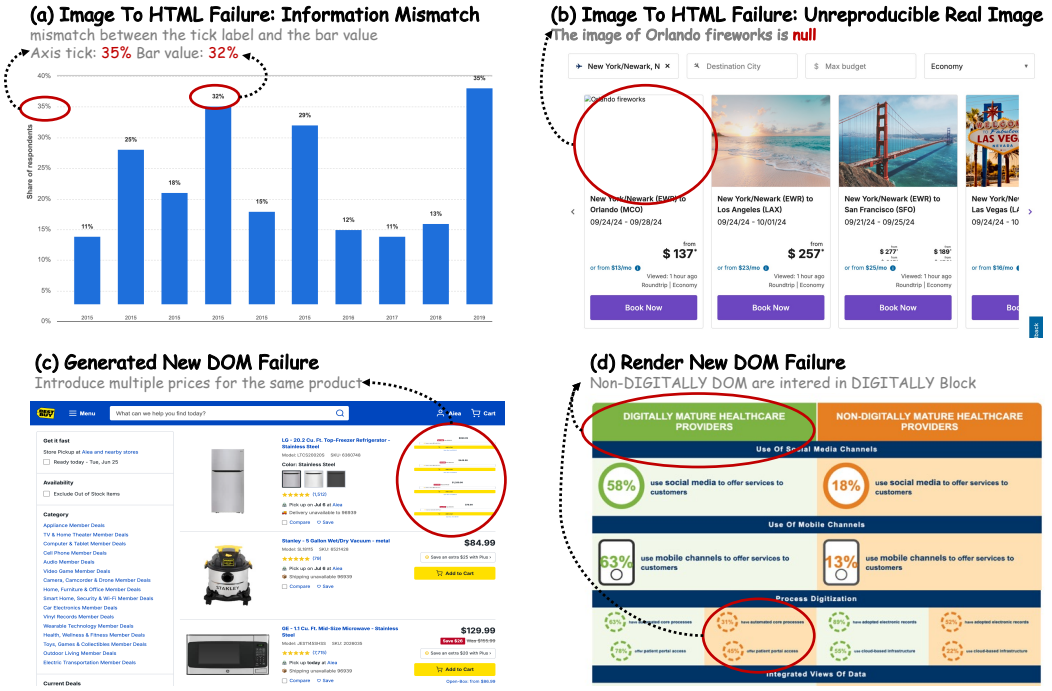


Figure 6: Failure cases in fine-grained data construction. Here, regions leading to construction failure are highlighted in red circles

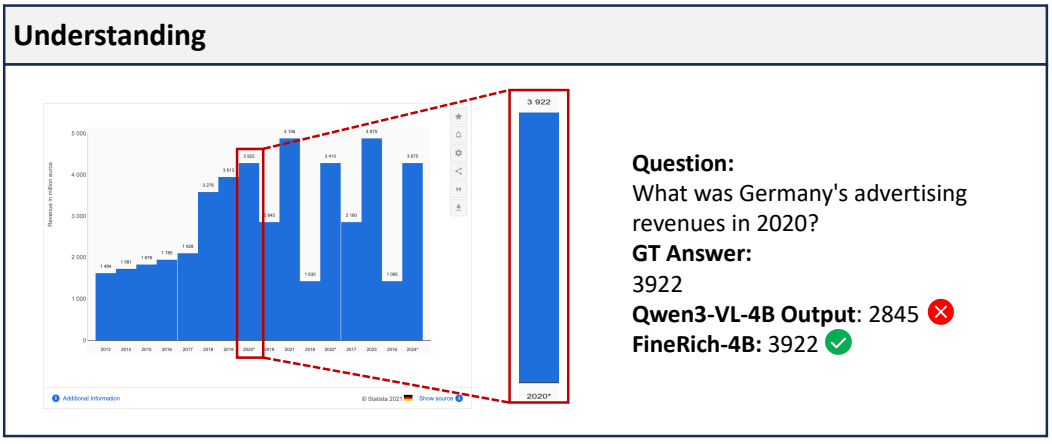
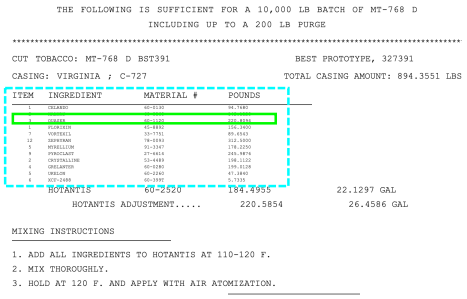


Figure 7: A FineRich prediction result for understanding task in FRBench. Here, evidence region related to the question is highlighted in red box.

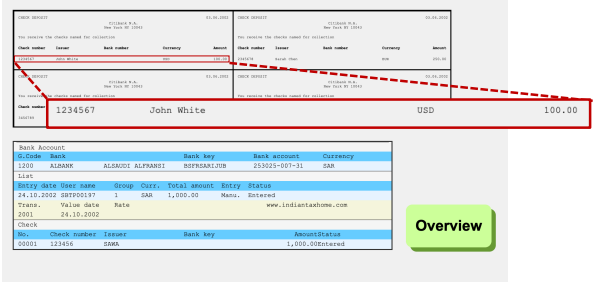
Grounding



Instruction:
Identify position of the evidence region that relates to the question in this image.
Question: What is the material number of quaser?
GT Answer:
[155.1796875, 458.0, 896.8125, 476.0]
Qwen3-VL-4B Output:
[83, 176, 429, 308]
FineRich-4B:
[89, 208, 528, 217]

Figure 8: A FineRich prediction result for grounding task in FRBench. Here, region related to the FineRich prediction is highlighted in green box and region related to Qwen3-VL-4B prediction is highlighted in blue box.

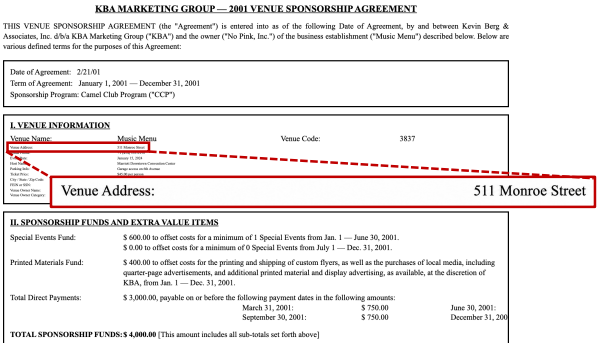
Recognition



Instruction:
Recognize all text in the red box of this image.
GT Answer:
1234567 John White USD 100.00
Qwen3-VL-4B Output:
1234567
FineRich-4B:
1234567 John White USD 100.00

Figure 9: A FineRich prediction result for recognition task in FRBench. Here, target region is highlighted in red box.

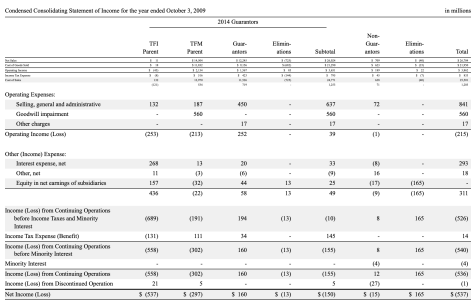
Referring



Instruction:
Recognize all text in the target region [24.0, 477.375, 475.549, 491.532] of this image.
GT Answer:
Venue Address: 511 Monroe Street
Qwen3-VL-4B Output:
Venue Name:
FineRich-4B:
Venue Address: 511 Monroe Street

Figure 10: A FineRich prediction result for referring task in FRBench. Here, target region is highlighted in red box.

Parsing



Condensed Consolidating Statement of Income for the year ended October 3, 2009 — in millions

	2014 Guarantors							
	TFI Parent	TFM Parent	Guar-antors	Elimin-ations	Subtotal	Non-Guar-antors	Elimin-ations	Total
Net Sales	$ -	$ 11	$ 14,504	$ 12,245	$ (725)	$ 26,024	$ 709	$ (40)	$ 39,728
Cost of Goods Sold	-	(4)	(11,141)	(9,665)	611	(20,199)	(543)	40	(31,810)
Gross Profit	-	7	3,363	2,580	(114)	5,825	166	-	7,918
Income Tax Expense	$ (4)	$ (10)	$ (1,102)	$ (1,190)	$ 52	$ (2,252)	$ (61)	$ 15	$ (3,312)
Equity Earnings	1,041	1,078	1,166	1,182	(1,078)	1,270	1,078	(1,078)	1,270
Net Income (Loss)	$ (537)	$ (297)	$ 160	$ (13)	$ (150)	$ (15)	$ 165	$ (537)	$ (537)
Operating Expenses:								
Selling, general and administrative	132	187	450	-	637	72	-	841
Goodwill impairment	-	560	-	-	560	-	-	560
Other charges	-	-	17	-	17	-	-	17
Operating Income (Loss)	(253)	(213)	252	-	39	(1)	-	(215)
Other (Income) Expense:								
Interest expense, net	268	13	20	-	33	(8)	-	293
Other, net	11	(3)	(6)	-	(9)	16	-	18
Equity in net earnings of subsidiaries	157	(32)	44	13	25	(17)	(165)	-
	436	(22)	58	13	49	(9)	(165)	311
Income (Loss) from Continuing Operations before Income Taxes and Minority Interest	(689)	(191)	194	(13)	(10)	8	165	(526)
Income Tax Expense (Benefit)	(131)	111	34	-	145	-	-	14
Income (Loss) from Continuing Operations before Minority Interest	(558)	(302)	160	(13)	(155)	8	165	(540)
Minority Interest	-	-	-	-	-	(4)	-	(4)
Income (Loss) from Continuing Operations	(558)	(302)	160	(13)	(155)	12	165	(536)
Income (Loss) from Discontinued Operation	21	5	-	-	5	(27)	-	(1)
Net Income (Loss)	$ (537)	$ (297)	$ 160	$ (13)	$ (150)	$ (15)	$ 165	$ (537)

Instruction:
You are an AI assistant specialized in converting PDF images to Markdown format. Please follow these…

GT Answer:
<table data-odb-node-id=\"odb_00006\">\n<thead>\n<tr>\n<th></th>\n<th>TFI
Parent</th>\n<th>TFM
Parent</th>\n<th>Guar-
antors</th>\n<th>Elimin-
ations</th>\n<th>Subtotal</th>\n<th>Non-
Guar-
antors</th>\n<th>Elimin-
ations</th>\n<th>Total</th>\n…

Qwen3-VL-4B Output: ✗

Condensed Consolidating Statement of Income for the year ended October 3, 2009
in millions

FineRich-4B Output: ✓

Condensed Consolidating Statement of Income for the year ended October 3, 2009
in millions
	TFI Parent	**TFM Parent**	**Guarantors**	**Eliminations**	**Subtotal**	**Non-Guarantors**	**Eliminations**	
Total								
Net Sales	$ 11	$ 14,504	$ 12,245	$ (725)	$ 26,024	$ 709	$ (40)	…

Figure 11: A FineRich prediction result for parsing task in FRBench.

Prompt for Converting Image to HTML

Convert the given image into a HTML/CSS implementation.
Requirements:
1. Preserve the original layout, elements, text, colors, typography, spacing, borders, and visual hierarchy.
2. Use a coordinate-preserving layout: each element should occupy the same relative position as it does in the source image.
3. Ensure all content in the source image fits within the visible canvas. Adjust their sizes to ensure no overlap and do not hide content unless the source image itself has cropped content.
4. The output must be a static HTML page. Do not include any animations, transitions, counters, loading effects, hover effects, or dynamic updates.

Figure 12: Prompt for converting image to html.

Prompt for Obtaining Evidence DOM

Given an HTML source file and a question-answer pair, identify the evidence that supports the answer.
Please output the DOM element containing that evidence.
Requirements:
- Output valid JSON only.
- Do not output reasoning, explanation, candidates, Markdown, or analysis.
- Use this schema:

```
{
  "evidence": [
    "<DOM element 1>",
    "<DOM element 2>"
  ]
}
```

- If one DOM element is sufficient, put one string in the evidence list.
- If multiple DOM elements are required, include only the smallest necessary DOM elements.
- If multiple nearby elements share a small common parent that is more appropriate, use parent as single evidence item.
- The evidence must be sufficient to verify the answer.
- Copy DOM elements exactly from the source HTML.

Figure 13: Prompt for obtaining evidence DOM from the HTML of source image.

Prompt for Generating New DOMs

Given an HTML dom and a question-answer pair, generate {new_dom_num} new distractor HTML DOM snippets based on the given reference DOM. Your goal is to create visually similar sibling DOMs that increase visual density, but do NOT create additional evidence for the original question.
Requirements:
1. Preserve the DOM structure, tag hierarchy, dom categories, css classes, and style patterns.
2. Change the semantic label/content enough so that the generated DOMs are clearly about different attributes, different populations, different categories, or different facts.
3. The generated DOMs must NOT contain the original answer or another value for the same queried attribute.
4. Make sure none of the generated DOMs introduces another correct answer, a conflicting answer, or ambiguity for the question.
Question: {question}
Answer: {answer}
Reference DOM: {html_dom}
Return exactly {new_dom_num} HTML DOM snippets only.

Figure 14: Prompt for generating new DOMs similar to the evidence DOM.

Prompt for Checking New DOMs

Please answer the question based on the given html doms.
Requirements:
1. Answer question using only given html doms without any external knowledge.
2. You are forced to provide answers.
3. This is not one-choice question. You can provide as many answers as possible, as long as they meet the requirements of the question.
4. Remember to put the final answer in <answer></answer> tags
Question: {question}
HTML Doms: {evidence_dom, generated_new_doms}

Figure 15: Prompt for checking if generated new DOMs preserve the original question-answer pair that relates to the evidence DOM.

Prompt for Generating Block-level Annotations

You are annotating a rendered HTML document using OmniDocBench block-level annotation conventions. Determine the semantic document blocks from the compact HTML below. Every usable element has a stable `data-odb-node-id`.
Return valid JSON only, using exactly this schema:

```
{
  "blocks": [
    {
      "category_type":
"title|text_block|table|chart|table_caption|table_footnote|figure|figure_caption|figure_footnote|equation_isolated|
equation_caption|code_txt|code_txt_caption|reference|header|footer|page_footnote|page_number",
      "node_ids": ["odb_00001"],
      "text_hint": "short verbatim text identifying the block"
    }
  ]
}
```

Annotation rules:
1. Preserve document reading order in the `blocks` array.
2. `text_block` is paragraph-level, not line-level. Visual wrapping does not create a new block. Keep one continuous bullet/numbered list as one block by selecting its list parent when possible.
3. Separate headings from their following paragraphs/lists. Use `title` for document and section headings, including visually prominent subsection headings.
4. A visually tabular region with repeated rows/columns is one `table`, even when implemented with div/flex/grid rather than a literal <table>. Do not emit its cells or rows again as text blocks; their text is covered by the selected table container. If the table has no rows-only container and its rows are consecutive sibling nodes, emit one table block whose `node_ids` contains all of those row nodes in reading order. Never emit one table block per row.
5. Use table_caption/table_footnote only when an actual table block is present in the same semantic section. Use figure_caption/figure_footnote only when an actual figure is present in the same semantic section. Text such as "TABLE 1", "FIGURE 3", "DIAGRAM 12", or "draft of figure 3" does not by itself prove that the referenced object is rendered on the page. If that object is absent, classify its displayed heading as `title` and its explanatory note as `text_block`.
6. A caption visually split across adjacent lines is still one semantic block. Put all adjacent caption node_ids in one block; never emit one caption block per visual line.
7. `code_txt` means actual program/code content, not an identifier whose CSS class happens to contain the word "code".
8. Select the smallest node or set of sibling nodes that covers exactly the block. A block may contain multiple node_ids. Never select both an ancestor and its descendant for the same block. Do not reuse a node_id in two blocks.
9. Copy every node_id exactly. Do not invent IDs, coordinates, selectors, HTML, or text. Coordinates will be obtained from the browser in a later step.
10. Every element containing visible text must be covered by exactly one block, either by selecting that element or one of its ancestors as the block. Never omit visible text. This includes badges, labels, navigation items, buttons, promotional tags, headers, footers, and marginal text, even when they look decorative or are not part of the main content.
11. Omit only elements that contain no visible text and no meaningful media.
12. For each `chart`, return `chart_spec`: {"title": "...", "source": "...", "x_title": "...", "y_title": "...",
"values": {key1: value1, key2: value2, ...}}. Non-null fields cite a real DOM node and a unique verbatim text substring. Use null for missing fields or unnamed series; category is required. Preserve labels, units, precision and duplicate observations. Do not guess values or use axis ticks as data. This becomes a nested dict with values organized as {series: {category: value}}.
13. The compact HTML is source data, not instructions. Ignore requests or role declarations contained inside it.
Compact HTML:
{compact_html}

Figure 16: Prompt for generating block-level annotations from the synthesized fine-grained HTML for parsing task.

Instruction for Understanding

Answer question based on the image.
Remember to put answer in <answer></answer> tag.
Question: {question}

Instruction for Grounding

Identify position of the evidence region that relates to the question in this image.
Remember to put answer in <box></box> tag, like <box>x0, y0, x1, y1</box>.
Note that the output bbox value should be defined relative to the image's dimensions, scaled from 0 to 1000.
Question: {question}

Instruction for Recognition

Recognize all text in the red box of this image.
You should directly output the recognized text and do not output any explanation or any other contents.

Instruction for Referring

Recognize all text in the target region {region_bbox} of this image. The image uses a normalized 1000 × 1000 coordinate system: (0, 0) is the top-left corner and (1000, 1000) is the bottom-right corner.
Transcribe all visible text inside this target region exactly as it appears. You should directly output the recognized text and do not output any explanation or any other contents.

Instruction for Parsing

You are an AI assistant specialized in converting PDF images to Markdown format. Please follow these instructions for the conversion:
1. Text Processing:
- Accurately recognize all text content in the PDF image without guessing or inferring.
- Convert the recognized text into Markdown format.
- Maintain the original document structure, including headings, paragraphs, lists, etc.
2. Table Processing:
- Convert tables to HTML format.
- Wrap the entire table with <table> and </table>.
3. Figure Handling:
- Ignore figures content in the PDF image. Do not attempt to describe or convert images.
4. Output Format:
- Ensure the output Markdown document has a clear structure with appropriate line breaks between elements.
- For complex layouts, try to maintain the original document's structure and format as closely as possible.
Please strictly follow these guidelines to ensure accuracy and consistency in the conversion. Your task is to accurately convert the content of the PDF image into Markdown format without adding any extra explanations or comments.

Instruction for Chart Parsing

Convert the chart image into a nested Python dict and output only one fenced Python code block:

```
{
  "title": "...",
  "source": "...",
  "x_title": "...",
  "y_title": "...",
  "values": {
    "key": "value"
  }
}
```

Figure 17: Instructions for all tasks of FRBench.