HighlightBench: Benchmarking and Diagnosing Markup-Driven Table Reasoning in Scientific Documents

Lexin Wang[1,2,3], Shenghua Liu[1,2,3⋆], Yiwei Wang[4], Yujun Cai[5], Yuyao Ge[1,2,3], Jiayu Yao[1,2,3], Jiafeng Guo[1,2,3], Xueqi Cheng[1,2,3]

[1] State Key Laboratory of AI Safety, China
[2] Institute of Computing Technology, Chinese Academy of Sciences, China
[3] University of Chinese Academy of Sciences, China
[4] University of California, Merced, USA
[5] University of Queensland, Australia
https://highlightbench.netlify.app/
lizzie011@163.com, liushenghua@ict.ac.cn

Abstract. Visual markups such as highlights, underlines, and bold text are common in table-centric documents. Although multimodal large language models (MLLMs) have made substantial progress in document understanding, their ability to treat such cues as explicit logical directives remains under-explored. More importantly, existing evaluations cannot distinguish whether a model fails to see the markup or fails to reason with it. This creates a key blind spot in assessing markup-conditioned behavior over tables. To address this gap, we introduce HighlightBench, a diagnostic benchmark for markup-driven table understanding that decomposes evaluation into five task families: Markup Grounding, Constrained Retrieval, Local Relations, Aggregation & Comparison, and Consistency & Missingness. We further provide a reference pipeline that makes intermediate decisions explicit, enabling reproducible baselines and finer-grained attribution of errors along the perception-to-execution chain. Experiments show that even strong models remain unstable when visual cues must be consistently aligned with symbolic reasoning under structured output constraints.

Keywords: Diagnostic Benchmark · Tabular Document Understanding · Multimodal Large Language Models · Structured Reasoning

1 Introduction

In academic papers, technical reports, and data analysis workflows, tables are often annotated with visual markups such as highlights, underlines, bold text, and arrows to emphasize key information. Reading such tables involves more than identifying values or parsing row–column structure: these markups are

⋆ Corresponding author.

routinely used as functional cues that determine which entries should be retrieved, compared, filtered, or verified. For multimodal large language models (MLLMs) [1,8,28], this setting is fundamentally different from standard TableQA and broader multimodal capability evaluation [13,19,23,39]. Beyond recognizing table content, a model must correctly bind a queried visual cue to the intended structural unit, translate the cue into a constraint that is executable over table structure, and preserve this constraint throughout downstream reasoning.

A concrete gap emerges in practice: recent studies report that even strong models can produce plausible or correct answers without faithfully grounding the queried cue to the intended table region, especially when the cue is weak, ambiguous, or absent [40]. In these cases, correctness does not certify cue-conditioned reasoning, and answer-only success can mask shortcut behaviors that bypass the cue. This mismatch motivates evaluations that disentangle cue perception from cue-conditioned execution, making markup-conditioned behavior measurable rather than implicit.

Recent multimodal table and document benchmarks have substantially improved realism and reasoning coverage [31, 34, 38, 41]. However, strong performance on these benchmarks still does not tell us whether a model truly uses embedded markups as reasoning constraints. It may instead arrive at the correct answer by exploiting dataset regularities, priors, or heuristics that are largely independent of the markup. This ambiguity is the core blind spot: answer-centric scores can conflate qualitatively different behaviors—failing to notice the markup, noticing it but mis-binding it to structure, or answering correctly while effectively ignoring it—and thus systematically misestimate capability in markup-sensitive settings.

To address this gap, we introduce **HighlightBench**, a diagnostic benchmark for markup-conditioned reasoning over tables. HighlightBench decomposes markup-conditioned table reasoning into five automatically scorable task families: Markup Grounding, Constrained Retrieval, Local Relations, Aggregation & Comparison, and Consistency & Missingness. This design separates grounding, constraint usage, and downstream execution, enabling fine-grained failure localization beyond end-to-end answer accuracy.

HighlightBench combines expert-collected real-world samples with controllable synthetic instances. The real-world subset contains markup-rich tables selected from scientific papers and paired with manually written questions, preserving natural markup patterns and realistic table irregularities. The synthetic subset is generated from structured templates, enabling controlled difficulty, reproducible settings, and counterfactual samples created by varying markup conditions under fixed table content.

We also provide a reference pipeline to support reproducible baselines and interpretable diagnostics. Given an input table image and its associated question, the pipeline produces a structured intermediate representation and a schema-constrained output, together with intermediate signals that expose where failures occur along the processing chain. This makes it possible to compare end-to-end

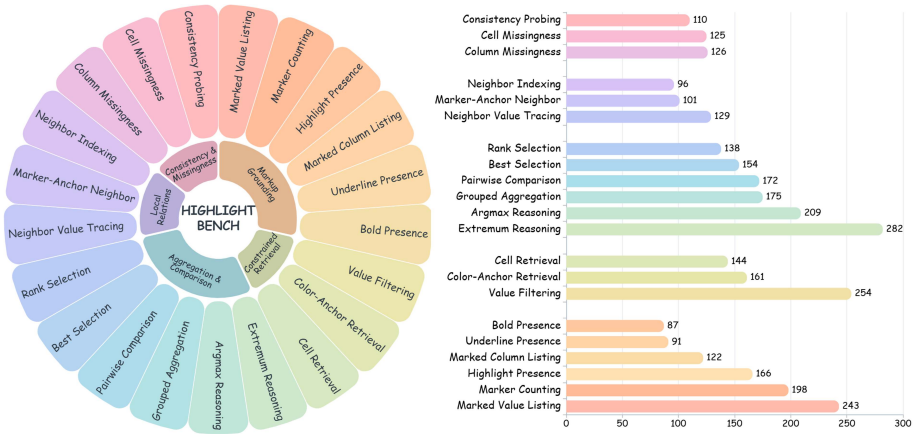


Fig. 1: Overview of **HighlightBench**. The benchmark contains five complementary task families with question counts. This design gives transparent coverage of markup-conditioned reasoning.

model behavior with a more inspectable alternative and to attribute errors to specific stages.

Systematic evaluation on HighlightBench shows that current multimodal models still exhibit persistent weaknesses in markup alignment and constraint-aware reasoning. These weaknesses become especially visible when accurate grounding, correct computation, and format-compliant outputs are required simultaneously. In contrast, the reference pipeline improves stability on a subset of tasks and provides clearer intermediate error signals than end-to-end answers alone. Overall, HighlightBench turns markup-conditioned table reasoning into a scalable, automatically scorable, and diagnosable evaluation problem.

Contributions.

- We formalize markup-conditioned reasoning over tables as a standalone evaluation target and introduce **HighlightBench**, a benchmark designed specifically for this setting.
- We construct a benchmark that combines expert-collected real-world samples with controllable synthetic instances, enabling automatically scorable evaluation, controlled difficulty, and finer-grained error attribution beyond end-to-end accuracy.
- We provide a reference pipeline that produces schema-constrained outputs with intermediate signals, serving both as a reproducible baseline and as a diagnostic interface for analyzing failure modes in current multimodal models.

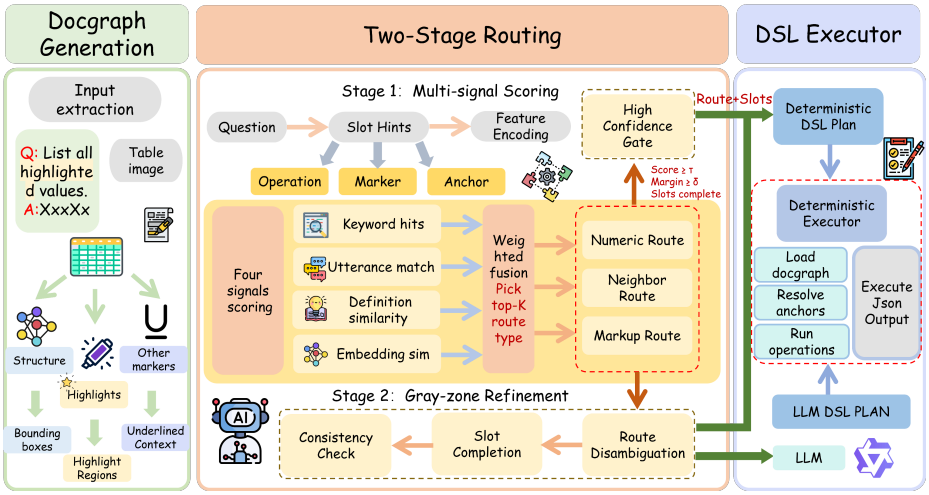


Fig. 2: Pipeline overview. The reference pipeline converts the input image into a unified docgraph, uses two-stage routing to determine the solving path, and executes the routed result as a DSL plan on the docgraph to produce the final structured output and its intermediate trace.

2 Related Work

2.1 Visual Markup Understanding in Text-Dense Documents

Recent progress in multimodal document understanding has greatly improved reading and reasoning over text-dense images. Representative systems include TextMonkey [20], which strengthens high-resolution visual reading for dense document content, and the mPLUG-DocOwl series [14, 15], which improves document modeling through stronger visual-language alignment and document-oriented instruction tuning. Other document-focused models, including DocPedia [11], Vary [32], Marten [30], and DocLayLLM [16], further show that multimodal systems can better handle layout-rich documents, long text regions, and structured extraction. Table structure recognition, especially for complicated layouts, has also been studied as a prerequisite for reliable downstream reasoning [6]. However, these models mainly target general document reading, extraction, and question answering, rather than table reasoning where embedded visual markups themselves carry task-relevant meaning.

A related line of work studies visual prompting and localized interaction. VIP-LLaVA [5], Draw-and-Understand [17], VP-Bench [35], and ControlMLLM [33] explore how arrows, points, masks, or free-form visual prompts can steer region-level grounding and localized reasoning. These methods show that explicit visual hints can substantially improve reference resolution. However, in most of these settings, the prompt is externally supplied at inference time. In our setting, by contrast, the cue is already embedded in the table and must be interpreted

as part of the original content, requiring the model to infer the cue's semantic role rather than follow an instruction-time hint. This makes markup-conditioned table reasoning qualitatively different from prompt-guided localization.

2.2 Structured Outputs and Evidence Grounding

A growing body of work suggests that answer-only evaluation can obscure important failure modes once tasks require structured outputs or explicit evidence. SO-Bench [10] demonstrates that schema-constrained multimodal outputs expose errors that are invisible under plain answer matching. ExtractBench [12] makes a similar point for structured multimodal extraction by emphasizing format compliance and faithful evidence usage. SlideVQA [24] further highlights evidence-aware document evaluation, where models must identify relevant evidence images within a slide deck in addition to answering the question. Similar diagnostic motivations have also been explored in other structured vision-language settings, such as chart question answering benchmarks that increase diversity and difficulty beyond answer-only scoring [21]. Our benchmark follows the same diagnostic spirit, but focuses on a different boundary: whether a model can correctly connect embedded visual markups, table structure, and downstream reasoning steps in a single table image.

While multimodal TableQA and document benchmarks have significantly advanced evaluation for table images and text-dense documents, they typically report end-to-end answer accuracy and do not explicitly separate cue-to-structure association from downstream reasoning behaviors. HighlightBench complements this line by making markup-conditioned evidence selection and subsequent execution directly scorable and diagnosable.

3 Methodology

Our aim is to develop a multimodal benchmark for markup-conditioned reasoning over tables under realistic scientific reading scenarios. HighlightBench is designed as a diagnostic framework that makes markup-conditioned behaviors explicit and automatically scorable, enabling analysis beyond end-to-end answer accuracy and helping identify where a model fails along the perception-to-execution chain.

For models, the central challenge is not merely detecting that a markup exists, but using it as an operational constraint over table structure. This requires correctly binding the cue to its corresponding structural units, interpreting the question under the induced scope, and executing the required table operations.

3.1 Problem Formulation

We first formalize each benchmark instance as a tuple (I, M, Q, A), where I is a table image, M denotes the visual markups, Q is the question, and A is the

target answer. The goal is to evaluate whether a model can produce the correct answer conditioned on both the table content and the markup cues:

$$P(A \mid I, M, Q). \tag{1}$$

Here, M includes visual cues relevant to reasoning, such as highlighted regions, underlined entries, bold text, arrows, or color-based emphasis. These cues interact with table structure and the question, shaping which evidence should be used and how the reasoning should proceed.

To expose failure sources, we use a diagnostic factorization with two intermediate variables. Let z denote the structural evidence indicated by the markups, such as marked cells, headers, rows, columns, or local neighborhoods. Let c denote the reasoning condition induced by that evidence under the question, such as a filtered subset, a local relation, or a comparison scope. The process can be written as

$$P(A \mid I, M, Q) = \sum_{z,c} P(A \mid I, c, Q)\, P(c \mid z, Q)\, P(z \mid I, M). \tag{2}$$

Intuitively, $P(z \mid I, M)$ captures whether visual cues are correctly mapped to table structure; $P(c \mid z, Q)$ captures whether the selected evidence is interpreted in a question-consistent way; and $P(A \mid I, c, Q)$ captures whether the model can execute the required table reasoning under that condition.

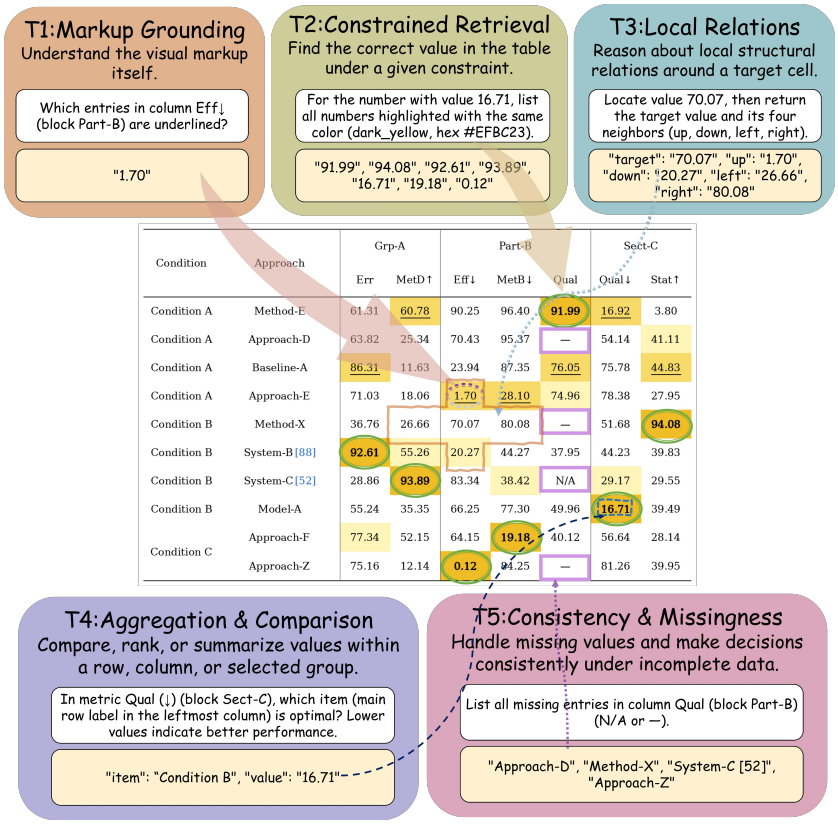
This factorization motivates our task taxonomy and supports coarse-grained attribution of errors to cue-to-structure binding, question-aware interpretation, or downstream execution.

3.2 Task Families

HighlightBench decomposes markup-conditioned table reasoning into five complementary task families. Each family emphasizes a different segment of Eq. 2, allowing targeted evaluation of specific failure modes while still operating on realistic table images.

T1. Markup Grounding. T1 evaluates cue-to-structure binding: whether highlights, underlines, bold text, and arrows are correctly associated with the intended cells, rows, columns, or headers. The questions require models to decide whether a queried cell carries a markup, count marked cells, or return the values selected by a visual style within the whole table or a specified column. It primarily probes the evidence selection term $P(z \mid I, M)$.

T2. Constrained Retrieval. T2 evaluates whether a question can be answered by retrieving entries from the table under an explicit constraint. The constraint may specify a cell location, a row and column label, a same-color highlight group anchored by a queried value, or a scoped subset such as a delta column under selected rows. In these cases, the model must identify the requested scope and return only the entries that satisfy it, rather than collecting visually nearby or semantically plausible values from the whole table. The dominant failure mode is to detect a cue but ignore or relax the constraint during selection, leading to retrieval from an incorrect scope. It primarily probes $P(c \mid z, Q)$.



Condition	Approach	Grp-A		Part-B			Sect-C	
		Err	MetD↑	Eff↓	MetB↓	Qual	Qual↓	Stat↑
Condition A	Method-E	61.31	60.78	90.25	96.40	91.99	16.92	3.80
Condition A	Approach-D	63.82	25.34	70.43	95.37	—	54.14	41.11
Condition A	Baseline-A	86.31	11.63	23.94	87.35	76.05	75.78	44.83
Condition A	Approach-E	71.03	18.06	1.70	28.10	74.96	78.38	27.95
Condition B	Method-X	36.76	26.66	70.07	80.08	—	51.68	94.08
Condition B	System-B [88]	92.61	55.26	20.27	44.27	37.95	44.23	39.83
Condition B	System-C [52]	28.86	93.89	83.34	38.42	N/A	29.17	29.55
Condition B	Model-A	55.24	35.35	66.25	77.30	49.96	16.71	39.49
Condition C	Approach-F	77.34	52.15	64.15	19.18	40.12	56.64	28.14
	Approach-Z	75.16	12.14	0.12	94.25	—	81.26	39.95

Fig. 3: Representative examples of the five task families in HighlightBench. Each card shows the core capability tested by a task family, together with a typical question format and its expected structured output.

T3. Local Relations. T3 evaluates whether local table topology is preserved, including neighborhood relations along rows and columns as well as local structural dependencies. The questions ask models to locate a target cell by index, value, or visual anchor, and then return its neighboring cells or a specified adjacent value. It bridges evidence localization and downstream reasoning by testing whether models maintain consistent local structure when moving from z to execution.

T4. Aggregation & Comparison. T4 evaluates comparison, ranking, and aggregation operations over subsets determined by the question and markups. The questions require models to compute extrema over rows or columns, find the optimal item, compare two entries, select a ranked item, or aggregate values after excluding invalid or visually marked cells. It emphasizes faithful execution under the intended scope and mainly probes $P(A \mid I, c, Q)$.

T5. Consistency & Missingness. T5 evaluates reasoning robustness under incomplete, conflicting, or missing evidence (e.g., `N/A`, dashes, or partially absent fields). The questions ask models to judge whether a specific cell is missing, list missing entries in a column, or identify values that satisfy a consistency condition such as positive deltas under a defined scope. It tests whether the model remains consistent with the intended selection and avoids shortcut heuristics when the table contains irregularities, again emphasizing $P(A \mid I, c, Q)$ under more challenging conditions.

Together, T1–T5 provide complementary diagnostics for the full pipeline from cue binding to retrieval and downstream execution, while maintaining a unified markup-conditioned table reasoning setting. Each family is further instantiated into multiple automatically scorable subtasks, yielding 21 subtasks in total. These subtasks share a unified structured output format, which makes results comparable across task families while preserving diagnostic signals specific to each family.

3.3 Benchmark Construction

HighlightBench contains 446 table images and 3,283 questions in total, including 266 real-world images with 2,268 questions and 180 synthetic images with 1,015 questions. It is constructed from two complementary sources: expert-collected real-world samples and synthetic samples generated through a structured question-driven process. This pairing preserves realistic markup patterns and table irregularities, while enabling controllable and reproducible counterfactual analysis.

Real-world samples. Human experts select markup-rich tables from CV and NLP papers and write questions for each instance. We verify that each question correctly refers to the markups present in the image and that the reference answer satisfies the corresponding markup-conditioned constraints. To improve annotation quality, candidate QA pairs are screened to remove ambiguous or underspecified instances, with roughly 90% retained after this step. Each retained instance is independently reviewed by two annotators, yielding a Cohen's kappa of 0.96; remaining disagreements are resolved through discussion and final adjudication. We further remove duplicate or near-duplicate questions to avoid redundancy. This subset preserves naturally occurring properties of scientific tables, including partial grids, merged cells, multi-level headers, and diverse markup styles.

Synthetic samples. For the synthetic subset, we start from pre-defined question templates and derive the structural roles required by each instance, including target, distractor, and anchor elements. We then instantiate table contents and visual markups so that the answer is deterministic and automatically scorable. Because this construction process is explicit and controllable, it is fully reproducible and also makes it easy to build counterfactual variants by changing markup conditions while keeping table content fixed. As a quality check, we automatically validate each synthetic instance by executing the underlying generation rule, which checks answer correctness and schema consistency before evaluation.

We also enforce diversity by removing duplicate or near-duplicate generations using template signatures and key entities. The same generation framework can be scaled by varying table contents, markup placements, templates, and distractor configurations, enabling larger verified diagnostic sets beyond the current release. This subset improves difficulty control and supports more precise analysis of model failure patterns, while following the same task taxonomy as the real-world subset.

4 Experiments

HighlightBench is used to evaluate end-to-end QA performance and to examine how multimodal large language models handle visual markups when translating them into reasoning constraints. We aim to understand both overall success rates and the sources of failure: whether they arise from unstable perception of visual cues, from incorrect reasoning with those cues, or from downstream decision errors after the cues have been identified.

This section has two complementary goals. First, we report full-benchmark results to provide an overall performance profile across the five task families. Second, we analyze the benchmark's controlled counterfactual subset, where table content is fixed while only the visual surface changes, allowing us to isolate how markup variations alone affect model behavior. Together, these experiments provide aggregate performance comparisons and direct evidence that HighlightBench can separate basic table reading from markup-conditioned reasoning failures.

4.1 Full-Benchmark Evaluation

We evaluate twelve representative MLLMs, including nine open-source models and three closed-source models. The open-source models are Gemma 3 4B [25], InternVL3.5-8B [29], Kimi-VL-A3B [26], LLaVA1.5-7B [18], MiniCPM-V 2.6 [36], MiniCPM-V 4.5 [37], Qwen2.5-VL-3B [4], Qwen3-VL-8B [3], and Qwen3.5-9B [27]. The closed-source models are Claude Sonnet 4 [2], Gemini 2.5 Flash [7], and GPT-4o mini [22]. For most model interfaces, we use VLMEvalKit [9] to standardize model invocation, input formatting, and result collection. All models are evaluated with the same input format, and scores are computed using a post-processed evaluator applied to cleaned JSON outputs.

Overall model behavior. Table 1 reports exact-match accuracy on both the synthetic and real-world subsets. Performance varies substantially across models, indicating that markup-conditioned table reasoning remains difficult even for recent MLLMs. Among the evaluated end-to-end models, Gemini 2.5 Flash achieves the strongest overall results on both subsets. However, its advantage is not uniform across task families: several models retain reasonable performance on direct retrieval-style tasks but degrade sharply when the answer depends on stable cue binding, local table topology, or multi-step constrained execution.

Table 1: Main results on HighlightBench. We report exact-match accuracy (%) for each task family and the overall score on two subsets. Yellow cells mark each model's best task family within a subset; bold numbers mark the best overall score within each evaluation group.

Model	Synthetic Acc. (%)						Real-world Acc. (%)					
	T1	T2	T3	T4	T5	All	T1	T2	T3	T4	T5	All
Open-source models												
Gemma 3 4B	4.3	10.2	2.4	20.3	23.3	12.4	22.2	9.9	2.5	10.1	16.5	13.7
InternVL3.5-8B	29.0	81.2	17.6	24.7	71.0	40.5	54.2	53.1	23.0	36.6	67.7	46.3
Kimi-VL-A3B	7.4	21.0	12.1	31.2	49.7	23.8	33.1	37.5	17.4	25.9	17.0	28.6
LLaVA1.5-7B	8.1	0.7	0.0	0.0	1.0	2.7	3.4	1.5	1.2	0.2	0.0	1.4
MiniCPM-V 2.6	17.6	59.0	7.9	13.1	14.5	20.4	17.2	29.0	9.9	17.5	3.4	17.5
MiniCPM-V 4.5	26.2	74.3	13.3	28.8	59.2	36.9	56.2	50.0	15.5	35.9	35.7	42.8
Qwen2.5-VL-3B	14.6	61.1	6.7	13.1	47.1	24.9	39.4	35.6	1.2	31.3	58.9	34.7
Qwen3-VL-8B	37.0	74.3	18.8	31.5	74.4	44.2	56.3	57.1	24.2	36.3	78.0	48.4
Qwen3.5-9B	27.7	28.5	26.7	45.5	60.5	37.7	55.3	50.4	23.6	42.8	70.1	48.8
Closed-source models												
Claude Sonnet 4	32.7	81.9	40.0	37.5	67.1	47.6	56.7	59.6	42.2	60.7	68.2	58.7
Gemini 2.5 Flash	51.9	90.3	89.1	67.8	93.9	**73.9**	81.3	75.3	62.7	72.2	78.5	75.3
GPT-4o mini	13.0	15.1	13.3	30.1	32.3	21.1	21.3	16.3	8.5	9.4	37.7	16.4
Reference pipeline												
Ours	53.4	88.9	73.3	81.5	78.9	72.3	72.9	74.2	70.8	82.2	79.8	**77.1**

Task-family difficulty is uneven. Across models, T2 is comparatively stable, as many instances mainly require locating the relevant row or column and retrieving values under a specified constraint. In contrast, T1 exposes failures in recognizing visual markups and binding them to the intended cells, rows, or columns. T3 stresses local row–column relationships, where errors often arise when the model loses the correct neighborhood or table position. T4 further amplifies upstream selection errors because aggregation or comparison requires a clean candidate set before computation. T5 is challenging in a different way: it requires reasoning over missing, ambiguous, or inconsistent entries without overcompleting unsupported structure. These patterns suggest that HighlightBench does not merely increase generic table-reading difficulty, but separates failures in cue grounding, constrained retrieval, local structural reasoning, aggregation, and evidence-faithful consistency checking.

Reference pipeline as a baseline. We also report results from our reference pipeline (**Ours**) as an additional diagnostic baseline, with design details summarized in Sec. 5. The pipeline exposes intermediate representations and execution traces, allowing the aggregate scores in Tab. 1 to be interpreted beyond final-answer accuracy. Its stronger performance on T3–T4 suggests that explicit intermediate structure helps when the task requires preserving local relations or computing over a selected candidate set. At the same time, remaining errors on T1–T2 indicate that recognizing markups and constructing the correct retrieval scope are still difficult. Together, these results make the pipeline useful for controlled error attribution and complement the end-to-end MLLM evaluation.

Table 2: Representative sub-tasks used in the counterfactual analysis.

Task	Family	Brief example
Bold Presence	T1	Is a queried cell boldfaced?
Underline Presence	T1	Is a queried cell underlined?
Highlight Presence	T1	Is a queried cell highlighted?
Cell Retrieval	T2	Read a specified table cell.
Extremum Reasoning	T4	Return the column-maximum item and value.

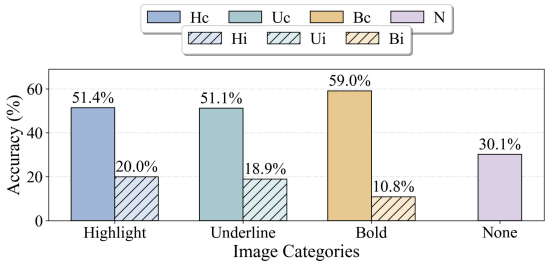


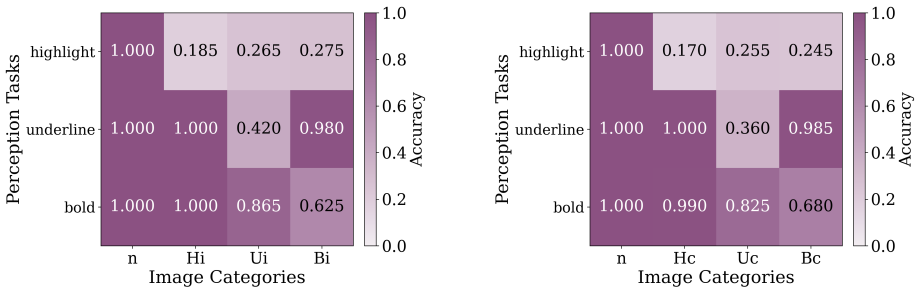
Fig. 4: Extremum Reasoning accuracy under the counterfactual variants.

4.2 Counterfactual Analysis

To probe how visual markups affect model behavior, we evaluate a controlled counterfactual subset where the table content and question text are fixed and only the visual markup is modified. For each source table, we generate seven image variants: `n` with no markup; `Hc`, `Uc`, and `Bc`, where highlight, underline, or bold is placed on the ground-truth target of the queried task; and `Hi`, `Ui`, and `Bi`, where the same marker is placed on a distractor. We conduct this analysis on Qwen3-VL-8B.

To make the intervention as controlled as possible for extremum queries, we standardize the comparison direction so that larger values are always preferred, and we ensure that the target is the unique maximum. Under this convention, the congruent variants `Hc`/`Uc`/`Bc` place the marker on the true maximum for the Extremum Reasoning task. This avoids confounding effects from mixed optimization directions and ensures that any observed shift is attributable to markup placement rather than to changes in task semantics.

We focus on five probes that capture complementary effects of markup interventions, with the specific task definitions listed in Tab. 2. Bold Presence, Underline Presence, and Highlight Presence test explicit recognition of common markup types. Cell Retrieval serves as a basic table-reading control because the queried content remains fixed when only markup placement changes. Extremum Reasoning tests whether markup placement can influence a downstream selection decision. Together, these probes separate perception, reading, and decision-level effects under controlled markup interventions.



(a) Incongruent: `n`, `Hi`, `Ui`, `Bi` **(b)** Congruent: `n`, `Hc`, `Uc`, `Bc`

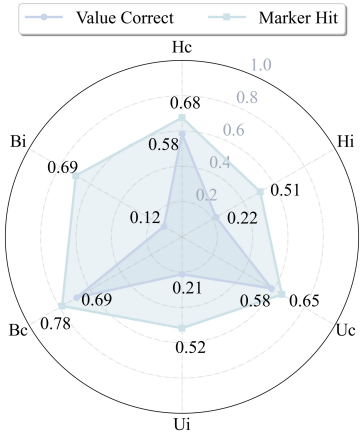
Fig. 5: Perception-task accuracy under counterfactual variants. Rows correspond to Bold Presence, Underline Presence, and Highlight Presence.

Markers materially change the final decision. The clearest effect appears in Extremum Reasoning. As shown in Fig. 4, changing only the marker placement produces large performance shifts: congruent settings consistently improve performance over the no-markup baseline, while incongruent settings consistently reduce it. Because table content and question text are identical across variants, these shifts are driven by the markup intervention itself, showing that visual markers can directly steer the final selection.

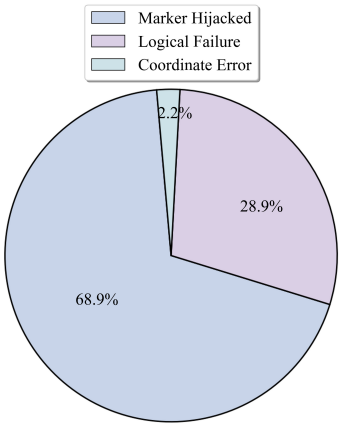
The main issue is not basic reading. This becomes clearer when compared with the Cell Retrieval control. Under the same counterfactual protocol and evaluation run, Cell Retrieval remains consistently high across the seven variants, staying within 94.5%–96.5%. This stability indicates that the large swings observed in T1 and T4 are unlikely to be caused by generic table-reading failures. Instead, the main difference lies in how the model reacts to the added markers once they are present.

Explicit recognition shows a negative bias and highlight is harder. A different pattern appears in the perception tasks (Fig. 5). For bold and underline, accuracy is very high in variants where the queried markup is absent, suggesting that the model often answers "not present" correctly and may default to negative responses. In contrast, highlight presence is substantially less stable across variants, indicating that explicit recognition of highlight is more difficult and more prone to confusion. This gap helps explain why a model can be influenced by markups in downstream decisions without reliably identifying those markups at the explicit reporting level.

Marked distractors dominate many failures. Fig. 6a shows a consistent gap between marker-hit rate and value-correct rate, indicating that predictions often land on marked values even when they are incorrect. Using value correctness and marker hit as observable signals, we further group failed cases into three attribution categories in Fig. 6b. Marker Hijacked refers to errors where the predicted value is incorrect but hits a marked cell, suggesting that the marker pulls the decision toward a distractor. Logical Failure refers to incorrect values without marker hit, indicating that the error is not directly driven by the



(a) Marker-hit vs. value-correct rates. Marker-hit checks whether the predicted value lands on a marked cell; value-correct checks whether the predicted value matches the ground-truth answer.



(b) Failure attribution under counterfactual variants.

Fig. 6: Mechanism probes under counterfactual variants. (a) Marker-hit rate versus value-correct rate across variants. (b) Error attribution of failed cases based on observable outcomes.

marked region. Coordinate Error refers to cases where the value is correct but the associated item is incorrect, suggesting a binding or localization mismatch. The breakdown shows that most failures fall into Marker Hijacked, with smaller portions explained by Logical Failure and Coordinate Error. This attribution is operational rather than mechanistic, but it provides concrete evidence that marked distractors are a dominant error source under conflicting visual cues.

Taken together, these results show that markups can systematically bias decisions under otherwise fixed evidence, motivating a more explicit inference interface in Sec. 5.

5 Pipeline Overview

Building on the counterfactual findings in Sec. 4.2, we introduce a reference pipeline as an explicit interface for markup-conditioned reasoning. The pipeline externalizes intermediate decisions that are otherwise hidden in a single response, making the solving process inspectable and enabling finer-grained failure attribution.

As shown in Fig. 2, the pipeline decomposes solving into three stages. It first converts the input into a structured docgraph that explicitly represents table content, table topology, and detected markups. It then selects a solving route and resolves task-specific constraints under a two-stage routing scheme. Finally, it executes the resulting plan with a deterministic DSL executor and returns both the final answer and intermediate traces in a unified schema.

Docgraph Generation. The first design choice is to separate the visual surface from the evidence used for reasoning. The docgraph stage converts the input image into an explicit intermediate representation that organizes textual content, table structure, and markup signals in a shared space. Markups are stored as structured attributes bound to specific cells or regions, rather than remaining as an implicit cue in a free-form response. This representation encourages later stages to operate on explicit units and relations, reduces reliance on raw visual salience, and makes binding mistakes observable.

Two-Stage Routing. A second design choice is to separate route commitment from constraint completion. The first routing stage performs coarse route selection to determine the route type and required operators based on the question form and readily available signals from the parsed context. The second stage is reserved for uncertainty resolution. It revisits cue-related conditions and completes missing constraints when the initial route is underspecified or internally inconsistent, for example when a task requires a markup-conditioned subset but the extracted cue evidence is weak or ambiguous. Compared with a single-stage router, this staged design reduces the chance that early interpretation errors are directly propagated into execution and makes failures easier to attribute to route selection versus constraint completion.

DSL Execution. The final design choice is to enforce constraint-faithful computation with deterministic execution. The routed outcome is converted into an executable DSL plan and executed on the docgraph by a deterministic executor. Constraint-conditioned operations such as filtering, extremum selection, aggregation, and consistency checks are implemented as explicit operators with fixed semantics. This reduces free-form variability, makes results reproducible, and produces intermediate traces that expose whether a failure arises from cue-to-structure binding, question-aware interpretation, or downstream execution.

Overall, the reference pipeline is a problem-driven interface for markup-conditioned reasoning that emphasizes explicit representations, staged interpretation, and traceable execution. By making intermediate decisions inspectable, it serves both as a reproducible baseline and as a diagnostic tool for understanding and mitigating failures in current multimodal models.

6 Conclusion

We present **HighlightBench**, a diagnostic benchmark for markup-conditioned reasoning over scientific tables. The benchmark covers five complementary task families and includes a controlled counterfactual subset where only markup placement varies under fixed table content and question text.

Across both full-benchmark evaluation and counterfactual analysis, we find that current MLLMs do not consistently treat visual markups as executable reasoning constraints. Models can be relatively strong at direct retrieval, yet remain fragile when correct cue grounding and constraint-conditioned execution must be combined. The counterfactual results show that marker placement alone

can systematically shift decisions, revealing a failure mode where visually salient cues can dominate outcomes even when the underlying evidence is unchanged.

Looking forward, these results suggest that progress in table understanding should be assessed and improved through diagnostic decomposition rather than answer-only scoring. Evaluations should separate cue grounding, constraint formation, and constrained execution, and systems should expose intermediate decisions to make errors traceable and correctable. More broadly, treating markups as first-class operational constraints is important for building multimodal tools that can reliably support scientific reading, verification, and table-centric document intelligence.

Acknowledgement. This work is supported in part by the National Key R&D Program of China under Grant Nos. 2025YFC3309700, the National Natural Science Foundation of China under Grant Nos. U25B2076, 62441229, and 62377043, and the Beijing Natural Science Foundation No. 4262033.

References

1. Alayrac, J.B., Donahue, J., Luc, P., Miech, A., Barr, I., Hasson, Y., Lenc, K., Mensch, A., Millican, K., Reynolds, M., et al.: Flamingo: a visual language model for few-shot learning. Advances in neural information processing systems **35**, 23716–23736 (2022)
2. Anthropic: System card: Claude opus 4 & claude sonnet 4. `https://www.anthropic.com/claude-4-system-card` (2025), accessed: 2026-03-04
3. Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., et al.: Qwen3-vl technical report. arXiv preprint arXiv:2511.21631 (2025)
4. Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., Lin, J.: Qwen2.5-vl technical report (2025), `https://arxiv.org/abs/2502.13923`, accessed: 2026-06-27
5. Cai, M., Liu, H., Mustikovela, S.K., Meyer, G.P., Chai, Y., Park, D., Lee, Y.J.: Vip-llava: Making large multimodal models understand arbitrary visual prompts. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 12914–12923 (2024)
6. Chi, Z., Huang, H., Xu, H.D., Yu, H., Yin, W., Mao, X.L.: Complicated table structure recognition. arXiv preprint arXiv:1908.04729 (2019)
7. Comanici, G., Bieber, E., Schaekermann, M., Pasupat, I., Sachdeva, N., Dhillon, I., Blistein, M., Ram, O., Zhang, D., Rosen, E., et al.: Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. arXiv preprint arXiv:2507.06261 (2025)
8. Dai, W., Li, J., Li, D., Tiong, A., Zhao, J., Wang, W., Li, B., Fung, P.N., Hoi, S.: Instructblip: Towards general-purpose vision-language models with instruction tuning. Advances in neural information processing systems **36**, 49250–49267 (2023)
9. Duan, H., Yang, J., Qiao, Y., Fang, X., Chen, L., Liu, Y., Dong, X., Zang, Y., Zhang, P., Wang, J., et al.: Vlmevalkit: An open-source toolkit for evaluating large

multi-modality models. In: Proceedings of the 32nd ACM International Conference on Multimedia. pp. 11198–11201 (2024)
10. Feng, D., Ma, K., Nan, F., Chen, H., Zhai, B., Griffiths, D., Gao, M., Gan, Z., Verma, E., Yang, Y., et al.: So-bench: A structural output evaluation of multimodal llms. arXiv preprint arXiv:2511.21750 (2025)
11. Feng, H., Liu, Q., Liu, H., Tang, J., Zhou, W., Li, H., Huang, C.: Docpedia: Unleashing the power of large multimodal model in the frequency domain for versatile document understanding. Science China Information Sciences **67**(12), 220106 (2024)
12. Ferguson, N., Pennington, J., Beghian, N., Mohan, A., Kiela, D., Agrawal, S., Nguyen, T.H.: Extractbench: A benchmark and evaluation methodology for complex structured extraction. arXiv preprint arXiv:2602.12247 (2026)
13. Herzig, J., Nowak, P.K., Müller, T., Piccinno, F., Eisenschlos, J.: Tapas: Weakly supervised table parsing via pre-training. In: Proceedings of the 58th annual meeting of the association for computational linguistics. pp. 4320–4333 (2020)
14. Hu, A., Xu, H., Ye, J., Yan, M., Zhang, L., Zhang, B., Zhang, J., Jin, Q., Huang, F., Zhou, J.: mplug-docowl 1.5: Unified structure learning for ocr-free document understanding. In: Findings of the Association for Computational Linguistics: EMNLP 2024. pp. 3096–3120 (2024)
15. Hu, A., Xu, H., Zhang, L., Ye, J., Yan, M., Zhang, J., Jin, Q., Huang, F., Zhou, J.: mplug-docowl2: High-resolution compressing for ocr-free multi-page document understanding. In: Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 5817–5834 (2025)
16. Liao, W., Wang, J., Li, H., Wang, C., Huang, J., Jin, L.: Doclayllm: An efficient multi-modal extension of large language models for text-rich document understanding. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4038–4049 (2025)
17. Lin, W., Wei, X., An, R., Gao, P., Zou, B., Luo, Y., Huang, S., Zhang, S., Li, H.: Draw-and-understand: Leveraging visual prompts to enable mllms to comprehend what you want. arXiv preprint arXiv:2403.20271 (2024)
18. Liu, H., Li, C., Li, Y., Lee, Y.J.: Improved baselines with visual instruction tuning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 26296–26306 (2024)
19. Liu, Y., Duan, H., Zhang, Y., Li, B., Zhang, S., Zhao, W., Yuan, Y., Wang, J., He, C., Liu, Z., et al.: Mmbench: Is your multi-modal model an all-around player? In: European conference on computer vision. pp. 216–233. Springer (2024)
20. Liu, Y., Yang, B., Liu, Q., Li, Z., Ma, Z., Zhang, S., Bai, X.: Textmonkey: An ocr-free large multimodal model for understanding document. IEEE Transactions on Pattern Analysis and Machine Intelligence (2026)
21. Masry, A., Islam, M.S., Ahmed, M., Bajaj, A., Kabir, F., Kartha, A., Laskar, M.T.R., Rahman, M., Rahman, S., Shahmohammadi, M., et al.: Chartqapro: A more diverse and challenging benchmark for chart question answering. In: Findings of the Association for Computational Linguistics: ACL 2025. pp. 19123–19151 (2025)
22. OpenAI: GPT-4o mini: Advancing cost-efficient intelligence. `https://openai.com/zh-Hans-CN/index/gpt-4o-mini-advancing-cost-efficient-intelligence/` (Jul 2024), accessed 27 June 2026
23. Pasupat, P., Liang, P.: Compositional semantic parsing on semi-structured tables. In: Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers). pp. 1470–1480 (2015)

24. Tanaka, R., Nishida, K., Nishida, K., Hasegawa, T., Saito, I., Saito, K.: Slidevqa: A dataset for document visual question answering on multiple images. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 13636–13645 (2023)
25. Team, G., Kamath, A., Ferret, J., Pathak, S., Vieillard, N., Merhej, R., Perrin, S., Matejovicova, T., Ramé, A., Rivière, M., Rouillard, L., Mesnard, T., Cideron, G., bastien Grill, J., Ramos, S., Yvinec, E., Casbon, M., Pot, E., Penchev, I., Liu, G., Visin, F., Kenealy, K., Beyer, L., Zhai, X., Tsitsulin, A., Busa-Fekete, R., Feng, A., Sachdeva, N., Coleman, B., Gao, Y., Mustafa, B., Barr, I., Parisotto, E., Tian, D., Eyal, M., Cherry, C., Peter, J.T., Sinopalnikov, D., Bhupatiraju, S., Agarwal, R., Kazemi, M., Malkin, D., Kumar, R., Vilar, D., Brusilovsky, I., Luo, J., Steiner, A., Friesen, A., Sharma, A., Sharma, A., Gilady, A.M., Goedeckemeyer, A., Saade, A., Feng, A., Kolesnikov, A., Bendebury, A., Abdagic, A., Vadi, A., György, A., Pinto, A.S., Das, A., Bapna, A., Miech, A., Yang, A., Paterson, A., Shenoy, A., Chakrabarti, A., Piot, B., Wu, B., Shahriari, B., Petrini, B., Chen, C., Lan, C.L., Choquette-Choo, C.A., Carey, C., Brick, C., Deutsch, D., Eisenbud, D., Cattle, D., Cheng, D., Paparas, D., Sreepathihalli, D.S., Reid, D., Tran, D., Zelle, D., Noland, E., Huizenga, E., Kharitonov, E., Liu, F., Amirkhanyan, G., Cameron, G., Hashemi, H., Klimczak-Plucińska, H., Singh, H., Mehta, H., Lehri, H.T., Hazimeh, H., Ballantyne, I., Szpektor, I., Nardini, I., Pouget-Abadie, J., Chan, J., Stanton, J., Wieting, J., Lai, J., Orbay, J., Fernandez, J., Newlan, J., yeong Ji, J., Singh, J., Black, K., Yu, K., Hui, K., Vodrahalli, K., Greff, K., Qiu, L., Valentine, M., Coelho, M., Ritter, M., Hoffman, M., Watson, M., Chaturvedi, M., Moynihan, M., Ma, M., Babar, N., Noy, N., Byrd, N., Roy, N., Momchev, N., Chauhan, N., Sachdeva, N., Bunyan, O., Botarda, P., Caron, P., Rubenstein, P.K., Culliton, P., Schmid, P., Sessa, P.G., Xu, P., Stanczyk, P., Tafti, P., Shivanna, R., Wu, R., Pan, R., Rokni, R., Willoughby, R., Vallu, R., Mullins, R., Jerome, S., Smoot, S., Girgin, S., Iqbal, S., Reddy, S., Sheth, S., Põder, S., Bhatnagar, S., Panyam, S.R., Eiger, S., Zhang, S., Liu, T., Yacovone, T., Liechty, T., Kalra, U., Evci, U., Misra, V., Roseberry, V., Feinberg, V., Kolesnikov, V., Han, W., Kwon, W., Chen, X., Chow, Y., Zhu, Y., Wei, Z., Egyed, Z., Cotruta, V., Giang, M., Kirk, P., Rao, A., Black, K., Babar, N., Lo, J., Moreira, E., Martins, L.G., Sanseviero, O., Gonzalez, L., Gleicher, Z., Warkentin, T., Mirrokni, V., Senter, E., Collins, E., Barral, J., Ghahramani, Z., Hadsell, R., Matias, Y., Sculley, D., Petrov, S., Fiedel, N., Shazeer, N., Vinyals, O., Dean, J., Hassabis, D., Kavukcuoglu, K., Farabet, C., Buchatskaya, E., Alayrac, J.B., Anil, R., Dmitry, Lepikhin, Borgeaud, S., Bachem, O., Joulin, A., Andreev, A., Hardin, C., Dadashi, R., Hussenot, L.: Gemma 3 technical report (2025), `https://arxiv.org/abs/2503.19786`, accessed: 2026-06-27
26. Team, K., Du, A., Yin, B., Xing, B., Qu, B., Wang, B., Chen, C., Zhang, C., Du, C., Wei, C., et al.: Kimi-vl technical report. arXiv preprint arXiv:2504.07491 (2025)
27. Team, Q.: Qwen3.5: Accelerating productivity with native multimodal agents (February 2026), `https://qwen.ai/blog?id=qwen3.5`, accessed: 2026-06-27
28. Wang, W., Lv, Q., Yu, W., Hong, W., Qi, J., Wang, Y., Ji, J., Yang, Z., Zhao, L., XiXuan, S., et al.: Cogvlm: Visual expert for pretrained language models. Advances in Neural Information Processing Systems **37**, 121475–121499 (2024)
29. Wang, W., Gao, Z., Gu, L., Pu, H., Cui, L., Wei, X., Liu, Z., Jing, L., Ye, S., Shao, J., et al.: Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. arXiv preprint arXiv:2508.18265 (2025)

30. Wang, Z., Guan, T., Fu, P., Duan, C., Jiang, Q., Guo, Z., Guo, S., Luo, J., Shen, W., Yang, X.: Marten: Visual question answering with mask generation for multi-modal document understanding. In: Proceedings of the Computer Vision and Pattern Recognition Conference. pp. 14460–14471 (2025)
31. Wang, Z., Xia, M., He, L., Chen, H., Liu, Y., Zhu, R., Liang, K., Wu, X., Liu, H., Malladi, S., et al.: Charxiv: Charting gaps in realistic chart understanding in multimodal llms. Advances in Neural Information Processing Systems **37**, 113569–113697 (2024)
32. Wei, H., Kong, L., Chen, J., Zhao, L., Ge, Z., Yang, J., Sun, J., Han, C., Zhang, X.: Vary: Scaling up the vision vocabulary for large vision-language model. In: European Conference on Computer Vision. pp. 408–424. Springer (2024)
33. Wu, M., Cai, X., Ji, J., Li, J., Huang, O., Luo, G., Fei, H., Jiang, G., Sun, X., Ji, R.: Controlmllm: Training-free visual prompt learning for multimodal large language models. Advances in Neural Information Processing Systems **37**, 45206–45234 (2024)
34. Wu, X., Yang, J., Chai, L., Zhang, G., Liu, J., Du, X., Liang, D., Shu, D., Cheng, X., Sun, T., et al.: Tablebench: A comprehensive and complex benchmark for table question answering. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 25497–25506 (2025)
35. Xu, M., Chen, J., Zhao, Y., Li, J.C.L., Qiu, Y., Du, Z., Wu, M., Zhang, P., Li, K., Yang, H., et al.: Vp-bench: A comprehensive benchmark for visual prompting in multimodal large language models. arXiv preprint arXiv:2511.11438 (2025)
36. Yao, Y., Yu, T., Zhang, A., Wang, C., Cui, J., Zhu, H., Cai, T., Li, H., Zhao, W., He, Z., et al.: Minicpm-v: A gpt-4v level mllm on your phone. arXiv preprint arXiv:2408.01800 (2024)
37. Yu, T., Wang, Z., Wang, C., Huang, F., Ma, W., He, Z., Cai, T., Chen, W., Huang, Y., Zhao, Y., Xu, B., Cui, J., Xu, Y., Ruan, L., Zhang, L., Liu, H., Tang, J., Liu, H., Guo, Q., Hu, W., He, B., Zhou, J., Cai, J., Qi, J., Guo, Z., Chen, C., Zeng, G., Li, Y., Cui, G., Ding, N., Han, X., Yao, Y., Liu, Z., Sun, M.: Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe (2025), https://arxiv.org/abs/2509.18154, accessed: 2026-06-27
38. Zheng, M., Feng, X., Si, Q., She, Q., Lin, Z., Jiang, W., Wang, W.: Multimodal table understanding. In: Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 9102–9124 (2024)
39. Zhong, V., Xiong, C., Socher, R.: Seq2sql: Generating structured queries from natural language using reinforcement learning. arXiv preprint arXiv:1709.00103 (2017)
40. Zhou, Y., Cheng, M., Mao, Q., Luo, Y., Liu, Q., Li, Y., Zhang, X., Liu, D., Li, X., Chen, E.: Benchmarking multimodal llms on recognition and understanding over chemical tables. arXiv preprint arXiv:2506.11375 (2025)
41. Zhu, J., Wang, J., Yu, B., Wu, X., Li, J., Wang, L., Xu, N.: TableEval: A real-world benchmark for complex, multilingual, and multi-structured table question answering. In: Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing. pp. 7126–7146. Association for Computational Linguistics (2025). https://doi.org/10.18653/v1/2025.emnlp-main.363