CAPTR: Caption-based Context Pruning for Multimodal Tabular Reasoning

Anonymous ACL submission

Abstract

Multimodal Large Language Models struggle with tabular reasoning on tables containing images as cells owing to their limited memory and the distractions posed by the surrounding irrelevant visual elements. Existing context-reduction methods reduce context size but degrade reasoning accuracy: text-only pruning neglects visual cells, token-level pruning discards valuable information, and retrieval cannot understand table structure. We propose **CAPTR**: Caption-based Context Pruning for Tabular Reasoning, a two-stage framework that uses image captions as a lightweight textual proxy to prune non-pertinent rows and columns and then restores images for reasoning. CAPTR can reduce the context length by 80% while achieving up to 8% accuracy improvement against competitive baselines. By analyzing the error cases, we find that entity confusion contributes to the majority of the remaining errors (about 50–60%). CAPTR is the first multimodal context-reduction method to improve performance on multimodal tabular reasoning.

1 Introduction

While Multimodal Large Language Models (MLLMs) have achieved reasoning capabilities for inputs that are composed of both language and visual modalities (Alayrac et al., 2022; Li et al., 2023a), their performance declines dramatically with increasing visual input sizes. This degradation is particularly important for multimodal tabular reasoning, in which a model is tasked to answer questions over tables where the cells contain text/images/visualizations in an interleaved fashion. Reasoning on such tables requires models to understand the table structure, textual context, visual evidence, and the relationships between rows and columns. In many cases, however, a question depends only on a small subset of the table, while the MLLM must process the full table. For example, on MMTabQA (Mathur et al., 2024), a single table with 324 images requires over 82,000 image tokens for Gemma 3 (Gemma Team, 2025). This creates two problems: **(1) Degraded reasoning**: MLLMs can be distracted by irrelevant visual content (Sharma et al., 2024; Wu et al., 2025). **(2) High peak memory**: all visual tokens must be encoded before reasoning, so memory grows with the number of table images.

Method	Modality	Tabular Reasoning Capabilities		
	Multimodal	Struct. Aware	Symbolic	Semantic
Multimodal Context Reduction				
SparseVLM (Zhang et al., 2024)	✓	✗	✗	✓
Retrieval	✓	✗	✗	✓
Tabular Reasoning (Text)				
H-STAR (Abhyankar et al., 2025)	✗	✓	✓	✓
WEAVER (Khoja et al., 2025)	✗	✓	✓	✓
CAPTR	✓	✓	✓	✓

Table 1: Comparison of CAPTR with existing multimodal context-reduction and table-reasoning methods.

The natural choice is to prune multimodal contexts before reasoning, but existing approaches are not designed for multimodal tables. Text-only table pruning methods such as H-STAR (Abhyankar et al., 2025) and ALTER (Zhang et al., 2025a) reduce table size by selecting query-relevant rows and columns, using symbolic reasoning, semantic filtering, SQL execution, or hybrid SQL-text reasoning (Pasupat and Liang, 2015a; Zhong et al., 2017a; Chen et al., 2020; Nan et al., 2022; Khoja et al., 2025). However, text-only methods cannot directly prune visual cells: SQL operates over structured text, while semantic filtering over images would require placing all images in the model context.

Multimodal retrieval systems choose rows and images using retrievers like ColPali (Faysse et al., 2025b), but the relevance of rows and columns is determined by the format, headers, nearby cells and symbolic rules rather than image-query similarity. Token-pruning methods like SparseVLM

Code available at https://anonymous.4open.science/r/CAPTR-MAY-ARR/README.md

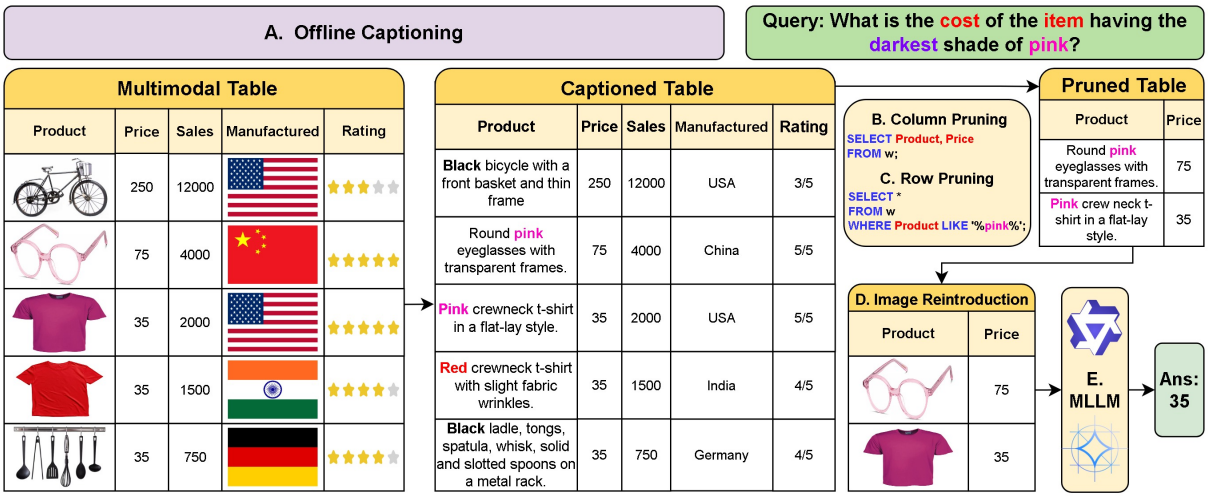


Figure 1: Overview of CAPTR. Images are replaced with captions, the captioned table is pruned via symbolic and semantic reasoning, and the original images are restored before final MLLM reasoning.

(Zhang et al., 2024), AdaFV (Han et al., 2025), and TRIM (Song et al., 2025) prune the vision tokens in a single image QA setting (Shao et al., 2025), but tabular reasoning demands pruning across rows and columns rather than image patches. Table 1 summarizes this gap: existing works are either multimodal but do not consider table structure or aware of structure yet restricted to text input.

We propose **CAPTR**, **Caption-based Context Pruning for Tabular Reasoning**, a two-stage framework illustrated in Figure 1. CAPTR replaces images with query-independent image captions as a light-weight textual representation that enables pruning of large tabular content. These captions are only needed to preserve enough information to determine whether a row or column is relevant. The first stage of CAPTR involves pruning the captioned table via symbolic SQL queries and semantic LLM-powered inference. The next stage involves bringing back the corresponding images from the pruned table for further multimodal reasoning. The two-stage design combines the efficiency of text-based pruning with the visual fidelity of multimodal reasoning.

CAPTR reduces context length while improving reasoning accuracy, outperforming both image-level retrieval and patch-level token compression. This establishes caption-based pruning as a third paradigm for multimodal context reduction.

Our main contributions are:

- We introduce **CAPTR**, a caption-based context-pruning approach for multimodal tabular reasoning.
- We show that CAPTR reduces context length upto 80% while improving QA accuracy upto 8%, with robustness across two benchmarks and three MLLMs.
- We compare against six strong baselines and show that CAPTR outperforms.
- We identify a limitation of exact-match evaluation for answer-mention questions and propose an LLM-as-a-judge with 96% agreement against human judgment.

2 CAPTR

CAPTR is a two-stage approach, illustrated in Figure 1, where the first stage replaces images with query-independent captions and performs table pruning to reduce irrelevant multimodal context, followed by a multimodal reasoner that conducts final query answering over the pruned table.

2.1 Stage 1: Caption-based Context Pruning

Table Preprocessing Not all tables are provided in SQL-friendly formats, especially those scraped off the Web or extracted as images that have more complex schemas and merged cells. CAPTR uses SQL queries to extract columns and rows, so all of the tables must be normalized to become SQL-compatible. Merged cells are broken by a rule-based algorithm: for every cell with attributes rowspan/colspan greater than 1, the cell is expanded, i.e., replicated to fill all spanned rows/columns to create a clean structure consisting of only regular cells. Since these tables are in HTML, DOM traversal is used to create the normalized tables. Additionally, all column names are checked to prevent possi-

ble naming clashes with SQL keyword names (for example, 'SELECT', 'FROM', 'WHERE'), and column names are renamed if necessary.

Offline Captioning. Each image cell is replaced with a query-independent caption, converting the multimodal table into a text-only table. Since captions are generated once per table and reused across all splits and multiple questions, this design avoids the latency and scalability costs of query-dependent captioning, making it more robust to out-of-distribution queries. Because the captions remain constant, the same index can be used for row and column selection.

We evaluate three captioning strategies: **(1) Naive captions** use surface-level descriptions, but it fails to capture fine-grained features. **(2) Detailed captions** contain fine-grained features about each object in the image as well as its features, which ensures that all details detected by the vision encoder are present in the text-only table. This makes the text longer. **(3) Short captions** provide keywords of the features. This acts like the middle ground between naive and detailed captions in terms of detail and length of text. We provide examples of each captioning strategy in Appendix G.1

Column and Row Pruning. Following H-STAR, we combine symbolic and semantic reasoning for pruning. Given the text-only table and query, we first prune columns by letting the LLM identify relevant columns through (a) SQL generation to select relevant columns and (b) direct semantic identification. Both use chain-of-thought reasoning. The final column set is the union of both. This is followed by row pruning, which is the same approach as column pruning except that it is applied to the column-filtered table.

2.2 Stage 2: Multimodal Reasoning

The pruned text table is converted back to multimodal format by replacing captions with their original images. This compact table is passed through the MLLM. Conceptually, CAPTR works with any multimodal reasoner. We opt to use the interleaved reasoner because it achieves the best results on MMTabQA and MMTBench. By operating on the pruned table, the MLLM benefits from reduced context length and reduced distraction from irrelevant images.

3 Experiments

Benchmarks **(1) MMTabQA.** (Mathur et al., 2024) is a large-scale benchmark for multimodal tabular reasoning, created by replacing textual entities with images across WikiSQL (Zhong et al., 2017b), WikiTableQuestions (Pasupat and Liang, 2015b), FeTaQA (Nan et al., 2022), and HybridQA (Chen et al., 2020). **(2) MMTBench.**(Titiya et al., 2025) consists of real-world multimodal tables where images naturally co-occur with text, hence having messier structures like merged cells and missing headers, providing a complementary setting for evaluating CAPTR. Further details can be found in Appendix A

We select these datasets because they contain images directly within structured HTML/JSON table cells rather than treating images and tables separately (MultimodalQA (Talmor et al., 2021), UniMMQA(Luo et al., 2023), SpiQA (Pramanick et al., 2024)). Unlike MMTab (Zheng et al., 2024), which uses table screenshots, this better reflects how databases store images on a real world setting.

Both datasets include four question types: **(1) Explicit Questions (EQ)** directly reference an entity whose image is in the table; in **(2) Answer-Mention Questions (AQ)**, answers contain an entity represented by an image in the table, while the question does not explicitly mention this entity; **(3) Implicit Questions (IQ)** are those where the image is not directly mentioned, however it is required for the intermediate reasoning; and **(4) Visual Questions (VQ)**, which depend on visual attributes such as color, shape, size, etc.

Baselines We compare CAPTR against the following baselines:
1) Partial Input. A text-only lower bound replacing images with placeholders, requiring context-based inference (Mathur et al., 2024). This baseline isolates the contribution of visual grounding and measures how much information can be inferred from text context and parametric knowledge alone. **2) Interleaved Reasoner.** A state-of-the-art method that processes the full table as interleaved text and images (Mathur et al., 2024; Titiya et al., 2025). Since it operates over the complete multimodal context without pruning, it serves as a strong information-access baseline against which context selection methods can be evaluated. **3) Row-wise Retrieval.** Retrieves top-k relevant rows as images using ColQwen2 before reasoning (Faysse et al., 2025a). This tests whether

coarse-grained row-level pruning is sufficient for reasoning. **4) Image+Text Retrieval.** Retrieves top-k rows separately via text and images using ColQwen2, then merges them (Faysse et al., 2025a). It evaluates whether modality-specific retrieval improves evidence coverage. **5) SparseVLM.** (Zhang et al., 2024) A token compression approach that prunes visual tokens using text-image alignment. This provides a comparison against token-level pruning methods, allowing us to compare with structural pruning. **6) Weaver.** (Khoja et al., 2025) A modular pipeline that dynamically combines SQL and LLMs for flexible table QA reasoning . It serves as a hybrid symbolic-neural baseline, enabling comparison against systems that perform structured reasoning without pruning. More details are in Appendix B.

We conduct every experiment using three models: Gemma-3-27B-Instruct (Gemma Team, 2025), Qwen3-VL-32B-Instruct (Bai et al., 2025) and Qwen-3.5-27B-Instruct(Qwen Team, 2026). More details on compute, memory etc in Appendix E

We focus on open-source MLLMs for reproducible evaluation of design choices. Efficiency is important because locally deployed models quickly exhaust VRAM when processing large multimodal tables, especially under multi-user workloads.

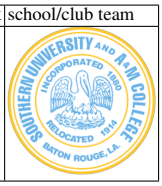
Prompt	Model	Gold	EM
pick: 139; school/club team:  **Q:** Which school or club team has a pick of 139?	southern university and a&m college	southern	0
team: ; best: 1:43.134 **Q:** What is the team with the best of 1:43.134?	PDVSA	minardi team usa	0

Table 2: Examples highlighting limitations of exact match (EM): semantically similar answers may be marked incorrect, and required information may be absent from the image.

Evaluation Metrics We identify two limitations of exact-match evaluation of these datasets: **(1)** semantically correct answers may score 0 if they do not exactly match the gold string, especially for AQ questions where answers come from images; and **(2)** models may return equivalent answers in different formats (e.g., abbreviations, full forms, or sentences). Since lexical metrics such as substring match do not fully address this (Table 11), we adopt an LLM-as-a-Judge framework following (Zhang et al., 2025b).

Because answers are typically short entities or numbers, evaluation is framed as binary correctness with prompts enforcing semantic equivalence. Manual evaluation on 100 random samples shows 96% agreement with human annotation. For FeTaQA, we retain ROUGE-L since long-form answers make binary LLM-based grading less reliable. We show some failure cases and provide our prompt in Appendix D

4 Results and Analysis

Performance across Baselines The baselines present three common failure modes: (1) Row-wise and image+text retrieval tend to retrieve similar rows visually but cannot include supportive tabular contexts like headers, neighbor cells, and symbolic constraints. (2) SparseVLM reduces visual context at the token level, but can remove useful visual information even for simple visuals such as flags and logos. Additionally, it prunes image tokens during inference, and thus the whole context needs to be fed into MLLM, resulting in similar issues as mentioned in Section 1. (3) Hybrid pipeline systems like Weaver have symbolic and semantic reasoning capabilities for tables, but their design is not suitable for multimodal reasoning of row and column selection for arbitrary tables. These limitations motivate CAPTR's caption-based pruning, which selects relevant rows and columns before final multimodal reasoning.

Performance of CAPTR vs Baselines As shown in Table 3, CAPTR shows consistent improvement across all models and all question types. The gains are most pronounced on visually grounded and multi-step reasoning queries, suggesting that CAPTR is more effective in retaining structural evidence in the process of pruning relevant rows and columns than the baselines, which depend on either coarse visual similarities or brittle prompting mechanisms.

Performance of CAPTR across Models. As can be observed from Table 3, CAPTR outperforms the interleaved reasoner in the majority of settings for all three MMLMs. In terms of the gains made by CAPTR compared to Gemma3,

Model	Baseline	MMTabQA-WikiTQ				MMTabQA-WikiSQL				MMTabQA-HybridQA				MMTBench			
		EQ	AQ	IQ	VQ	EQ	AQ	IQ	VQ	EQ	AQ	IQ	VQ	EQ	AQ	IQ	VQ
ColQwen2 Retrieval	Row-wise	43.12	40.75	27.79	35.35	53.39	56.38	58.10	50.85	52.35	49.53	53.69	44.45	37.36	42.15	26.46	26.62
	Image+Text	47.00	40.71	36.09	40.94	54.81	53.61	58.41	50.08	54.90	50.84	57.20	46.69	39.74	46.83	26.59	31.47
Gemma3	Partial Input	40.63	31.04	64.14	33.29	39.57	28.14	70.48	27.71	68.43	62.43	69.14	49.14	28.51	37.19	23.61	22.06
	SparseVLM	35.71	36.43	32.00	33.29	43.29	38.71	52.38	34.86	60.71	53.00	59.43	39.86	34.71	35.26	23.61	28.09
	WEAVER	22.47	27.80	46.26	21.71	37.86	48.43	65.08	19.18	58.14	29.64	62.00	62.16	28.31	31.01	17.15	19.85
	Interleaved	62.86	51.86	67.00	55.86	63.00	57.43	74.60	62.14	74.43	68.00	73.71	57.14	43.24	46.56	30.94	33.68
	CAPTR	67.29	55.00	67.86	57.14	64.29	64.43	75.56	64.29	77.71	68.57	81.00	67.29	44.77	47.66	31.6	34.71
	Δx	+4.43%	+3.14%	+0.86%	+1.28%	+1.29%	+7.00%	+0.96%	+2.15%	+3.28%	+0.57%	+7.29%	+5.13%	+1.53%	+1.10%	+0.66%	+1.03%
Qwen3 VL	Partial Input	40.63	32.05	64.57	30.29	40.86	27.21	71.43	27.43	67.71	64.14	69.14	48.43	24.74	28.65	18.18	17.21
	SparseVLM	59.00	48.71	67.14	53.71	61.00	54.14	73.65	59.14	72.43	66.57	74.86	56.43	32.91	36.06	22.39	26.03
	WEAVER	28.14	34.78	45.74	21.71	43.86	48.29	60.32	34.25	44.00	31.43	56.71	32.43	28.97	29.36	22.10	21.80
	Interleaved	64.00	51.00	66.14	54.43	62.71	58.43	75.24	63.57	73.00	66.29	75.71	58.43	38.75	38.84	29.85	32.79
	CAPTR	70.14	54.71	70.14	58.00	65.71	62.00	77.06	65.86	81.86	71.86	82.14	63.29	43.38	41.87	31.21	35.15
	Δx	+6.14%	+3.71%	+4.00%	+3.57%	+3.00%	+3.57%	+1.82%	+2.29%	+8.86%	+5.57%	+6.43%	+4.86%	+4.63%	+3.03%	+1.36%	+2.36%
Qwen3.5	Partial Input	36.19	29.47	60.14	21.57	34.86	27.00	67.94	28.00	68.57	58.14	70.14	47.14	33.00	33.33	21.71	20.88
	SparseVLM	58.57	47.43	63.63	51.43	60.43	55.57	71.29	60.86	71.57	66.86	71.38	55.71	33.74	36.65	23.24	27.21
	WEAVER	27.57	21.88	48.17	18.28	28.71	16.14	60.32	24.93	48.92	36.55	64.08	44.14	31.97	35.64	23.69	22.86
	Interleaved	59.43	50.57	64.86	49.57	65.00	58.00	75.24	60.00	75.43	65.14	76.29	62.86	46.21	47.66	33.24	35.00
	CAPTR	66.71	56.29	70.29	54.71	67.14	63.13	77.06	62.86	83.00	70.43	83.86	64.43	49.13	49.63	34.85	36.03
	Δx	+7.28%	+5.72%	+5.43%	+5.14%	+2.14%	+5.13%	+1.82%	+2.86%	+7.57%	+5.29%	+7.57%	+1.57%	+2.92%	+1.97%	+1.61%	+1.03%

Table 3: Main results across models, datasets, and question types.

Model	Baseline	MMTabQA-FetaQA			
		EQ	AQ	IQ	VQ
ColQwen2 Retrieval	Row-wise	0.52	0.32	0.42	0.48
	Image+Text	**0.53**	0.33	**0.43**	**0.49**
Gemma3	Partial Input	0.32	0.31	0.27	0.26
	SparseVLM	0.32	0.29	0.24	0.26
	WEAVER	0.26	0.29	0.29	0.25
	Interleaved	0.50	0.46	0.37	0.34
	CAPTR	**0.53**	**0.48**	0.42	0.48
Qwen3 VL	Partial Input	0.49	0.40	0.37	0.26
	SparseVLM	0.55	0.47	0.42	0.42
	WEAVER	0.39	0.36	0.28	0.27
	Interleaved	0.56	**0.49**	0.43	**0.44**
	CAPTR	**0.58**	**0.49**	**0.45**	**0.44**
Qwen3.5	Partial Input	0.50	0.46	0.37	0.34
	SparseVLM	0.55	0.48	0.43	0.42
	WEAVER	0.38	0.34	0.34	0.32
	Interleaved	0.56	**0.49**	0.43	**0.44**
	CAPTR	**0.58**	**0.49**	**0.45**	**0.44**

Table 4: Results on MMTabQA-FetaQA.

the greatest improvements were for WikiSQL-AQ, HybridQA-IQ, and HybridQA-VQ. In contrast, the results are relatively uniform in the case of Qwen3 VL, where the model improved for all considered datasets and types of questions, achieving especially large improvements in HybridQA-EQ (+8.86), HybridQA-IQ, and MMTBench-EQ. As for Qwen3.5, similarly to its predecessor, the most noticeable improvements were achieved in WikiTQ-EQ (+7.28), WikiTQ-AQ, HybridQA-EQ (+7.57), and HybridQA-IQ (+7.57). These results show that CAPTR is model-agnostic, and caption-based pruning helps boost reasoning on different open-source MMLM families

Performance of CAPTR across Question Types. CAPTR shows the largest gains on EQ and IQ question types, particularly on WikiTQ and HybridQA. For EQ, improvements include +8.86 for Qwen3.5 on WikiTQ and Qwen3 on HybridQA, while IQ gains reach +7.57 for Qwen3.5, +7.29 for Gemma3, and +6.43 for Qwen3 on HybridQA. AQ gains are strongest on WikiSQL, with improvements of +7.00 for Gemma3 and +5.13 for Qwen3.5. Although VQ gains are smaller, they remain consistent across models and datasets, including MMTBench, showing that pruning preserves visual reasoning ability. Overall, CAPTR is most effective when accurate evidence selection is required, while still improving visually grounded tasks.

Effect of Captions on Models The context reduction effectiveness of the CAPTR pruning process is provided in Table 6. On MMTabQA, CAPTR maintains around 66–69% of the rows and 50–53% of the columns, thereby retaining 76–79% of the overall tokens. In the case of MMTBench, the row retention ratio remains comparable (48–62%) while the token reduction rate is stricter (70–80%), suggesting higher density of the table contents with less number of rows and columns pertinent to each question. In both sets of experiments, the naive caption approach proves to be the best-performing in the pruning step, beating the shorter and longer versions by 1–4 points.

Effect of Pruning Table 6 quantifies the context reduction achieved by CAPTR's pruning stage. On MMTabQA, CAPTR retains approximately 66–69% of rows and 50–53% of columns, resulting in

Model	Setting	MMTabQA-WikiTQ				MMTabQA-WikiSQL				MMTabQA-HybridQA				MMTBench			
		EQ	AQ	IQ	VQ	EQ	AQ	IQ	VQ	EQ	AQ	IQ	VQ	EQ	AQ	IQ	VQ
Gemma3	w/Img	67.29	55.00	67.86	57.14	64.29	64.43	75.56	64.29	77.71	68.57	81.00	67.29	44.77	46.56	31.6	34.71
	w/o Img	55.00	53.86	53.71	54.29	60.00	61.43	69.84	62.43	73.14	66.86	75.00	64.43	43.74	44.35	28.29	31.53
	Δ	-12.29%	-1.14%	-14.15%	-2.85%	-4.29%	-3.00%	-5.72%	-1.86%	-4.57%	-1.71%	-5.00%	-2.86%	-1.03%	-2.21%	-3.31%	-3.18%
Qwen3 VL	w/Img	70.14	54.71	70.14	58.00	65.71	62.00	77.06	65.86	81.86	71.86	82.14	63.29	43.38	41.87	31.21	35.15
	w/o Img	59.00	50.14	61.57	50.43	59.29	56.43	70.79	54.29	75.00	68.29	73.29	62.00	38.08	37.19	27.00	28.38
	Δ	-11.14%	-4.57%	-8.57%	-7.57%	-6.42%	-5.57%	-6.27%	-11.57%	-6.86%	-3.57%	-8.85%	-1.29%	-5.30%	-4.68%	-4.21%	-6.77%
Qwen3.5	w/Img	66.71	56.29	70.29	54.71	67.14	63.13	77.06	62.86	83.00	70.43	83.86	64.43	49.13	49.63	34.85	36.03
	w/o Img	59.14	52.57	55.14	52.86	61.71	58.29	72.06	60.71	73.29	65.14	74.14	59.29	39.20	39.67	28.36	28.82
	Δ	-7.57%	-3.72%	-15.15%	-1.85%	-5.43%	-4.84%	-5.00%	-2.15%	-9.71%	-5.29%	-9.72%	-4.86%	-9.93%	-9.96%	-6.49%	-7.21%

Table 5: Captioning strategy ablation with and without image reintroduction (w/ vs w/o).

Model	Cap.	MMTabQA			MMTBench		
		%R	%C	%T	%R	%C	%T
Gemma3	Naive	66.60	52.41	79.23	55.20	60.03	77.92
	Short	67.13	52.24	79.15	56.78	60.06	70.64
	Det.	65.21	52.39	76.65	56.27	59.67	74.19
Qwen3 VL	Naive	68.50	52.51	79.92	48.40	59.33	74.01
	Short	67.70	50.80	79.28	50.28	59.15	74.23
	Det.	65.13	52.49	76.22	50.60	59.23	70.62
Qwen3.5	Naive	68.42	52.87	79.86	60.36	60.24	80.40
	Short	69.29	52.89	79.90	60.28	59.91	80.19
	Det.	69.18	52.91	78.28	62.21	60.6	77.03

Table 6: Effect of pruning across two datasets. %R, %C, and %T denote the percentage of rows, columns, and tokens removed relative to the original table.

76–79% of the total tokens. On MMTBench, row retention is similar (48–62%) but token reduction is more aggressive (70–80%), indicating denser tables with fewer relevant rows and columns per query. Across both datasets, the naive caption strategy consistently achieves the strongest pruning-stage accuracy, outperforming the short and detailed variants by 1-4 points. This suggests that unstructured keyword lists are more effective for relevance detection than either sparse or verbose captions, avoiding both information loss and noise. The effect of pruning for each subset for each question type is given in Appendix G.1

Image Reintroduction Captions act as cheap textual proxies that let us prune the table under a much smaller context budget than the full interleaved table would require. However, they may fail to capture all fine-grained visual information, such as colors, positioning of elements, number values, and chart visualizations. Hence, after caption-based pruning, we replace the captions with their original images. The MLLM then attends only to the small set of relevant images rather than all images in the table. As shown in Table 5, incorporating the original images after pruning improves overall QA accuracy by up to 15%. The largest gains are observed for IQ and VQ question types, where access to both the exact image provides the information necessary for reasoning. This also explains why in some cases the performance on caption-only pruned tables is lower than that of the origial table, despite pruning a large chunk of the table.

Model	MTD.	Ent. + Chart	Ent. + Map	Maps	Mult. Charts	Mult. Entities	Single Chart	Single Entity	Viz
Gemma3	Int.	58.33	46.37	32.85	29.09	39.94	29.71	41.43	32.39
	Det.	61.11	47.41	39.42	29.09	40.17	32.59	47.68	38.03
	Naive	58.33	45.60	31.39	30.30	43.10	35.46	40.10	30.99
	Short	57.22	40.93	32.12	26.67	37.15	29.39	42.02	37.68
Qwen3 VL	Int.	53.67	44.53	34.42	31.56	39.47	35.13	43.43	35.19
	Det.	55.56	45.85	36.50	41.82	41.22	42.81	44.96	41.90
	Naive	52.78	45.60	32.85	29.70	42.20	33.23	45.62	39.08
	Short	58.33	52.59	43.07	44.85	47.85	42.49	51.57	41.55
Qwen3.5	Int.	54.78	41.47	38.78	35.77	40.55	38.65	43.19	35.19
	Det.	55.56	46.63	41.61	42.42	43.20	42.81	48.79	45.07
	Naive	50.00	39.56	25.00	41.82	41.60	38.10	43.38	42.05
	Short	63.89	53.37	43.80	42.22	46.87	43.13	50.70	46.83

Table 7: Performance breakdown across different table types for MMTBench.

Robustness across Table Types Table 7 shows that CAPTR improves the performance across all the table types in MMTBench. Tables are categorizes into eight types: (1) *Single Entity*, containing a single entity type such as persons, flags, or maps; (2) *Multiple Entities*, containing combinations of entities such as person+flag; (3) *Single Chart*, containing one chart type such as bar or pie charts; (4) *Multiple Charts*, containing multiple chart types within the same table, e.g., bar+line charts; (5) *Entity+Chart*; (6) *Entity+Map*; (7) *Maps Only*; and (8) *Visualizations*.

While detailed captions are preferable for entity and map-oriented tables, where maintaining more context can be helpful in choosing the correct evidence during the pruning step, short captions perform better on dense visuals such as multi-chart tables. Overall, the breakdown demonstrates that CAPTR remains robust across diverse table types, while also indicating that no single captioning strategy consistently performs best for every table type.

5 Ablations

How Captioning Strategy affects pruning Two trends are observed in Table 28 from the three cap-

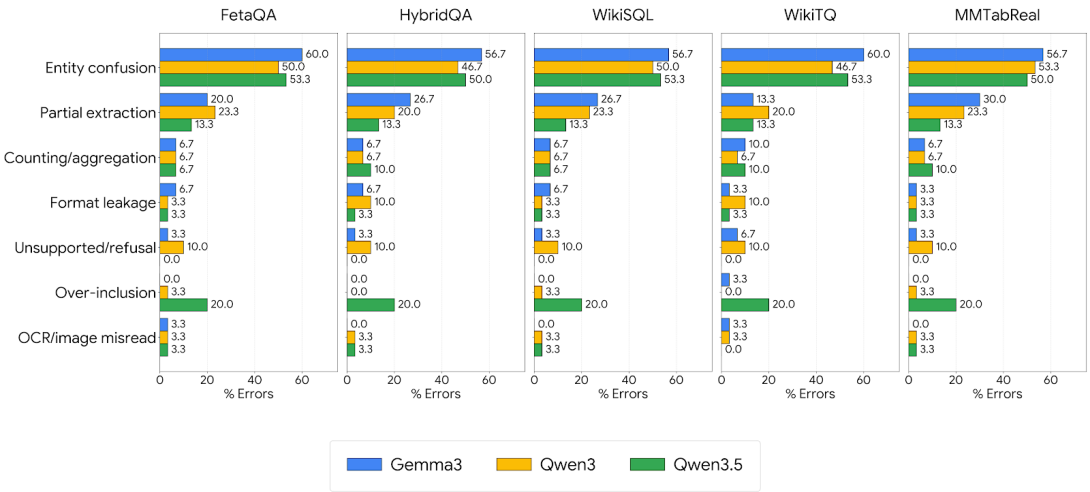


Figure 2: Error Distribution

Strategy	Dataset	EQ	AQ	IQ	VQ
Select	WikiTQ	61.00	50.36	59.30	56.00
	WikiSQL	61.86	57.43	69.84	60.55
	HybridQA	71.14	61.14	71.84	56.95
	FeTaQA	0.533	0.484	0.436	0.489
	MMTBench	42.42	44.35	30.26	33.97
Remove	WikiTQ	52.86	49.14	49.29	47.57
	WikiSQL	62.29	59.29	67.30	62.57
	HybridQA	72.43	63.57	71.00	62.71
	FeTaQA	0.459	0.528	0.414	0.486
	MMTBench	38.21	40.77	26.05	30.44

Table 8: Selector vs. Remover using Gemma3.

tioning approaches (Naive, Short, and Detailed). First, caption style determines the efficiency of pruning: the Detailed caption allows for more severe pruning, while Short captions preserve more rows and tokens (Table 6). This is because the semantic pruner can identify which part of the table is relevant. Secondly, caption quality has different impacts on the downstream QA task depending on the model and the question type: Short and Detailed captions with the reintroduction of the image consistently obtain higher QA results compared to Naive captions, even though the latter performs well at the pruning stage.

While captioning enables efficient context reduction, naive captions in particular produce sparse descriptions that miss important visual cues, leading to accidental pruning of relevant images that are important for intermediate reasoning and downstream QA.

Selecting vs. Removing Pruning can be considered as either *selection* or *removal*. Table 8 presents the performance comparison for both methods using Gemma3. While removal seems to be safer due to the ambiguity of content, selection is a superior method. Selection shows significant improvements on MMTabQA-WikiTQ with an average gain of 6.95 across all types of questions, while slightly improving FeTaQA with ROUGE-L 0.014. On WikiSQL and HybridQA, the results vary depending on the type of questions, and removal proves to be effective for some types of questions. In other words, although LLMs have a greater ability to recognize which information they need rather than to know which information is not required, the outcome highly depends on the dataset configuration. Thus, CAPTR implements selection as its pruning technique.

6 Error Analysis

To understand where CAPTR fails, we randomly selected 30 failure cases by the model for each split (total of 600 unique questions) and classified each error into one of seven categories: (1) entity confusion - the wrong selection of a nearby entity/row; (2) partial extraction - the answer is incomplete; (3) counting/aggregation mistake - calculation errors made by the model; (4) format leak - the output is still a draft version that includes the reasoning or the parts of the SQL query; (5) unsupported/unanswered - the question cannot be answered or there's no answer available due to overpruning or poor reasoning; (6) over-inclusion - there are some additional elements included besides those from the actual answer; (7) OCR/image reading mistake. Figure 2

Key Patterns and Insights. CAPTR mostly clusters errors in terms of *entity confusion* and *partial extraction*. Entity confusion remains dominant among all data splits, whereas over-inclusion dom-

inates in Qwen3.5. The biggest impact made by the CAPTR is that the remaining errors become more semantically oriented as opposed to being general structural mistakes. Gemma3 is the most stable when it comes to CAPTR. Qwen3 exhibits a slightly more varied error spectrum, where apart from entity confusion, there are format leaks and refusals.

7 Related Works

Tabular Reasoning. LLM-based table QA systems combine semantic reasoning, symbolic execution, and hybrid SQL-LLM pipelines (Wei et al., 2022; Liu et al., 2024; Zhu et al., 2024; Wu and Feng, 2024; Wang et al., 2025). Methods such as H-STAR (Abhyankar et al., 2025), ALTER (Zhang et al., 2025a), and WEAVER (Khoja et al., 2025) select or reason over relevant regions of textual tables using SQL execution, semantic filtering, or modular execution plans. These systems provide strong structure-aware reasoning modules, but they assume textual cell values and therefore cannot directly prune or reason over image cells.

Multimodal Tabular Reasoning. Recent benchmarks such as MMTabQA (Mathur et al., 2024) and MMTBench (Titiya et al., 2025) expose the need to reason jointly over table structure, text, and embedded visual content. Existing methods address different parts of this setting. Table-LLaVA (Zheng et al., 2024) focuses on structured recognition from rendered table images, while HProPro (Shi et al., 2024) uses code generation for multimodal table reasoning but processes the full context. Training-based approaches improve structure awareness: TURBO (Jiang et al., 2026) uses structured table data as privileged supervision, and DiSCo with Table-GLS (Zhu et al., 2026) separates structural abstraction from semantic grounding. These methods improve multimodal table understanding, but they do not provide an inference-time context-pruning mechanism for large multimodal tables.

Multimodal Context Reduction. Most MLLM context-reduction methods compress visual tokens inside images rather than remove semantically irrelevant table regions. A recent survey (Shao et al., 2025) categorizes these methods into transformation-, similarity-, attention-, and query-based approaches. Methods such as FastV and PyramidDrop (Xing et al., 2025) prune visual tokens using attention or staged compression, while SparseVLM (Zhang et al., 2024), AdaFV (Han et al., 2025), and TRIM (Song et al., 2025) use text-image alignment to remove patch tokens. These methods reduce visual-token cost, but they do not perform semantic row-column selection. Retrieval models such as ColPali and ColQwen2 (Faysse et al., 2025b) operate at a coarser granularity by retrieving pages or rows from visual embeddings, but they do not explicitly model table operations or row-column dependencies.

CAPTR fills this gap by using captions as a pruning-only proxy: it performs semantic row-column selection before final multimodal reasoning, rather than compressing patch tokens, retrieving visually similar rows, or pruning text-only tables. To our knowledge, CAPTR is the first multimodal table context-reduction method to improve performance, establishing caption-based pruning as a reliable paradigm.

8 Conclusion

In this work, we propose CAPTR - a caption-based approach to context pruning for multimodal tabular reasoning that deals with the scalability and memory problem, wherein most questions can be answered using only a few rows and columns, thus making the provision of all rows and columns infeasible because it would result in a larger number of tokens and higher VRAM. With the help of query-independent captions, CAPTR performs the task of table pruning before image restoration and answering questions. For the three MLLMs considered, CAPTR manages to significantly decrease the token size by 70-80% while outperforming the full interleaved reasoner by up to 10% in MMTabQA and 4% in MMTBench. To our knowledge, CAPTR is the first multimodal context-reduction method to improve multimodal tabular reasoning, establishing caption-based pruning as a reliable paradigm for scalable multi-image reasoning.

Future work includes extending CAPTR to document and video QA, as well as multilingual and domain-specific settings involving technical terminology and complex visual content. The results and error analysis also suggest opportunities for fine-tuning MLLMs to better support SQL-based and semantic pruning, including post-pruning adaptation for multimodal table QA. Finally, integrating more advanced table transformation methods (Nahid and Rafiei, 2025; Li et al., 2023b; Najpande et al., 2026) could improve compatibility with SQL-based pruning, reducing reliance on semantic pruning and lowering token costs.

Limitations

Since CAPTR uses query-independent captions as the basis for the pruning phase, it performs poorly whenever the critical information is encoded using pixels rather than words. This is the case with small bits of text in a table, labels and notes in graphs, keys to legends, logos, stamps, and visual attributes like colors and shapes. In such situations, the pruning phase will effectively filter out any hint distinguishing the right row/column from a potential alternative.

Another limitation is semantic row pruning, which requires providing the entire column-pruned captioned table to the LLM used for row pruning. Although this increases token usage at this stage, it is necessary because query-independent linking between captions and images is highly challenging. For example, in Figure 1, the query asks *"What is the cost of the item having the darkest shade of pink?"* while the relevant caption may instead use terms such as *magenta, rose, fucsia, or flamingo*. Moreover, this step is performed after column pruning, which already reduces the overall token count.

Our evaluation focuses on accuracy and on how much the table is pruned %peak per-example context length (the maximum across all pipeline stages) rather than end-to-end latency. Because CAPTR distributes work across multiple LLM calls, a fair wall-clock comparison against single-call baselines depends on serving choices (batching, caching, model co-location) that are orthogonal to our research questions, and we leave such analysis to future work.

Ethical Statement

We note several potential risks associated with this work. First, models may reflect or amplify biases present in their training data, which can lead to unfair or inaccurate outputs. Second, applying these methods to tables containing private or sensitive information could result in privacy harms. To mitigate these risks, we recommend limiting the use of these methods to research settings and requiring human review before deploying outputs in high-stakes or decision-making contexts.

All datasets and tools used in this work are publicly available and open source, and our use of these artifacts was consistent with their intended purposes and applicable licenses. The datasets employed do not contain private, sensitive, or offensive content. For the artifacts released as part of this work, we intend them solely for research and evaluation purposes. Any derivatives of the accessed data should remain within research contexts and should not be redistributed in ways that violate the terms of the original datasets. Additionally, we used AI-assisted writing tools to improve the clarity and readability of the manuscript by obtaining feedback and suggestions during the writing process.

References

Nikhil Abhyankar, Vivek Gupta, Dan Roth, and Chandan K. Reddy. 2025. H-STAR: LLM-driven hybrid SQL-text adaptive reasoning on tables. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8841–8863, Albuquerque, New Mexico. Association for Computational Linguistics.

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, and 8 others. 2022. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, and 1 others. 2025. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. HybridQA: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036, Online. Association for Computational Linguistics.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2025a. Colpali: Efficient document retrieval with vision language models. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Manuel Faysse, Hugues Sibille, Tony Wu, Bilel Omrani, Gautier Viaud, Céline Hudelot, and Pierre Colombo. 2025b. Colpali: Efficient document retrieval with vision language models. *Preprint*, arXiv:2407.01449.

Gemma Team. 2025. Gemma 3 technical report. *arXiv preprint*, abs/2503.19786.

Jiayi Han, Liang Du, Yiwen Wu, Xiangguo Zhou, Hongwei Du, and Weibo Zheng. 2025. Adafv: Rethinking of visual-language alignment for VLM acceleration. *arXiv preprint*, abs/2501.09532.

Jun-Peng Jiang, Yu Xia, Hai-Long Sun, Shiyin Lu, Qing-Guo Chen, Weihua Luo, Kaifu Zhang, De-Chuan Zhan, and Han-Jia Ye. 2026. Multimodal tabular reasoning with privileged structured information. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Rohit Khoja, Devanshu Gupta, Yanjie Fu, Dan Roth, and Vivek Gupta. 2025. Weaver: Interweaving SQL and LLM for table reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 28282–28308, Suzhou, China. Association for Computational Linguistics.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles, SOSP 2023, Koblenz, Germany, October 23-26, 2023*, pages 611–626. ACM.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023a. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. *Preprint*, arXiv:2301.12597.

Peng Li, Yeye He, Cong Yan, Yue Wang, and Surajit Chaudhuri. 2023b. Auto-tables: Synthesizing multi-step transformations to relationalize tables without using examples. *Preprint*, arXiv:2307.14565.

Shicheng Liu, Jialiang Xu, Wesley Tjangnaka, Sina Semnani, Chen Yu, and Monica Lam. 2024. SUQL: Conversational search over structured and unstructured data with large language models. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4535–4555, Mexico City, Mexico. Association for Computational Linguistics.

Haohao Luo, Ying Shen, and Yang Deng. 2023. Unifying text, tables, and images for multimodal question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9355–9367, Singapore. Association for Computational Linguistics.

Suyash Vardhan Mathur, Jainit Sushil Bafna, Kunal Kartik, Harshita Khandelwal, Manish Shrivastava, Vivek Gupta, Mohit Bansal, and Dan Roth. 2024. Knowledge-aware reasoning over multimodal semi-structured tables. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 14054–14073, Miami, Florida, USA. Association for Computational Linguistics.

Md Mahadi Hasan Nahid and Davood Rafiei. 2025. Normtab: Improving symbolic reasoning in llms through tabular data normalization. *Preprint*, arXiv:2406.17961.

Gaurav Najpande, Tampu Ravi Kumar, Manan Roy Choudhury, Neha Valeti, Yanjie Fu, and Vivek Gupta. 2026. Quiett: Query-independent table transformation for robust reasoning. *Preprint*, arXiv:2602.20017.

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, Mutethia Mutuma, Ben Rosand, Isabel Trindade, Renusree Bandaru, Jacob Cunningham, Caiming Xiong, and Dragomir Radev. 2022. FeTaQA: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

Panupong Pasupat and Percy Liang. 2015a. Compositional semantic parsing on semi-structured tables. *Preprint*, arXiv:1508.00305.

Panupong Pasupat and Percy Liang. 2015b. 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)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Shraman Pramanick, Rama Chellappa, and Subhashini Venugopalan. 2024. Spiqa: A dataset for multimodal question answering on scientific papers. In *Advances in Neural Information Processing Systems*, volume 37, pages 118807–118833. Curran Associates, Inc.

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

Kele Shao, Keda Tao, Kejia Zhang, Sicheng Feng, Mu Cai, Yuzhang Shang, Haoxuan You, Can Qin, Yang Sui, and Huan Wang. 2025. When tokens talk too much: A survey of multimodal long-context token compression across images, videos, and audios. *arXiv preprint*, abs/2507.20198.

Aditya Sharma, Michael Saxon, and William Yang Wang. 2024. Losing visual needles in image haystacks: Vision language models are easily distracted in short and long contexts. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5429–5451, Miami, Florida, USA. Association for Computational Linguistics.

Qi Shi, Han Cui, Haofeng Wang, Qingfu Zhu, Wanxiang Che, and Ting Liu. 2024. Exploring hybrid question answering via program-based prompting. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11035–11046, Bangkok, Thailand. Association for Computational Linguistics.

Dingjie Song, Wenjun Wang, Shunian Chen, Xidong Wang, Michael X. Guan, and Benyou Wang. 2025. Less is more: A simple yet effective token reduction method for efficient multi-modal LLMs. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7614–7623, Abu Dhabi, UAE. Association for Computational Linguistics.

Alon Talmor, Ori Yoran, Amnon Catav, Dan Lahav, Yizhong Wang, Akari Asai, Gabriel Ilharco, Hannaneh Hajishirzi, and Jonathan Berant. 2021. Multimodal{qa}: complex question answering over text, tables and images. In *International Conference on Learning Representations*.

Prasham Yatinkumar Titiya, Jainil Trivedi, Chitta Baral, and Vivek Gupta. 2025. MMTBENCH: A unified benchmark for complex multimodal table reasoning. *arXiv preprint*, abs/2505.21771.

Yuxiang Wang, Jianzhong Qi, and Junhao Gan. 2025. Accurate and regret-aware numerical problem solver for tabular question answering. pages 12775–12783.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Tsung-Han Wu, Giscard Biamby, Jerome Quenum, Ritwik Gupta, Joseph E. Gonzalez, Trevor Darrell, and David M. Chan. 2025. Visual haystacks: A vision-centric needle-in-a-haystack benchmark. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Zirui Wu and Yansong Feng. 2024. ProTrix: Building models for planning and reasoning over tables with sentence context. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 4378–4406, Miami, Florida, USA. Association for Computational Linguistics.

Long Xing, Qidong Huang, Xiaoyi Dong, Jiajie Lu, Pan Zhang, Yuhang Zang, Yuhang Cao, Conghui He, Jiaqi Wang, Feng Wu, and Dahua Lin. 2025. Conical visual concentration for efficient large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 14593–14603.

Han Zhang, Yuheng Ma, and Hanfang Yang. 2025a. ALTER: augmentation for large-table-based reasoning. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL 2025 - Volume 1: Long Papers, Albuquerque, New Mexico, USA, April 29 - May 4, 2025*, pages 179–198. Association for Computational Linguistics.

Qiyuan Zhang, Yufei Wang, Yuxin Jiang, Liangyou Li, Chuhan Wu, Yasheng Wang, Xin Jiang, Lifeng Shang, Ruiming Tang, Fuyuan Lyu, and Chen Ma. 2025b. Crowd comparative reasoning: Unlocking comprehensive evaluations for LLM-as-a-judge. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5059–5074, Vienna, Austria. Association for Computational Linguistics.

Yuan Zhang, Chun-Kai Fan, Junpeng Ma, Wenzhao Zheng, Tao Huang, Kuan Cheng, Denis A. Gudovskiy, Tomoyuki Okuno, Yohei Nakata, Kurt Keutzer, and Shanghang Zhang. 2024. Sparsevlm: Visual token sparsification for efficient vision-language model inference. *arXiv preprint*, abs/2410.04417.

Mingyu Zheng, Xinwei Feng, Qingyi Si, Qiaoqiao She, Zheng Lin, Wenbin Jiang, and Weiping Wang. 2024. Multimodal table understanding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9102–9124, Bangkok, Thailand. Association for Computational Linguistics.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017a. Seq2sql: Generating structured queries from natural language using reinforcement learning. *Preprint*, arXiv:1709.00103.

Victor Zhong, Caiming Xiong, and Richard Socher. 2017b. Seq2sql: Generating structured queries from natural language using reinforcement learning. *arXiv preprint*, abs/1709.00103.

Fengbin Zhu, Ziyang Liu, Fuli Feng, Chao Wang, Moxin Li, and Tat-Seng Chua. 2024. TAT-LLM: A specialized language model for discrete reasoning over financial tabular and textual data. In *Proceedings of the 5th ACM International Conference on AI in Finance, ICAIF 2024, Brooklyn, NY, USA, November 14-17, 2024*, pages 310–318. ACM.

Yingjie Zhu, Xuefeng Bai, Kehai Chen, Yang Xiang, Youcheng Pan, Xiaoqiang Zhou, and Min Zhang. 2026. Decoupling skeleton and flesh: Efficient multimodal table reasoning with disentangled alignment and structure-aware guidance. *Preprint*, arXiv:2602.03491.

A Dataset Statistics

We summarize the datasets and question-type coverage used in our experiments. The collections combine tables with varying amounts of visual content and structural complexity; Table 9 reports the number of evaluation questions per dataset and split, while Table 10 lists the distribution of table types.

Dataset	EQ	AQ	IQ	VQ
WikiTQ	700	700	700	700
WikiSQL	700	700	700	700
HybridQA	700	700	700	343
FeTaQA	700	700	700	700
MMTBench	2232	737	685	365

Table 9: No of Queries per Dataset split.

Table Type	Number
entity+map	44
entity+chart	4
multiple_charts	23
single_chart	45
maps	17
viz	43
single_ent	169
multiple_ents	155

Table 10: Distribution of table types (from folders.json).

B Baselines

We provide additional implementation details and design choices for all baselines.

B.1 Partial Input

All images are replaced with deterministic placeholders of the form `ENTITY-N`, preserving positional structure while removing visual content. We adopt the 8-shot prompting setup from Mathur et al. (2024) to ensure comparability. This baseline serves as a strict lower bound, capturing how much the model can infer from text alone. Performance in this setting reflects how well the model can recover missing entities from the surrounding table context, as well as whether it can correctly guess or reconstruct the missing information. It also provides a rough measure of the model's parametric knowledge, including whether it has been exposed to similar data during training.

B.2 Interleaved Reasoner

We follow Mathur et al. (2024); Titiya et al. (2025) and serialize the full table as an interleaved sequence of text cells and images, processed end-to-end by the MLLM. We use the same 8-shot prompting configuration for fairness. This setup represents an upper bound in terms of information access but is inefficient due to long context length and redundant visual tokens. Any improvements over this baseline indicate more effective information selection rather than increased capacity.

B.3 Row-wise Retrieval

Each table row is rendered as a single image, and ColQwen2 (Faysse et al., 2025a) is used to retrieve the top-k most relevant rows given the query. We set $k = 5$, balancing recall and context budget: smaller k risks omitting relevant evidence, while larger k increases noise and token cost. The retrieved rows are passed to the Interleaved Reasoner. This baseline tests whether coarse-grained pruning at the row level is sufficient for multimodal reasoning.

B.4 Image+Text Retrieval

We independently retrieve top-k rows using (i) text similarity and (ii) image relevance via ColQwen2 (Faysse et al., 2025a), and take the union of both sets. We again use $k = 5$ for each modality, resulting in up to $2k$ rows before deduplication. This design improves recall over single-channel retrieval but may introduce redundancy. It evaluates whether separating modality-specific retrieval signals yields better coverage than unified ranking.

B.5 SparseVLM

We apply SparseVLM (Zhang et al., 2024) on top of the interleaved representation, pruning visual tokens within images based on text-image alignment scores. This operates at a finer granularity than row filtering, reducing visual token count while preserving potentially relevant regions. The original method is designed for a LLaVA-based architecture; however, since we work with Gemma, direct token removal is not supported. Instead, tokens identified for pruning are replaced with the average of the remaining visual tokens. This issue does not arise in the Qwen family, but we adopt this modification for fairness and to enable a consistent comparison across models.

We further cap pruning at a maximum of 30% of visual tokens per image. This choice is motivated by the fact that multiple images are associated with each table; higher pruning rates tend to remove useful information, while lower rates provide limited compression benefits. This baseline allows us to compare CAPTR against token-level compression methods and assess whether structural pruning (ours) is more effective than intra-image sparsification.

B.6 Weaver

Weaver (Khoja et al., 2025) is a modular pipeline that decomposes queries into step-by-step execu-

tion plans combining SQL operations and LLM-based reasoning. Structured components (e.g., filtering, aggregation) are handled via SQL, while semantic understanding is delegated to the LLM. Originally, Weaver is not designed for multimodal inputs; however, in our setup we extend it by introducing an MLLM-based reasoning module to handle visual and textual information jointly.

Unlike our setting, Weaver assumes access to structured tables and focuses on hybrid execution rather than multimodal pruning. We use default configurations as reported in prior work. This baseline provides insight into how far hybrid symbolic-neural pipelines can go without explicit multimodal selection mechanisms.

C Captioning Examples

This section shows representative caption styles used in the pruning stage (Naive, Detailed, Short).

C.1 Example 1: Image of a Logo of a Football Club

Figure 3: Example of a logo being captioned

Naive VARBERG FOTBOLL FRIDROTT GYMNASTIK

Detailed A circular logo featuring the letters "V" and "F" prominently displayed within a diamond shape. The diamond is divided into four quadrants, colored red and blue diagonally. The letter "V" is white and centered on the blue quadrant, while the letter "F" is white and centered on the red quadrant. The letters "G" and "I" are also white and appear on the remaining red and blue quadrants respectively. The diamond is encircled by two golden rings. Above the circle, curved text reads "VARBERG" in bold, dark letters. Below the circle, curved text reads "FOTBOLL • FRIIDROTT • GYMNASTIK" in smaller, dark letters. The overall color scheme is dominated by red, blue, gold, and white.

Short Varberg, logo, football, gymnastics, crest, blue, yellow, red, square, circle

C.2 Example 2: Image of a Flag of a US State

Figure 4: Example of a flag being captioned

Naive The state flag of Arizona.

Detailed The official state flag of Arizona, divided horizontally into two equal halves. The bottom half is a solid field of dark liberty blue. The top half features thirteen alternating rays of red and yellow that radiate from a central focal point, representing an Arizona sunset. Superimposed directly over the center of the flag is a large, five-pointed star colored in a solid copper-orange to represent Arizona's copper mining industry.

Short Arizona flag, copper star, sun rays, yellow, red, dark blue, southwest, emblem

D LLM-as-a-Judge

This section includes the LLM-as-a-Judge prompt used to evaluate semantic equivalence between model and oracle answers.

LLM-as-a-Judge Prompt

You are an impartial, precise, and expert judge.
Your task is to evaluate the factual correctness of a model's response against an oracle (ground-truth) answer in the context of a question asked about a table.

Gold Answer	Model Answer	EM	Sub. Match	LLM Judge
IBM	International Business Machines	0	0.18	1
Southern University and A&M College	The student attends Southern University.	0	0.51	1
Minardi Team USA	Team USA motorsport team	0	0.40	0
1:43.134	1 min 43.134 sec	0	0.58	1
Technical University of Darmstadt	Darmstadt Technical University	0	0.75	0
Michael Gary Scott	Micael J Scott	0	0.81	0

Table 11: Evaluation of lexical vs. semantic metrics.

Background Information
The model being evaluated answers questions based on a table where some cells contain images (e.g., logos, photos) instead of text.
The Oracle Answer represents the ground-truth entity or fact originally contained in the cell.
The model must interpret the image to identify this information.
Crucially, the Oracle Answer is only one possible textual realization of the underlying information.
Your goal is to determine whether the Model Answer identifies the same underlying entity or fact.

Input Data

- Table Metadata: `{table_metadata}`
- Question: `{question}`
- Oracle Answer (Ground Truth): `{oracle_answer}`
- Model Answer (Prediction): `{our_answer}`

Evaluation Criteria
You must determine whether the Model Answer is **CORRECT** or **INCORRECT**.
The prediction is **CORRECT** if and only if:

1. **Factual Accuracy:** The answer correctly addresses the question based on the ground truth.
2. **Equivalence:** The answer identifies the exact same entity, value, or fact as the Oracle Answer.

Guidelines for Equivalence and Specificity

1. **Named Entities** (organizations, people, locations, etc.)
   - The Model Answer does not need to match the specificity of the Oracle Answer as long as it correctly and unambiguously identifies the same entity.
   - Accept synonyms, acronyms, full names, or commonly recognized alternative names.
   - Over-specification is acceptable if the same entity is identified.
   - Under-specification is acceptable only if the entity remains unambiguous in context.
2. **Numerical and Date Information**
   - Numerical values and dates must match exactly, although formatting variations are acceptable.
3. **Formatting and Phrasing**
   - Minor punctuation or phrasing differences are acceptable if the meaning is preserved.

The prediction is INCORRECT if:

1. A different entity or value is identified.
2. The response contains unrelated hallucinated facts or unsupported details.
3. The response is too ambiguous or overly general to determine the intended entity.

Instructions

1. Analyze the Question and Oracle Answer to identify the required entity or fact.
2. Compare the Model Answer against the Oracle Answer using the equivalence rules.
3. Provide a detailed step-by-step chain-of-thought explaining the reasoning.

4. Explicitly discuss naming or specificity variations and explain whether they are acceptable.

5. Conclude with one of the following exact verdicts:

   - `[[CORRECT]]`
   - `[[INCORRECT]]`

E Implementation Details

This section describes the runtime environment, major libraries, and high-level inference settings used in our experiments.

We use vLLM (Kwon et al., 2023) for inference. Generation hyperparameters were kept consistent across all experiments to avoid tuning bias: maximum output length is capped at 1500 tokens per call, temperature is set to 0.1, nucleus sampling uses $p = 0.95$, and we use a small top-k of 0.4 with a repetition penalty of 1.0. With Gemma 3 27B (Gemma Team, 2025), CAPTR runs on a single H200 GPU and achieves approximately 1500 output tokens per second; for the much longer interleaved-reasoner prompts generation speed is typically between 50 and 100 tokens per second. SQL- and symbolics-based pruning steps use conservative execution limits (single-query row caps and timeouts) to avoid uncontrolled scans during preprocessing.

F Weaver Failures

The original Weaver pipeline includes a plan verification step after the planner stage, in which a second LLM call reviews and refines the generated reasoning plan before code execution. To assess whether this step is beneficial in our multimodal setting, we ran a controlled ablation on 500 questions (25 per question type across MMTabQA and MMTBench datasets), stratified equally across Explicit, Answer-Mention, Implicit, and Visual question types, using Qwen3.5-32B-Instruct as the verifier model and the same normalized exact-match metric used for both.

As shown in Table 12, the verifier improves performance on Explicit questions (+4.8pp overall), where plan structure is straightforward, but consistently degrades Implicit (−7.2pp) and Visual (−7.2pp) question types, precisely those requiring multi-step reasoning over image-bearing cells. We attribute this to two failure modes: the verifier tends to oversimplify plans that require subtle visual lookups, and for Implicit questions, it frequently misidentifies the correct reasoning chain when answers must combine visual and textual evidence. WikiSQL shows the largest drop (−15.0pp), while FeTaQA is the only dataset with a net benefit (+4.0pp), consistent with its text-heavy, free-form answer style. Given that three of four question types are hurt and the aggregate effect is −3.2pp, we omit the verifier from the Weaver baseline reported in the main paper.

Dataset	Q Type	W/ Verifier	W/o Verifier	Δ (pp)
WikiTQ	EQ	28.0	36.0	−8.0
	AQ	20.0	20.0	0.0
	IQ	52.0	52.0	0.0
	VQ	12.0	8.0	+4.0
	Total	**28.0**	**29.0**	−1.0
WikiSQL	EQ	32.0	40.0	−8.0
	AQ	8.0	20.0	−12.0
	IQ	44.0	64.0	−20.0
	VQ	20.0	40.0	−20.0
	Total	**26.0**	**41.0**	−15.0
HybridQA	EQ	60.0	52.0	+8.0
	AQ	32.0	32.0	0.0
	IQ	52.0	64.0	−12.0
	VQ	36.0	44.0	−8.0
	Total	**45.0**	**48.0**	−3.0
FeTaQA	EQ	40.0	24.0	+16.0
	AQ	28.0	16.0	+12.0
	IQ	28.0	28.0	0.0
	VQ	28.0	40.0	−12.0
	Total	**31.0**	**27.0**	+4.0
MMTBench	EQ	36.0	20.0	+16.0
	AQ	16.0	32.0	−16.0
	IQ	28.0	32.0	−4.0
	VQ	20.0	20.0	0.0
	Total	**25.0**	**26.0**	−1.0

Table 12: Weaver verifier ablation across datasets and question types (Qwen3.5). All values are accuracy (%). W/o Verifier corresponds to the Weaver results reported in the main paper.

G Additional Results

This appendix contains extended tables, breakdowns, and ablation details to aid reproducibility.

G.1 Effect of pruning for MMTabQA across all Data Splits and all captioning strategy

G.1.1 MMTabQA-WikiTQ

Model	% Pruned	EQ	AQ	IQ	VQ
Gemma3	Rows	63.69	73.52	58.58	66.81
	Cols	62.97	62.17	65.23	59.86
	Tokens	82.27	85.65	83.91	81.64
Qwen3VL	Rows	65.57	81.00	67.32	62.59
	Cols	64.41	64.33	35.34	61.46
	Tokens	83.42	90.20	87.20	80.06
Qwen3.5	Rows	68.95	71.83	60.95	67.42
	Cols	64.53	62.31	66.17	61.51
	Tokens	84.75	85.61	84.70	82.73

Table 13: WikiTQ-Short: Effect of pruning different components on question categories.

Model	% Pruned	EQ	AQ	IQ	VQ
Gemma3	Rows	64.36	72.94	56.52	65.59
	Cols	63.59	62.96	65.25	60.03
	Tokens	79.19	83.42	82.22	77.78
Qwen3VL	Rows	63.79	80.91	67.07	56.61
	Cols	64.18	62.26	64.56	60.97
	Tokens	79.01	88.72	85.25	74.17
Qwen3.5	Rows	69.89	70.85	60.45	66.72
	Cols	65.23	62.62	66.34	60.85
	Tokens	82.64	81.89	83.56	78.72

Table 14: WikiTQ-Detailed: Effect of pruning different components on question categories.

Model	% Pruned	EQ	AQ	IQ	VQ
Gemma3	Rows	60.62	73.05	59.58	66.00
	Cols	61.90	62.64	65.74	60.24
	Tokens	80.20	85.81	84.24	81.66
Qwen3VL	Rows	64.33	80.37	66.39	70.43
	Cols	64.91	63.29	64.88	61.44
	Tokens	83.14	89.83	86.63	87.06
Qwen3.5	Rows	64.91	72.59	60.47	70.74
	Cols	64.07	63.11	66.54	60.78
	Tokens	83.45	86.29	84.93	83.73

Table 15: WikiTQ-Naive: Effect of pruning different components on question categories.

G.1.2 MMTabQA-WikiSQL

Model	% Pruned	EQ	AQ	IQ	VQ
Gemma3	Rows	63.44	86.00	73.77	80.92
	Cols	59.83	61.40	66.61	57.29
	Tokens	83.13	92.97	89.77	88.79
Qwen3VL	Rows	58.98	86.86	79.38	60.02
	Cols	60.38	61.26	68.29	58.32
	Tokens	80.41	93.21	92.22	78.92
Qwen3.5	Rows	63.05	90.75	81.77	68.44
	Cols	60.85	61.95	68.43	58.16
	Tokens	82.72	94.58	92.96	80.63

Table 16: WikiSQL-Short: Effect of pruning different components on question categories.

Model	% Pruned	EQ	AQ	IQ	VQ
Gemma3	Rows	63.17	84.74	75.06	77.09
	Cols	60.55	60.94	67.14	57.89
	Tokens	82.32	92.05	89.96	86.01
Qwen3VL	Rows	49.59	86.50	78.28	60.82
	Cols	60.06	64.49	68.83	57.89
	Tokens	74.15	92.82	92.02	76.50
Qwen3.5	Rows	64.56	90.82	82.04	71.05
	Cols	60.52	61.40	67.76	58.81
	Tokens	81.68	94.64	92.22	80.58

Table 17: WikiSQL-Detailed: Effect of pruning different components on question categories.

Model	% Pruned	EQ	AQ	IQ	VQ
Gemma3	Rows	58.26	85.58	72.29	81.88
	Cols	59.85	61.42	66.38	57.56
	Tokens	80.41	92.70	88.82	89.16
Qwen3VL	Rows	53.47	86.39	76.92	78.72
	Cols	59.81	61.12	67.81	58.27
	Tokens	76.92	92.92	90.43	87.32
Qwen3.5	Rows	55.76	90.88	79.19	73.18
	Cols	60.94	62.46	68.08	57.85
	Tokens	79.14	94.79	92.14	83.91

Table 18: WikiSQL-Naive: Effect of pruning different components on question categories.

G.1.3 MMTabQA-HybridQA

Model	% Pruned	EQ	AQ	IQ	VQ
Gemma3	Rows	82.20	85.88	84.56	83.23
	Cols	49.77	52.91	46.45	46.45
	Tokens	88.32	90.31	88.51	88.76
Qwen3VL	Rows	82.00	83.11	82.84	70.64
	Cols	49.19	52.65	45.71	45.68
	Tokens	87.39	88.33	86.97	80.42
Qwen3.5	Rows	84.36	82.14	83.78	71.48
	Cols	48.90	53.47	47.31	47.37
	Tokens	87.59	88.14	88.61	81.24

Table 19: HybridQA-Short: Effect of pruning different components on question categories.

Model	% Pruned	EQ	AQ	IQ	VQ
Gemma3	Rows	78.32	82.98	82.61	78.89
	Cols	48.81	52.14	46.07	46.11
	Tokens	84.84	86.72	86.45	84.75
Qwen3VL	Rows	79.03	77.67	79.66	66.57
	Cols	48.78	51.80	45.27	45.81
	Tokens	84.98	83.19	84.32	74.49
Qwen3.5	Rows	82.32	81.57	82.34	73.91
	Cols	49.83	52.87	47.60	47.37
	Tokens	86.23	85.73	86.09	80.58

Table 20: HybridQA-Detailed: Effect of pruning different components on question categories.

Model	% Pruned	EQ	AQ	IQ	VQ
Gemma3	Rows	83.65	84.61	85.97	84.78
	Cols	49.40	52.93	46.16	47.10
	Tokens	89.00	89.97	90.26	89.20
Qwen3VL	Rows	82.00	80.67	82.09	76.88
	Cols	49.09	52.21	46.19	46.61
	Tokens	87.66	87.04	87.44	83.63
Qwen3.5	Rows	81.38	81.89	83.29	75.45
	Cols	49.49	53.06	46.77	47.86
	Tokens	87.40	88.17	88.13	83.28

Table 21: HybridQA-Naive: Effect of pruning different components on question categories.

G.1.4 MMTabQA-FetaQA

Model	% Pruned	EQ	AQ	IQ	VQ
Gemma3	Rows	61.34	63.25	55.14	60.90
	Cols	46.28	47.20	48.39	42.34
	Tokens	74.54	78.45	74.22	74.09
Qwen3VL	Rows	68.80	71.60	64.01	65.16
	Cols	46.15	47.60	49.45	44.09
	Tokens	78.41	83.07	78.83	76.20
Qwen3.5	Rows	73.55	73.46	67.89	74.57
	Cols	46.55	45.66	50.21	44.25
	Tokens	81.40	83.18	80.20	81.97

Table 22: FetaQA-Short: Effect of pruning different components on question categories.

Model	% Pruned	EQ	AQ	IQ	VQ
Gemma3	Rows	55.53	59.53	52.97	54.56
	Cols	47.47	47.50	48.65	41.60
	Tokens	68.29	73.84	72.73	66.14
Qwen3VL	Rows	62.59	70.89	62.80	59.38
	Cols	46.37	46.41	48.78	44.02
	Tokens	71.83	80.59	76.62	69.06
Qwen3.5	Rows	70.05	75.44	67.31	68.22
	Cols	47.29	45.78	49.50	43.31
	Tokens	76.57	82.64	79.88	75.71

Table 23: FetaQA-Detailed: Effect of pruning different components on question categories.

Model	% Pruned	EQ	AQ	IQ	VQ
Gemma3	Rows	58.59	64.41	55.53	59.77
	Cols	47.89	47.80	48.85	42.23
	Tokens	75.08	79.64	76.56	74.70
Qwen3VL	Rows	65.88	72.03	63.73	63.88
	Cols	46.05	46.06	49.45	43.04
	Tokens	77.74	82.89	80.02	79.09
Qwen3.5	Rows	69.30	74.65	64.61	72.11
	Cols	47.57	45.91	48.88	42.13
	Tokens	80.60	84.47	79.07	81.62

Table 24: FetaQA-Naive: Effect of pruning different components on question categories.

G.1.5 MMTBench

Model	% Pruned	EQ	AQ	IQ	VQ
Gemma3	Rows	64.37	62.88	47.37	52.52
	Cols	55.54	60.07	66.74	57.90
	Tokens	82.23	80.28	76.71	71.35
Qwen3VL	Rows	55.58	55.78	43.72	46.05
	Cols	59.58	55.08	66.77	55.18
	Tokens	77.62	78.33	75.15	65.83
Qwen3.5	Rows	66.88	66.67	55.11	54.83
	Cols	60.06	55.89	67.09	56.61
	Tokens	83.12	83.34	80.10	74.22

Table 25: MMTBench-Short: Effect of pruning different components on question categories.

Model	% Pruned	EQ	AQ	IQ	VQ
Gemma3	Rows	61.81	62.78	49.21	51.27
	Cols	59.75	55.62	66.23	57.08
	Tokens	76.88	79.67	74.11	69.00
Qwen3VL	Rows	57.26	56.84	42.92	45.38
	Cols	59.16	55.16	65.99	56.59
	Tokens	73.71	76.22	67.80	64.75
Qwen3.5	Rows	67.91	67.70	56.22	57.02
	Cols	60.34	56.66	67.28	58.12
	Tokens	79.74	79.88	75.78	72.73

Table 26: MMTBench-Detailed: Effect of pruning different components on question categories.

Model	% Pruned	EQ	AQ	IQ	VQ
Gemma3	Rows	59.83	64.09	44.57	52.32
	Cols	60.53	56.23	66.07	57.27
	Tokens	80.69	83.29	77.44	70.27
Qwen3VL	Rows	54.82	53.07	41.14	44.57
	Cols	59.86	55.58	66.58	55.32
	Tokens	77.94	77.59	75.41	65.08
Qwen3.5	Rows	65.51	67.01	55.55	53.38
	Cols	60.58	56.12	66.97	57.27
	Tokens	83.23	83.52	80.62	74.21

Table 27: MMTBench-Naive: Effect of pruning different components on question categories.

Model	Setting	MMTabQA-WikiTQ				MMTabQA-WikiSQL				MMTabQA-HybridQA				MMTBench			
		EQ	AQ	IQ	VQ	EQ	AQ	IQ	VQ	EQ	AQ	IQ	VQ	EQ	AQ	IQ	VQ
Gemma3	Naive (w/)	50.86	57.14	58.86	51.00	55.57	63.00	67.30	58.86	80.00	66.43	79.43	62.86	44.77	46.56	32.16	34.71
	Naive (w/o)	55.29	47.29	56.43	48.00	54.86	60.00	66.03	56.57	72.43	59.73	72.14	60.14	39.74	38.57	29.04	30.00
	Short (w/)	66.71	56.57	66.29	55.71	64.86	60.71	74.84	63.44	81.81	67.71	81.29	68.86	42.88	44.08	31.34	33.53
	Short (w/o)	59.63	53.43	55.14	54.43	59.57	60.71	66.98	61.68	74.17	66.09	74.71	64.57	43.02	42.70	30.19	31.91
	Detailed (w/)	67.29	55.00	67.86	57.14	64.29	64.43	75.56	64.29	77.71	68.57	81.00	67.29	43.78	44.90	30.53	34.56
	Detailed (w/o)	55.00	53.86	53.71	54.29	60.00	61.43	69.84	62.43	73.14	66.86	75.00	64.43	43.74	44.35	28.29	32.74
Qwen3	Naive (w/)	57.14	47.86	60.57	50.29	53.71	57.29	70.79	55.29	80.43	64.14	79.57	60.29	40.86	39.94	30.66	31.32
	Naive (w/o)	56.57	44.43	60.43	49.29	53.71	50.86	69.21	52.57	73.29	61.14	70.57	57.57	35.11	35.81	24.69	24.85
	Short (w/)	70.14	54.71	70.14	58.00	65.71	62.00	77.06	65.86	81.86	71.86	82.14	63.29	38.35	39.39	27.14	29.41
	Short (w/o)	59.00	50.14	61.57	50.43	59.29	56.43	70.79	54.29	75.00	68.29	73.29	62.00	38.08	37.19	27.00	28.38
	Detailed (w/)	69.86	52.28	69.14	57.85	63.14	59.00	76.11	65.14	81.86	70.29	81.71	66.43	43.38	41.87	31.21	35.15
	Detailed (w/o)	59.14	51.57	59.29	53.43	58.57	57.29	68.57	57.86	76.14	67.86	75.86	62.86	41.18	40.77	28.36	31.76
Qwen3.5	Naive (w/)	59.29	51.43	61.14	52.57	54.29	53.14	67.94	58.14	81.57	59.71	80.57	62.29	43.95	42.12	28.26	33.82
	Naive (w/o)	59.29	51.29	56.71	47.57	53.43	51.29	68.89	46.29	71.43	41.71	72.57	57.86	36.87	37.74	24.69	26.18
	Short (w/)	66.71	56.29	70.29	54.71	67.14	63.14	77.06	62.86	83.00	70.43	83.86	64.43	45.16	42.14	28.49	29.71
	Short (w/o)	59.14	52.57	55.14	52.86	61.71	58.29	72.06	60.71	80.43	66.00	79.14	62.57	39.20	39.67	28.36	28.82
	Detailed (w/)	65.14	55.71	67.43	53.71	65.43	61.00	76.06	62.43	81.71	69.71	82.29	67.86	49.13	49.63	34.85	36.03
	Detailed (w/o)	56.86	52.57	61.43	53.29	56.00	56.71	72.06	58.71	73.29	65.14	74.14	59.29	42.03	41.60	27.68	33.09

Table 28: Captioning strategy ablation with and without image reintroduction (w/ vs w/o).

G.2 Image Reintroduction for all captioning strategies

H Prompts for Interleaved Reasoner

H.1 MMTabQA

MMTabQA Interleaved Baseline Prompt

Answer in a sentence, using the table data given. The table consists of data in the form of text and images. Each row of the table has been represented using [] with data for each column in the row separated by a semi-colon.

In the table, some entities (mentioned in text form originally) have been replaced by images that represent them. Based upon the context of the table while using real-world knowledge, your task is to identify the entities corresponding to the images in the table and answer the question. You must perform this task in the following steps:

Step 1: Reason about what should be the answer to the question by identifying the relevant entities represented by images using the context of the table and the question. The reasoning should be detailed and should be based upon the context of the table and the question, using real-world knowledge for answering the question.

IMPORTANT: You must explore any kind of reasoning – numerical, logical, knowledge-based needed for disambiguating the entities and answering the question. **Step 2:** Based upon the reasoning provided, provide the answer to the question.

You are also provided with some question-answer examples for better understanding the format of providing the answer:

{Few-Shot Examples}

Now, based upon the examples given above, you must understand the image given and follow the steps 1-2 to answer the question corresponding to the table represented in the image.

It is **IMPORTANT** that you perform all the both the steps to the best possible extent to get the correct answer. You must follow the format of answers as demonstrated by the examples above. IMPORTANT: You must give the answer in the format 'Step 2:',

Table context: **{Table Metadata}**

Question: **{QUESTION}**

Table: **{INTERLEAVED TABLE}**

H.2 MMTBench

MMTBench Interleaved Baseline Prompt

You will be provided a pipe-seperated table where some cells are images. Your task is to:

Step 1: UNDERSTAND THE TABLE CONTEXT - Carefully analyze the table structure and understand the intricate relationship between image and text.

Step 2: ANALYZE THE QUESTIONS - Read all the questions provided and explore **ALL TYPES OF REASONING** to find answers.

Step 3: PROVIDE ANSWERS IN FORMAT Ensure that all answers adhere strictly to the FORMAT specified. Avoid deviating from this format or including unnecessary explanations.
{Answer Formatting Guidelines}
ALWAYS PROVIDE YOUR ANSWERS IN THIS FORMAT.

IMPORTANT ALL answers are there in the image.

Now I will provide you with the table.
Note: within these prompt examples, SQL fragments can be highlighted using ... and user variables using ... for emphasis.
{INTERLEAVED TABLE}
For this table, answer the following question.
{QUESTION}
Based on the steps I mentioned above, answer the question.

I Prompts for Table Pruning

Column Selection using SQL

You are given the table schema, and the table along with the corresponding statement. Write a simple SQLite program for selecting only the required columns to help answer the question correctly.

The SQLite program need not directly answer the question.
Assume you always have enough information when executing the SQLite query. Fuzzy match data if unsure.
If a column name contains spaces or special characters (such as `/`, `-`, `%`, or parentheses), wrap that identifier in backticks in SQL (for example: `‘town/city‘`).

1. **Plan:**
   - Identify critical values and ranges from the table related to the statement.
   - Make use of domain knowledge to determine the correct approach for solving the question.
   - Always select the column containing special aggregate values such as `total`.
2. **Retrieval:**
   - Generate a simple SQL program extracting the relevant columns.
   - SQL format: use the ... macro for SQL queries and ... for user variables. Example: `SELECT <COLUMNS> FROM w;`
   - Evidence format: `f_col([column names])`

Response Format
Begin your response with `Output` and include:

- **Plan:** Write the plan for column extraction along with a reasoning chain.
- **Retrieval:** Write a simple SQL query.
- **Evidence:** `f_col([column names])`

Before returning the answer, review your outputs and ensure that:

- All instructions have been followed.
- Only columns are selected in the `SQL` query.

<input>

```
CREATE TABLE Płock Governorate(
    row_id int,
    language text,
    number int,
    percentage (%) text,
    males int,
    females int)
```

Q: how many male and female german speakers are there?

<output>

```
Output:
1. Plan:
 - Need male and female counts.
 - 'language' contains aggregate
   value 'total'.
 - Select language, males,
   females.

2. Retrieval:
 - SQL:
   SELECT `language`,
          `males`,
          `females`
   FROM w;

 - Evidence:
   f_col(['language',
          'males',
          'females'])

\inputlabel
\begin{verbatim}
CREATE TABLE Electricity in Sri Lanka(
    row_id int,
    filledcolumnname text,
    2005 int,
    2006 int,
    2007 int,
    2008 int,
    2009 int)
```

Q: how did the other renewables fare till 2009?

<output>

```
Output:
1. Plan:
 - Analyze 'other renewables'
   till 2009.
 - Select yearly columns up to
   2009.

2. Retrieval:
 - SQL:
   SELECT `filledcolumnname`,
          `2005`,
          `2006`,
          `2007`,
          `2008`,
          `2009`
   FROM w;

 - Evidence:
   f_col(['filledcolumnname',
          '2005',
          '2006',
          '2007',
          '2008',
          '2009'])
```

Column Selection using Text

You must select all relevant columns from the table and obtain a relevant sub-table for answering the question.
The sub-table should then be used to answer the question.
Please ensure that the final answer matches the reasoning.
Your task is to perform the following actions:

1. **Plan:**
   - Identify critical values and ranges from the table related to the question.
   - Note all required columns needed from the table.
2. **Retrieval:**
   - Identify all table columns containing question-relevant information.
   - Explain the relevance of each selected piece of evidence.

Response Format

Your response must begin with `Output` and always include:

- **Plan:** Write the plan for column extraction along with a reasoning chain.

- **Retrieval:** Based on the plan, identify the columns followed by a reasoning chain.
- **Evidence:** extttf_col([column names])

Please output only the column names inside f_col([column names]), with no additional details.

<input>

```
/*
row: row 1 | row 2 | row 3 | row 4 | row 5
pick: 1 | 2 | 3 | 4 | 5
player: jay washington | alex cabagnot |
        dennis miranda | ato ular |
        kameron vales
country of origin:
united states | united states |
philippines | philippines |
united states
pba team:
air21 express | sta lucia realtors |
coca-cola tigers |
blackwater bossing |
converge fiberxers
college:
eckerd | hawaii - hilo |
feu | mpbl | ateneo
*/

Q: Which college has most players
belonging to asian countries?

\outputlabel
Output:
1. Plan:
 - Identify colleges associated
   with players from Asian
   countries.
 - Use player, country of origin,
   and college columns.

2. Retrieval:
Explanation:
 - 'player' identifies players.
 - 'country of origin' identifies
   Asian countries.
 - 'college' identifies the
   associated colleges.

Evidence:
f_col([player,
       country of origin,
       college])
```

<input>

```
/*
row : row 0 | row 1 | row 2
season : 2003 | 2004 | 2005
division : 1 | 1 | 1
tms. : 12 | 13 | 13
pos. : 11 | 10 | 8
fa cup :
quarter final |
round of 32 |
quarter final
afc cl : none | none | none
*/

Q: how far did they make it
in the fa cup after 2009?

\outputlabel
Output:
1. Plan:
 - Filter rows using the
   'season' column for years
   after 2009.
 - Use the 'fa cup' column
   for tournament progress.

2. Retrieval:
Explanation:
 - 'season' enables temporal
   filtering.
 - 'fa cup' contains the
   tournament result.

Evidence:
f_col([season,
       fa cup])

\inputlabel
\begin{verbatim}
/*
rank: 1 | 2 | 3 | 4 | 5
lane: 6 | 5 | 4 | 1 | 8
athlete:
manjeet kaur |
olga tereshkova |
pinki pramanik |
```

```
tang xiaoyin |
marina maslyonko
time:
52.17 | 51.86 |
53.06 | 53.66 |
53.99
country:
india | kazakhstan |
india | china |
kazakhstan
*/

Q: who finished immediately
after manjeet kaur?

\outputlabel
Output:
1. Plan:
 - Determine finishing order
   using ranks.
 - Identify athlete names.

2. Retrieval:
Explanation:
 - 'rank' gives finishing order.
 - 'athlete' provides names.

Evidence:
f_col([rank,
       athlete])
```

Row Selection using SQL

You are given a table schema and a question. Write exactly one `SQLite SELECT` statement that extracts the rows needed to answer the question.

Rules:

- Output only a single line beginning with `SQL:`
- Do not include decomposition, explanation, markdown, or code fences.
- Use only one `SELECT` statement.
- Use the column names exactly as given.
- If a column name contains spaces or special characters, wrap it in backticks.
- Prefer the smallest row subset that still preserves the relevant evidence.

Format:

`SQL: SELECT <column names> FROM w WHERE <conditions>;`

Row Selection using Text

Task Description:
You are provided with an initial response containing rows from a table along with a question.
Your task is to provide feedback about the initial response and identify omitted rows, if any.

Procedure:

1. **Assessment**
   - Assess the question type.
2. **Feedback**

   Based on the question, provide feedback.
   - **Calculation/comparison-based question**
     - Check whether the relevant rows are limited or whether the entire table is required.
   - **Order/temporal-based question**
     - Check whether the response includes the row containing the mentioned entity.
     - Check whether the response contains rows satisfying the required relative position.
   - **Entity-specific question**
     - Check whether any row related to the question-relevant entity has been omitted.

   Add rows to the response if they are not present.

Rows Format:

`Rows: f_row([rows])`

Response Format

Begin your response with `<output>` and include:

- **Assessment:** Assess the question and the response.
- **Feedback:** Provide step-by-step feedback on the suggested answers.
- **Rows:** f_row([rows])

Be careful to follow all instructions exactly.

<input>

```
/*
col : Rank | Athlete
row 0 : 1 | Manjeet Kaur
row 1 : 2 | Olga Tereshkova
row 2 : 3 | Pinki Pramanik
row 3 : 4 | Tang Xiaoyin
row 4 : 5 | Marina Maslyonko
*/

Q: tell me who finished immediately
after manjeet kaur

Initial Response: ['row 1']

\outputlabel
Output:
1. Assessment:
 - The question is temporal.
 - We need the row containing
   Manjeet Kaur and the row
   immediately after it.

2. Feedback:
 - Is the row for Manjeet Kaur
   included?
   A: No; add(row 0)

 - What is the row after
   Manjeet Kaur?
   A: row 1; add(row 1)

 - Is row 1 already included?
   A: Yes

Rows:
f_row([row 0, row 1])
```

<input>

```
/*
col : away team |
      away team score

row 0 : melbourne |
        13.11 (89)

row 1 : footscray |
        11.13 (79)

row 2 : fitzroy |
        13.22 (100)

row 3 : collingwood |
        17.9 (111)

row 4 : carlton |
        23.11 (149)

row 5 : essendon |
        12.15 (87)
*/

Q: what is the away team
with the highest score?

Initial Response:
(['row 0', 'row 1',
  ... , 'row 5'])

\outputlabel
Output:
1. Assessment:
 - The question is
   calculation-based.
 - The comparison requires
   all rows.

2. Feedback:
 - Are all rows included?
   A: No; add(*)

Rows:
f_row([*])
```