TABQAWORLD: Optimizing Multimodal Reasoning for Multi-Turn Table Question Answering

Anonymous ACL submission

Abstract

Multi-turn table reasoning agents must repeatedly read, transform, and verify evolving table states across tool-use trajectories. While existing methods improve reasoning accuracy through tool use and reward modeling, they rely on fixed text serializations that distort table topology, causing representation errors that accumulate across steps. Such accumulation is alleviated by tabular grounding at additional inference cost, limiting real-world deployment. To address this, we introduce TABQAWORLD, a training-free framework that jointly optimizes what the agent observes and how many actions it executes per step. For representation, an action-conditioned multimodal selection policy dynamically switches between visual and textual views to maximize readout reliability. For estimation, metadata-guided trajectory optimization batches low-complexity actions using lightweight table metadata, reducing conversation turns and latency. TABQAWORLD achieves a **5.42%** accuracy gain and **33.35%** inference latency reduction over static unoptimized baselines in controlled within-backbone ablations, and outperforms the strongest adaptive-modality and trained trajectory-optimizing baselines by up to **10.81%** in cross-paper comparisons. Our code is available at https://anonymous.4open.science/r/tabworldqa-8C90/.

1 Introduction

Multi-turn table reasoning requires agents to execute sequences of tool-use operations over structured tables to answer complex questions (Wang et al., 2024; Yang et al., 2025d; Zhang et al., 2025). Unlike free-form text, tables encode information in 2D row-column grids, so maintaining an accurate view of the evolving table state across each reasoning step is both necessary and non-trivial (Wang et al., 2025b). Consider a representative example: given a table of LNG plants and the question "How many plants are in Algeria?", a multi-step agent must filter relevant rows, count them, and produce a final answer, while maintaining an accurate view of the evolving table state after each operation.

In practice, multi-step table reasoning struggles to track evolving table states (Yu et al., 2026). Existing methods share three limitations: (1) tables encode data in 2D grids that text tokens cannot natively preserve (Singha et al., 2023), (2) fixed serializations such as `JSON` or `Markdown` flatten this structure, weakening row-column grounding and cell retrieval (Wang et al., 2025b), and (3) existing fixes such as stepwise reflection (Ji et al., 2025) and process reward modeling (Zou et al., 2026), address individual links but require trained modules, adding inference or training cost (Jiang et al., 2023; Nguyen et al., 2025; Zhou et al., 2025b).

These limitations expose two challenges: (1) *Representation*: grounding errors compound across steps (Laban et al., 2026), and no single serialization reliably preserves 2D structure for all operation types. (2) *Estimation*: full table-state prediction is brittle, as frontier LLMs hallucinate cell values. Early errors cascade without state verification. This motivates a unified, training-free approach that controls both observation quality and trajectory length.

We introduce TABQAWORLD (Figure 1), which jointly optimizes *representation* (what to observe) and *estimation* (what to expect) (Xu et al., 2026; Yu et al., 2026) via two training-free components. (1) An **action-conditioned multimodal selection policy** routes each step's table observation to the most reliable modality (text, image, or multimodal) based on the planned action. (2) **Metadata-guided trajectory optimization** batches low-risk actions by verifying state consistency via lightweight table metadata, reducing conversation turns and latency.

TABQAWORLD achieves strong and consistent performance across evaluated benchmarks. Backbone-

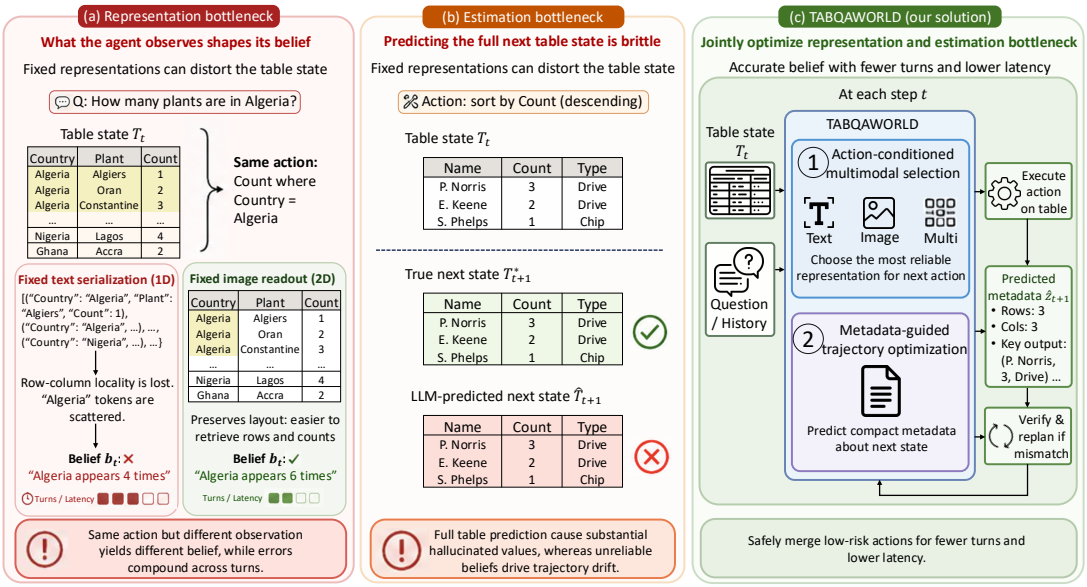

Figure 1: TABQAWORLD resolves two compounding bottlenecks in multi-turn table reasoning. *Representation bottleneck:* fixed serializations distort row-column structure while fixed image readouts can incur resolution or cost errors. No single format reliably serves all operation types. *Estimation bottleneck:* full next-table prediction is brittle and prone to hallucination. TABQAWORLD addresses both by conditioning modality selection on the planned action at each step and using lightweight metadata prediction to gate and verify action batching.

controlled ablations show a **5.42%** accuracy gain from adaptive modality routing and a **33.35%** end-to-end latency reduction from the full framework (Section 7). Cross-paper comparisons show up to **10.81%** gain over adaptive modality baselines, with backbone differences disclosed in Section 6. Our contributions are:

1. **Bottleneck analysis.** We identify representation and estimation as two complementary bottlenecks governing multi-turn table reasoning reliability and efficiency (Section 3).

2. **Training-free framework: TABQAWORLD.** A plug-and-play framework with (1) an action-conditioned multimodal selection policy routing each step's observation to the most reliable modality, and (2) metadata-guided trajectory optimization compressing low-risk actions to reduce latency (Section 4).

3. **Strong and consistent performance.** Controlled within-backbone ablations show consistent gains over fixed-modality and routing baselines (Section 5).

2 Related Work

Multi-turn table reasoning has been advanced through tool-use agents (Wang et al., 2024; Yang et al., 2025d; Ji et al., 2025) and post-training trajectory optimizers (Zou et al., 2026; Guo et al., 2026), both of which are direct baselines for TABQAWORLD. Representation format is known to affect downstream reasoning quality (Deng et al., 2024; Xu et al., 2026), motivating adaptive modality selection for table tasks (Xing et al., 2026). TABQAWORLD extends this by conditioning modality on the planned action at each step rather than once per question. For trajectory efficiency, step-level reward signals (Zou et al., 2026) and table pruning (Guo et al., 2026) improve reasoning quality but require trained modules. TABQAWORLD instead uses lightweight metadata consistency as a training-free batching signal. Additional related work on table pre-training, semantic parsing, and multi-step planning is in Appendix B.

3 Motivating Analysis: Bottlenecks in Multi-turn Table Reasoning

3.1 Preliminary

We follow standard table reasoning setup (Wang et al., 2024; Yang et al., 2025d). A table $T = (H, R)$ has headers H and rows R. Given question q and initial state T_0, the agent executes a trajectory $\tau = \{a_1, \ldots, a_S\}$ of tool-use actions, observing a modality-conditioned encoding $o_t = g(T_t; m_t)$ at each step t. We additionally define $z_t = \phi(T_t)$ as a low-dimensional metadata projection of the table state for Section 4's lightweight trajectory verification. Full formulation is in Appendix A.1.

Table 1: Vision-supported representations improve reasoning accuracy at modest token cost. Multimodal input achieves the best balance across benchmarks.

	WTQ	MMQA	MMTU
Accuracy ± SD under different perceived modality			
`json`	0.722 (0.032)	0.594 (0.035)	0.303 (0.033)
`latex`	0.707 (0.032)	0.574 (0.035)	0.333 (0.034)
`Markdown`	0.712 (0.032)	0.584 (0.035)	0.364 (0.034)
Image	**0.818 (0.027)**	0.614 (0.035)	**0.374** (0.034)
Multimodal	0.788 (0.029)	**0.635 (0.034)**	0.364 (0.034)
Accuracy ± SD under different estimated state			
No estimate	**0.818 (0.027)**	0.614 (0.035)	**0.374** (0.034)
Table estimate	0.753 (0.031)	0.543 (0.036)	0.303 (0.033)
Token costs (in 10^3 tokens) / Conversation Turns			
`json`	14.6 / 8.83	7.4 / 5.13	23.5 / 3.55
`latex`	12.6 / 9.53	6.1 / 5.20	27.5 / 3.41
`Markdown`	11.8 / 9.01	6.5 / 4.54	17.5 / 4.16
Image	8.4 / 5.58	**5.3** / 2.86	**16.8** / 2.16
Multimodal	**6.0** / 4.66	5.8 / 2.41	17.0 / 2.51

3.2 Pilot Study

Table reasoning faces two compounding failure modes: unreliable state readout (misunderstanding due to suboptimal encoding modality) and trajectory drift (executing too many turns without state verification). To understand existing bottlenecks, we study one key question: *to what extent does table representation affect information accessibility?* We extend prior work's finding that representation choice affects extraction quality (Wang et al., 2025b) to the stepwise reasoning setting. Specifically, we compare text-based serializations (`json`, `latex`, and `markdown`) against vision-supported parsing, including an image view (table rendered as `jpg`) and a multimodal view containing image and text headers. We analyze using Qwen3-VL-8B (Bai et al., 2025a) on three complex reasoning datasets, namely WikiTableQuestions (WTQ) (Pasupat and Liang, 2015), MMQA (Wu et al., 2025b), and MMTU (Xing et al., 2025b), with the latter two datasets released in 2025 to reduce contamination risk relative to established benchmarks. Table 1 shows a consistent accuracy improvement and compute reduction with vision-supported representations.

Observation. Vision-supported representations often improve accuracy and reduce conversation turns. Image leads on WTQ and MMTU while multimodal leads on MMQA, but visual grounding reliably improves both accuracy and turn efficiency.

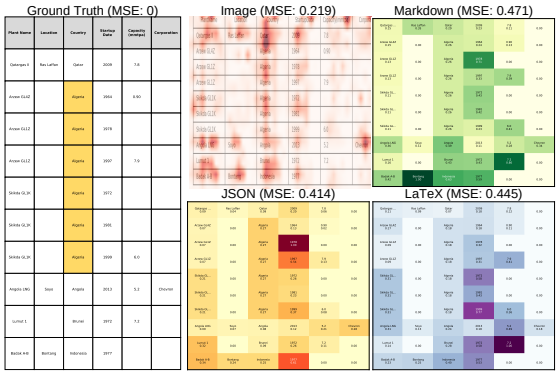

Figure 2: An illustrative example of how image-based parsing facilitates more human-preferred column attention than text-serialized tables. The value below each table indicates the mean-squared error (MSE) against human-preferred binary attention. Attention is extracted at the answer-generation position and aggregated across all 36 decoder layers. See Appendix D for the full methodology and case study.

3.3 Performance Analysis

This motivates error localization on incorrect trajectories, allowing us to identify two existing table reasoning bottlenecks:

(1) Serializing tables weakens 2D structure awareness for row computation. The WTQ question "How many plants are in Algeria?" requires counting "Algeria" occurrence within "Country" column. Figure 2 shows sharper attention from image-based parsing on relevant entries compared to diffused attention from text serializations. Lower total token usage and fewer turns with visual representations in Table 1 suggest that state representation impacts information accessibility and prevents drift in long interactions (Laban et al., 2026). This is consistent with prior evidence on preserved information under visual compression (Shi et al., 2026). We include detailed case studies on how the agent attends to different table representations in Appendix D and E.

(2) Full table state estimation is challenging for frontier MLLMs. While modeling belief state is considered a better objective for planning under POMDP assumption (Kaelbling et al., 1998), Table 1 shows degraded accuracy when the MLLM incorporates table estimates to assist each reasoning step. Here, *Table estimate* refers to asking the MLLM to explicitly regenerate the full table state after each action and prepend it to the next step's context as a forward prediction. *No estimate* uses the table only as observed, without any forward prediction. The degraded accuracy suggests that

full-table regeneration is fragile even for frontier LLMs due to the structural complexity of tabular data. In Figure 3, we ask GPT-5.4 (OpenAI, 2026) to estimate the table state after sorting. While the sort order is correct, the model hallucinates all numerical values as '1'.

(a) Original	(b) Truth	(c) GPT-5.4
E. Keene: 2	P. Norris: 3	P. Norris: 1
P. Norris: 3	E. Keene: 2	E. Keene: 1
K. Biller: 2	K. Biller: 2	K. Biller: 1
M. Fresco: 2	M. Fresco: 2	M. Fresco: 1
others (11): 1	others (11): 1	others (11): 1

Figure 3: GPT-5.4 hallucination in full table estimation after sorting: sort order is correct but all numerical values are collapsed to '1', motivating lightweight metadata verification over full-table regeneration.

Takeaway. MLLMs are sensitive to table representation: visual grounding improves information accessibility, producing higher accuracy and fewer interaction rounds. Full table-state estimation is fragile to hallucination, motivating lightweight metadata-based verification rather than full-table regeneration.

4 Methodology: TabQAWorld

Our findings motivate TabQAWorld with (i) action-conditioned multimodal selection and (ii) metadata-guided trajectory optimization to improve table representation and estimation.

4.1 Action-conditioned Multimodal Selection Policy

Figure 4 illustrates the full per-step reasoning loop of TabQAWorld. A key failure mode in table reasoning is an observation mismatch, where the action may be correct in principle, but the chosen representation obscures the evidence needed for reliable execution. TabQAWorld addresses this by selecting *what to observe next*. Let $g(\cdot; m)$ be a modality-conditioned encoder with $m \in \mathcal{M}$. At step t, the agent predicts a triplet $a_t = \{\alpha_t, r_t, m_t\}$, where α_t is the executable operation, r_t is step-level feedback, and m_t is the modality used to encode the next observation.

$$x_t \mapsto a_t = \{\alpha_t, r_t, m_t\}, \text{ where}$$
$$T_{t+1} = a_t(T_t), \quad o_{t+1} = \Omega(T_{t+1}),$$
$$x_{t+1} = g(o_{t+1}; m_t).$$

This formulation turns modality into a decision policy rather than a fixed input format. In practice, the policy offers the MLLM the flexibility to dynamically select the preferred representation, in order to improve information representativeness under a fixed context budget. As a result, later decisions are conditioned on more faithful state encoding, which reduces downstream error propagation across long trajectories.

Concretely, m_t is produced by the same MLLM call that generates the full action triplet via structured JSON output (Appendix A.3), requiring no additional model call or separate routing classifier. The framework uses a pandas-backed execution engine: each tool call is transpiled to a pandas operation, executed in a sandboxed environment, and any execution error triggers a corrective re-prompt. Invalid JSON responses or tool-name mismatches are caught and the step is retried with explicit error feedback. The complete set of supported table operations (filter, sort, compute column, groupby, merge, and more) is enumerated in the step prompt tool catalogue in Appendix A.3. Image rendering parameters (DPI, row/column limits, font size) and a sensitivity discussion are provided in Appendix A.3 and Table 6.

4.2 Metadata-guided Trajectory Optimization

As full table state estimation is infeasible (Figure 3), we use the metadata projection $z_t = \phi(T_t)$ defined in Section 3.1 as a lightweight state verification shortcut (Figure 5). This is distinct from the belief b_t: while b_t captures the agent's accumulated context, z_t is an external, low-dimensional summary (row count direction, expected column names, expected key output value) computed from the realized table state. Nevertheless, even with simplified state estimation, strict step-by-step execution still incurs large interaction overhead and can amplify drift when the plan grows long. To address this, we enable *metadata-guided action batching*: the MLLM, under prompt guidance, may group multiple low-risk actions into a single step when its predicted metadata aligns with the realized state. At each step t, the agent predicts $\hat{z}_{t+1}$ and compares it against the realized state z_{t+1}. A mismatch triggers replanning before further batching is allowed. Formally:

$$(z_t, \hat{z}_{t+1}, z_{t+1}) = \big(\phi(T_t), \phi(\hat{T}_{t+1}), \phi(T_{t+1})\big),$$
$$c(a_t) = \nabla(z_t, \hat{z}_{t+1}),$$
$$\nu(a_t) = \nabla(z_{t+1}, \hat{z}_{t+1}).$$

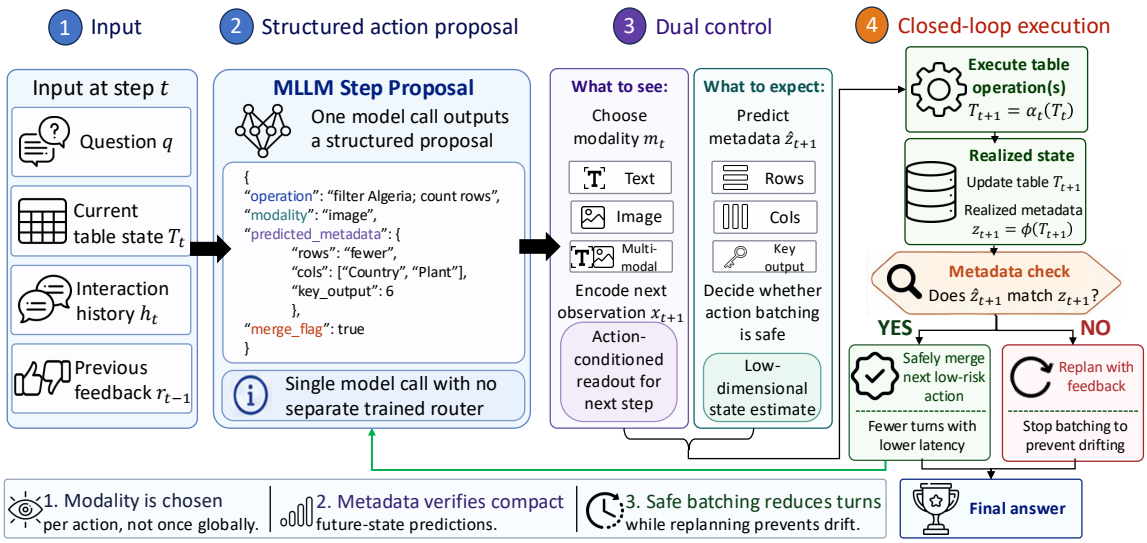

Figure 4: TABQAWORLD reduces turn count without sacrificing correctness by gating action batching on metadata consistency. At each step, the MLLM proposes a triplet (an executable operation, an observation modality, and a predicted next-state metadata), and the modality determines how the next table state is encoded. A metadata match with the realized state enables batching of subsequent low-risk operations. A mismatch triggers replanning before any further batching is allowed.

Plan: `"filter Algeria and sort by year"`
$\hat{z}$: `{rows:"fewer", cols:[...], key_output: "1964"}`
z: `{rows:3, cols:[...], key_output: "1968"}`
if `key_output` mismatch ($\hat{z} \neq z$) **then**
$\Rightarrow$ trigger replanning
end if

Figure 5: Metadata-guided execution. Predicted `"1964"` vs. realized `"1968"` triggers replanning.

The metadata schema (`rows`: directional change, "more", "same", or "fewer". `cols`: expected column name set, matched by exact string equality. `key_output`: expected extreme or target cell value, matched by exact string equality) is specified as a structured JSON field in the step prompt (Appendix A.3), so no additional training or separate model is required.

The `key_output` field extends verification beyond row-count and column-name checks to semantic value mismatches, as illustrated in Figure 5. Order-sensitive operations, which carry higher metadata uncertainty, naturally tend not to be batched (Figure 7). Remaining limitations of the metadata schema are discussed in the Limitations section.

5 Experimental Setup

Evaluation Datasets and Metrics. We evaluate TABQAWORLD by following established table reasoning evaluation protocol from (Zheng et al., 2024; Wang et al., 2026) on seven diverse benchmarks across Table Question Answering (TQA) and Table Fact Verification (TFV) tasks. For TQA, which requires interpreting complex queries (Ren et al., 2020, 2021), we evaluate on five benchmarks, namely WTQ (Pasupat and Liang, 2015), TABMWP (Lu et al., 2023), TAT-QA (Zhu et al., 2021), HiTab (Cheng et al., 2022), and FeTaQA (Nan et al., 2022). For TFV tasks requiring grounded reasoning, we use TabFact (Chen et al., 2020) and InfoTabs (Gupta et al., 2020). Regarding evaluation metrics, standard protocol uses BLEU score (Papineni et al., 2002) on FeTaQA for its free-form response task and binary accuracy for the remaining six datasets.

Baselines. We compare TABQAWORLD against six groups: **(1)** table-as-text LLMs, **(2)** table-as-image MLLMs, **(3)** table-as-multimodality MLLMs, **(4)** training-free agents (GPT-3.5), **(5)** adaptive-modality routing, and **(6)** trained trajectory-optimizing agents. Full model lists, backbone matching rationale, and result sourcing are in Appendix A.5. Proprietary models (GPT-5-nano, GPT-5.4) appear in ablations only.

Table 2: Benchmark results for TABQAWORLD and representative baselines. †Average is over the six accuracy-based benchmarks. FeTaQA (BLEU) excluded to avoid mixing metrics. *Result from the published paper. Backbone and settings differ from ours (see Appendix A.5). Unmarked baselines are re-evaluated by us. Training-free agent baselines (Mix-SC, TIDE, CIT-DP) use GPT-3.5. Accuracy gaps reflect both framework design and backbone capacity. Full baseline landscape in Appendix Table 7.

Method	TQA					TFV		Summary	
	WTQ (Acc.)	TABMWP (Acc.)	TAT-QA (Acc.)	HiTab (Acc.)	FeTaQA (BLEU)	TabFact (Acc.)	InfoTabs (Acc.)	WTQ + TabFact (Acc.)	Average† (Acc.)
Table-as-Text Baselines									
Llama3-Instruct-8B*	21.24	42.01	13.08	6.97	12.66	73.89	54.00	47.57	35.20
TableLlama-7B*	24.97	10.10	19.04	46.57	38.38	79.37	46.57	52.17	37.77
Table-as-Image Baselines									
MiniCPM-V-2.6-8B*	47.97	83.68	51.55	56.53	32.68	78.48	73.03	63.23	65.21
Qwen2.5-VL-7B	54.37	63.69	51.94	62.69	10.99	75.81	70.13	65.09	63.11
Qwen3-VL-8B	81.82	79.90	59.80	65.83	13.89	82.41	75.88	82.12	73.77
Table-as-Multimodality Baselines									
HIPPO-8B*	55.77	87.50	60.75	63.00	33.18	82.27	75.74	69.02	70.84
Gemini 2.0 Flash	63.56	46.29	35.62	60.41	10.57	81.33	54.31	72.45	56.92
Training-free Table Reasoning Agent Baselines (GPT-3.5)									
Mix-SC*	73.7	—	—	—	—	88.50	—	81.10	—
TIDE*	75.0	—	—	—	—	89.82	—	82.41	—
CIT-DP*	76.4	—	—	—	**36.34**	91.30	—	83.85	—
Dynamic Adaptive Routing (TableDART, TableGPT2-7B base)									
TableDART (TG2-7B+Qwen2.5-VL-7B)*	69.29	72.61	59.07	71.13	29.87	77.94	71.46	73.62	70.25
TableDART (TG2-7B+Ovis2-8B)*	70.58	84.54	62.05	74.37	36.11	81.37	76.22	75.98	74.86
Trained trajectory-optimizing agents (TATTOO: DS-R1-Qwen-14B base, TabTrim: Qwen3 family)									
DS-Q14B + TATTOO (Qwen3-8B)*	69.80	—	—	—	—	58.79	—	64.30	—
TabTrim-4B*	76.80	—	—	—	—	89.40	—	83.10	—
TabTrim-8B*	79.40	—	—	—	—	91.20	—	85.30	—
TABQAWORLD *(Training-free)*									
Qwen2.5-VL-7B + TABQAWORLD	70.20	86.43	71.36	76.88	20.47	84.42	76.38	77.31	77.61
Qwen3-VL-8B + TABQAWORLD	**88.89**	**94.97**	**75.88**	**81.41**	31.78	**91.45**	**81.41**	**90.17**	**85.67**

Implementation Details. To test for generalizability, we incorporate a standardized prompt for all test datasets to avoid inductive bias on specific dataset formats. We conduct local inference with Ollama (team, 2025) for the two open-source models, with hyperparameters following the protocol of prior work (Zou et al., 2026; Xing et al., 2026). The complete details regarding prompts, data construction, hyperparameter settings, and computational environment are provided in Appendix A.

6 Main Results

Results Overview The primary evidence for framework effectiveness is the controlled within-backbone ablation (Section 7). Cross-paper comparisons are provided for context but involve backbone differences. Against the strongest adaptive-modality baseline (TableDART TG2+Ovis2-8B), TABQAWORLD (Qwen3-VL-8B) improved by 10.81% on the six-benchmark average. Against the strongest trained trajectory-optimizing agents within the Qwen3 family, the gain was 4.87%. The full baseline landscape and per-backbone breakdown are in Appendix 7.

Per-dataset patterns. Examining Table 2, gains over the same-backbone image baseline (Qwen3-VL-8B) are consistent across all six accuracy benchmarks: TABMWP improves from 79.90 to 94.97 (+15.1 pp), TAT-QA from 59.80 to 75.88 (+16.1 pp), and HiTab from 65.83 to 81.41 (+15.6 pp), reflecting that compositional multi-step tasks benefit most from adaptive modality routing. TabFact and InfoTabs also improve substantially (82.41→91.45 and 75.88→81.41), confirming gains in structured fact-verification reasoning. On FeTaQA (BLEU), the same-backbone score improves from 13.89 to 31.78, though TABQAWORLD trails single-turn generation baselines with stronger backbones (HIPPO: 33.18, CIT-DP: 36.34). Tool-use overhead reduces n-gram overlap in free-form tasks, consistent with the answer extraction error analysis in Appendix G.

7 Ablation Studies

The ablation study follows similar protocol as Section 5 and (Xing et al., 2026) to evaluate component effectiveness with accuracy, and inference latency (in seconds), adding proprietary GPT-5-nano (OpenAI, 2025a) and GPT-5.4 as additional models. We first evaluate the model-agnostic gains jointly brought by the two components, followed by specific benefits on contribution of each component. Lastly, we evaluate the inference efficiency of TABQAWORLD. Detailed motivation for dataset

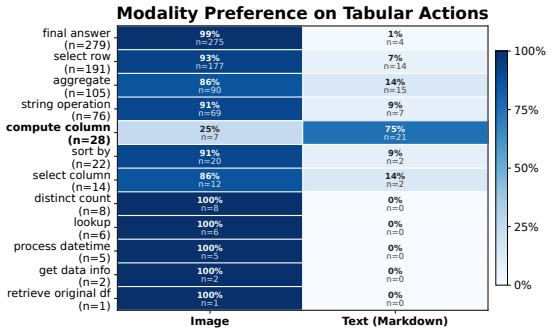

Figure 6: Action-conditioned modality preferences under TABQAWORLD. The agent selects image in most steps but switches to text for `compute_column`, reflecting emergent tool-aware representation selection.

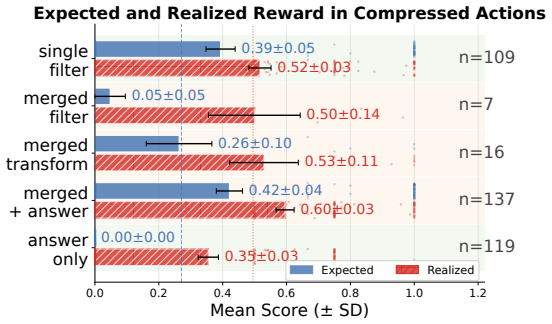

Figure 7: Action compression preference by batch type under TABQAWORLD. The x-axis shows the metadata alignment score (proportion of fields matching predicted vs. realized state), not a policy reward.

selection and additional BLEU and ROUGE-L (Lin, 2004) evaluations are provided in Appendix A.4 and C respectively.

7.1 Model-agnostic gains from two components

Across all four model groups, dynamic modality choice yielded an average 5.42% accuracy improvement over the best fixed-modality baseline, confirming that reliable table-state readout improves tabular reasoning. While visual-included settings reported lengthened per-step latency, trajectory optimization via metadata reduced average latency by 33.35%, offsetting the overhead and enabling more efficient inference overall. Table 3 reports the nine-condition breakdown for Qwen3-VL-8B. The *oracle best static* row (per-dataset best fixed modality, avg. 60.9) is an upper bound for any static strategy. Adaptive modality alone (avg. 63.7) exceeded it by 2.8 pp, confirming that per-step routing captures variation no single fixed modality can.

Three controls further tightened this: *random* routing (avg. 58.0), *heuristic* routing (avg. 56.5), and *question-level* routing (avg. 54.5) all fell below adaptive modality, confirming that per-step action-conditioned adaptation drives the gain. Trajectory optimization alone yielded modest accuracy changes, with a small WTQ dip (81.8 to 79.8) from occasional premature batching, while delivering the latency gains in Section 7.3. Full TABQAWORLD achieves the best result across all conditions.

Table 3: Nine-condition component ablation (Qwen3-VL-8B). Averages over WTQ, MMQA, and MMTU. Values are accuracy (SE $= \sqrt{p(1-p)/n}$), sample sizes in column headers. ¶Oracle: best fixed modality per dataset (WTQ/MMTU: Image, MMQA: Multimodal). ‡Random: uniform modality per step. §Heuristic: compute/transform to text, else to image. †Question-level: one modality fixed per question at step 1.

Condition	WTQ (n=200)	MMQA (n=200)	MMTU (n=100)	Avg.
Static text (Markdown)	71.2 (3.2)	58.4 (3.5)	36.4 (4.8)	55.3
Static image	81.8 (2.7)	61.4 (3.4)	37.4 (4.8)	60.2
Static multimodal	79.3 (2.9)	63.5 (3.4)	36.4 (4.8)	59.7
Oracle best static¶	81.8 (2.7)	63.5 (3.4)	37.4 (4.8)	60.9
Random modality‡	79.2 (2.9)	58.7 (3.5)	36.2 (4.8)	58.0
Heuristic routing§	76.4 (3.0)	60.0 (3.5)	33.0 (4.7)	56.5
Question-level routing†	79.7 (2.8)	56.5 (3.5)	27.3 (4.5)	54.5
Adaptive modality only	83.3 (2.6)	66.5 (3.3)	41.4 (4.9)	63.7
Traj. opti. only (fixed image)	79.8 (2.8)	66.5 (3.3)	38.4 (4.9)	61.6
Full TABQAWORLD	**88.9** (2.2)	**76.7** (3.0)	**42.4** (4.9)	**69.3**

7.2 Emergent modal selection and action compression preferences

Figures 6 and 7 showed consistent prompt-induced preferences for both components. Figure 6 showed emergent tool-aware modality selection: image dominated most steps, but `compute_column` strongly favored text. Figure 7 showed that high metadata alignment correlated with higher compression (`single filter`, `merged + answer`), while riskier batches (`merged transform`) were naturally avoided. Quantitatively, 26.4% of steps were multi-action batches (29.9%/26.7%/18.7% on WTQ/MMQA/MMTU), confirming that metadata alignment drove real compression. A per-question-type modality breakdown is in Appendix C.3.

7.3 Inference efficiency

We benchmark TABQAWORLD against a static-image unoptimized baseline (no action batching), measuring the combined gain from both components. Evaluation protocol is in Appendix A.8. Table 4 summarises the results. TABQAWORLD reduced wall-clock latency by 33.35% (41.55 to 27.29s per sample), turns by 38% (3.61 to 2.24),

Table 4: Inference efficiency: TABQAWORLD vs. unoptimized baseline (Qwen3-VL-8B). ⋆Mean total tokens (prompt + completion) per trajectory, averaged over WTQ/MMQA/MMTU. Baseline measured on the static-image unoptimized condition.

	Baseline	**TABQAWORLD**
Latency (s / sample)	41.55	27.29
Turns / model calls	3.61	2.24
Tokens / trajectory ($\times 10^3$)*	21.2	15.1
Latency reduction	—	−33.35%
Token reduction	—	−28.7%

and tokens by 28.7% (21.2K to 15.1K), with fewer model calls as the primary driver. The metadata prediction overhead (one extra JSON field per step) was outweighed by the turn reduction. To isolate trajectory optimization's contribution, Appendix Table 9 compares the adaptive-modality unoptimized (col. 6) and full TABQAWORLD (col. 13) conditions, showing 10-27% additional latency reduction. Full accuracy-latency results across all 13 conditions are in Appendix Tables 7 and 9.

8 Summary and Future Work

We have presented TABQAWORLD, a training-free framework for multi-turn table reasoning that jointly addresses representation and estimation bottlenecks without additional training. The central finding is that neither modality nor trajectory decisions need to be fixed at the question level: conditioning observation modality on the planned action and gating batching on lightweight metadata consistency each contribute independently, and together yield accuracy and efficiency gains that neither component achieves alone. The ablation further shows that no single representation is universally optimal: visual grounding dominates most operation types while text serialization remains preferable for arithmetic column computation, suggesting that representation-aware observation is a productive design axis for future agentic systems. Looking ahead, we plan to examine whether the routing and verification principles generalise to broader document and heterogeneous agentic reasoning settings.

Limitations

TABQAWORLD is a training-free framework. The paragraphs below identify where the current design falls short and where future work (including RL post-training) may close remaining gaps.

Metadata scope. The metadata-guided trajectory optimization relies on row-count and column-name checks that do not fully capture operations preserving table dimensions (e.g., `sort`). The `key_output` field provides a partial semantic check, but internal cell-value mutations outside these fields remain outside the current detection scope. Critically, high-risk operations such as `f_compute_column` and `f_aggregate` reach **90-100% error rates when batched**, because these operations mutate cell values in ways that row-count and column-name checks cannot detect. The metadata guard verifies *metadata consistency*, with reliable batching confined to operations that preserve row counts and column names. Appendix G.2 provides a full empirical breakdown, showing that batched steps fail at 1.2-1.4 times the rate of single-action steps overall, with per-operation breakdown revealing the concentration of risk in aggregations and string transformations.

Answer extraction. Answer extraction errors are the dominant failure mode overall (58.3%), and are particularly pronounced on free-form generation tasks (FeTaQA), where multi-turn tool-use overhead reduces BLEU relative to single-turn generation baselines. This reflects a scope limitation of the method: the tool-use framework is designed for structured table manipulation and is less suited to open-ended text generation tasks. A detailed breakdown is provided in Appendix G.

Trajectory optimisation. Trajectory optimization component produces a small accuracy dip on WTQ (0.818 to 0.798) due to occasional premature action batching, indicating the need to further tune metadata confidence threshold on certain datasets.

Pre-emptive modality selection. The action-conditioned modality selection is pre-emptive: the model chooses the next-step modality before observing the post-action table state, which may be suboptimal when an action yields an unexpectedly large or structurally modified table.

Backbone comparison caveat. Comparisons against training-free agent baselines (DATER, ReAcTable, Mix-SC, TIDE, CIT-DP) use GPT-3.5 as their backbone, while TABQAWORLD uses Qwen3-VL-8B. The reported accuracy gaps reflect both framework design and backbone capacity.

Attention analysis. The attention analysis in Section D is correlational and should not be interpreted as causal evidence.

Ethics Statement

This work studies how to improve table reasoning for language models in interactive multi-turn conversation environments. Our experiments exclusively use publicly available benchmarks and packages, and do not involve any non-public data or personally identifiable information.

LLM Usage

In this work, we utilize LLMs for code debugging, codebase restructuring, and writing polishing. Specifically, we also leverage the coding ability of LLMs to generate structured figures (Figure 2, 6, and 7). We have prepared a repository restructured by LLM that contains 20 questions for all used datasets in this work for sample reproducibility. Details are in Appendix D.

References

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, and 45 others. 2025a. Qwen3-vl technical report. *Preprint*, arXiv:2511.21631.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, and 8 others. 2025b. Qwen2.5-vl technical report. *Preprint*, arXiv:2502.13923.

Wenhu Chen. 2023. Large language models are few(1)-shot table reasoners. In *Findings of the Association for Computational Linguistics: EACL 2023*, pages 1120–1130, Dubrovnik, Croatia. Association for Computational Linguistics.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2020. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2022. HiTab: A hierarchical table dataset for question answering and natural language generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1094–1110, Dublin, Ireland. Association for Computational Linguistics.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2023. Binding language models in symbolic languages. In *The Eleventh International Conference on Learning Representations*.

Naihao Deng, Zhenjie Sun, Ruiqi He, Aman Sikka, Yulong Chen, Lin Ma, Yue Zhang, and Rada Mihalcea. 2024. Tables as texts or images: Evaluating the table reasoning ability of LLMs and MLLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 407–426, Bangkok, Thailand. Association for Computational Linguistics.

Google DeepMind. 2025. Genie 3: A general-purpose world model. https://deepmind.google/models/genie/. Accessed: 2026-03-17.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yu Guo, Shenghao Ye, Shuangwu Chen, Zijian Wen, Tao Zhang, Qirui Bai, Dong Jin, Yunpeng Hou, Huasen He, Jian Yang, and Xiaobin Tan. 2026. Rethinking table pruning in tableqa: From sequential revisions to gold trajectory-supervised parallel search. *Preprint*, arXiv:2601.03851.

Vivek Gupta, Maitrey Mehta, Pegah Nokhiz, and Vivek Srikumar. 2020. INFOTABS: Inference on tables as semi-structured data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2309–2324, Online. Association for Computational Linguistics.

David Ha and Jürgen Schmidhuber. 2018. World models.

Danijar Hafner, Timothy Lillicrap, Ian Fischer, Ruben Villegas, David Ha, Honglak Lee, and James Davidson. 2019. Learning latent dynamics for planning from pixels. In *International Conference on Machine Learning*, pages 2555–2565.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Eisenschlos. 2020. TaPas: Weakly supervised table parsing via pre-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4320–4333, Online. Association for Computational Linguistics.

Yushi Hu, Weijia Shi, Xingyu Fu, Dan Roth, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Ranjay Krishna. 2024. Visual sketchpad: Sketching as a visual chain of thought for multimodal language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Deyi Ji, Lanyun Zhu, Siqi Gao, Peng Xu, Hongtao Lu, Jieping Ye, and Feng Zhao. 2025. Tree-of-table: Unleashing the power of LLMs for enhanced large-scale table understanding.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. 2023. StructGPT: A general framework for large language model to reason over structured data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9237–9251, Singapore. Association for Computational Linguistics.

Zhengbao Jiang, Yi Mao, Pengcheng He, Graham Neubig, and Weizhu Chen. 2022. OmniTab: Pretraining with natural and synthetic data for few-shot table-based question answering. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 932–942, Seattle, United States. Association for Computational Linguistics.

Leslie Pack Kaelbling, Michael L. Littman, and Anthony R. Cassandra. 1998. Planning and acting in partially observable stochastic domains. *Artif. Intell.*, 101(1–2):99–134.

Koray Kavukcuoglu and Google DeepMind. 2025. Gemini 2.5: Our most intelligent ai model. Last updated March 26, 2025; accessed June 26, 2025.

Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, Quan Vuong, Thomas Kollar, Benjamin Burchfiel, Russ Tedrake, Dorsa Sadigh, Sergey Levine, Percy Liang, and Chelsea Finn. 2024. Openvla: An open-source vision-language-action model. *Preprint*, arXiv:2406.09246.

Tung Sum Thomas Kwok, Xinyu Wang, Hengzhi He, Xiaofeng Lin, Peng Lu, Liheng Ma, Chunhe Wang, Chun Ho Mak, Yuyu Luo, Ying Nian Wu, Lei Ding, and Guang Cheng. 2026. Enhancing table reasoning with deterministic table-state rewards. *Preprint*, arXiv:2601.22530.

Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2026. LLMs get lost in multi-turn conversation. In *The Fourteenth International Conference on Learning Representations*.

Jihyung Lee, Jin-Seop Lee, Jaehoon Lee, YunSeok Choi, and Jee-Hyong Lee. 2025. DCG-SQL: Enhancing in-context learning for text-to-SQL with deep contextual schema link graph. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15397–15412, Vienna, Austria. Association for Computational Linguistics.

Xuanyu Lei, Zonghan Yang, Xinrui Chen, Peng Li, and Yang Liu. 2025. Scaffolding coordinates to promote vision-language coordination in large multi-modal models. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2886–2903, Abu Dhabi, UAE. Association for Computational Linguistics.

Chin-Yew Lin. 2004. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Weizhe Lin, Rexhina Blloshmi, Bill Byrne, Adria de Gispert, and Gonzalo Iglesias. 2023. An inner table retriever for robust table question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 9909–9926, Toronto, Canada. Association for Computational Linguistics.

Tianyang Liu, Fei Wang, and Muhao Chen. 2024. Rethinking tabular data understanding with large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 450–482, Mexico City, Mexico. Association for Computational Linguistics.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2023. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *International Conference on Learning Representations (ICLR)*.

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.

Giang Nguyen, Ivan Brugere, Shubham Sharma, Sanjay Kariyappa, Anh Totti Nguyen, and Freddy Lecue. 2025. Interpretable LLM-based table question answering. *Transactions on Machine Learning Research*.

OpenAI. 2025a. Gpt-5 nano model documentation. `https://platform.openai.com/docs/models/gpt-5-nano`. Accessed: 2025-12-30.

OpenAI. 2025b. Introducing openai o3 and o4-mini: Our smartest and most capable models to date. `https://openai.com/index/introducing-o3-and-o4-mini/`. Accessed: 2026-03-17.

OpenAI. 2026. Introducing gpt-5.4. `https://openai.com/index/introducing-gpt-5-4/`. OpenAI blog post, accessed 2026-03-14.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, page 311–318, USA. Association for Computational Linguistics.

Panupong Pasupat and Percy Liang. 2015. 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.

Jingang QU, David Holzmüller, Gaël Varoquaux, and Marine Le Morvan. 2025. TabICL: A tabular foundation model for in-context learning on large data. In *Forty-second International Conference on Machine Learning*.

Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Zi Huang, and Kai Zheng. 2021. Learning to ask appropriate questions in conversational recommendation. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '21, page 808–817, New York, NY, USA. Association for Computing Machinery.

Xuhui Ren, Hongzhi Yin, Tong Chen, Hao Wang, Nguyen Quoc Viet Hung, Zi Huang, and Xiangliang Zhang. 2020. Crsal: Conversational recommender systems with adversarial learning. *ACM Trans. Inf. Syst.*, 38(4).

Yuling Shi, Chaoxiang Xie, Zhensu Sun, Yeheng Chen, Chenxu Zhang, Longfei Yun, Chengcheng Wan, Hongyu Zhang, David Lo, and Xiaodong Gu. 2026. Codeocr: On the effectiveness of vision language models in code understanding. *Preprint*, arXiv:2602.01785.

Ananya Singha, José Cambronero, Sumit Gulwani, Vu Le, and Chris Parnin. 2023. Tabular representation, noisy operators, and impacts on table structure understanding tasks in LLMs. In *NeurIPS 2023 Second Table Representation Learning Workshop*.

Yuan Sui, Jiaru Zou, Mengyu Zhou, Xinyi He, Lun Du, Shi Han, and Dongmei Zhang. 2024. TAP4LLM: Table provider on sampling, augmenting, and packing semi-structured data for large language model reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10306–10323, Miami, Florida, USA. Association for Computational Linguistics.

Ollama team. 2025. Ollama.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Haolan Wang, Zhenghao Liu, Xinze Li, Xiaocui Yang, Yu Gu, Yukun Yan, Qi Shi, Fangfang Li, Chong Chen, and Ge Yu. 2026. Hippo: Enhancing the table understanding capability of llms through hybrid-modal preference optimization. *Preprint*, arXiv:2502.17315.

Kangrui Wang, Pingyue Zhang, Zihan Wang, Yaning Gao, Linjie Li, Qineng Wang, Hanyang Chen, Yiping Lu, Zhengyuan Yang, Lijuan Wang, Ranjay Krishna, Jiajun Wu, Li Fei-Fei, Yejin Choi, and Manling Li. 2025a. VAGEN: Reinforcing world model reasoning for multi-turn VLM agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Lanrui Wang, Mingyu Zheng, Hongyin Tang, Zheng Lin, Yanan Cao, Jingang Wang, Xunliang Cai, and Weiping Wang. 2025b. NeedleinATable: Exploring long-context capability of large language models towards long-structured tables. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, and Tomas Pfister. 2024. Chain-of-table: Evolving tables in the reasoning chain for table understanding. In *The Twelfth International Conference on Learning Representations*.

Jialong Wu, Shaofeng Yin, Ningya Feng, and Mingsheng Long. 2025a. RLVR-world: Training world models with reinforcement learning. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Jian Wu, Yicheng Xu, Yan Gao, Jian-Guang Lou, Börje F. Karlsson, and Manabu Okumura. 2023. TACR: A table alignment-based cell selection method for HybridQA. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 6535–6549, Toronto, Canada. Association for Computational Linguistics.

Jian Wu, Linyi Yang, Dongyuan Li, Yuliang Ji, Manabu Okumura, and Yue Zhang. 2025b. MMQA: Evaluating LLMs with multi-table multi-hop complex questions. In *The Thirteenth International Conference on Learning Representations*.

Zhenhe Wu, Jian Yang, Jiaheng Liu, Xianjie Wu, Changzai Pan, Jie Zhang, Yu Zhao, Shuangyong Song, Yongxiang Li, and Zhoujun Li. 2025c. Table-r1: Region-based reinforcement learning for table understanding. *Preprint*, arXiv:2505.12415.

Eric Xing, Mingkai Deng, Jinyu Hou, and Zhiting Hu. 2025a. Critiques of world models. *Preprint*, arXiv:2507.05169.

Junjie Xing, Yeye He, Mengyu Zhou, Haoyu Dong, Shi Han, Lingjiao Chen, Dongmei Zhang, Surajit Chaudhuri, and H. V. Jagadish. 2025b. MMTU: A massive multi-task table understanding and reasoning benchmark. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Xiaobo Xing, Wei Yuan, Tong Chen, Quoc Viet Hung Nguyen, Xiangliang Zhang, and Hongzhi Yin. 2026. TableDART: Dynamic adaptive multi-modal routing for table understanding. In *The Fourteenth International Conference on Learning Representations*.

Yi Xu, Chengzu Li, Han Zhou, Xingchen Wan, Caiqi Zhang, Anna Korhonen, and Ivan Vulić. 2026. Visual planning: Let's think only with images. In *The Fourteenth International Conference on Learning Representations*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei,

Huan Lin, Jialong Tang, and 41 others. 2025a. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Saisai Yang, Qingyi Huang, Jing Yuan, Liangyu Zha, Kai Tang, Yuhang Yang, Ning Wang, Yucheng Wei, Liyao Li, Wentao Ye, Hao Chen, Tao Zhang, Junlin Zhou, Haobo Wang, Gang Chen, and Junbo Zhao. 2025b. Tablegpt-r1: Advancing tabular reasoning through reinforcement learning. *Preprint*, arXiv:2512.20312.

Zhen Yang, Ziwei Du, Minghan Zhang, Wei Du, Jie Chen, Zhen Duan, and Shu Zhao. 2025c. Triples as the key: Structuring makes decomposition and verification easier in LLM-based tableQA. In *The Thirteenth International Conference on Learning Representations*.

Zhen Yang, Ziwei Du, Minghan Zhang, Wei Du, Jie Chen, Fulan Qian, and Shu Zhao. 2025d. Causality meets the table: Debiasing LLMs for faithful tableQA via front-door intervention. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Yuan Yao, Tianyu Yu, Ao Zhang, Chongyi Wang, Junbo Cui, Hongji Zhu, Tianchi Cai, Chi Chen, Haoyu Li, Weilin Zhao, Zhihui He, Qianyu Chen, Ronghua Zhou, Zhensheng Zou, Haoye Zhang, Shengding Hu, Zhi Zheng, Jie Zhou, Jie Cai, and 5 others. 2025. Efficient gpt-4v level multimodal large language model for deployment on edge devices. *Nature Communications*, 16(1):5509.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, page 174–184, New York, NY, USA. Association for Computing Machinery.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. TaBERT: Pretraining for joint understanding of textual and tabular data. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8413–8426, Online. Association for Computational Linguistics.

Yangbin Yu, Mingyu Yang, Junyou Li, Yiming Gao, Feiyu Liu, Yijun Yang, Zichuan Lin, Jiafei Lyu, Yicheng Liu, Zhicong Lu, Deheng Ye, and Jie Jiang. 2026. Proact: Agentic lookahead in interactive environments. *Preprint*, arXiv:2602.05327.

Liangyu Zha, Junlin Zhou, Liyao Li, Rui Wang, Qingyi Huang, Saisai Yang, Jing Yuan, Changbao Su, Xiang Li, Aofeng Su, Tao Zhang, Chen Zhou, Kaizhe Shou, Miao Wang, Wufang Zhu, Guoshan Lu, Chao Ye, Yali Ye, Wentao Ye, and 6 others. 2023. Tablegpt: Towards unifying tables, nature language and commands into one gpt. *Preprint*, arXiv:2307.08674.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024a. TableLlama: Towards open large generalist models for tables. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6024–6044, Mexico City, Mexico. Association for Computational Linguistics.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024b. Tablellama: Towards open large generalist models for tables. *Preprint*, arXiv:2311.09206.

Xuanliang Zhang, Dingzirui Wang, Longxu Dou, Qingfu Zhu, and Wanxiang Che. 2025. A survey of table reasoning with large language models. *Frontiers of Computer Science*, 19(9):199348.

Yunjia Zhang, Jordan Henkel, Avrilia Floratou, Joyce Cahoon, Shaleen Deep, and Jignesh M. Patel. 2024c. Reactable: Enhancing react for table question answering. *Proc. VLDB Endow.*, 17(8):1981–1994.

Zhuosheng Zhang, Aston Zhang, Mu Li, hai zhao, George Karypis, and Alex Smola. 2024d. Multimodal chain-of-thought reasoning in language models. *Transactions on Machine Learning Research*.

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.

Bangbang Zhou, Zuan Gao, Zixiao Wang, Boqiang Zhang, Yuxin Wang, Zhineng Chen, and Hongtao Xie. 2025a. Syntab-llava: Enhancing multimodal table understanding with decoupled synthesis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 24796–24806.

Qiji Zhou, Ruochen Zhou, Zike Hu, Panzhong Lu, Siyang Gao, and Yue Zhang. 2024. Image-of-thought prompting for visual reasoning refinement in multimodal large language models. *Preprint*, arXiv:2405.13872.

Wei Zhou, Bolei Ma, Annemarie Friedrich, and Mohsen Mesgar. 2025b. Table question answering in the era of large language models: A comprehensive survey of tasks, methods, and evaluation. *Preprint*, arXiv:2510.09671.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287, Online. Association for Computational Linguistics.

Jiaru Zou, Soumya Roy, Vinay Kumar Verma, Ziyi Wang, David Wipf, Pan Lu, Sumit Negi, James Zou, and Jingrui He. 2026. TaTToo: Tool-grounded thinking PRM for test-time scaling in tabular reasoning. In

The Fourteenth International Conference on Learning Representations.

A Implementation Details

A.1 Full Problem Formulation

We model multi-turn table reasoning as a trajectory policy over partial observations of an evolving table state, following the partially observable Markov Decision Process (POMDP) formalism (Wang et al., 2025a; Xing et al., 2025a). A table $T = (H, R)$ consists of column headers H and rows R aligned with H. Given an initial table $T_0 = T$, a natural-language question q, and ground-truth answer α, the goal is to predict $\hat{\alpha}$ that maximizes a task metric $G(\hat{\alpha}, \alpha)$. At step t, we denote T_t the current table state (an instantiated table after $t-1$ edits), o_t the partial observation exposed to the agent under the chosen modality m, and b_t the agent's internal belief state (the accumulated context carried across steps). The agent executes a reasoning trajectory $\tau = \{a_1, \ldots, a_S\}$, where $a_1, \ldots, a_{S-1}$ are intermediate tool-use steps and $a_S = \hat{\alpha}$ is the final-answer action, inducing a conditional policy $\pi(\tau \mid T_0, q)$. We additionally define $z_t = \phi(T_t)$ as a low-dimensional metadata projection of the table state, distinct from the belief b_t. This projection is used for lightweight state verification in the trajectory optimization component. This formulation identifies two core failure modes: (i) selecting actions inconsistent with the current belief b_t (Kim et al., 2024), addressed by the action-conditioned modality selection policy, and (ii) maintaining a faithful belief over evolving table states (Wang et al., 2025a), addressed by the metadata-guided trajectory optimization.

A.2 TabQAWorld Algorithm

In our experiment, we instantiate modality m_0 as image. The MLLM agent then selects modality m at each step.

Algorithm 1 TabQAWorld

Require: $q, df, \mathcal{M}, \mathcal{T}, T, m_0$
Ensure: $\hat{a}$

$env \leftarrow df,\ m \leftarrow m_0,\ fb_0 \leftarrow \emptyset$
for $t = 1, \ldots, T$ **do**
  $o_t \leftarrow \begin{cases} \text{Ser}(env) & m = \text{txt} \\ \text{Rend}(env) \parallel \text{cols} & m = \text{multi} \\ \text{Rend}(env) & m = \text{img} \end{cases}$
  $\mathcal{A}_t \leftarrow \mathcal{M}(o_t, q, \mathcal{T}, fb_{t-1})$ *// prompt encourages grouping low-complexity operations (e.g., filter+sort, select+rename), with ν_t as the safety signal*
  $env_{\text{prev}} \leftarrow \text{snapshot}(env)$ *// pre-batch checkpoint for rollback*
  for each $a \in \mathcal{A}_t$ **do**
    if $a = \text{f_ans}$ **then return** $a.\text{ans}$
    end if
    if $a = \text{f_swi}$ **then** $m \leftarrow a.\text{mode}$
    else
      $(env, out) \leftarrow \text{Exec}(a, env)$
      if not $\text{ExecOk}(env)$ **and** $|\mathcal{A}_t| > 1$ **then**
        $env \leftarrow env_{\text{prev}}$; re-execute $\mathcal{A}_t$ one-by-one to isolate failure *// batch rollback*
        break
      end if
    end if
  end for
  $\nu_t \leftarrow \text{Score}(a.\text{pred}, env)$ *// compare predicted vs. realized metadata; low ν_t suppresses batching in fb_t*
  $fb_t \leftarrow (out,\ \text{ExecOk}(env),\ \nu_t)$
end for

A.3 Inference Prompts

We follow standardized prompting on all testing datasets, leaving the agent to answer the query as-is.

A.3.1 Role and Default Prompt

> You are a table reasoning agent. Use tools to

transform the table and answer the question. You may batch multiple simple actions together in one step to save time.

Respond with JSON only using the schema:

```
{
  "reasoning": str,
  "next_modality":
      "image|text|multimodal",
  "predicted_metadata": {
      "rows": "fewer|same|more",
      "cols": [str] or null,
      "key_output": str or null
  },
  "actions": [
    {"tool_name": str,
     "tool_args": object}, ...]
}
```

`next_modality` selects the table representation for the *next* step; the agent updates it at every turn.

Action batching rules:

- Group low-complexity operations in one step (e.g. filter + sort, select + rename).
- Use a single action when the result of one step determines the next (e.g. check row count after a filter).
- To answer, include `{"tool_name": "f_final_answer", "tool_args": {"answer": "..."}}` as the *last* action.
- To revert to the original table, use `f_retrieve_original_df`.

A.3.2 Table Observation

Table observation is prepended to the message modality-dependent

Mode	What is attached
`image`	JPEG render of `env[df]` (200 dpi, up to 30 rows × 5 cols) prepended as `<image>` token
`multimodal`	Same JPEG *plus* column names appended as text
`text`	Markdown table (via `to_markdown`); no image

A.3.3 Step Prompt

⟨*step budget note* b_t⟩ : one of:

- `[URGENT] This is your last step (`t/T`). You MUST call f_final_answer now.`
- `[NOTE]` k `steps remaining (`t/T`). Start converging toward f_final_answer.`
- `Step` t `of` T`.`

`Question:` ⟨q⟩

`Table shape:` ⟨(r, c)⟩

`Columns:` ⟨$[col_1, col_2, \ldots]$⟩

`Preview rows (JSON):` ⟨*first 15 rows of* `env[df]` *as JSON*⟩

`Last step output:` ⟨*truncated to 400 chars;* `N/A` *at* $t = 1$⟩

`Last error:` ⟨`N/A` *if none*⟩

Cross-step action ledger (drift guard)

⟨*Prior actions this episode:*⟩

`step`k`: tool_name(args)` ← *REPEATED* $n\times$*, avoid calling again* (if $n \geq 2$)

[DRIFT WARNING] tool+args combos called 2+ times: ... You MUST try a different approach or call f_final_answer. (if any repeats)

Tools:

⟨*formatted tool catalogue* $\mathcal{S}$ *(name, description, parameter list)*⟩

A.4 Datasets and Metrics

Our experimental evaluation is conducted on seven diverse benchmarks across TQA and TFV. We follow the established protocol of previous works (Zheng et al., 2024; Wang et al., 2026; Xing et al., 2026) for evaluation metrics, using Accuracy for most tasks and BLEU score for the generative FeTaQA benchmark. Table 5 provides a detailed breakdown of these datasets, including the number of instances used for our experiment.

Ablation datasets. Aside from WTQ, which is a standard TQA benchmark, we conduct ablation studies specifically on MMQA and MMTU, two datasets released in 2025, chosen to simulate a scenario where the backbone LLM has not seen the datasets during pre-training.

MMQA is a large-scale benchmark for evaluating

Table 5: Dataset statistics by task type.

Task Type	Dataset	Train	Validation	Test
Table Question Answering (TQA)	TABMWP	500	200	1,000
	WTQ	500	200	1,000
	HiTab	500	200	1,000
	TAT-QA	—	—	772
	FeTaQA	500	200	1,000
Table Fact Verification (TFV)	TabFact	500	200	1,000
	InfoTabs	500	200	1,000
Total		**3,000**	**1,400**	**6,772**

LLMs on multi-table and multi-hop question answering. The benchmark includes a total of 3,312 relational tables across 138 domains, where each instance consists of two or three interlinked tables. The dataset features 5,000 multi-table samples, annotated with natural language questions, SQL queries, gold answers, and explicit primary/foreign key relations. To ensure annotation quality, foreign and primary keys were labeled by human experts with inter-annotator agreement exceeding 80%. MMQA questions span four main categories, including numerical, list, count, and select, with an average length of 77-85 tokens, reflecting their compositional complexity.

MMTU is a large-scale benchmark with around 28K questions across 25 real-world table tasks. These tasks are drawn from decades' worth of computer science research on tabular data, with a focus on complex table tasks faced by professional users, where even frontier reasoning models like OpenAI GPT-5 and DeepSeek R1 score only around 69% and 57% respectively, suggesting significant room for improvement. For MMTU, we specifically extract the question subgroups Table-QA, Table-Fact-Verification and Table-Lookup to replicate the same table reasoning tasks only on datasets unreleased by the time the base models were released. Because both datasets are recent, most baseline works have not evaluated on them. We therefore conduct a detailed ablation study to report baseline performance on these two benchmarks.

A.5 Comparative Models and Rationale

We compare TABQAWORLD against a comprehensive set of baselines, selected to validate our core contributions across multiple axes.

Constituent models. To demonstrate that TABQAWORLD's performance arises from its dynamic framework rather than any single component, we evaluate its constituent models as standalone baselines. These include: Qwen2.5-VL-7B (Bai et al., 2025b) and Qwen3-VL-8B (Bai et al., 2025a). Evaluating the VLM in isolation establishes direct performance baselines.

Multimodal baselines. To compare the advantage of action-conditioned multimodal selection over static, one-modality-fits-one-attempt approaches, we compare against two multimodal LLM baselines and recent work on dynamic adaptive routing TableDART (Xing et al., 2026). The first MLLM is HIPPO (Wang et al., 2026) which jointly processes both text and image representations for all inputs. The second is Gemini 2.0 Flash (Kavukcuoglu and DeepMind, 2025) which serves as an additional MLLM baseline in regards to its constituent support on TableDART fusion path. Lastly, we compare against TableDART, which routes modality per question attempt with a trained model, for an isolated comparison between action-conditioned and question-level modality selection.

Trajectory optimizing baselines. To compare the advantage of our proposed metadata-guided trajectory optimization over existing trajectory optimization methods, we compare against two key baselines that optimize trajectories via a process reward model (Zou et al., 2026) and stepwise table pruning (Guo et al., 2026). The first trains a process reward model from Qwen3-8B (Yang et al., 2025a) to provide stepwise reward evaluation, while the second trains a pruner from derived gold pruning trajectories. We initiate these comparisons to evaluate the effectiveness of projecting high-dimensional structured data to low-dimensional metadata for trajectory optimization.

Broader competitive landscape. To position TABQAWORLD within the broader literature, we benchmark it against an extensive suite of single-modality baselines in the 7-8B parameter range, including table-as-text baselines under standard LLMs such as Llama-2-7B (Touvron et al., 2023), Llama3-Instruct-8B (Grattafiori et al., 2024), TableLlama-7B (Zhang et al., 2024a). We also include various table-as-image models aside from our base constituent models with Table-LLaVA-7B (Zheng et al., 2024), SynTab-LLaVA-7B (Zhou et al., 2025a), MiniCPM-V-2.6-8B (Yao et al., 2025). Additional training-free table reason-

ing agent baselines are also included, containing DATER (Ye et al., 2023), ReAcTable (Zhang et al., 2024c), Mix-SC (Liu et al., 2024), TIDE (Yang et al., 2025c) and CIT-DP (Yang et al., 2025d)

Result sources and reporting. For a comprehensive and fair comparison, we report several baseline results directly from prior work. Specifically, results for Llama2-7B, Llama3-Instruct-8B, TableLlama-7B, TableLLaVA-7B, MiniCPM-V-2.6-8B, and HIPPO are taken from (Wang et al., 2026). Results for dynamic adaptive routing, TableDART, are taken from its proposal paper (Xing et al., 2026), while results for trained trajectory-optimizing agents are taken from the proposed TATTOO PRM paper (Zou et al., 2026) and TabTrim pruner paper (Guo et al., 2026). Lastly, the training-free table reasoning agents are taken from recent published paper (Yang et al., 2025d). All remaining results not covered above are generated from our own experimental runs.

A.6 Hyperparameter Configuration

Table 6 summarizes the complete hyperparameter settings for experiments. These hyperparameters are selected through preliminary experiments to balance inference time and efficiency.

A.7 Computational Configuration

Inference is conducted on a single NVIDIA RTX4090 GPU with FP16 precision, as optimized by Ollama (team, 2025).

A.8 Efficiency Benchmark Protocol

This section provides a detailed account of the protocol used for the efficiency analysis presented in Section 7.3. We include data sampling methodology and precise definitions for reported metrics to ensure full reproducibility, as reference to protocol in prior work (Xing et al., 2026).

A.8.1 Benchmark Setup and Data Sampling

To create a representative testbed, we constructed an evaluation set via stratified random sampling from the three benchmark test sets. We randomly sampled a balanced set of 200 instances from each dataset, resulting in a comprehensive benchmark suite of 600 unique samples. To ensure statistical stability, the entire measurement process was repeated three times with different random seeds, and all reported metrics are the average across these independent runs. All benchmarks were executed on a single NVIDIA RTX4090 GPU.

A.8.2 Metric Definitions

We use the following three primary metrics to report efficiency:

- **Conversation Turns:** The total conversation turns to complete the stepwise reasoning, calculated as number of actions. This helps us identify the effectiveness of action compression in combining multiple actions into one. Lower values are better.
- **Latency (s):** Total time in seconds to process a single sample. This is our primary metric to evaluate end-to-end inference speed. Lower values are better.
- **Tokens per Second (TPS):** A measure of throughput calculated by dividing number of generated output tokens, including all reasoning tokens, by the total latency. The token count is collected during generation. Higher values are better.

B Additional Related Work

Table pre-training and semantic parsing. Early table reasoning relied on schema-aware encoders and table-specific pre-training (Herzig et al., 2020; Yin et al., 2020; Jiang et al., 2022; Lee et al., 2025). LLM-based methods later expanded the paradigm with prompting, in-context learning, retrieval, and structured tool use (Cheng et al., 2023; Chen, 2023; Sui et al., 2024; Wang et al., 2024; Yang et al., 2025c; Lin et al., 2023; Wu et al., 2023). Post-training via supervised fine-tuning and reinforcement learning further improved reasoning quality (Zha et al., 2023; Zhang et al., 2024b; Wu et al., 2025c; Yang et al., 2025b). TABQAWORLD is orthogonal to these: it is training-free and can be applied on top of any such backbone.

Multimodal representation for tables. Representation format is known to affect reasoning quality (Deng et al., 2024; Zhang et al., 2024d; Lei et al., 2025; Hu et al., 2024; Zhou et al., 2024; OpenAI, 2025b). Serialization-robustness analyses show linearized tables induce large representational drift (Deng et al., 2024). Parsing-and-retrieval pipelines preserve 2D structure via structured-to-text bridging (Wang et al., 2024). TABQAWORLD complements these by routing modality per reasoning step rather than at retrieval time.

Multi-step planning and world models. Explicit state estimation and transition modeling have

Table 6: Summary of image rendering, model generation, and experiment hyperparameters.

Parameter	Value / Default	Description
Image Rendering Hyperparameters		
Max rows displayed	30	Truncate overly wide tables
Max cols displayed	5	Truncate overly wide tables
Font size	8pt	Standardize patch size for each word
Row scale	1.2×	Standardize patch size for each row
DPI	200	Standardize resolution
Format	`.jpg`	Standardize image format
Model / Generation Hyperparameters		
Max tokens	8192	Max output tokens per call
Temperature	0.7	Sampling temperature (Mandatory 1 for GPT-5 models)
Seed	42	Random seed (optional)
Experiment / Agentic Loop Hyperparameters		
Max steps	12	Max tool-use steps per question
Runtime retries	2	Retries per step for JSON parse errors
Attempts per question	1	Independent attempts per question
Seed	42	Random seed (optional)

been explored in multi-step interaction (Xing et al., 2025a; Wu et al., 2025a), tabular grounding (QU et al., 2025; Jiang et al., 2023; Nguyen et al., 2025), and step-level reward modeling (Liu et al., 2024; Kwok et al., 2026). Action-conditional generative world models (Ha and Schmidhuber, 2018; Google DeepMind, 2025; Hafner et al., 2019) and lookahead planning (Yu et al., 2026) inform TABQAWORLD's trajectory optimization design, though TABQAWORLD replaces learned dynamics with lightweight metadata verification.

C Supporting Information of Section 7 Ablation Studies

C.1 Details of Section 7 Ablation Studies

This section includes detailed numerical figures for Main Results (Figures 6 and 7). Specifically, Figure 8 illustrates the specific performance on each dataset, where similar performance patterns are observed across the three datasets, indicating the dataset-agnostic improvement by TABQAWORLD. In particular, the accuracy gain from action-conditioned multimodal selection is larger across the three smaller models than for the large proprietary GPT-5.4. This suggests that a larger model is naturally more capable of understanding text-serialized tabular data and attending to more distant tokens, offsetting the inherent advantage of the action-conditioned multimodal selection policy. Nevertheless, under a resource-constrained budget, we observe a consistent accuracy improvement in having action-conditioned multimodal to represent tabular state.

C.2 Supporting ROUGE and BLEU evaluation for Section 7

In our additional evaluation with ROUGE (Lin, 2004) and BLEU (Papineni et al., 2002), we observe that ROUGE scores indicate similar performance patterns as the accuracy-based evaluation, with a much less consistent result on the BLEU evaluation metrics. This suggests that lexical overlapping metrics may not be well suited for evaluating table reasoning correctness, as they fail to measure the semantic equivalence of the generated answer, especially in scenarios of paraphrasing, formatting variation or abbreviations. As ROUGE shows to be more consistent with the accuracy results, we attribute this to BLEU's design for machine translation and emphasis of exact n-gram precision. This renders BLEU harsh on cases with short ground truth, which happen to be the case given existing TQA datasets.

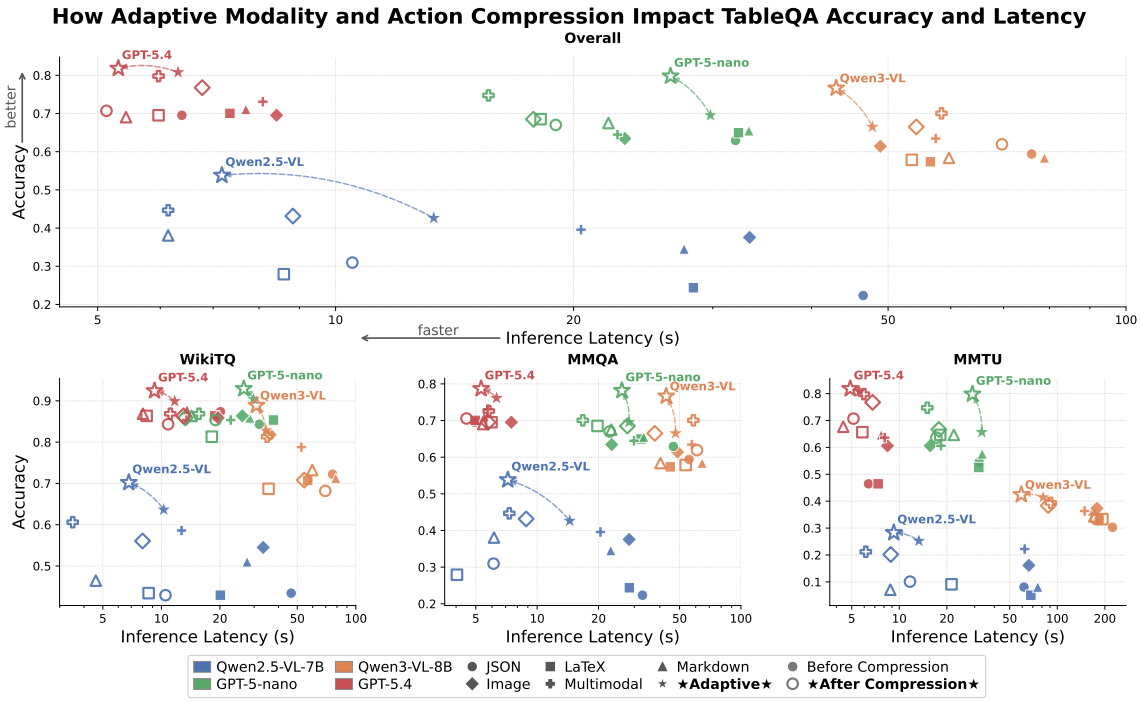

Figure 8: TABQAWORLD consistently improves accuracy and reduces latency across all four backbone models on WTQ, MMQA, and MMTU. Detailed per-model numerical results are in Table 7.

C.3 Modality Accuracy by Question Type

Following Yang et al. (2025d), we group questions into nine types spanning lookup and arithmetic behaviors. As shown in Figure 9 (left), vision-supported modalities consistently outperform text-serialized tables in "which" and most arithmetic-based tasks, which require cross-row handling and computation. Operations requiring cross-row aggregation improve the most under visual representations. Nevertheless, all modalities still struggle in arithmetic operations that require a particular order (ratio, difference), indicating that multi-cell ordered computation remains a reasoning bottleneck for existing MLLMs.

D Case Study: How Table Representation Affects Attention

Figure 9 reports a case study on how MLLMs attend to the table when given queries requiring different actions. By extending the 'Algeria' question from Figure 2 into 20 *1-step* questions, we annotate a human-preferred attention standard as follows: two co-authors independently labeled each table cell in binary (1 = query-relevant, 0 = not relevant) for all 20 questions. Each annotator followed the same written guideline ("mark cells whose values are directly needed to answer the query"). Disagreements were resolved by discussion to produce the final binary mask. We then measure the MSE between actual cell-level attention and this standard. Image modality consistently achieves lower MSE than text modalities, consistent with more precise grounding over relevant table regions. We note that attention maps are an indirect proxy for model behavior and should be interpreted as correlational rather than causal evidence. Details of the questions are in Table 11.

E Case study on how different text serialization formats affect attention on tables

Both empirical results in Table 1 and Figure 8 indicate that LLMs' understanding of table structure is sensitive to different text serialization formats, motivating us to conduct an in-depth attention-level case study.

We evaluate the WTQ (Pasupat and Liang, 2015) using 4-bit quantized Qwen3-VL-8B-Instruct (Bai et al., 2025a). For each table-question pair, we extract decoder attention at the answer-generation position and aggregate it across the 36 decoder layers of the chosen MLLM to measure the mean attention distribution over the prompt tokens. This allows us to examine how the MLLM agent distributes the attention budget across the table con-

Table 7: Detailed accuracy of Figure 8

Accuracy													
Modality	`json`	`latex`	`Markdown`	Image	Multimodal	Adaptive	`json`	`latex`	`Markdown`	Image	Image	Multimodal	Adaptive
Optimization	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓(Metadata)	✓(Table)	✓	✓
WTQ													
Qwen2.5-VL-7B	0.4343	0.4293	0.5101	0.5450	0.5859	0.6364	0.4293	0.4343	0.4646	0.5606	0.4814	0.6061	**0.7020**
Qwen3-VL-8B	0.7222	0.7071	0.7121	0.8182	0.7929	0.8333	0.7071	0.6869	0.7323	0.7980	0.7525	0.8131	**0.8889**
GPT-5-nano	0.8434	0.8535	0.8586	0.8636	0.8535	0.9040	0.8535	0.8131	0.8636	0.8586	0.8131	0.8687	**0.9293**
GPT-5.4	0.8737	0.8636	0.8788	0.8586	0.8535	0.8990	0.8434	0.8636	0.8687	0.8636	0.8384	0.8687	**0.9242**
MMQA													
Qwen2.5-VL-7B	0.2234	0.2437	0.3452	0.3756	0.3959	0.4264	0.3096	0.2792	0.3807	0.4315	0.3056	0.4467	**0.5381**
Qwen3-VL-8B	0.5939	0.5736	0.5838	0.6142	0.6345	0.6650	0.6193	0.5787	0.5838	0.6650	0.5431	0.7005	**0.7665**
GPT-5-nano	0.6294	0.6497	0.6548	0.6345	0.6447	0.6954	0.6701	0.6853	0.6751	0.6850	0.6294	0.7005	**0.7817**
GPT-5.4	0.6954	0.7005	0.7107	0.6954	0.7310	0.7614	0.7056	0.6954	0.6904	0.6954	0.6244	0.7259	**0.7868**
MMTU													
Qwen2.5-VL-7B	0.0808	0.0505	0.0808	0.1616	0.2222	0.2525	0.1010	0.0909	0.0707	0.2020	0.1053	0.2121	**0.2828**
Qwen3-VL-8B	0.3030	0.3333	0.3636	0.3737	0.3636	0.4141	0.3333	0.3333	0.3434	0.3838	0.3030	0.3939	**0.4242**
GPT-5-nano	0.5455	0.5253	0.5758	0.6061	0.6061	0.6566	0.6364	0.6465	0.6465	0.6667	0.4646	0.7475	**0.7980**
GPT-5.4	0.4646	0.4646	0.6465	0.6061	0.6364	0.8081	0.7071	0.6566	0.6768	0.7677	0.5253	0.7980	**0.8182**

Table 8: Detailed standard deviation for Figure 8

Standard Deviation													
Modality	`json`	`latex`	`Markdown`	Image	Multimodal	Adaptive	`json`	`latex`	`Markdown`	Image	Image	Multimodal	Adaptive
Optimization	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓(Metadata)	✓(Table)	✓	✓
WTQ													
Qwen2.5-VL-7B	0.0352	0.0352	0.0355	0.0354	0.0350	0.0342	0.0352	0.0352	0.0354	0.0353	0.0355	0.0347	**0.0325**
Qwen3-VL-8B	0.0318	0.0323	0.0322	0.0274	0.0288	0.0265	0.0323	0.0330	0.0315	0.0285	0.0307	0.0277	**0.0223**
GPT-5-nano	0.0258	0.0251	0.0248	0.0244	0.0251	0.0209	0.0251	0.0277	0.0244	0.0248	0.0277	0.0240	**0.0182**
GPT-5.4	0.0236	0.0244	0.0232	0.0248	0.0251	0.0214	0.0258	0.0244	0.0240	0.0244	0.0262	0.0240	**0.0188**
MMQA													
Qwen2.5-VL-7B	**0.0299**	0.0308	0.0339	0.0345	0.0348	0.0352	0.0329	0.0320	0.0346	0.0353	0.0328	0.0354	0.0356
Qwen3-VL-8B	0.0350	0.0352	0.0351	0.0347	0.0343	0.0336	0.0346	0.0352	0.0351	0.0336	0.0355	0.0326	**0.0301**
GPT-5-nano	0.0344	0.0340	0.0339	0.0343	0.0341	0.0328	0.0335	0.0331	0.0334	0.0331	0.0344	0.0326	**0.0294**
GPT-5.4	0.0328	0.0326	0.0323	0.0328	0.0316	0.0304	0.0325	0.0328	0.0329	0.0328	0.0345	0.0318	**0.0292**
MMTU													
Qwen2.5-VL-7B	0.0193	**0.0155**	0.0193	0.0260	0.0294	0.0307	0.0213	0.0203	0.0181	0.0284	0.0217	0.0289	0.0318
Qwen3-VL-8B	**0.0327**	0.0336	0.0343	0.0345	0.0343	0.0351	0.0336	0.0336	0.0338	0.0346	**0.0327**	0.0348	0.0352
GPT-5-nano	0.0352	0.0353	0.0349	0.0346	0.0346	0.0336	0.0340	0.0338	0.0338	0.0333	0.0353	0.0307	**0.0284**
GPT-5.4	0.0353	0.0353	0.0338	0.0346	0.0340	0.0278	0.0322	0.0336	0.0331	0.0299	0.0353	0.0284	**0.0273**

tent as well as the format-specific structural tokens. We standardize the attention evaluation by measuring the attention ratio of different components of a table reasoning query. Specifically, we have the following four evaluation diagnostics:

- Format overhead ratio that measures the fraction of table token budget spent on syntax $\frac{N_{\text{structural tokens}}}{N_{\text{table tokens}}}$
- Fraction of header attention to data attention to evaluate if the data rows receive sufficient attention: $\frac{\frac{1}{N_{\text{header tokens}}}\sum_{i\in\text{header tokens}}\text{attn}(i)}{\frac{1}{N_{\text{data tokens}}}\sum_{j\in\text{data tokens}}\text{attn}(j)}$
- Fraction of table attention to all tokens to evaluate how much prompt attention is directed to the table: $\frac{\frac{1}{N_{\text{table tokens}}}\sum_{i\in\text{table tokens}}\text{attn}(i)}{\frac{1}{N_{\text{tokens}}}\sum_{j\in\text{tokens}}\text{attn}(j)}$
- Table attention entropy which evaluates the Shannon entropy of attention on each content token to evaluate whether agent's attention is focused on specific cells, where a lower entropy means that the attention is more focused on specific cells

We observe that the attention metrics for `LaTeX` and `Markdown` are more similar to each other than to `JSON`: both share similar table attention fractions, entropy values, and header-data attention ratios, while `JSON`'s format overhead ratio is higher and its header-data attention ratio is lower. This pattern is consistent with `JSON`'s distinctive format, where the column name is repeated in every serialized row (Table 12), whereas `LaTeX` and `Markdown` list headers once at the beginning. The

Table 9: Detailed time latency of Figure 8

Inference Latency (in seconds)													
Modality	`json`	`latex`	`Markdown`	Image	Multimodal	Adaptive	`json`	`latex`	`Markdown`	Image	Image	Multimodal	Adaptive
Optimization	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓(Metadata)	✓(Table)	✓	✓
WTQ													
Qwen2.5-VL-7B	46.47	20.05	27.62	33.35	12.73	10.28	10.50	8.58	4.59	7.95	12.68	**3.50**	6.75
Qwen3-VL-8B	75.96	56.56	78.85	36.22	52.48	34.44	69.67	35.59	59.70	54.28	40.89	35.00	**30.83**
GPT-5-nano	31.86	37.71	28.48	26.07	22.73	29.80	18.99	18.17	14.31	**13.27**	27.34	15.61	26.53
GPT-5.4	20.13	18.82	13.55	19.58	13.47	11.71	10.83	8.40	**8.04**	13.00	8.09	11.12	9.22
MMQA													
Qwen2.5-VL-7B	32.88	28.36	22.97	28.27	20.44	14.45	6.11	**4.04**	6.14	8.83	6.83	7.30	7.18
Qwen3-VL-8B	55.68	45.00	64.48	48.89	57.44	47.77	60.89	53.58	40.28	37.79	36.01	58.42	**42.98**
GPT-5-nano	46.59	32.34	33.33	23.22	29.87	28.24	22.68	19.77	23.12	27.72	20.98	**16.75**	26.06
GPT-5.4	4.94	4.97	5.50	7.47	5.72	6.32	**4.50**	5.97	5.43	5.74	5.66	5.79	5.31
MMTU													
Qwen2.5-VL-7B	61.53	68.03	75.26	66.10	62.18	13.31	11.69	21.40	8.78	8.83	8.29	**6.14**	9.23
Qwen3-VL-8B	223.76	183.87	166.59	178.51	148.91	81.37	176.81	193.02	171.19	87.65	53.17	90.83	**59.22**
GPT-5-nano	32.06	31.86	33.47	15.69	18.36	33.30	17.16	18.15	22.14	17.79	39.01	**15.05**	29.02
GPT-5.4	6.39	7.35	7.70	8.42	8.09	5.60	5.13	5.84	**4.42**	6.78	9.61	5.97	4.89

Table 10: ROUGE (Lin, 2004) and BLEU (Papineni et al., 2002) score of Main Result Figure 8

ROUGE / BLEU													
Modality	`json`	`latex`	`Markdown`	Image	Multimodal	Adaptive	`json`	`latex`	`Markdown`	Image	Image	Multimodal	Adaptive
Optimization	✗	✗	✗	✗	✗	✗	✓	✓	✓	✓(Metadata)	✓(Table)	✓	✓
WTQ													
Qwen2.5-VL-7B	0.608 / 0.029	0.618 / **0.039**	0.580 / 0.035	0.599 / 0.037	0.640 / 0.024	0.582 / 0.026	0.546 / 0.009	0.496 / 0.024	0.512 / 0.016	0.573 / 0.024	0.467 / 0.000	0.565 / 0.029	**0.642** / 0.038
Qwen3-VL-8B	0.709 / 0.038	0.713 / 0.030	0.715 / 0.028	0.769 / 0.034	0.768 / 0.034	0.810 / **0.040**	0.717 / 0.036	0.722 / 0.031	0.753 / 0.034	0.774 / 0.026	0.780 / 0.031	0.775 / 0.035	**0.852** / 0.035
GPT-5-nano	0.745 / **0.045**	0.742 / 0.033	0.758 / 0.037	0.764 / 0.031	0.746 / 0.025	0.785 / 0.040	0.772 / 0.035	0.732 / 0.040	0.738 / 0.037	0.734 / 0.038	0.701 / 0.031	0.744 / 0.040	**0.793** / 0.039
GPT-5.4	0.764 / 0.031	0.746 / **0.034**	0.760 / 0.033	0.717 / 0.033	0.711 / 0.026	**0.794** / 0.032	0.704 / **0.034**	0.705 / 0.029	0.736 / 0.031	0.690 / 0.032	0.727 / 0.024	0.679 / 0.030	0.784 / 0.031
MMQA													
Qwen2.5-VL-7B	0.320 / 0.030	0.398 / 0.049	0.404 / 0.034	0.393 / 0.024	0.407 / 0.027	0.436 / 0.030	0.387 / 0.039	0.353 / 0.035	0.396 / 0.033	0.385 / 0.032	0.327 / 0.004	0.402 / 0.040	**0.477** / **0.045**
Qwen3-VL-8B	0.590 / 0.054	0.566 / 0.038	0.556 / 0.046	0.584 / 0.051	0.579 / 0.043	0.615 / 0.047	0.585 / 0.046	0.568 / 0.047	0.588 / 0.050	0.630 / **0.051**	0.564 / 0.044	0.638 / 0.048	**0.698** / 0.049
GPT-5-nano	0.572 / **0.057**	0.590 / 0.055	0.590 / 0.054	0.588 / 0.048	0.605 / 0.054	0.636 / 0.050	0.594 / 0.050	0.599 / 0.054	0.573 / 0.048	0.580 / 0.045	0.586 / 0.036	0.596 / 0.048	**0.669** / 0.056
GPT-5.4	0.590 / **0.054**	0.592 / 0.053	0.585 / 0.049	0.589 / 0.043	0.594 / 0.046	**0.626** / 0.048	0.553 / 0.039	0.540 / 0.031	0.572 / 0.041	0.558 / 0.046	0.562 / 0.045	0.574 / 0.049	0.579 / 0.047
MMTU													
Qwen2.5-VL-7B	0.079 / 0.000	0.084 / 0.000	0.087 / 0.000	0.145 / 0.000	0.166 / 0.000	**0.218** / 0.000	0.121 / 0.000	0.101 / 0.000	0.067 / 0.000	0.132 / 0.000	0.082 / 0.000	0.177 / 0.000	0.206 / 0.000
Qwen3-VL-8B	0.258 / 0.000	0.296 / 0.000	0.321 / 0.000	0.309 / 0.000	0.315 / 0.000	0.341 / 0.000	0.283 / 0.000	0.288 / 0.000	0.370 / 0.000	0.327 / 0.000	0.306 / 0.000	0.330 / 0.000	**0.355** / 0.000
GPT-5-nano	0.276 / 0.000	0.303 / 0.000	0.339 / 0.000	0.283 / 0.000	0.315 / 0.000	0.290 / 0.000	0.305 / 0.000	**0.346** / 0.000	0.314 / 0.000	0.323 / 0.000	0.297 / 0.000	0.337 / 0.000	0.316 / 0.000
GPT-5.4	0.285 / 0.000	0.286 / 0.000	**0.341** / 0.000	0.288 / 0.000	0.305 / 0.000	0.300 / 0.000	0.324 / 0.000	0.335 / 0.000	0.317 / 0.000	0.303 / 0.000	0.296 / 0.000	0.280 / 0.000	0.333 / 0.000

similar accuracy between `LaTeX` and `Markdown` suggests that delimiter choice alone has limited impact, consistent with prior findings (Kwok et al., 2026). We hypothesize that the repeated column-name prefix in `JSON` allows the MLLM to interpret each row more independently, without needing to cross-reference distant headers, which may explain its modest accuracy advantage despite higher token cost.

The error localization analysis provides further evidence consistent with this hypothesis: `JSON` shows a 51%/55% format overhead ratio for correct/incorrect questions respectively, versus (38%/34%) for `LaTeX` and (48%/45%) for `Markdown`. The lower header-data attention ratio for `JSON` is consistent with the agent connecting data entries and headers more directly, and the lower table attention entropy suggests more focused cell-level attention. We note that these attention patterns are indirect measurements and do not constitute causal proof. They are consistent with the hypothesis that per-row column repetition improves row-level semantic accessibility in long tables. We also seek to understand whether these specific factors of different formatting would interact with different question types.

Break down performance by question types The question-type analysis in Figure 10 shows the weakness of the header-repeating `JSON`, as this action increases the contextual length of each row. While this facilitates unordered retrieval (illustrated by its outperformance in `sum` and `who`), it underperforms in tasks which require retrieving multiple row values (`which` and `order`). This suggests that as an increased structural syntax preserves individual semantic meaning of each row indepen-

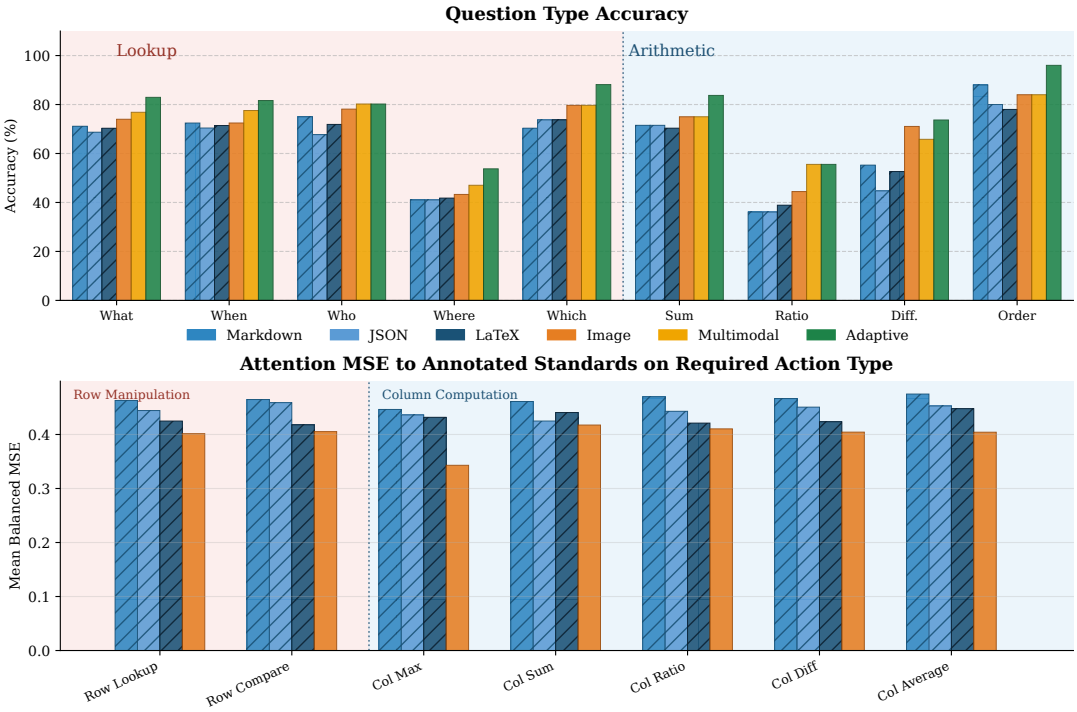

Figure 9: In this small single-table case study (20 questions, one table), visual-included representations yield both higher accuracy across question types and lower MSE relative to human-preferred attention, suggesting an association between precise grounding and table reasoning quality. Causal claims are not supported by this study design.

dently, it also results in more attention being absorbed by structural syntax instead of semantic content, hence hurting performance on tasks that require multi-row retrievals, where JSON fails to disperse attention effectively across cell values. This further suggests that integrating diverse serialization formats into the action-conditioned modality selection policy is a promising direction for improving tabular understanding.

F Case Study: How Table Size Affects Modality Performance

To understand when different table representations succeed or fail, we conduct a case study on the effect of table size on reasoning accuracy. We measure table size by the number of serialised tokens, as some tables contain long text cells, where simple row/column counts fail to capture contextual length within a cell. Within our evaluated datasets and models, we observe a non-monotonic relationship between table size and accuracy in Table 14. Specifically, we observe relatively low performance in the smallest token bin. Performance peaks at the second-smallest token bin and declines steadily as the table grows larger. Aside from the proposed action-conditioned multimodal selection policy, which consistently outperforms all fixed-modality settings as shown in Figure 6, visual-grounded modalities such as image and multimodal show more graceful performance degradation. We attribute this to visual rendering, which partially compresses the table before reasoning and releases some of the attention burden on symbolic delimiters for table structuring, which are unavoidable in text serialization formats. Regarding the underperformance in the smallest bin, in contrast to the overall decreasing trend, we conduct a more in-depth study on each dataset.

Break down performance by datasets The low accuracy in the smallest token bin is partly attributable to dataset composition. In that bin, MMTU, which has the lowest per-dataset accuracy, contributes a relatively high 31% of samples, compared to only 20% from WTQ (which achieves 91% accuracy in that bin). We hypothesize that these small MMTU tables present disproportionately difficult scenarios: compact tabular fragments requiring dense numerical reasoning (e.g., percentage growth over two rows) combined with ambiguous question keywords that prevent direct column retrieval from headers. WTQ and MMQA tables in

Table 11: Designed questions and required reasoning actions.

Question	Ground Truth	Required Actions
How many plants are in Algeria?	6	sum_over_column
What is the average capacity of Algeria plants?	4.933	average_over_column
What is the name and capacity of the plant in Brunei?	Lumut 1 with 7.2 mmtpa	lookup_row
What is the average capacity of Malaysia plants?	6.267	lookup_row, average_over_column
What is the latest plant built in Qatar?	2010	lookup_row, max_over_column
What is the average capacity of all plants?	8.508	average_over_column
Which country has the most plants?	Algeria and Indonesia	sum_over_column
Are all plants in Egypt located in the same location?	No	lookup_row, comparison_across_rows
Is the average capacity of Malaysian plants higher than that of Egyptian plants?	No	lookup_row, average_over_column, comparison_across_rows
Which country has the highest-capacity plant?	Qatar	max_over_column, lookup_row
What is the age difference between Algeria's oldest and newest plant?	35	lookup_row, difference_within_column
Between Algeria's oldest and newest plants, how many times as large is the capacity of the newest plant as the oldest?	6.667	ratio_within_column, lookup_row
Between plants from Angola and Brunei, which country's plant(s) has its location stated?	Angola	comparison_across_rows
Between the plant in Indonesia and Yemen, how many times as large is the capacity of the Indonesia plant as the Yemen plant?	1.134	lookup_row, ratio_within_column
Which country has the newest plant?	Angola	lookup_row, comparison_across_rows
Considering only plants with given capacity, is any Algeria plant larger in capacity than any Indonesia plant?	No	lookup_row, comparison_across_rows, difference_within_column
What is the age difference between the oldest and newest plants in the table?	49	difference_within_column
What is the absolute capacity difference between the Brunei plant and the Nigeria plant?	16.3	lookup_row, difference_within_column
What is the relative capacity difference between the Brunei plant and the Nigeria plant?	2.264	lookup_row, ratio_within_column
Which country has the widest age gap between oldest and newest plant?	Algeria	difference_within_column, lookup_row, comparison_across_rows

the same bin appear more standardized, which we expect facilitates table understanding. A more controlled per-dataset split would be needed to confirm this attribution quantitatively.

G Error Analysis

We conduct a systematic error analysis on failure cases of TABQAWORLD across WTQ, MMQA, and MMTU using Qwen3-VL-8B, GPT-5-nano, and GPT-5.4 (the three backbones with complete multi-turn reasoning traces. Qwen2.5-VL-7B is excluded due to incomplete trace coverage). GPT-4.1-mini served as the LLM judge, producing 247 failures out of 1,491 questions in total. We categorize each failure by inspecting the per-step `error` field and question type in the reasoning trace. *Tool runtime error* covers predicate execution failures (`UnboundLocalError` in `f_select_row`), empty result sets, and serialisation issues. *Tool invocation error* covers missing or malformed tool arguments.

Answer extraction error (58.3% overall, 68% on MMQA/MMTU). This is the dominant failure mode across all datasets. The tool sequence completes without error and the intermediate table state is correct, but the final answer produced by `f_final_answer` is incomplete or misaligned with the gold label, for example returning a column name instead of a cell value, omitting a required entity, or producing a verbose explanation rather than a concise answer. The prevalence is markedly higher on MMQA and MMTU than on WTQ, which we attribute to MMQA/MMTU answers often requiring precise multi-entity or cell-coordinate formats that leave little tolerance for

Table 12: Canonical serialization templates for the same 3-column row.

Format	Example serialization
JSON	`["Col1":"val1","Col2":"val2","Col3":"val3"]`
LaTeX	`begin{tabular}{lll} Col1 & Col2 & Col3 <ROW> val1 & val2 & val3 end{tabular}`
Markdown	`| Col1 | Col2 | Col3 | <ROW> | val1 | val2 | val3 |`

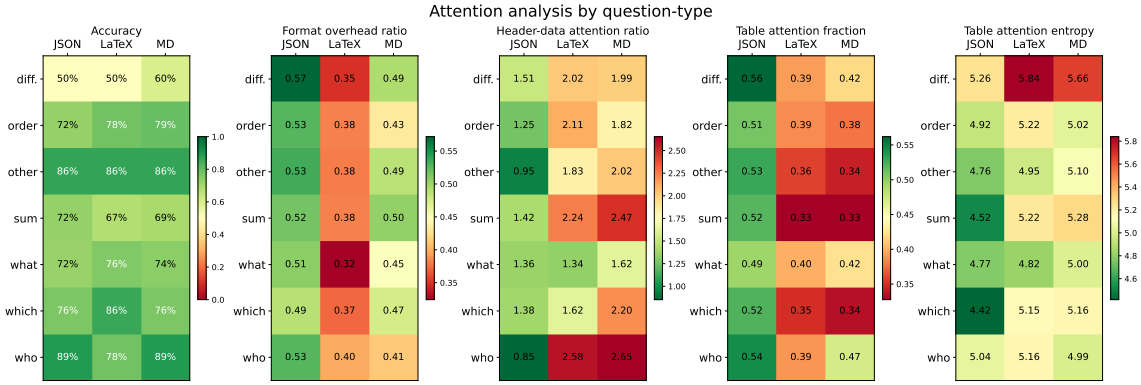

Figure 10: Heatmap of question-type performance and attention statistics across `JSON`, `LaTeX` and `Markdown` table serializations. Results show that while `JSON` devotes more attention to tables to improve lookup tasks, its heavier format overhead leads to underperformance in ordered computation tasks across multiple rows.

Table 13: Accuracy and attention diagnostics across three text serialization formats on WTQ. Each cell shows the overall metric (Correct Q / Wrong Q) where split results are provided.

Overall Metrics (Correct Q / Wrong Q)			
Bin (Tokens)	**JSON**	**LaTeX**	**Markdown**
Accuracy (%)	72.2	70.7	71.2
Average Structural Token	231	147	121
Total Token	469	416	417
Format overhead ratio (%)	51.8 (51/55)	37.0 (38/34)	46.9 (48/45)
Header-data attention ratio	1.29 (1.3/1.2)	1.95 (2.0/1.7)	2.10 (2.3/1.6)
Table attention fraction (%)	51.6 (51/52)	36.3 (34/43)	36.7 (34/43)
Table attention entropy	4.71 (4.6/4.9)	5.13 (5.1/5.2)	5.14 (5.1/5.1)

Table 14: LLM Accuracy (%) by Token-Size Bin: Aggregated and Per-Dataset Breakdown

Overall Accuracy (%) (WTQ/MMQA/MMTU)						
Bin (Tokens)	**JSON**	**LaTeX**	**Markdown**	**Image**	**Multimodal**	**Adaptive**
41-263	59.5% (80/60/21)	60.2% (82/59/21)	58.3% (81/59/20)	62.8% (85/63/30)	63.1% (84/63/30)	**70.1%** (91/68/41)
264-375	74.5% (83/65/75)	75.0% (85/67/79)	76.9% (85/70/75)	77.7% (86/63/82)	80.0% (85/72/82)	**83.5%** (88/74/86)
376-510	73.5% (88/67/74)	74.8% (84/66/74)	76.1% (89/68/86)	74.3% (88/69/81)	74.8% (88/67/86)	**78.5%** (92/73/83)
513-816	64.9% (69/62/30)	66.8% (70/65/33)	69.7% (73/69/34)	69.0% (73/64/34)	70.7% (74/65/38)	**75.5%** (79/69/43)
817-1695	67.3% (74/67/27)	66.0% (73/66/31)	68.9% (75/66/32)	71.4% (80/68/34)	72.9% (78/69/38)	**77.8%** (82/77/42)
1720-94774	45.7% (73/47/31)	45.4% (72/48/24)	47.0% (71/48/32)	48.7% (72/50/37)	51.0% (73/56/37)	**55.7%** (77/62/39)

verbosity.

Tool runtime error (25.5%). Runtime failures arise from three sub-cases: (i) `f_select_row` predicates that reference an intermediate variable before it is bound, (ii) over-restrictive filters that produce an empty intermediate table, blocking downstream rendering, and (iii) date or timestamp values that are not JSON-serialisable. These errors are framework-specific: static single-turn baselines incur zero tool errors by construction. Nevertheless, TABQAWORLD's overall WTQ error rate (7.5%) is still less than half that of the best static baseline (15.2%), confirming that the tool overhead is outweighed by the reasoning gains. These errors are also in principle recoverable with a rephrased predicate, and represent a promising direction for future work.

Arithmetic error (11.7%, 38.6% on WTQ). On WTQ this is the most common failure: the model selects the correct rows and columns but produces the wrong final numerical result for `sum` or `difference` questions. Critically, this failure mode is task-inherent rather than

Table 15: Error breakdown by dataset (Qwen3-VL-8B + GPT-5-nano + GPT-5.4 combined). n is the number of questions per model. Failures include both incorrect and partial verdicts.

Dataset	Failures	Answ. extr.	Tool runtime	Arithmetic	Tool invoc.
WTQ (n=200)	44	18.2%	34.1%	38.6%	9.1%
MMQA (n=197)	114	68.4%	27.2%	1.8%	2.6%
MMTU (n=100)	89	65.2%	19.1%	11.2%	4.5%
Overall	247	58.3%	25.5%	11.7%	4.5%

framework-specific: the static image baseline already fails on 20.4% of WTQ arithmetic questions, whereas TABQAWORLD reduces this to 13.2% (a 35% relative improvement), and arithmetic questions account for 44-61% of static-image failures depending on the backbone. The low rate on MMQA/MMTU (under 2-11%) reflects those benchmarks' lower proportion of purely aggregation-type questions.

Tool invocation error (4.5%). A small fraction of errors stem from the MLLM generating tool calls with missing required arguments (e.g., omitting `col_B` for `f_compute_column`). These failures are relatively rare across all backbone models, reflecting that modern MLLMs reliably follow structured tool-call formats after sufficient instruction tuning.

G.1 Error Rate by Table Representation

To understand how modality choice affects failure rates, we compare the LLM-judged error rate across five conditions (the four static baselines and the full TABQAWORLD) using the same three backbone models (Qwen3-VL-8B, GPT-5-nano, GPT-5.4) and three datasets. Each cell in Table 16 reports the fraction of questions receiving an *incorrect* or *partial* verdict from the LLM judge.

Table 16: LLM-judged error rate (%) by table representation across WTQ, MMQA, and MMTU (Qwen3-VL-8B + GPT-5-nano + GPT-5.4 combined: $n\approx$198-200 per model per dataset). *Adaptive only* = action-conditioned modality selection without trajectory optimisation.

Representation	WTQ	MMQA	MMTU
Static text (Markdown)	16.5%	33.3%	45.7%
Static image	15.2%	33.5%	45.9%
Static multimodal	16.0%	31.2%	44.6%
Adaptive modality only	12.0%	27.3%	36.3%
Full TABQAWORLD	**7.5%**	**19.7%**	**30.5%**

Three observations stand out. First, static text and static image produce nearly identical error rates (within 1-2 pp) across all datasets, confirming that neither representation is inherently better. The per-step action context is what makes modality selection beneficial. Second, adaptive modality selection alone reduces errors by $\sim$21% relative on WTQ (best static baseline: image, 15.2%) and $\sim$12% on MMQA (best static baseline: multimodal, 31.2%), a consistent gain that holds across all three datasets. Third, adding trajectory optimisation (full TABQAWORLD) provides a further *complementary* reduction: the error rate drops by an additional 38% relative on WTQ (12.0% to 7.5%), 28% on MMQA (27.3% to 19.7%), and 16% on MMTU (36.3% to 30.5%). The two components together achieve a cumulative 51% relative error reduction over the best static single-modality baseline on WTQ, validating that modality routing and trajectory optimisation address complementary failure modes.

G.2 Action Batching Error Breakdown

To quantify the error cost of action batching, we analyse step-level reasoning traces for Qwen3-VL-8B under two conditions: the full TABQAWORLD (adaptive modality + metadata-guided batching n=200/200/100) and the random-modality ablation (random per-step modality, partial counts WTQ n=128, MMQA n=131, MMTU n=84). For each reasoning step we classify it as *single-action* (one tool call per step) or *batched* (two or more tool calls issued in the same step), and record whether an execution error occurred.

Table 17: Step-level error rates for single-action vs. batched steps. "Full TABQAWORLD" uses complete run counts (n=200/200/100). "Random mod." are partial run counts. Batching rate = fraction of steps that are multi-action.

Condition	Dataset	Batch rate	Single err.	Batched err.	Ratio
Full TABQAWORLD	WTQ	29.9%	39.6%	52.2%	1.3×
	MMQA	26.7%	28.8%	34.1%	1.2×
	MMTU	18.7%	40.8%	57.1%	1.4×
Random mod.	WTQ	32.7%	23.6%	55.9%	2.4×
	MMQA	35.2%	18.1%	31.8%	1.8×
	MMTU	17.8%	18.0%	62.5%	3.5×

Three findings stand out. First, batched steps consistently fail at higher rates than single-action steps in both conditions, confirming that action batching introduces additional execution risk. Second, the error amplification ratio is markedly lower under full TABQAWORLD (1.2-1.4×) than under random modality (1.8-3.5×), indicating that the metadata consistency feedback discourages high-risk batching in subsequent steps. Third, at

the per-operation level, `f_compute_column`, `f_string_operation`, and `f_aggregate` reach 90-100% error rates when batched in both conditions, whereas `f_sort_by` shows near-zero error rates, consistent with it being a dimension-preserving, order-only transform. These high-risk operations silently mutate cell values in ways that row-count and column-name checks cannot detect. This is precisely the gap noted in the Limitations section, and it motivates tightening the metadata confidence threshold for aggregation and string-transform actions.

G.3 Modality Pre-emption under Random Routing

In the random-modality ablation, each reasoning step is assigned a modality drawn uniformly from {text, image, multimodal}, independently of the action type. We cross-tabulate the resulting modality distribution against final answer correctness to assess whether the modality signal matters beyond action-type context.

Under random routing, the empirical modality distribution is approximately 30-40% image, 29-37% multimodal, and 24-31% text-consistent with uniform sampling over three categories. Correct answers on WTQ show slightly higher multimodal usage (41.9% vs. 34.8% for incorrect), while on MMTU correct answers favour text (36.8% vs. 27.8% for incorrect), with smaller differences on MMQA. Modality switches across consecutive steps are rare, averaging 0.19-0.25 per question, suggesting that the randomly assigned modality tends to persist even when it does not match the action type. The small per-modality accuracy differences under random routing further validate that the performance gains of adaptive modality selection stem from *action-conditioned* routing (e.g., switching to text for `compute_column`) rather than a global preference for any single representation.

H WTQ Training-Data Contamination Discussion

WikiTableQuestions (WTQ) (Pasupat and Liang, 2015) was released in 2015 and has since become one of the most widely used TQA benchmarks. As a result, it is very likely present in the pretraining corpora of modern large language models, including the Qwen family. The high accuracy achieved by TabQAWorld on WTQ (0.889 with Qwen3-VL-8B) may therefore partly reflect pretraining memorization in addition to genuine generalization from the framework's design.

We raise three mitigating considerations.

Controlled ablations isolate framework contributions. The ablation study in Section 7 evaluates *all* conditions-static modalities, random routing, heuristic routing, question-level routing, and full TabQAWorld, under the same backbone and the same evaluation set. Contamination, if present, affects all conditions equally, so the incremental accuracy gains attributed to each framework component (e.g., adaptive modality: $+5.42\%$, trajectory optimization: latency -33.35%) are not confounded by contamination.

MMQA and MMTU simulate a zero-shot environment. Precisely to mitigate contamination risk, the ablation study was designed around MMQA (Wu et al., 2025b) and MMTU (Xing et al., 2025b), two datasets released in 2025 that are not plausibly present in the pretraining data of Qwen3-VL-8B, the backbone used for all controlled ablations. The consistent performance improvements on these datasets in the within-backbone ablation (Table 3 and Table 7) therefore provide lower-contamination-risk evidence for the framework's generalization ability. Results involving proprietary backbones such as GPT-5.4 should be interpreted as exploratory given their unknown training data cutoffs.

WTQ results should be interpreted contextually. We retain WTQ in the main comparison table (Table 2) to allow comparison with the large body of prior work that also evaluates on WTQ, but caution readers that the absolute accuracy numbers on WTQ should be interpreted as an upper-bound estimate. The contamination-robust evidence for TabQAWorld's effectiveness is the controlled within-backbone ablation on MMQA and MMTU reported in Table 3.