From Table to Cell: Attention for Better Reasoning with TABALIGN

Tung Sum Thomas Kwok [*1,2] **Zeyong Zhang** [*3] **Xinyu Wang** [4,2] **Chunhe Wang** [1] **Xiaofeng Lin** [1] **Hanwei Wu** [5,2] **Lei Ding** [6,2] **Guang Cheng** [1] **Zhijiang Guo** [*7,8]

Abstract

Multi-step LLM reasoning over structured tables fails because planning and execution share no explicit cell-grounding contract. Existing methods constrain the planner to a left-to-right factorization at odds with table permutation invariance, and score intermediate states by generated content alone, overlooking cell grounding. We conduct a pilot study showing that diffusion language models (DLMs) produce more human-aligned and permutation-stable cell attention on tables than autoregressive models, with a 40.2% median reduction in attention-AUROC variability under row reordering. Motivated by this, we propose TABALIGN, a planned table reasoning framework that operationalizes the contract. TABALIGN pairs a masked DLM planner, whose bidirectional denoising emits plan steps as binary cell masks, with TABATTN, a lightweight verifier trained on 1,600 human-verified attention standards to score each step by its attention overlap with the plan-designated mask. Across eight benchmarks covering table question answering and fact verification, TABALIGN improves average accuracy by 15.76 percentage points over the strongest open-source baseline at comparable 8B-class scale, with a matched-backbone ablation attributing 2.87 percentage points of this gain to the DLM planner over an AR planner on a fixed reasoner. Cleaner DLM plans also accelerate downstream reasoning execution by 44.64%.

[*]Equal contribution [1]University of California, Los Angeles [2]SimpleWay [3]New Jersey Institute of Technology [4]McGill University, Montréal [5]Université de Montréal [6]University of Manitoba [7]The Hong Kong University of Science and Technology [8]The Hong Kong University of Science and Technology (Guangzhou). Correspondence to: Tung Sum Thomas Kwok <tk1018@ucla.edu>, Zhijiang Guo <zhijiangguo@hkust-gz.edu.cn>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

1. Introduction

Table reasoning has evolved from single-shot semantic parsing to *interactive structured reasoning*, where agents select actions to transform table states and accumulate evidence to answer [40; 8; 55; 56]. Although tool-use agents [53; 19] further extend this setting, they remain brittle on tables, where sparse evidence, permutation invariance of rows and columns, and complex schema dependencies cause early-step errors to compound across the trajectory. To mitigate this, *planned reasoning* has emerged [37; 42], requiring agents to generate a natural-language plan before executing tool calls. Despite its appeal, this approach remains brittle because current pipelines lack **an explicit cell-grounding contract between planning and execution**. Because tabular evidence localizes to specific cells, this contract must operate at cell granularity. Without such a contract to supervise the planner's cell designations and verify the reasoner's cell-level attention, prior efforts often fall into a two-stage bottleneck where plausible reasoning steps target incorrect cells (Figure 1).

This bottleneck first appears at planning time. Autoregressive (AR) generation produces plans under a left-to-right factorization, which carries an order-sensitive prior over which cells become relevant. Because tables are permutation-invariant, this prior couples plans to arbitrary serialization choices, so the same table under a different row order can yield a different plan. An early filtering step may only be valid given a later aggregation, but a causal planner commits to it before the relevant context is in scope. Our pilot study (Section 3) shows that diffusion-pretrained bidirectional attention produces more human-aligned and permutation-stable cell attention than state-of-the-art AR models.

The same bottleneck reappears at reasoning time, where existing process reward models *do not check* whether each step consults plan-designated cells. Final-answer rewards [30; 37] are too sparse to localize a step that read the wrong cell, and executor-based rewards [56] require answer-checking environments unavailable for many tabular tasks. Existing process rewards either score the produced state's lexical or semantic quality [23; 41], or rely on LLM-as-a-judge step-correctness labels that face an upstream labelability ceiling [78], and neither family checks whether

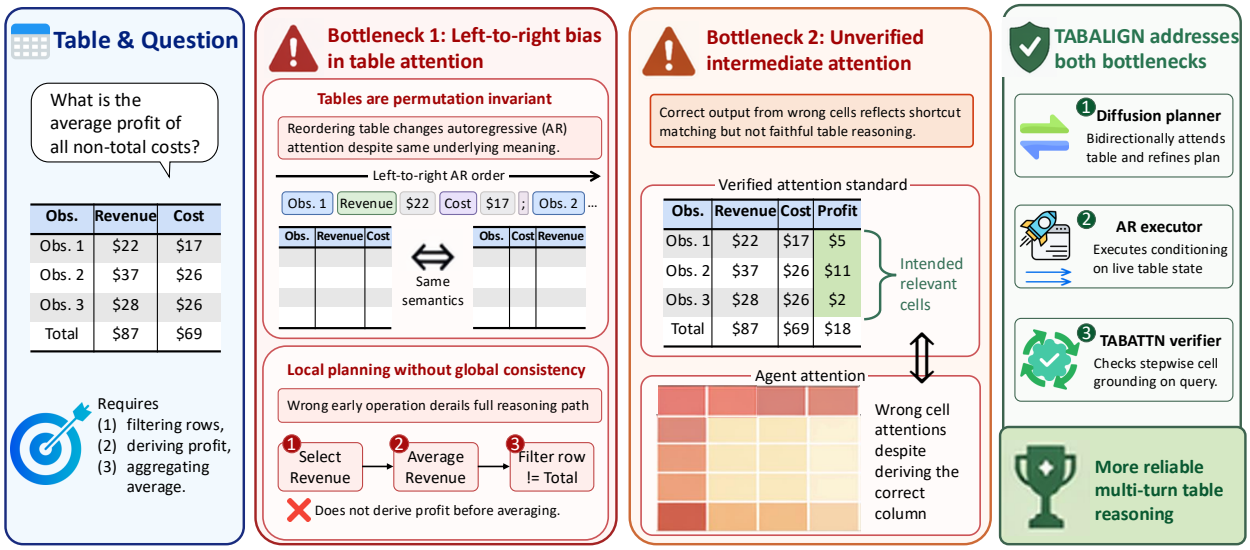


Figure 1. **Existing methods leave the cell-grounding contract implicit, producing failures at both planning and reasoning stages.** At planning time, AR models impose an order-sensitive prior that conflicts with table permutation invariance, producing a globally inconsistent plan. At reasoning time, content-based PRMs score the generated state alone, leaving cell grounding to chance. TABALIGN operationalizes the contract through a diffusion-based bidirectional planner and the TABATTN verifier.

the model attends to the right cells. Our reward-component ablation (Table 5(b)) confirms that an attention-grounded signal closes this gap with consistent accuracy gains over content-based rewards.

Read together, the two failures trace to the same missing cell-grounding contract. A planned step commits to consult a specific set of cells, so the reasoner's cell-level attention forms an inspectable, labelable proxy for whether the step is grounded in the intended evidence rather than a complete causal explanation of the model's output. Our insight is that *attention overlap with plan-designated cells yields a query-grounded, executor-free signal that exposes right-content-wrong-grounding failures invisible to content-based rewards and factorizes across planning and reasoning.*

Guided by this insight, we introduce TABALIGN, a planned table reasoning framework whose two components operationalize this contract. For *understanding*, TABALIGN uses a masked diffusion language model (DLM) to denoise a globally consistent tool-use blueprint, with bidirectional denoising capturing mutual cell dependencies schema-wide. Each step terminates with a `[target: col]` tag that compiles into a binary cell mask $\mathbf{m}_t$, making the plan's cell commitment explicit. For *reasoning*, we introduce TABATTN, a lightweight *verifier* that scores each step by the reasoner's per-cell attention overlap with $\mathbf{m}_t$. Fit on 1,600 human-verified standards from eight benchmarks, TABATTN supplies in-loop feedback with a stagnation halt on non-improving attention.

Across eight benchmarks, TABALIGN outperforms the strongest open-source baseline at comparable 8B-class scale by 15.76 pp on the six-benchmark accuracy average. A matched-backbone ablation (Table 7) attributes 2.87 pp of this gain to the DLM planner over an AR planner on a fixed Qwen3-VL-8B reasoner, with the remainder attributable to planning-vs-end-to-end execution and the attention-grounded reward. Cleaner DLM plans further accelerate downstream reasoning by 44.64 %. Our pilot study isolates two drivers behind these gains: bidirectional attention reduces attention-AUROC variability under row reordering by 40.2 %, and cell-relevance labels achieve 19.8 pp higher LLM-vs-human agreement than step-correctness labels. Component-level analyses (Section 5.2) attribute the framework gain to planner choice, reasoner backbone, and the attention-grounded reward, ruling out backbone scaling. Our contributions are:

(1) **Cell-level attention pilot study.** We conduct a pilot study showing that bidirectional attention is more human-aligned and permutation-stable than AR attention, and that cell relevance is more labelable than step correctness across eight benchmarks (Section 3).

(2) **Planned reasoning with attention grounding.** We propose TABALIGN, featuring a bidirectional DLM planner and an attention-overlap verifier (TABATTN). Our framework captures grounding failures missed by content-based rewards while utilizing optimized DLM plans to reduce column hallucinations and accelerate downstream reasoning (Section 4).

(3) **Consistent gains across benchmarks.** TABALIGN consistently outperforms existing open-source baselines at comparable 8B-class scale across eight table reasoning bench-

marks and ablates cleanly across both planner and reasoner backbones (Section 5).

2. Related Work

Table reasoning agents and process supervision. Table reasoning has evolved from single-shot semantic parsing [40; 18; 69] to interactive agent reasoning across fact verification [8; 16], multi-hop QA [76; 55], and unified evaluation benchmarks [9; 56]. Prior work [53; 19; 30; 63; 64] formulates this as a partially observable Markov Decision Process [79] that transforms table states through tool actions. Tool-using LLM agents [65; 53] predefine a closed action set but make local decisions without verifying intermediate states. PRMs [28; 49; 51; 33; 77] have been introduced into table reasoning through lexical rewards [23], learned tool-grounded supervision [78], and step-level execution rewards for agentic analysis [41], while multimodal studies [24; 57] adopt table representation selection as a reasoning component. These approaches supervise generated content alone. We complement them by adding input attention as a process signal for structured table reasoning.

Tabular data with diffusion models. Early diffusion work on tabular data focuses on data synthesis [21; 29]. TabNAT [70] extends this with a continuous-discrete joint generative framework, noting that AR generation order introduces structural misalignment for heterogeneous tabular columns. We share this observation but shift the focus to table reasoning and process supervision. With the rise of discrete diffusion [32], masked DLMs have evolved from early controllable generators [26; 2] to scaled models competitive with AR systems on structured generation [38; 67; 44], enabling planned reasoning [5] that pairs diffusion planners with AR executors. Existing diffusion-based reasoning primarily focuses on text and vision, leaving table reasoning underexplored. We address this by using masked bidirectional attention to better match the permutation invariance of tables, producing more stable and human-aligned cell attention [30] (Section D.2).

3. Pilot Analysis: Revisiting Attention Mechanism on Tables

We conduct a pilot study to identify supervision difficulty of table reasoning, guided by the following research question:

RQ: Which attention mechanism best aligns cell-level information accessibility with table semantics?

Table attention for understanding. To investigate the research question, we evaluate models with different attention mechanisms, including AR Qwen3-8B [60] and its multimodal counterpart Qwen3-VL-8B [3], and bidirectional attention models. We begin with ModernBERT-149M [54], trained via random masking. We then extend to diffusion-based models with scheduled masking, DiffuLLaMA-7B [12], which adapts AR models through diffusion post-training, and LLaDA-8B [38], a fully diffusion-pretrained model. All models operate on serialized table text tokens for modality control. DiffuLLaMA specifically serves as a within-text causal control to isolate the effect of bidirectional *pre-training* from multimodal architecture differences. We curate 1,600 human-verified table attention standards from eight datasets (details in Section B.3) Each map contains the table with each cell assigned a binary label to indicate its relevance to the question.

Table 1. Mean AUROC and permutation stability σ_{AUROC} (K=5 row permutations). Per-dataset breakdown in Table 24.

Model	Bidir.	Diff.	Overall ↑	σ_{AUROC} ↓
ModernBERT (149M)	✓	✗	0.569	**0.046**
Qwen3-VL-8B	✗	✗	0.628	0.123
Qwen3-8B	✗	✗	0.623	0.138
DiffuLLaMA-7B	✓	✓	0.639	0.123
LLaDA-8B	✓	✓	**0.677**	0.074

Table 1 shows that bidirectional attention improves cell-level AUROC against human annotations by 4.9 pp (0.628 → 0.677), with DiffuLLaMA and LLaDA both outperforming Qwen3-VL-8B despite older backbones (e.g., LLaMA-2-7B [48]). The text-only Qwen3-8B and multimodal Qwen3-VL-8B score nearly identically, and Table 15 confirms that image-pad tokens receive zero attention mass in Qwen3-VL-8B, so the AR-DLM gap reflects attention mechanism rather than multimodal pretraining. While ModernBERT's smaller scale prevents a clean attribution of its AUROC, its permutation stability ($\sigma_{\text{AUROC}} = 0.046$) exceeds even LLaDA-8B, evidence that bidirectional pre-training improves stability independently of model capacity. DiffuLLaMA's failure in stability further indicates that bidirectional *pre-training* is the load-bearing property, motivating our use of a masked DLM as the planner.

Figure 2(a) shows that LLaDA's advantage is largest with sparse relevant cells (density <10%) and narrows above 40%, while leading across all bins. This supports the intuition that bidirectional attention helps when reasoning depends on a few decisive cells, the dominant table reasoning regime. Figure 2(b) shows a WTQ example where LLaDA aligns with ground-truth cells while Qwen3-VL disperses.

Takeaway 1 (Attention Mechanism on Tables): Existing AR LLMs produce structurally misaligned cell attention on tables, whereas diffusion-pretrained bidirectional attention consistently improves alignment with human annotations and is more stable under row reordering.

From attention alignment to attention-grounded supervision. Building on Takeaway 1, we ask whether this attention can directly serve as process-level supervision, which hinges on labelability of its natural ground truth, cell-relevance. The two dominant families, *content-similarity*

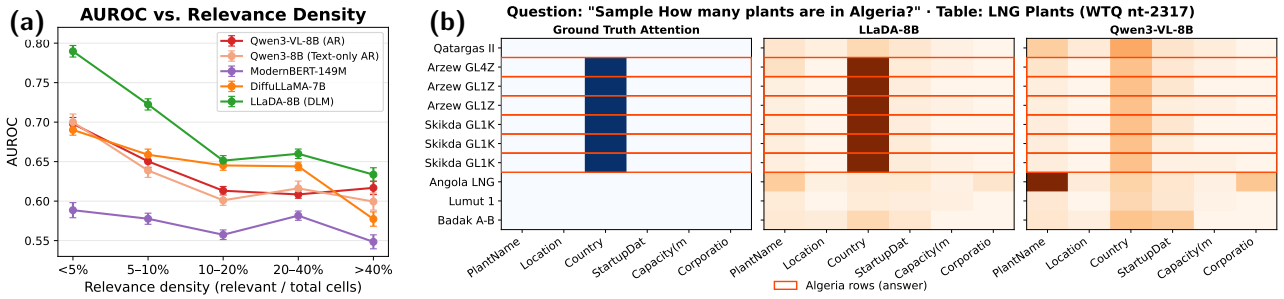


Figure 2. **Pilot analysis: bidirectional attention is more human-aligned than AR attention, with the largest gap on tables where relevant cells are sparse. (a)** LLaDA's alignment advantage is largest with sparse relevant cells (density <10%) and narrows above 40% density, leading across all bins. **(b)** Question-cell attention for "How many plants are in Algeria?": LLaDA concentrates on relevant cells while Qwen3-VL disperses across irrelevant rows.

rewards [43; 76; 23] and *step-correctness PRMs* [78], both score outputs of reasoning. We pivot to the input side, asking which cells the question depends on, a per-cell binary view supported by recent evidence that attention statistics predict downstream correctness [27]. To compare under matched conditions, we follow Krishna et al. [22] and Bologna et al. [7] in evaluating each side at its natural decision unit (one binary per step, one per cell) and run an identical LLM-as-a-judge pipeline on both, with full setup details in Section E.7.

Table 2 shows that content-similarity rewards miss content-preserving table-state changes where correct cell selection is indistinguishable from spurious overlap. Zou et al. [78] report the same pattern in their error analysis, with table-retrieval and schema-interaction errors as the dominant failure categories. TaTToo's authors further prepend the correct sub-table to schema-interaction steps and manually correct retrievals to make their LLM-judge labels tractable, a context-engineering workaround that itself signals step-correctness as a hard target.

Table 2. Representative failures of existing reward signals. Content-based rewards can miss numeric, semantic, or structural errors in table reasoning.

Metric	Failure case
Cosine Sim [43]	**Q:** *"How long did Cap Anson play?"* **GT:** 26 years **Ans:** 126 years **Score:** 0.749 **Issue:** Same semantic frame ("N years") hides the numeric mismatch [47].
Numerical F1 [76]	**Q:** *"Which player scored the most?"* **GT:** Lionel Messi, 25 **Ans:** Karim Benzema, 25 **Score:** 1.000 **Issue:** Ignores units and entity semantics, giving full credit despite the qualitative mismatch.
TABROUGE [23]	**Q:** *"What is the element before Cu?"* **GT:** Fe **Step:** `sort('atom_num')` **Score:** 0.0167 → 0.0167 **Issue:** LCS rewards query-token presence rather than row order, leaving sorting steps unrewarded.

Table 3 quantifies this gap directly. Cell-relevance reaches 93.0% per-decision agreement and Cohen's $\kappa = 0.68$ pooled across 76,157 cell decisions, while step-correctness hits a structural ceiling. 46.9% of human step labels are marked "unsure," and on the human-confident subset LLM-vs-human agreement is only 73.2% with $\kappa = 0.48$ (n=190). A step-correctness classifier trained on 1,548 LLM-judge step labels reaches held-out F1=0.77 / AUROC=0.88, capped by the noisy supervision signal it inherits. The pooled advantage holds on six of eight benchmarks. On InfoTabs and TabMWP, where step labels reduce to short-form classification or template-aligned arithmetic, step-correctness is easier to label than cell-relevance (-10.6 and -5.9 pp respectively, Table 25), so the claim is that cell-relevance is more labelable on average and on most regimes rather than uniformly. Cell decisions are local and extractive, while step decisions are multi-conditional and evaluative. Prior faithfulness annotation works [22; 7] identify the same asymmetry as the source of finer-unit labeling reliability.

Table 3. Pooled labelability under identical LLM-as-a-judge pipelines. Per-dataset breakdown in Table 25.

Method	Agree (%)	κ
Step-correctness	73.2	0.48
Cell-relevance	**93.0**	**0.68**
Δ	+19.8	+0.20

Takeaway 2 (Labelability of Supervision Targets): Step-correctness hits a labelability ceiling under LLM-as-a-judge whereas cell-relevance admits reliable per-decision labeling, making cell-grounded attention a labelable proxy for step quality on most table reasoning regimes.

4. Methodology

Section 3 motivates TABALIGN with two findings: bidirectional table attention is more human-aligned and permutation-stable than AR attention, and cell relevance is more labelable than step correctness. TABALIGN operationalizes this cell-grounding contract with two components: a bidirectional denoising DLM planner that assigns cell-level targets (Section 4.1), and a TABATTN-based executor

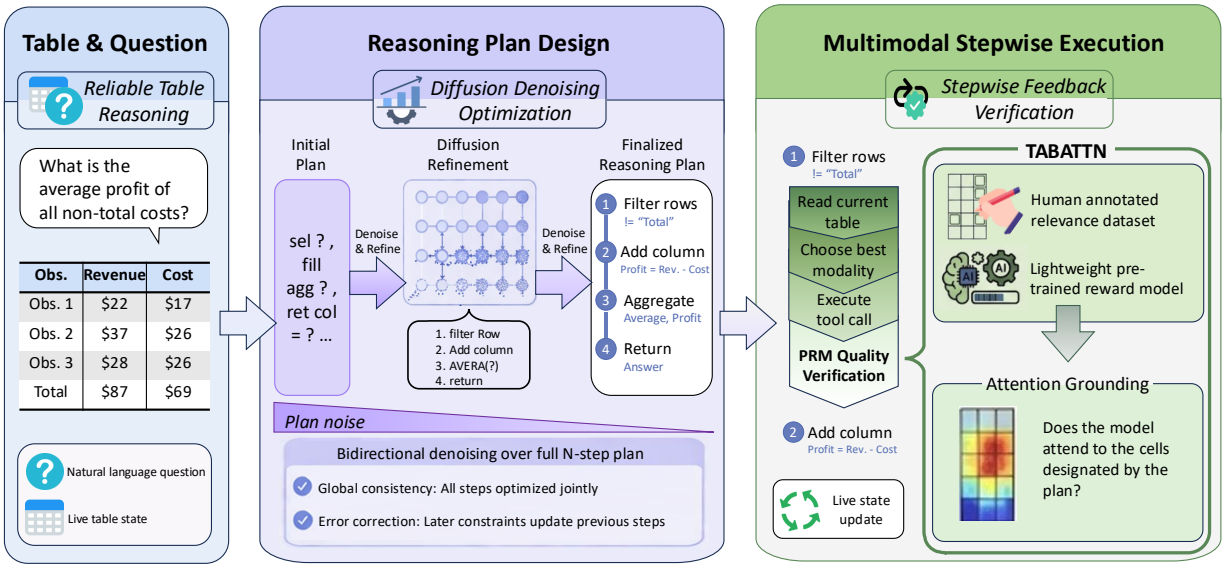


Figure 3. **TABALIGN pairs a masked DLM planner with the TABATTN verifier to enforce the cell-grounding contract.** The DLM denoises a tool-use plan over compact metadata into a step-by-step blueprint, with each step terminating in a `[target: col]` tag that compiles into a cell mask $\mathbf{m}_t$. TABATTN then scores each reasoning step by the overlap between the reasoner's per-cell attention and $\mathbf{m}_t$, supplying in-loop process feedback.

that verifies the reasoner's attention against those targets using a reward fit on cell-relevance labels (Section 4.2).

4.1. Reasoning Plan Design

Table understanding through bidirectional masking. To align with Section 3's conclusion, we adopt bidirectional attention through a masked DLM. Given an input sequence x_0 of table tokens (headers, cell values) and the question, the forward process masks each token independently and symmetrically [11]. This treats all positions identically and reduces order-sensitivity:

$$q_{t|0}(x_t \mid x_0) = \prod_{i=1}^{L} q_{t|0}(x_t^i \mid x_0^i)$$

$$q_{t|0}(x_t^i \mid x_0^i) = \begin{cases} \alpha_t & x_t^i = x_0^i \\ 1 - \alpha_t & x_t^i = [\mathtt{m}] \end{cases}$$

Plan generation via joint denoising. The model denoises a fully masked plan canvas into a natural-language step sequence, with each token attending bidirectionally over the full canvas [11]. Joint denoising lets requirements in later steps constrain earlier ones, enforcing inter-step dependencies that AR generation cannot. Canvas length, block schedule, and padding conventions follow [31; 17], with full details in Section B.2.

$$p_\theta(x_0 \mid x_t) = \prod_{i=1}^{L} p_\theta(x_0^i \mid x_t).$$

Tool-integrated planning. The planning prompt presents the DLM with a semi-closed tool vocabulary alongside the table metadata, requiring each step to conclude with a `[target: col]` tag designating which cells it operates on. These tags are parsed after generation into the binary cell mask $\mathbf{m}_t \in \{0,1\}^C$, later processed to evaluate stepwise attention alignments by TABATTN (Section 4.2). The full tool vocabulary and prompt format are in Section B.2.

4.2. Multimodal Stepwise Execution

Human-verified attention standard curation. The 1,600 human-verified table attention standards from Section 3 double as the training set for the stepwise attention reward. We seed with 100 manual annotations of which cells should be attended to per question, bootstrap with strong LLMs, and verify all generated examples by human annotators. This human-in-the-loop generation pipeline is feasible because cell relevance is a labelable target (Table 3), reaching 93.0% LLM-vs-human per-cell agreement under our identical pipeline protocol, compared to step-correctness pipeline which inherits the upstream labelability ceiling measured in Section 3. Full annotation protocols are in Section B.3.

TABATTN: lightweight learned attention verifier. TABATTN monitors whether each reasoning step attends to the cells designated by the DLM plan. For step t, the planner emits a binary cell mask $\mathbf{m}_t \in \{0,1\}^C$, while the multimodal reasoner provides token-level attention $\alpha_t^{(\tau)}$ from the next-token-emit position over the serialized table. We aggregate to per-cell scores by summing attention mass over the tokens of each cell, $a_t^{(c)} = \sum_{\tau \in \mathcal{T}(c)} \alpha_t^{(\tau)}$, where $\mathcal{T}(c)$ is the contiguous token span of cell c in the serialization (header tokens are pooled into their column, value

tokens into the cell at the corresponding row and column), and we then compute the overlap:

$$R_{\text{attn}}(a_t; \mathbf{m}_t) = \frac{\sum_{c=1}^{C} a_t^{(c)}\, \mathbf{m}_t^{(c)}}{\max\left(\sum_{c=1}^{C} a_t^{(c)},\ \epsilon\right)}, \qquad (1)$$

with $\epsilon = 10^{-6}$ for zero table-attention mass. A density-preserving falsification (Section D.4) confirms that R_{attn} tracks the correct cell set rather than attention sharpness (1.5 to $3.1\times$ over cell-shuffled nulls). Such uninformative steps are excluded from training. If a target tag cannot be parsed (3.6 pp DLM-vs-AR parse-error increase, Section E.4), $\mathbf{m}_t$ defaults to a uniform mask. Finally, TABATTN applies a two-parameter logistic calibration to R_{attn}, fitted by binary cross-entropy against terminal trajectory correctness on the 1,600 standards with an 80/20 early-stopping split, preserving an interpretable calibrated cell-overlap reward. We use text-side cell attention because Qwen3-VL-8B assigns negligible attention to image-pad tokens (Section D.3).

Grounded AR execution. At each step t, the multimodal AR reasoner receives the current table state s_t and plan step p_t, selects an observation modality, and issues a tool call to produce s_{t+1}. Following prior per-step modality routing [57; 24], it chooses between text serialization (cheap, exact strings) and rendered images (layout and merged-cell structure): lookups and string comparisons use text by default, while layout-sensitive operations use images, with escalation to images after text-branch failure. TABATTN scores each transition via Equation (1) and returns the score as in-loop feedback. To cap unproductive iteration, a stagnation halt emits the highest-scoring answer when R_{attn} improves by less than 0.02 for unchanged states. We follow Xing et al. [57] and report Pass@1. Full pseudocode and parse-error/stagnation accounting are in Section E.4.

5. Empirical Evaluations

All experiments follow the same evaluation protocol and baseline set adopted by recently published table reasoning works [57; 78].

Baselines. We compare TABALIGN against five categories of prior work: table-as-text LLMs [48; 72; 62], table-as-image VLMs [74; 75; 66; 4; 3], multimodal frameworks [57], trajectory-optimization agents [78; 15], and training-free GPT-3.5 agents [68; 73; 30; 63; 64]. We further evaluate the contemporary Qwen3-VL-8B [3] as both stronger baseline and reasoner backbone of TABALIGN. Proprietary models appear for context only, as TABATTN requires open attention weights, so the direct comparisons are 8B-class systems (TableDART, TATTOO). Full baseline details are in Section A.

Datasets. We evaluate on eight benchmarks covering two task types: TQA on WTQ [40], MMQA [55], TabMWP [34], TAT-QA [76], HiTab [9], FeTaQA [36], and TFV on TabFact [8] and InfoTabs [16]. We use accuracy for all benchmarks except FeTaQA, which uses BLEU [39] after the LLM free form generation [57]. The headline *Avg* column in Table 4 averages the six accuracy benchmarks, with MMQA and FeTaQA excluded since not all baselines report them. Per-dataset statistics are in Section A.

Implementation. TABALIGN uses LLaDA-8B-Instruct [38] as planner and Qwen3-VL-8B [3] as reasoner. The 1,600 TABATTN attention standards come from train/dev splits and do not overlap with the test sets in Table 4. We report all TABALIGN accuracies in Tables 4, 5 and 7 with a fixed decoding seed: greedy denoising for the planner (T=0) and temperature 0.7 for the reasoner. Pilot AUROC comparisons (Table 1) include paired Wilcoxon p-values over n=319 matched records (Section D.2). Each accuracy entry in Tables 4 and 5 is computed over n=200 evaluation questions per dataset. Per-cell volatility analysis is in Table 9. Hyperparameters, ablation backbones, and computational environment are in Section A.

5.1. Main Results

Headline Accuracy across Eight Benchmarks. TABALIGN reaches 90.62% on the six-benchmark accuracy average, which is 15.76 pp above the strongest dynamic-modality baseline TableDART (on Ovis2, 74.86%) and 16.35 pp above the strongest single-modality baseline Qwen3-VL-8B (74.27%) at comparable 8B-class scale (Table 4). At the per-benchmark level, TABALIGN ranks first on six of the eight columns (WTQ, MMQA, TabMWP, TAT-QA, HiTab, InfoTabs) and remains within 1.21 pp of the best on TabFact. FeTaQA is the only column where the table-specialized TableLlama-7B retains a meaningful lead under BLEU. Consistent with recent work framing modality as the key bottleneck [59], backbone ablations (Table 7) confirm this is not pure backbone scaling.

Generalization across Component-specific Baselines. TABALIGN also outperforms baselines targeting specific parts of the table-reasoning pipeline. Against dynamic modality routing, TABALIGN beats TableDART (Ovis2) on six of seven accuracy benchmarks and stays within 3.48 BLEU on FeTaQA. Against trajectory-optimization agents with auxiliary verifiers or pruners over a frozen reasoner, TABALIGN exceeds TabTrim-8B by 14.72 pp on WTQ and stays within 1.11 pp on TabFact, while outperforming TATTOO by 24.32 pp on WTQ and 31.30 pp on TabFact. These gains suggest that grounding cell-level attention recovers more headroom than auxiliary trajectory supervision under fixed-attention reasoners.

Table 4. Results on eight table reasoning benchmarks. All numbers are accuracy except FeTaQA (BLEU). *Avg* measures accuracy mean of WTQ, TabMWP, TAT-QA, HiTab, TabFact, InfoTabs. Baselines that do not report all six are marked "—". **Bold**/Underline: best/second per column. TABALIGN uses LLaDA-8B-Instruct as the planner and Qwen3-VL-8B as the reasoner.

	TQA						TFV		
Method	WTQ	MMQA	TabMWP	TAT-QA	HiTab	FeTaQA	TabFact	InfoTabs	**Avg**
Table-as-Text Baselines									
Llama-2-7B	16.39	—	22.82	13.73	10.72	10.93	9.20	38.92	18.63
Llama3-Instruct-8B	21.24	—	42.01	13.08	6.97	12.66	73.89	54.00	35.20
TableLlama-7B	24.97	—	10.10	19.04	46.57	**38.38**	79.37	46.57	37.77
Table-as-Image Baselines									
Table-LLaVA-7B	18.43	—	57.78	12.82	10.09	25.60	59.85	65.26	37.37
SynTab-LLaVA-7B	39.59	—	88.30	51.94	35.66	35.45	70.78	69.42	59.28
MiniCPM-V-2.6-8B	47.97	—	83.68	51.55	56.53	32.68	78.48	73.03	65.21
Qwen2.5-VL-7B	54.37	—	63.69	51.94	62.69	10.99	75.81	70.13	63.11
Qwen3-VL-8B	81.82	—	79.90	59.80	65.83	13.89	82.41	75.88	74.27
Joint Multimodal Baselines									
HIPPO-8B	55.77	—	87.50	60.75	63.00	33.18	82.27	75.74	70.84
Gemini 2.0 Flash	63.56	—	46.29	35.62	60.41	10.57	81.33	54.31	56.92
Training-free GPT-3.5 Agents									
DATER	65.9	—	—	—	—	30.92	85.60	—	—
ReAcTable	68.0	—	—	—	—	30.43	86.10	—	—
Mix-SC	73.7	—	—	—	—	—	88.50	—	—
TIDE	75.0	—	—	—	—	—	89.82	—	—
CIT-DP	76.4	—	—	—	—	36.34	**91.30**	—	—
Dynamic Modality Routing									
TableDART (on Q2.5VL)	69.29	—	72.61	59.07	71.13	29.87	77.94	71.46	70.25
TableDART (on Ovis2)	70.58	—	84.54	62.05	74.37	36.11	81.37	76.22	74.86
Trajectory-Optimization Agents									
TATTOO	69.80	25.10	—	—	—	—	58.79	—	—
TabTrim-4B	76.80	—	—	—	—	—	89.40	—	—
TabTrim-8B	79.40	—	—	—	—	—	91.20	—	—
TABALIGN (Ours)	**94.12**	**71.07**	**95.13**	**91.41**	**82.48**	32.63	90.09	**90.45**	**90.62**

5.2. In-Depth Analyses on TABALIGN

This section reports five analyses: planner × reasoner backbone (Table 7), reasoning configuration with DLM error taxonomy (Table 5(a), Table 6), reward signal (Table 5(b)), inference latency (Table 5(a)), and plan quality (Table 8). Permutation stability is established earlier in Section 3.

Backbone Ablation: Disentangling Planner and Reasoner Gains. To rule out backbone scaling as the driver of TABALIGN's gain, we ablate planner choice (AR vs DLM) crossed with reasoner backbone (Q2.5-VL vs Q3-VL) on the six-benchmark average (Table 7). Holding the reasoner fixed, the DLM planner contributes +2.07 to +2.87 pp across both backbones, with the strongest configuration (DLM + Q3-VL) reaching 90.62%. This consistency across reasoners rules out backbone scaling alone as the source of TABALIGN's gain over the AR baseline.

Reasoning Ablation and Error-Mode Taxonomy. Table 5(a) compares end-to-end DLM/LLM reasoning with their planned variants under a shared AR reasoner. End-to-end DLM reasoning collapses, mainly due to duplicate actions (56%) and format errors (51%) (Table 6). Using the DLM only for planning improves accuracy by 4.69 pp over AR planning, suggesting current DLMs are stronger at table-structure understanding than tool execution. This likely reflects instruction-tuning gaps in existing DLMs rather than diffusion-specific limits [61; 6]. The main remaining failure is *plan bypass*: in 67% of error episodes, the AR reasoner emits `f_final_answer` at step 1 and ignores the DLM plan (Table 23). Plan-following episodes have lower schema hallucination (Table 8) and shorter trajectories (Table 5(a)), motivating future plan-adherence reward shaping (Section E.4).

Inference Latency and Accuracy–Cost Tradeoff. The latency rows of Table 5(a) split wall-clock time into understanding and reasoning phases. Consistent with prior planned-reasoning work [37; 42; 10], planning improves accuracy at added inference cost. DLM planning is slower than AR planning because multi-step denoising is expensive, especially on long, multi-column tables such as WTQ, MMQA, and TabFact. However, better DLM plans reduce downstream execution time from 102.6s to 56.8s on average, a 44.64% drop. This makes DLM acceleration, e.g., batched denoising [46; 52] or flow-matching distillation [45], a promising future direction.

Reward-Signal Ablation: Cell-Grounded vs Content-Based Supervision. Section 3 motivated cell-grounded

Table 5. Component ablations on six benchmarks. **(a) Backbone ablation:** accuracy (%) with avg. reasoning turns in parentheses, and latency (s/q) shown in small as understand + reason = total. **(b) Reward-signal ablation:** accuracy (%) with the AR planner (Qwen3-VL-8B) and Qwen3-VL-8B executor fixed, varying only the reward signal. TABATTN outperforms all content-based verifiers, with the largest gains on numerically and structurally dense tasks. The bottom block swaps the cell mask for the human Oracle and a 20% noised Oracle (Section D.5) as upper bounds.

Configuration	WTQ	MMQA	InfoT	TFact	TAT	HiTab	Avg.
(a) Backbone ablation							
DLM Reasoning	30.61 (—)	33.33 (—+17.7)	40.28 (—+22.8)	41.94 (—+23.3)	46.91 (—+21.1)	33.33 (—+21.0)	37.73 (—+21.2†)
LLM Reasoning	81.54 (—+54.7)	60.10 (—+36.6)	79.78 (—+21.5)	87.05 (—+21.6)	79.80 (—+49.9)	73.87 (—+58.6)	77.02 (—+40.5)
LLM Planned Reasoning	90.32 (2.13) 3.5+124.7=128.2	61.34 (1.60) 8.5+142.6=151.1	83.41 (1.50) 3.4+70.3=73.7	**92.39** (2.86) 3.5+58.6=62.1	85.50 (2.08) 13.4+127.6=141.0	78.50 (2.38) 3.6+92.0=95.6	81.91 (2.09) 6.0+102.6=108.6
DLM Planned Reasoning	**94.12** (1.89) 85.6+71.8=157.4	**71.07** (1.46) 95.8+85.8=181.6	**90.45** (1.49) 41.4+24.2=65.6	90.09 (1.99) 86.4+64.2=150.6	**91.41** (1.58) 78.6+47.0=125.6	**82.48** (1.38) 80.2+47.5=127.7	**86.60** (1.63) 78.0+56.8=134.8
(b) Reward-signal ablation							
No Reward (Baseline)	81.8	54.3	75.9	82.4	59.8	65.8	70.0
Cosine Sim [43]	77.8	56.6	78.2	83.8	81.5	67.0	74.2
Numerical F1 [76]	76.5	56.3	78.4	84.3	79.4	66.5	73.6
Cosine + Numerical	75.9	60.1	77.3	85.3	79.0	65.5	73.9
TABROUGE [23]	81.2	57.7	80.2	**92.4**	85.9	65.5	77.1
TABATTN **(ours)**	**90.3**	**61.3**	**83.4**	**92.4**	**85.5**	**78.5**	**81.9**
TABATTN + 20% Noise	91.2	76.8	85.0	93.3	87.9	83.3	86.3
TABATTN + Oracle	93.0	78.6	88.3	95.0	87.9	86.7	88.3

Table 6. Failure mode taxonomy for LLaDA-8B end-to-end reasoning.

Failure mode	Description	% wrong	Root cause
Duplicate action generation	Same tool call emitted 2–4× per step	56%	Fixed canvas force denoising on unused positions
Output format and parsing error	Wrong label set or argument syntax	51%	Weaker instruction alignment vs. AR
Canvas overflow	≥5 actions packed into one step	17%	Fixed length forces action list compression
Empty answer	`f_final_answer` emits empty string	9%	Tool-call chain fails with no recovery path
Over-enumeration	Full column returned instead of single value	7%	No intermediate filter/select step executed

Table 7. Backbone ablation: planner (AR Qwen3-VL-8B (Q3-VL) versus DLM (LLaDA-8B) × reasoner (Qwen2.5-VL-7B vs Q3-VL) combinations on the six-benchmark accuracy average.

Planner / Reasoner	Q2.5-VL	Q3-VL	Δ
AR (Q3-VL)	78.22	87.75	+9.53
DLM (LLaDA-8B)	80.29	**90.62**	+10.33
Δ	+2.07	+2.87	—

Table 8. Plan quality: AR vs. DLM column hallucination rates, measured per plan and per step.

Dataset	Avg. halluc. ↓		% steps w/ halluc. ↓	
	AR	DLM	AR	DLM
WTQ	64.5	**41.5**	74.0	**50.0**
MMQA	82.0	**40.5**	87.4	**50.3**
TabFact	86.9	**46.8**	94.9	**57.5**
HiTab	79.2	**44.1**	91.5	**48.1**
TabMWP	59.4	**23.8**	67.4	**27.6**
TAT-QA	**78.2**	78.7	87.1	**83.5**
Avg.	75.0	**45.9**	83.7	**52.8**

attention as a stronger supervision target than content-similarity rewards. We confirm this empirically in Table 5(b) by holding the AR planner and executor as Qwen3-VL-8B, varying only the reward signal. TABATTN reaches 81.9% average accuracy, 4.8 pp above TABROUGE (77.1%). Per-dataset, the gap widens on numerically and structurally dense tasks, with 9.1 pp and 13 pp over TABROUGE on WTQ and HiTab, where attention-grounded supervision distinguishes correct cell selection from spurious content overlap. Additional human-verified and 20% noised oracles in Table 5(b) indicates that TABATTN is within 6.4 pp of the ceiling (full setup in Section D.5).

Plan Quality Rubric: Column Hallucination Rate. To test whether bidirectional attention improves table understanding, we compare AR (Qwen3-VL-8B) and DLM (LLaDA-8B-Instruct) planners by column hallucination rate. Table 8 shows that across six datasets, the DLM reduces average hallucination from 75.0% to 45.9%, and hallucinatory steps from 83.7% to 52.8%. Gains are largest on structurally rich datasets (MMQA, TabFact, HiTab), while shallow-schema TAT-QA shows similar rates. This aligns with Section 3 as bidirectional attention improves adherence on non-trivial schemas, supporting DLM as planner.

6. Conclusion

We presented TABALIGN, a table reasoning framework that enforces cell grounding through a masked diffusion planner and TABATTN stepwise verifier. TABALIGN outperforms strongest 8B-class baseline by 15.76 pp on the accuracy average, with diagnostics attributing gains to attention-grounded understanding (Section 5.2). Our future work includes extending to complex tabular settings, and DLM acceleration in table reasoning (Section F).

Limitations. Two limitations still persist. 67% of TABALIGN error trajectories bypass the plan by answering at step 1 (Table 23), mainly on single-cell lookups, suggesting plan-adherence reward shaping. On FeTaQA, TABALIGN

trails TableLlama-7B in BLEU largely due to BLEU penalizing correct paraphrases. Our LLM-judge analysis finds 78.6% factual correctness, indicating the BLEU gap overstates the capability gap (Section A.1).

References

[1] Arriola, M., Venkat, N., Granskog, J., and Germanidis, A. Adapting autoregressive vision language models for parallel diffusion decoding, 2025. URL https://runwayml.com/research/autoregressive-to-diffusion-vlms.

[2] Austin, J., Johnson, D. D., Ho, J., Tarlow, D., and van den Berg, R. Structured denoising diffusion models in discrete state-spaces. In Beygelzimer, A., Dauphin, Y., Liang, P., and Vaughan, J. W. (eds.), *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=h7-XixPCAL.

[3] Bai, S., Cai, Y., Chen, R., Chen, K., Chen, X., Cheng, Z., Deng, L., Ding, W., Gao, C., Ge, C., Ge, W., Guo, Z., Huang, Q., Huang, J., Huang, F., Hui, B., Jiang, S., Li, Z., Li, M., Li, M., Li, K., Lin, Z., Lin, J., Liu, X., Liu, J., Liu, C., Liu, Y., Liu, D., Liu, S., Lu, D., Luo, R., Lv, C., Men, R., Meng, L., Ren, X., Ren, X., Song, S., Sun, Y., Tang, J., Tu, J., Wan, J., Wang, P., Wang, P., Wang, Q., Wang, Y., Xie, T., Xu, Y., Xu, H., Xu, J., Yang, Z., Yang, M., Yang, J., Yang, A., Yu, B., Zhang, F., Zhang, H., Zhang, X., Zheng, B., Zhong, H., Zhou, J., Zhou, F., Zhou, J., Zhu, Y., and Zhu, K. Qwen3-vl technical report, 2025. URL https://arxiv.org/abs/2511.21631.

[4] Bai, S., Chen, K., Liu, X., Wang, J., Ge, W., Song, S., Dang, K., Wang, P., Wang, S., Tang, J., Zhong, H., Zhu, Y., Yang, M., Li, Z., Wan, J., Wang, P., Ding, W., Fu, Z., Xu, Y., Ye, J., Zhang, X., Xie, T., Cheng, Z., Zhang, H., Yang, Z., Xu, H., and Lin, J. Qwen2.5-vl technical report. *ArXiv*, abs/2502.13923, 2025. URL https://api.semanticscholar.org/CorpusID:276449796.

[5] Berrayana, L., Heakl, A., Sohail, M. A., Hofmann, T., Khan, S., and Chen, W. Planner and executor: Collaboration between discrete diffusion and auto-regressive models in reasoning, 2026. URL https://openreview.net/forum?id=AhggxqDQCb.

[6] Bie, T., Cao, M., Chen, K., Du, L., Gong, M., Gong, Z., Gu, Y., Hu, J., Huang, Z., Lan, Z., Li, C., Li, C., Li, J., Li, Z., Liu, H., Liu, L., Lu, G., Lu, X., Ma, Y., Tan, J., Wei, L., Wen, J.-R., Xing, Y., Zhang, X., Zhao, J., Zheng, D., Zhou, J., Zhou, J., Zhou, Z., Zhu, L., and Zhuang, Y. Llada2.0: Scaling up diffusion language models to 100b, 2025. URL https://arxiv.org/abs/2512.15745.

[7] Bologna, F., Pan, T., Wilkens, M., Guo, Y., and Wang, L. L. Cqa-eval: Designing reliable evaluations of multi-paragraph clinical qa under resource constraints, 2026. URL https://arxiv.org/abs/2510.10415.

[8] Chen, W., Wang, H., Chen, J., Zhang, Y., Wang, H., Li, S., Zhou, X., and Wang, W. Y. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rkeJRhNYDH.

[9] Cheng, Z., Dong, H., Wang, Z., Jia, R., Guo, J., Gao, Y., Han, S., Lou, J.-G., and Zhang, D. HiTab: A hierarchical table dataset for question answering and natural language generation. In Muresan, S., Nakov, P., and Villavicencio, A. (eds.), *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1094–1110, Dublin, Ireland, May 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.78. URL https://aclanthology.org/2022.acl-long.78/.

[10] Deng, Z., Deng, M., Liang, C., Gao, Z., Ma, C., Lin, C., Zhang, H., Mei, S., Shen, S., and Wang, C. Planu: Large language model reasoning through planning under uncertainty. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=r3LYE0Ct3G.

[11] Feng, G., Geng, Y., Guan, J., Wu, W., Wang, L., and He, D. Theoretical benefit and limitation of diffusion language model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=fGBCRZQVse.

[12] Gong, S., Agarwal, S., Zhang, Y., Ye, J., Zheng, L., Li, M., An, C., Zhao, P., Bi, W., Han, J., Peng, H., and Kong, L. Scaling diffusion language models via adaptation from autoregressive models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=j1tSLYKwg8.

[13] Google. Gemini 3 flash. https://blog.google/products-and-platforms/products/gemini/gemini-3-flash/, 2025. Accessed: 2026-04-22.

[14] Grattafiori, A., Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., Mathur, A., Schelten, A., Vaughan, A., Yang, A., Fan, A., Goyal, A., Hartshorn, A., Yang, A., Mitra, A., Sravankumar, A., Korenev, A., Hinsvark, A., Rao, A., Zhang, A., Rodriguez, A., Gregerson, A., Spataru, A., Roziere, B., Biron, B., Tang, B., Chern, B., Caucheteux, C., Nayak, C., Bi, C., Marra, C., McConnell, C., Keller, C., Touret, C., Wu, C., Wong, C., Ferrer, C. C., Nikolaidis, C., Allonsius, D., Song, D., Pintz, D., Livshits, D., Wyatt, D., Esiobu, D., Choudhary, D., Mahajan, D., Garcia-Olano, D., Perino, D., Hupkes, D., Lakomkin, E., AlBadawy, E., Lobanova, E., Dinan, E., Smith, E. M., Radenovic, F., Guzmán, F., Zhang, F., Synnaeve, G., Lee, G., Anderson, G. L., Thattai, G., Nail, G., Mialon, G., Pang, G., Cucurell, G., Nguyen, H., Korevaar, H., Xu, H., Touvron, H., Zarov, I., Ibarra, I. A., Kloumann, I., Misra, I., Evtimov, I., Zhang, J., Copet, J., Lee, J., Geffert, J., Vranes, J., Park, J., Mahadeokar, J., Shah, J., van der Linde, J., Billock, J., Hong, J., Lee, J., Fu, J., Chi, J., Huang, J., Liu, J., Wang, J., Yu, J., Bitton, J., Spisak, J., Park, J., Rocca, J., Johnstun, J., Saxe, J., Jia, J., Alwala, K. V., Prasad, K., Upasani, K., Plawiak, K., Li, K., Heafield, K., Stone, K., El-Arini, K., Iyer, K., Malik, K., Chiu, K., Bhalla, K., Lakhotia, K., Rantala-Yeary, L., van der Maaten, L., Chen, L., Tan, L., Jenkins, L., Martin, L., Madaan, L., Malo, L., Blecher, L., Landzaat, L., de Oliveira, L., Muzzi, M., Pasupuleti, M., Singh, M., Paluri, M., Kardas, M., Tsimpoukelli, M., Oldham, M., Rita, M., Pavlova, M., Kambadur, M., Lewis, M., Si, M., Singh, M. K., Hassan, M., Goyal, N., Torabi, N., Bashlykov, N., Bogoychev, N., Chatterji, N., Zhang, N., Duchenne, O., Çelebi, O., Alrassy, P., Zhang, P., Li, P., Vasic, P., Weng, P., Bhargava, P., Dubal, P., Krishnan, P., Koura, P. S., Xu, P., He, Q., Dong, Q., Srinivasan, R., Ganapathy, R., Calderer, R., Cabral, R. S., Stojnic, R., Raileanu, R., Maheswari, R., Girdhar, R., Patel, R., Sauvestre, R., Polidoro, R., Sumbaly, R., Taylor, R., Silva, R., Hou, R., Wang, R., Hosseini, S., Chennabasappa, S., Singh, S., Bell, S., Kim, S. S., Edunov, S., Nie, S., Narang, S., Raparthy, S., Shen, S., Wan, S., Bhosale, S., Zhang, S., Vandenhende, S., Batra, S., Whitman, S., Sootla, S., Collot, S., Gururangan, S., Borodinsky, S., Herman, T., Fowler, T., Sheasha, T., Georgiou, T., Scialom, T., Speckbacher, T., Mihaylov, T., Xiao, T., Karn, U., Goswami, V., Gupta, V., Ramanathan, V., Kerkez, V., Gonguet, V., Do, V., Vogeti, V., Albiero, V., Petrovic, V., Chu, W., Xiong, W., Fu, W., Meers, W., Martinet, X., Wang, X., Wang, X., Tan, X. E., Xia, X., Xie, X., Jia, X., Wang, X., Goldschlag, Y., Gaur, Y., Babaei, Y., Wen, Y., Song, Y., Zhang, Y., Li, Y., Mao, Y., Coudert, Z. D., Yan, Z., Chen, Z., Papakipos, Z., Singh, A., Srivastava, A., Jain, A., Kelsey, A., Shajnfeld, A., Gangidi, A., Victoria, A., Goldstand, A., Menon, A., Sharma, A., Boesenberg, A., Baevski, A., Feinstein, A., Kallet, A., Sangani, A., Teo, A., Yunus, A., Lupu, A., Alvarado, A., Caples, A., Gu, A., Ho, A., Poulton, A., Ryan, A., Ramchandani, A., Dong, A., Franco, A., Goyal, A., Saraf, A., Chowdhury, A., Gabriel, A., Bharambe, A., Eisenman, A., Yazdan, A., James, B., Maurer, B., Leonhardi, B., Huang, B., Loyd, B., Paola, B. D., Paranjape, B., Liu, B., Wu, B., Ni, B., Hancock, B., Wasti, B., Spence, B., Stojkovic, B., Gamido, B., Montalvo, B., Parker, C., Burton, C., Mejia, C., Liu, C., Wang, C., Kim, C., Zhou, C., Hu, C., Chu, C.-H., Cai, C., Tindal, C., Feichtenhofer, C., Gao, C., Civin, D., Beaty, D., Kreymer, D., Li, D., Adkins, D., Xu, D., Testuggine, D., David, D., Parikh, D., Liskovich, D., Foss, D., Wang, D., Le, D., Holland, D., Dowling, E., Jamil, E., Montgomery, E., Presani, E., Hahn, E., Wood, E., Le, E.-T., Brinkman, E., Arcaute, E., Dunbar, E., Smothers, E., Sun, F., Kreuk, F., Tian, F., Kokkinos, F., Ozgenel, F., Caggioni, F., Kanayet, F., Seide, F., Florez, G. M., Schwarz, G., Badeer, G., Swee, G., Halpern, G., Herman, G., Sizov, G., Guangyi, Zhang, Lakshminarayanan, G., Inan, H., Shojanazeri, H., Zou, H., Wang, H., Zha, H., Habeeb, H., Rudolph, H., Suk, H., Aspegren, H., Goldman, H., Zhan, H., Damlaj, I., Molybog, I., Tufanov, I., Leontiadis, I., Veliche, I.-E., Gat, I., Weissman, J., Geboski, J., Kohli, J., Lam, J., Asher, J., Gaya, J.-B., Marcus, J., Tang, J., Chan, J., Zhen, J., Reizenstein, J., Teboul, J., Zhong, J., Jin, J., Yang, J., Cummings, J., Carvill, J., Shepard, J., McPhie, J., Torres, J., Ginsburg, J., Wang, J., Wu, K., U, K. H., Saxena, K., Khandelwal, K., Zand, K., Matosich, K., Veeraraghavan, K., Michelena, K., Li, K., Jagadeesh, K., Huang, K., Chawla, K., Huang, K., Chen, L., Garg, L., A, L., Silva, L., Bell, L., Zhang, L., Guo, L., Yu, L., Moshkovich, L., Wehrstedt, L., Khabsa, M., Avalani, M., Bhatt, M., Mankus, M., Hasson, M., Lennie, M., Reso, M., Groshev, M., Naumov, M., Lathi, M., Keneally, M., Liu, M., Seltzer, M. L., Valko, M., Restrepo, M., Patel, M., Vyatskov, M., Samvelyan, M., Clark, M., Macey, M., Wang, M., Hermoso, M. J., Metanat, M., Rastegari, M., Bansal, M., Santhanam, N., Parks, N., White, N., Bawa, N., Singhal, N., Egebo, N., Usunier, N., Mehta, N., Laptev, N. P., Dong, N., Cheng, N., Chernoguz, O., Hart, O., Salpekar, O., Kalinli, O., Kent, P., Parekh, P., Saab, P., Balaji, P., Rittner, P., Bontrager, P., Roux, P., Dollar, P., Zvyagina, P., Ratanchandani, P., Yuvraj, P., Liang, Q., Alao, R., Rodriguez, R., Ayub, R., Murthy, R., Nayani, R., Mitra, R., Parthasarathy, R., Li, R., Hogan, R., Battey, R., Wang, R., Howes, R., Rinott, R., Mehta, S., Siby, S., Bondu, S. J., Datta, S., Chugh, S., Hunt, S., Dhillon, S., Sidorov, S., Pan, S., Mahajan, S., Verma, S., Yamamoto, S., Ramaswamy, S.,

Lindsay, S., Lindsay, S., Feng, S., Lin, S., Zha, S. C., Patil, S., Shankar, S., Zhang, S., Zhang, S., Wang, S., Agarwal, S., Sajuyigbe, S., Chintala, S., Max, S., Chen, S., Kehoe, S., Satterfield, S., Govindaprasad, S., Gupta, S., Deng, S., Cho, S., Virk, S., Subramanian, S., Choudhury, S., Goldman, S., Remez, T., Glaser, T., Best, T., Koehler, T., Robinson, T., Li, T., Zhang, T., Matthews, T., Chou, T., Shaked, T., Vontimitta, V., Ajayi, V., Montanez, V., Mohan, V., Kumar, V. S., Mangla, V., Ionescu, V., Poenaru, V., Mihailescu, V. T., Ivanov, V., Li, W., Wang, W., Jiang, W., Bouaziz, W., Constable, W., Tang, X., Wu, X., Wang, X., Wu, X., Gao, X., Kleinman, Y., Chen, Y., Hu, Y., Jia, Y., Qi, Y., Li, Y., Zhang, Y., Zhang, Y., Adi, Y., Nam, Y., Yu, Wang, Zhao, Y., Hao, Y., Qian, Y., Li, Y., He, Y., Rait, Z., DeVito, Z., Rosnbrick, Z., Wen, Z., Yang, Z., Zhao, Z., and Ma, Z. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[15] Guo, Y., Ye, S., Chen, S., Wen, Z., Zhang, T., Bai, Q., Jin, D., Hou, Y., He, H., Yang, J., and Tan, X. Rethinking table pruning in tableqa: From sequential revisions to gold trajectory-supervised parallel search, 2026. URL https://arxiv.org/abs/2601.03851.

[16] Gupta, V., Mehta, M., Nokhiz, P., and Srikumar, V. INFOTABS: Inference on tables as semi-structured data. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2309–2324, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.210. URL https://aclanthology.org/2020.acl-main.210/.

[17] He, G., Nie, S., Zhu, F., Zhao, Y., Bai, T., Yan, R., Fu, J., Li, C., and Yuan, B. UltraLLaDA: Scaling the context length to 128k for diffusion large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=68DGlhlvD9.

[18] Herzig, J., Nowak, P. K., Müller, T., Piccinno, F., and Eisenschlos, J. TaPas: Weakly supervised table parsing via pre-training. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4320–4333, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.398. URL https://aclanthology.org/2020.acl-main.398/.

[19] Ji, D., Zhu, L., Gao, S., Xu, P., Lu, H., Ye, J., and Zhao, F. Tree-of-table: Unleashing the power of LLMs for enhanced large-scale table understanding, 2025. URL https://openreview.net/forum?id=Yv8FrCY87H.

[20] Kim, J., Kit, L. C., Domingo-Enrich, C., Du, Y., Kakade, S. M., Ngotiaoco, T., Chen, S., and Albergo, M. S. Any-order flexible length masked diffusion. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ttuNnMRI6H.

[21] Kotelnikov, A., Baranchuk, D., Rubachev, I., and Babenko, A. Tabddpm: modelling tabular data with diffusion models. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org, 2023. URL https://dl.acm.org/doi/10.5555/3618408.3619133.

[22] Krishna, K., Bransom, E., Kuehl, B., Iyyer, M., Dasigi, P., Cohan, A., and Lo, K. LongEval: Guidelines for human evaluation of faithfulness in long-form summarization. In Vlachos, A. and Augenstein, I. (eds.), *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 1650–1669, Dubrovnik, Croatia, May 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.eacl-main.121. URL https://aclanthology.org/2023.eacl-main.121/.

[23] Kwok, T. S. T., Wang, X., He, H., Lin, X., Lu, P., Ma, L., Wang, C., Wu, Y. N., Ding, L., and Cheng, G. Enhancing tableqa through verifiable reasoning trace reward, 2026. URL https://arxiv.org/abs/2601.22530.

[24] Kwok, T. S. T., Wang, X., Lin, X., Lu, P., Wang, C., Li, C., Wu, H., Tang, N., Kreiss, E., and Cheng, G. Tabqaworld: Optimizing multimodal reasoning for multi-turn table question answering, 2026. URL https://arxiv.org/abs/2604.03393.

[25] Li, J., Dong, X., Zang, Y., Cao, Y., Wang, J., and Lin, D. Beyond fixed: Training-free variable-length denoising for diffusion large language models. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Ic2A2gCseC.

[26] Li, X. L., Thickstun, J., Gulrajani, I., Liang, P., and Hashimoto, T. Diffusion-LM improves controllable text generation. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho, K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=3s9IrEsjLyk.

[27] Li, Y., Dong, Z., Sun, Y., Wang, W., Xiong, S., Luo, Y., Liu, J., Lu, H., Wang, J., Su, W., Zheng, B., and Yan, J. Attention illuminates llm reasoning: The preplan-and-anchor rhythm enables fine-grained policy optimization, 2025. URL https://arxiv.org/abs/2510.13554.

[28] Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

[29] Lin, X., Xu, C., Yang, M., and Cheng, G. CTSyn: A foundation model for cross tabular data generation. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Sh4FOyZRpv.

[30] Liu, T., Wang, F., and Chen, M. Rethinking tabular data understanding with large language models. In Duh, K., Gomez, H., and Bethard, S. (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 450–482, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.26. URL https://aclanthology.org/2024.naacl-long.26/.

[31] Liu, X., Song, Y., Liu, Z., Huang, Z., Guo, Q., He, Z., and Qiu, X. Longllada: Unlocking long context capabilities in diffusion llms. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. URL https://doi.org/10.1609/aaai.v40i38.40491.

[32] Lou, A., Meng, C., and Ermon, S. Discrete diffusion modeling by estimating the ratios of the data distribution. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24, 2024. URL https://dl.acm.org/doi/10.5555/3692070.3693403.

[33] Lu, J., Dou, Z., WANG, H., Cao, Z., Dai, J., Feng, Y., and Guo, Z. AutoPSV: Automated process-supervised verifier. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=eOAPWWOGs9.

[34] Lu, P., Qiu, L., Chang, K.-W., Wu, Y. N., Zhu, S.-C., Rajpurohit, T., Clark, P., and Kalyan, A. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=DHyHRBwJUTN.

[35] Lu, S., Li, Y., Xia, Y., Hu, Y., Zhao, S., Ma, Y., Wei, Z., Li, Y., Duan, L., Zhao, J., Han, Y., Li, H., Chen, W., Tang, J., Hou, C., Du, Z., Zhou, T., Zhang, W., Ding, H., Li, J., Li, W., Hu, G., Gu, Y., Yang, S., Wang, J., Sun, H., Wang, Y., Sun, H., Huang, J., He, Y., Shi, S., Zhang, W., Zheng, G., Jiang, J., Gao, S., Wu, Y.-F., Chen, S., Chen, Y., Chen, Q.-G., Xu, Z., Luo, W., and Zhang, K. Ovis2.5 technical report, 2025. URL https://arxiv.org/abs/2508.11737.

[36] Nan, L., Hsieh, C., Mao, Z., Lin, X. V., Verma, N., Zhang, R., Kryściński, W., Schoelkopf, H., Kong, R., Tang, X., Mutuma, M., Rosand, B., Trindade, I., Bandaru, R., Cunningham, J., Xiong, C., and Radev, D. FeTaQA: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49, 2022. doi: 10.1162/tacl_a_00446. URL https://aclanthology.org/2022.tacl-1.3/.

[37] Nguyen, G., Brugere, I., Sharma, S., Kariyappa, S., Nguyen, A. T., and Lecue, F. Interpretable LLM-based table question answering. *Transactions on Machine Learning Research*, 2025. ISSN 2835-8856. URL https://openreview.net/forum?id=2eTsZBoU2W.

[38] Nie, S., Zhu, F., You, Z., Zhang, X., Ou, J., Hu, J., Zhou, J., Lin, Y., Wen, J.-R., and Li, C. Large language diffusion models, 2025. URL https://arxiv.org/abs/2502.09992.

[39] Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pp. 311–318, USA, 2002. Association for Computational Linguistics. doi: 10.3115/1073083.1073135. URL https://doi.org/10.3115/1073083.1073135.

[40] Pasupat, P. and Liang, P. Compositional semantic parsing on semi-structured tables. In Zong, C. and Strube, M. (eds.), *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1470–1480, Beijing, China, July 2015. Association for Computational Linguistics. doi: 10.3115/v1/P15-1142. URL https://aclanthology.org/P15-1142/.

[41] Qiu, Z., Qiao, S., Xu, K., Zhu, Y., Du, L., Zhang, N., and Chen, H. Rewarding the scientific process: Process-level reward modeling for agentic data analysis, 2026. URL https://arxiv.org/abs/2604.24198.

[42] Rawat, M., Gupta, A., Goomer, R., Bari, A. D., Gupta, N., and Pieraccini, R. Pre-act: Multi-step planning and reasoning improves acting in llm agents, 2025. URL https://arxiv.org/abs/2505.09970.

[43] Reimers, N. and Gurevych, I. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In Inui, K., Jiang, J., Ng, V., and Wan, X. (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 3982–3992, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1410. URL https://aclanthology.org/D19-1410/.

[44] Sahoo, S. S., Arriola, M., Gokaslan, A., Marroquin, E. M., Rush, A. M., Schiff, Y., Chiu, J. T., and Kuleshov, V. Simple and effective masked diffusion language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=L4uaAR4ArM.

[45] Sahoo, S. S., Deschenaux, J., Gokaslan, A., Wang, G., Chiu, J. T., and Kuleshov, V. The diffusion duality. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=9P9Y8FOSOk.

[46] Song, Y., Zhang, Z., Luo, C., Gao, P., Xia, F., Luo, H., Li, Z., Yang, Y., Yu, H., Qu, X., Fu, Y., Su, J., Zhang, G., Huang, W., Wang, M., Yan, L., Jia, X., Liu, J., Ma, W.-Y., Zhang, Y.-Q., Wu, Y., and Zhou, H. Seed diffusion: A large-scale diffusion language model with high-speed inference, 2025. URL https://arxiv.org/abs/2508.02193.

[47] Steck, H., Ekanadham, C., and Kallus, N. Is cosine-similarity of embeddings really about similarity? In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, pp. 887–890, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400701726. doi: 10.1145/3589335.3651526. URL https://doi.org/10.1145/3589335.3651526.

[48] Touvron, H., Martin, L., Stone, K., Albert, P., Almahairi, A., Babaei, Y., Bashlykov, N., Batra, S., Bhargava, P., Bhosale, S., Bikel, D., Blecher, L., Ferrer, C. C., Chen, M., Cucurull, G., Esiobu, D., Fernandes, J., Fu, J., Fu, W., Fuller, B., Gao, C., Goswami, V., Goyal, N., Hartshorn, A., Hosseini, S., Hou, R., Inan, H., Kardas, M., Kerkez, V., Khabsa, M., Kloumann, I., Korenev, A., Koura, P. S., Lachaux, M.-A., Lavril, T., Lee, J., Liskovich, D., Lu, Y., Mao, Y., Martinet, X., Mihaylov, T., Mishra, P., Molybog, I., Nie, Y., Poulton, A., Reizenstein, J., Rungta, R., Saladi, K., Schelten, A., Silva, R., Smith, E. M., Subramanian, R., Tan, X. E., Tang, B., Taylor, R., Williams, A., Kuan, J. X., Xu, P., Yan, Z., Zarov, I., Zhang, Y., Fan, A., Kambadur, M., Narang, S., Rodriguez, A., Stojnic, R., Edunov, S., and Scialom, T. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

[49] Uesato, J., Kushman, N., Kumar, R., Song, H. F., Siegel, N. Y., Wang, L., Creswell, A., Irving, G., and Higgins, I. Solving math word problems with process-based and outcome-based feedback, 2023. URL https://openreview.net/forum?id=MND1kmmNy0O.

[50] Wang, H., Liu, Z., Li, X., Yang, X., Gu, Y., Yan, Y., Shi, Q., Li, F., Chen, C., and Yu, G. Hippo: Enhancing the table understanding capability of llms through hybrid-modal preference optimization, 2026. URL https://arxiv.org/abs/2502.17315.

[51] Wang, P., Li, L., Shao, Z., Xu, R., Dai, D., Li, Y., Chen, D., Wu, Y., and Sui, Z. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

[52] Wang, X., Xu, C., Jin, Y., Jin, J., Zhang, H., Yu, K., and Deng, Z. Diffusion LLMs can do faster-than-AR inference via discrete diffusion forcing. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=t5uLZSRjhF.

[53] Wang, Z., Zhang, H., Li, C.-L., Eisenschlos, J. M., Perot, V., Wang, Z., Miculicich, L., Fujii, Y., Shang, J., Lee, C.-Y., and Pfister, T. Chain-of-table: Evolving tables in the reasoning chain for table understanding. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=4L0xnS4GQM.

[54] Warner, B., Chaffin, A., Clavié, B., Weller, O., Hallström, O., Taghadouini, S., Gallagher, A., Biswas,

R., Ladhak, F., Aarsen, T., Adams, G. T., Howard, J., and Poli, I. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 2526–2547, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.127. URL https://aclanthology.org/2025.acl-long.127/.

[55] Wu, J., Yang, L., Li, D., Ji, Y., Okumura, M., and Zhang, Y. MMQA: Evaluating LLMs with multi-table multi-hop complex questions. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=GGlpykXDCa.

[56] Xing, J., He, Y., Zhou, M., Dong, H., Han, S., Chen, L., Zhang, D., Chaudhuri, S., and Jagadish, H. V. 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*, 2025. URL https://openreview.net/forum?id=ryUzgwD6UQ.

[57] Xing, X., Yuan, W., Chen, T., Nguyen, Q. V. H., Zhang, X., and Yin, H. TableDART: Dynamic adaptive multi-modal routing for table understanding. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=4aZTiLH3fm.

[58] Xiong, Z., Cai, Y., Li, Z., and Wang, Y. Unveiling the potential of diffusion large language model in controllable generation. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=qhd0qv6L0k.

[59] Xu, Y., Li, C., Zhou, H., Wan, X., Zhang, C., Korhonen, A., and Vulić, I. Visual planning: Let's think only with images. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=wsnse46kRO.

[60] Yang, A., Li, A., Yang, B., Zhang, B., Hui, B., Zheng, B., Yu, B., Gao, C., Huang, C., Lv, C., Zheng, C., Liu, D., Zhou, F., Huang, F., Hu, F., Ge, H., Wei, H., Lin, H., Tang, J., Yang, J., Tu, J., Zhang, J., Yang, J., Yang, J., Zhou, J., Zhou, J., Lin, J., Dang, K., Bao, K., Yang, K., Yu, L., Deng, L., Li, M., Xue, M., Li, M., Zhang, P., Wang, P., Zhu, Q., Men, R., Gao, R., Liu, S., Luo, S., Li, T., Tang, T., Yin, W., Ren, X., Wang, X., Zhang, X., Ren, X., Fan, Y., Su, Y., Zhang, Y., Zhang, Y., Wan, Y., Liu, Y., Wang, Z., Cui, Z., Zhang, Z., Zhou, Z., and Qiu, Z. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

[61] Yang, J., Jiang, Y., Hu, X., Cheng, S., Qi, B., and Shao, J. Dare: Diffusion large language models alignment and reinforcement executor, 2026. URL https://arxiv.org/abs/2604.04215.

[62] Yang, S., Huang, Q., Yuan, J., Zha, L., Tang, K., Yang, Y., Wang, N., Wei, Y., Li, L., Ye, W., Chen, H., Zhang, T., Zhou, J., Wang, H., Chen, G., and Zhao, J. Tablegpt-r1: Advancing tabular reasoning through reinforcement learning, 2025. URL https://arxiv.org/abs/2512.20312.

[63] Yang, Z., Du, Z., Zhang, M., Du, W., Chen, J., Duan, Z., and Zhao, S. Triples as the key: Structuring makes decomposition and verification easier in LLM-based tableQA. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=UwcZEoNP19.

[64] Yang, Z., Du, Z., Zhang, M., Du, W., Chen, J., Qian, F., and Zhao, S. Causality meets the table: Debiasing LLMs for faithful TableQA via front-door intervention. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://neurips.cc/virtual/2025/loc/san-diego/poster/115020.

[65] Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., and Cao, Y. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.03629.

[66] Yao, Y., Yu, T., Zhang, A., Wang, C., Cui, J., Zhu, H., Cai, T., Chen, C., Li, H., Zhao, W., He, Z., Chen, Q., Zhou, R., Zou, Z., Zhang, H., Hu, S., Zheng, Z., Zhou, J., Cai, J., Han, X., Zeng, G., Li, D., Liu, Z., and Sun, M. Efficient gpt-4v level multimodal large language model for deployment on edge devices. *Nature Communications*, 16(1):5509, 2025. doi: 10.1038/s41467-025-61040-5. URL https://doi.org/10.1038/s41467-025-61040-5.

[67] Ye, J., Xie, Z., Zheng, L., Gao, J., Wu, Z., Jiang, X., Li, Z., and Kong, L. Dream 7b: Diffusion large language models, 2025. URL https://arxiv.org/abs/2508.15487.

[68] Ye, Y., Hui, B., Yang, M., Li, B., Huang, F., and Li, Y. 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, pp. 174–184, New York, NY, USA, 2023. Association for Computing Machinery. ISBN 9781450394086. doi: 10.1145/3539618.3591708. URL https://doi.org/10.1145/3539618.3591708.

[69] Yin, P., Neubig, G., Yih, W.-t., and Riedel, S. TaBERT: Pretraining for joint understanding of textual and tabular data. In Jurafsky, D., Chai, J., Schluter, N., and Tetreault, J. (eds.), *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 8413–8426, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.745. URL https://aclanthology.org/2020.acl-main.745/.

[70] Zhang, H., Fang, L., Wu, Q., and Yu, P. S. TabNAT: A continuous-discrete joint generative framework for tabular data. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=WbfbT2BH6F.

[71] Zhang, T., Yue, X., Li, Y., and Sun, H. Tablellama: Towards open large generalist models for tables, 2024. URL https://arxiv.org/abs/2311.09206.

[72] Zhang, X., Luo, S., Zhang, B., Ma, Z., Zhang, J., Li, Y., Li, G., Yao, Z., Xu, K., Zhou, J., Zhang-Li, D., Yu, J., Zhao, S., Li, J., and Tang, J. TableLLM: Enabling tabular data manipulation by LLMs in real office usage scenarios. In Che, W., Nabende, J., Shutova, E., and Pilehvar, M. T. (eds.), *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10315–10344, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.538. URL https://aclanthology.org/2025.findings-acl.538/.

[73] Zhang, Y., Henkel, J., Floratou, A., Cahoon, J., Deep, S., and Patel, J. M. Reactable: Enhancing react for table question answering. *Proc. VLDB Endow.*, 17 (8):1981–1994, April 2024. ISSN 2150-8097. doi: 10.14778/3659437.3659452. URL https://doi.org/10.14778/3659437.3659452.

[74] Zheng, M., Feng, X., Si, Q., She, Q., Lin, Z., Jiang, W., and Wang, W. Multimodal table understanding. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9102–9124, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.493. URL https://aclanthology.org/2024.acl-long.493/.

[75] Zhou, B., Gao, Z., Wang, Z., Zhang, B., Wang, Y., Chen, Z., and Xie, H. Syntab-llava: Enhancing multimodal table understanding with decoupled synthesis. In *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 24796–24806, 2025. doi: 10.1109/CVPR52734.2025.02309. URL https://cvpr.thecvf.com/virtual/2025/poster/33173.

[76] Zhu, F., Lei, W., Huang, Y., Wang, C., Zhang, S., Lv, J., Feng, F., and Chua, T.-S. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In Zong, C., Xia, F., Li, W., and Navigli, R. (eds.), *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)*, pp. 3277–3287, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.254. URL https://aclanthology.org/2021.acl-long.254/.

[77] Zhu, X., Zhou, X., Zhu, B., Hu, H., Du, M., Zhang, H., Wang, H., and Guo, Z. Codescaler: Scaling code llm training and test-time inference via execution-free reward models, 2026. URL https://arxiv.org/abs/2602.17684.

[78] Zou, J., Roy, S., Verma, V. K., Wang, Z., Wipf, D., Lu, P., Negi, S., Zou, J., and He, J. TaTToo: Tool-grounded thinking PRM for test-time scaling in tabular reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=zc1ezBrr5m.

[79] Åström, K. J. Optimal control of markov processes with incomplete state information. *Journal of Mathematical Analysis and Applications*, 10(1): 174–205, 1965. ISSN 0022-247X. doi: https://doi.org/10.1016/0022-247X(65)90154-X. URL https://www.sciencedirect.com/science/article/pii/0022247X6590154X.

A. Empirical Setup: Datasets, Baselines, and Implementation Details

Dataset statistics. WTQ contains 2,108 test questions over heterogeneous web tables. MMQA adds cross-modal multi-hop questions over tables paired with images and text. TabMWP covers grade-school math word problems with tabular context. TAT-QA pairs tabular and textual evidence for financial reasoning. HiTab targets hierarchical tables requiring multi-level lookups. FeTaQA requires fluent free-form answers graded by BLEU. TabFact and InfoTabs are binary entailment benchmarks over Wikipedia and infobox tables, respectively.

Baseline details. We organize the baselines into six categories that span the dominant table reasoning paradigms in the recent literature. For each category we first define the paradigm, then describe how each individual baseline within the category operates.

Table-as-text LLMs serialize the input table into a textual format (typically Markdown or HTML) and feed it to a text-only LLM that produces the answer in a single forward pass. This is the dominant pre-multimodal deployment pattern for tabular QA. Llama-2-7B [48] is a generalist instruction-tuned chat model with no table-specific adaptation, included as a baseline for the floor of generalist LLM capability at the 7B scale. Llama3-Instruct-8B [14] is the analogous third-generation generalist chat backbone. TableLlama-7B [71] is an LLM further post-trained on large-scale table instruction data, and TableGPT-R1 [62] extends this line by adding reinforcement learning on tabular reasoning trajectories. TableLLM [72] additionally targets real-office spreadsheet manipulation tasks. Together this group quantifies how much table-specific post-training matters when the table is presented purely as text.

Table-as-image VLMs render the table as an image and feed it to a vision language model that perceives the cell grid visually rather than through text serialization. The motivation is to bypass the brittleness of long Markdown serialization on wide tables and to retain visual cues such as merged cells and hierarchical headers [74]. Table-LLaVA-7B [74] and SynTab-LLaVA-7B [75] are LLaVA-style VLMs further trained on real and synthetic table images. MiniCPM-V-2.6-8B [66], Qwen2.5-VL-7B [4], and Qwen3-VL-8B [3] are general-purpose VLMs evaluated on table-as-image inputs without table-specific tuning. We include Qwen3-VL-8B explicitly because it postdates the VLMs used by competing baselines and serves as the strongest contemporary open-source VLM at our parameter scale.

Joint multimodal frameworks process both the textual serialization and the rendered image of the table jointly inside a single model, on the assumption that the two modalities expose complementary information (text preserves exact strings while image preserves layout). HIPPO-8B [50] is a recent open-source multimodal LLM that aligns the two modalities for table understanding through hybrid-modal preference optimization, conditioning the answer on a fused text-and-image representation for every query. Gemini 2.0 Flash [13] is a proprietary multimodal LLM included for research context, but it cannot serve as a primary comparison point for TABALIGN because TABATTN requires access to internal attention weights that proprietary endpoints do not expose.

Dynamic modality-routing frameworks train a lightweight router that, per query, selects either the text branch, the image branch, or a fusion of both, rather than committing to a single modality for every input. TableDART [57] is the representative method in this category. It keeps two frozen single-modality experts (a text expert built on TableGPT2-7B [62] and an image expert built on either Qwen2.5-VL-7B [4] or Ovis2-8B [35]) and trains a 2-layer MLP gating network to pick the cheapest inference path that still answers correctly. We report both published TableDART variants. TableDART is the closest competing baseline to TABALIGN in spirit, since both methods explicitly refuse to commit to a static modality choice.

Trained trajectory-optimization agents train an auxiliary model that supervises or prunes the multi-step reasoning trajectory produced by a base reasoner, with the goal of suppressing wrong intermediate steps before they propagate into the final answer. TATTOO [78] is a tool-grounded process reward model (PRM) that scores each reasoning step of a DeepSeek-R1-Distill-Qwen-14B policy reasoner and drives best-of-N selection at inference, using external tool execution to ground the per-step verification signal. TabTrim [15] is a parallel-search table pruner that trims irrelevant rows and columns from intermediate states under gold-trajectory supervision. These two methods isolate the contribution of trajectory-level supervision, and they form the most direct comparison point for TABATTN, which also operates as an in-the-loop step-level verifier but grounds its scores in attention rather than in an extra reward model or pruning network.

Training-free GPT-3.5 agents prompt a closed-source GPT-3.5 backbone with handcrafted reasoning scaffolds and external tools, without any fine-tuning of the underlying LLM. We include this category for historical completeness, since the published numbers from these methods on WTQ, TabFact, and TabMWP set the prior open-domain accuracy bar. DATER [68] decomposes a complex question into a sequence of sub-table operations. ReAcTable [73] adapts the ReAct

paradigm to tabular QA by interleaving SQL and Python tool calls with chain-of-thought reasoning. Mix-SC [30] aggregates self-consistency across mixed text and symbolic reasoning paths. TIDE [63] structures the table into triples to make question decomposition and step verification more tractable, and CIT-DP [64] debiases the LLM through front-door causal intervention. Because these baselines are GPT-3.5 era, they are not the most direct comparison point for TABALIGN, but they remain useful for situating absolute accuracy on the long-running benchmarks.

Baselines are reproduced from published checkpoints under the same evaluation protocol where possible. Numbers taken directly from the original papers are marked with †.

Baseline selection rationale. Our baseline pool is intentionally aligned with the most recently published peer-reviewed table reasoning works, in particular TableDART [57] and TATTOO [78], so that the numbers in Table 4 are directly comparable to the latest published table reasoning results without re-implementation drift. We acknowledge that several baselines in this pool, such as Llama-2-7B, Table-LLaVA-7B, and Gemini 2.0 Flash, use backbones that are now one to two generations behind the frontier. To address the concern that this could understate current open-source capability and bias the comparison in our favor, we additionally evaluate Qwen3-VL-8B [3] both as a standalone table-as-image VLM baseline and as the reasoner backbone in TABALIGN. Qwen3-VL-8B postdates the Qwen2.5-VL-7B and Ovis2-8B variants used by TableDART and the text-only Qwen3-8B backbone used by TATTOO, so the gains we report are measured against the strongest publicly available open-source VLM at our parameter scale at the time of submission, not only against the legacy backbones inherited from prior baseline pools.

Implementation details. The DLM understanding model (LLaDA-8B-Instruct) conditions on the question, column names, data types, and up to 15 sampled cell values. It runs 128 denoising steps with block length 32 and greedy decoding. The AR reasoner (Qwen3-VL-8B) runs at most 6 tool steps with a maximum of 16,384 output tokens per step and temperature 0.7. The AR understanding model ablation uses the same Qwen3-VL-8B backbone to isolate the DLM contribution. TABALIGN commits **one trajectory per query** (N=1): the DLM emits a plan via greedy denoising, the reasoner executes it step by step, and TABATTN acts as an in-the-loop step-level verifier whose score and rationale are passed to the reasoner before the next step. A stagnation halt emits the highest-scoring step's answer when R_{attn} fails to improve by more than 0.02 for two consecutive steps and the table state is unchanged. This design redirects misaligned steps through in-loop feedback rather than paying the $N\times$ cost of an outer best-of-N re-ranker over independently sampled trajectories. TABATTN's learned attention verifier (Equation (1)) is fit by binary cross-entropy on terminal trajectory correctness using the curated 1,600 attention standards (Section B.3), split 80/20 with the inner 20% reserved solely for early stopping. This is the single training procedure used in all reported results. Separately, to test cross-dataset generalisation, we run a transfer evaluation in which TABATTN is trained on WTQ examples only and measured on held-out performance on the remaining seven benchmarks (see Section B.4 and the transfer results below).

TABATTN cross-dataset transfer. TABATTN trained on WTQ attention standards transfers to all seven held-out benchmarks without degradation ($\Delta\rho \leq 0$ in all cases), with AUROC of 0.95 to 1.00 on six of seven benchmarks. TabMWP is the outlier (AUROC $= 0.779$), consistent with its near-ceiling accuracy (95.1%) limiting binary discrimination. MMQA shows the sharpest transfer benefit (ρ=0.888 vs. 0.834 when retrained on MMQA), suggesting WTQ's trajectory diversity provides a stronger training signal than the target domain's own data.

Volatility of reported accuracies. We adopt n=200 evaluation questions per dataset under a single deterministic decoding pass (Section 5), matching the per-dataset budget used by recently accepted table reasoning works such as TaTToo [78] and TableDART [57]. To surface volatility for every reported accuracy under this protocol, Table 9 attaches a per-cell binomial standard deviation $\sigma=\sqrt{p(1-p)/n}$ to every accuracy in Table 4 and Table 5, with the Avg. column reporting the corresponding macro-mean alongside its propagated standard error under independent binomials. Per-cell SDs sit between 1.3 and 3.5 pp at n=200, narrowing on the macro-mean. We report this volatility analysis as a transparency aid since prior table reasoning works do not disclose run-to-run variance under matched evaluation budgets, and we use the binomial form rather than seed-level σ because the dominant source of variance under deterministic decoding is finite-sample evaluation noise rather than stochastic generation.

A.1. FeTaQA: LLM-Judge Supplementary Analysis

FeTaQA is evaluated with BLEU throughout this paper, consistent with standard practice [36]. The BLEU score of 32.63 reported in Table 4 is computed after a post-processing step that rewrites fragmentary fact-extraction outputs into complete

Table 9. **Volatility study via binomial-approximation standard deviations.** For every accuracy value reported in Table 4 (Part a) and Table 5 (Parts b1, b2), we report the binomial standard deviation $\sigma = \sqrt{p(1-p)/n}$ at n=200 evaluations per dataset, the unified evaluation budget under our protocol. Cells are written *accuracy* $\pm\, \sigma$ in percentage points (pp). The Avg. column reports the macro-average mean and the propagated standard error of that mean under independent binomials, $\mathrm{SE}(\bar{p}) = \frac{1}{k}\sqrt{\sum_i p_i(1-p_i)/n}$, with k the number of datasets averaged. FeTaQA is omitted because BLEU is not a binomial accuracy. For baselines re-evaluated under our pipeline n=200 matches the actual sample size, for numbers cited from prior work n=200 is used as a consistency-level estimate (original protocols may differ). Avg. matches each source table's convention (Part a excludes MMQA, Parts b1, b2 exclude TABMWP).

Method	WTQ	MMQA	TABMWP	TAT-QA	HiTab	TabFact	InfoT	Avg.
Part (a): Main benchmark results (mirrors Table 4)								
Table-as-Text Baselines								
Llama-2-7B	16.39 ± 2.62	—	22.82 ± 2.97	13.73 ± 2.43	10.72 ± 2.19	9.20 ± 2.04	38.92 ± 3.45	18.63 ± 1.09
Llama3-Instruct-8B	21.24 ± 2.89	—	42.01 ± 3.49	13.08 ± 2.38	6.97 ± 1.80	73.89 ± 3.11	54.00 ± 3.52	35.20 ± 1.20
TableLlama-7B	24.97 ± 3.06	—	10.10 ± 2.13	19.04 ± 2.78	46.57 ± 3.53	79.37 ± 2.86	46.57 ± 3.53	37.77 ± 1.23
Table-as-Image Baselines								
Table-LLaVA-7B	18.43 ± 2.74	—	57.78 ± 3.49	12.82 ± 2.36	10.09 ± 2.13	59.85 ± 3.47	65.26 ± 3.37	37.37 ± 1.22
SynTab-LLaVA-7B	39.59 ± 3.46	—	88.30 ± 2.27	51.94 ± 3.53	35.66 ± 3.39	70.78 ± 3.22	69.42 ± 3.26	59.28 ± 1.31
MiniCPM-V-2.6-8B	47.97 ± 3.53	—	83.68 ± 2.61	51.55 ± 3.53	56.53 ± 3.51	78.48 ± 2.91	73.03 ± 3.14	65.21 ± 1.32
Qwen2.5-VL-7B	54.37 ± 3.52	—	63.69 ± 3.40	51.94 ± 3.53	62.69 ± 3.42	75.81 ± 3.03	70.13 ± 3.24	63.10 ± 1.37
Qwen3-VL-8B	81.82 ± 2.73	—	79.90 ± 2.83	59.80 ± 3.47	65.83 ± 3.35	82.41 ± 2.69	75.88 ± 3.03	74.27 ± 1.24
Table-as-Multimodality Baselines								
HIPPO-8B	55.77 ± 3.51	—	87.50 ± 2.34	60.75 ± 3.45	63.00 ± 3.41	82.27 ± 2.70	75.74 ± 3.03	70.84 ± 1.27
Gemini 2.0 Flash	63.56 ± 3.40	—	46.29 ± 3.53	35.62 ± 3.39	60.41 ± 3.46	81.33 ± 2.76	54.31 ± 3.52	56.92 ± 1.37
Training-free TableQA Agents (Proprietary GPT3.5, cited)								
DATER	65.90 ± 3.35	—	—	—	—	85.60 ± 2.48	—	75.75 ± 2.09
ReAcTable	68.00 ± 3.30	—	—	—	—	86.10 ± 2.45	—	77.05 ± 2.05
Mix-SC	73.70 ± 3.11	—	—	—	—	88.50 ± 2.26	—	81.10 ± 1.92
TIDE	75.00 ± 3.06	—	—	—	—	89.82 ± 2.14	—	82.41 ± 1.87
CIT-DP	76.40 ± 3.00	—	—	—	—	91.30 ± 1.99	—	83.85 ± 1.80
Routing on TableGPT2-7B / Ovis2-8B (cited)								
TG2-7B (text)	61.42 ± 3.44	—	83.87 ± 2.60	50.39 ± 3.54	70.27 ± 3.23	77.80 ± 2.94	71.07 ± 3.21	69.14 ± 1.30
Ovis2-8B (image)	58.76 ± 3.48	—	87.00 ± 2.38	47.67 ± 3.53	68.59 ± 3.28	80.80 ± 2.79	74.11 ± 3.10	69.49 ± 1.27
TableDART (Q2.5VL)	69.29 ± 3.26	—	72.61 ± 3.15	59.07 ± 3.48	71.13 ± 3.20	77.94 ± 2.93	71.46 ± 3.19	70.25 ± 1.31
TableDART (Ovis2-8B)	70.58 ± 3.22	—	84.54 ± 2.56	62.05 ± 3.43	74.37 ± 3.09	81.37 ± 2.75	76.22 ± 3.01	74.86 ± 1.23
Trajectory-optimizing agents (cited)								
DS + Qwen2.5-Math-PRM-72B	69.20 ± 3.26	24.40 ± 3.04	—	—	—	57.90 ± 3.49	—	63.55 ± 2.39
DS + TATTOO (Qwen3-8B)	69.80 ± 3.25	25.10 ± 3.07	—	—	—	58.79 ± 3.48	—	64.30 ± 2.38
TabTrim-4B	76.80 ± 2.98	—	—	—	—	89.40 ± 2.18	—	83.10 ± 1.85
TabTrim-8B	79.40 ± 2.86	—	—	—	—	91.20 ± 2.00	—	85.30 ± 1.75
TABALIGN								
LLM + Q2.5VL	69.89 ± 3.24	47.22 ± 3.53	87.94 ± 2.30	74.87 ± 3.07	75.38 ± 3.05	83.33 ± 2.64	77.89 ± 2.93	78.22 ± 1.18
DLM + Q2.5VL	73.76 ± 3.11	52.53 ± 3.53	88.50 ± 2.26	78.37 ± 2.91	79.00 ± 2.88	82.41 ± 2.69	79.71 ± 2.84	80.29 ± 1.14
LLM + Q3VL	90.32 ± 2.09	61.34 ± 3.44	96.35 ± 1.33	85.50 ± 2.49	78.50 ± 2.90	92.39 ± 1.87	83.41 ± 2.63	87.75 ± 0.93
DLM + Q3VL	94.12 ± 1.66	71.07 ± 3.21	95.13 ± 1.52	91.41 ± 1.98	82.48 ± 2.69	90.09 ± 2.11	90.45 ± 2.08	90.61 ± 0.83
Part (b1): Backbone ablation (mirrors Table 5(a))								
DLM Reasoning	30.61 ± 3.26	33.33 ± 3.33	—	46.91 ± 3.53	33.33 ± 3.33	41.94 ± 3.49	40.28 ± 3.47	37.73 ± 1.39
LLM Reasoning	81.54 ± 2.74	60.10 ± 3.46	—	79.80 ± 2.84	73.87 ± 3.11	87.05 ± 2.37	79.78 ± 2.84	77.02 ± 1.19
LLM Planned Reasoning	90.32 ± 2.09	61.34 ± 3.44	—	85.50 ± 2.49	78.50 ± 2.90	92.39 ± 1.87	83.41 ± 2.63	81.91 ± 1.07
DLM Planned Reasoning	94.12 ± 1.66	71.07 ± 3.21	—	91.41 ± 1.98	82.48 ± 2.69	90.09 ± 2.11	90.45 ± 2.08	86.60 ± 0.96
Part (b2): Reward-signal ablation (mirrors Table 5(b))								
Cosine Sim	77.80 ± 2.94	56.60 ± 3.50	—	81.50 ± 2.75	67.00 ± 3.32	83.80 ± 2.61	78.20 ± 2.92	74.15 ± 1.23
Numerical F1	76.50 ± 3.00	56.30 ± 3.51	—	79.40 ± 2.86	66.50 ± 3.34	84.30 ± 2.57	78.40 ± 2.91	73.57 ± 1.24
Cosine + Numerical	75.90 ± 3.02	60.10 ± 3.46	—	79.00 ± 2.88	65.50 ± 3.36	85.30 ± 2.50	77.30 ± 2.96	73.85 ± 1.24
TABROUGE	81.20 ± 2.76	57.70 ± 3.49	—	85.90 ± 2.46	65.50 ± 3.36	92.40 ± 1.87	80.20 ± 2.82	77.15 ± 1.16
TABATTN (ours)	90.30 ± 2.09	61.30 ± 3.44	—	85.50 ± 2.49	78.50 ± 2.90	92.40 ± 1.87	83.40 ± 2.63	81.90 ± 1.07

descriptive sentences using a secondary LLM call with no gold reference (natural mode), following the dataset-specific answer generation approach used in prior work [57]. The diagnostic analysis below is conducted on raw agent outputs

before this refinement step, on a 185-example subset, to illustrate why unrefined outputs score poorly on BLEU despite being factually correct. BLEU measures n-gram overlap against a single human reference phrasing and is insensitive to semantically equivalent answers expressed differently, so fact-extraction outputs systematically under-score even when the factual content is correct.

To quantify this gap, we ran a post-hoc LLM judge (GPT-5-nano) on these 185 pre-refinement FeTaQA trajectories, asking whether each predicted answer is *factually correct* regardless of phrasing. The judge was given the question, a table excerpt, the reference answer, and the predicted answer, and asked to respond with exactly one word (CORRECT or INCORRECT). Results are summarised below.

Metric	**Value**
Trajectories judged	185
Judge-correct	125 (67.6%)
Judge-incorrect	34 (18.4%)
Unresolved (judge abstained)	26 (14.1%)
Mean BLEU (all)	0.089 ± 0.165
Judge-correct but BLEU < 0.05	81 (43.8%)

The key figure is the last row: **81 predictions (43.8%) are judged semantically correct yet score near-zero BLEU**, confirming that the low BLEU reflects a phrasing mismatch rather than a factual failure. Among the 159 trajectories where the judge reached a verdict, 78.6% were judged correct.

We emphasise that this analysis does *not* replace the BLEU score reported in Table 4. It supplements it. The LLM-judge methodology does not follow the standard FeTaQA evaluation protocol, and direct comparison with baselines that report BLEU should be made only via the BLEU column. The judge results are offered as diagnostic evidence that the capability gap is smaller than the metric gap suggests.

B. Method Details

This section consolidates implementation and design details for the three components of TABALIGN: the architectural choice that splits planning from execution (Section B.1), the diffusion-based understanding model that emits plans (Section B.2), the curated attention standards that train the verifier (Section B.3), and the design justification for the TABATTN verifier itself (Section B.4).

B.1. Why Planning and Execution Should Be Split

Table reasoning decomposes into a global *planning phase* and a reactive *execution phase* that place different computational demands on the model. Four reasons justify assigning these phases to different architectures.

Reason 1: execution is inherently tool-conditioned and sequential. During execution, the model must read the latest tool output (often a transformed dataframe printed as text) and generate the next tool call conditioned on that observed state. This creates an observe $\rightarrow$ respond $\rightarrow$ observe loop, not a one-shot structured prediction problem. LLaDA-style masked diffusion models instead generate by iteratively denoising a fully specified masked canvas, which is well suited to holistic sequence construction but not naturally suited to reactive conditioning on newly arriving external observations [38]. Autoregressive models natively condition each next token on all previously revealed context, including newly returned tool outputs.

Reason 2: DLMs require fixed output length, which is unknown at execution time. Tool outputs vary substantially across steps: a selection operation may return a short list of columns, a filter may return many rows, and an aggregation may collapse the state to a scalar. The reasoner must consume these variable-length observations and decide when to terminate with a final answer. This is a poor match for masked diffusion inference, where the generation canvas length is typically fixed before decoding begins. Recent work makes this limitation explicit: DAEDAL frames statically specified generation length as a core inference bottleneck for diffusion language models, while FlexMDM starts from the same fixed-canvas assumption and requires substantial retraining to support flexible-length generation [25; 20]. We verify this directly: LLaDA-8B running end-to-end table reasoning with a 256-token canvas achieves 40.28% exact match on InfoTabs and 41.94% on TabFact, compared with 79.78% and 87.05% for AR Reasoning (Table 5), a gap of 39 to 45 pp. We note that 256 tokens is the largest canvas that reliably fits within LLaDA-8B's fixed-canvas constraint for multi-step table inputs. Larger canvases that exceed the model's pre-trained length distribution cause degenerate outputs.

Reason 3: the DLM attention advantage is a comprehension advantage, not a generation advantage. The correct tool call at step t depends on the live tool output o_{t-1} returned by step $t-1$. That output is not available before denoising begins, because it is produced by executing the prior action in the environment. A DLM cannot condition step t's tool call on o_{t-1} without first observing o_{t-1}, creating a fundamental circular dependency. Autoregressive decoding resolves this naturally: each token is conditioned on all previously revealed context, including freshly returned tool outputs interleaved between generation steps. Masked diffusion, which denoises a fixed canvas in parallel, has no mechanism for this interleaved conditioning without restarting denoising from scratch after every tool call.[1]

Reason 4: AR reasoners are better aligned to multimodal grounding, instruction following, and error recovery. The reasoner must perceive rendered table images, and no current diffusion language model supports image inputs. LLaDA, the DLM used as the understanding model, is text-only. Only an autoregressive VLM can handle the multimodal perception that action-conditioned modality routing requires [24]. Beyond this structural gap, instruction-tuned AR VLMs are better suited to following tool specifications exactly, remaining format-faithful, and recovering from imperfect intermediate states. The practical cost of using a DLM as reasoner is visible in Table 4: the DLM-only baseline (LLaDA-8B) falls more than 30 pp below AR Reasoning (Qwen3-VL-8B) on average. Current diffusion VLMs more broadly still lag instruction-tuned AR VLMs in stepwise interactive generation [1].

[1] Schema-scaffolding approaches such as S^3 [58] cannot side-step this: the output schema at execution time is data-dependent (whether the next action is `f_aggregate` or `f_filter_row` is determined by the prior tool output that has not yet been observed).

B.2. Diffusion-based Understanding Model: Implementation Details

Fixed-length canvas and block denoising. The plan canvas has a fixed length of $L{=}256$ tokens [38], chosen to be long enough that no plan is truncated. Positions not consumed by the plan remain as padding tokens. Denoising proceeds over semi-autoregressive blocks of 32 tokens: within each block all masked tokens are refined in parallel, and blocks are processed left-to-right, following the long-context diffusion practice of Liu et al. [31]; He et al. [17]. We run 128 denoising steps by default (256 for higher-quality plans at $2\times$ cost) with classifier-free guidance scale 1.5 and greedy decoding ($T{=}0$).

Tool vocabulary. The planning prompt presents six operations: `filter` (select rows satisfying a column condition), `sort` (order rows by a column), `aggregate` (sum/count/average/min/max over a column), `lookup` (retrieve a specific cell), `compare` (compare values across rows or columns), and `select` (extract a column subset). Each generated step must name one of these operations and end with a `[target: col]` tag (or `[target: col, row N]` for cell-level steps) identifying the table cells it operates on. Example: *"Aggregate: count rows where* COUNTRY = *Algeria. [target: Country]"*.

Cell mask extraction. After generation, each `[target:]` tag is parsed to identify the designated column(s) and, where specified, row index. All cells matching the designation are set to 1 in $\mathbf{m}_t \in \{0,1\}^C$, and all other cells are set to 0. If a step produces no parseable tag ($+3.6$ pp DLM-vs-AR parse-error increase, Section E.4), $\mathbf{m}_t$ defaults to a uniform mask and R_{attn} is treated as uninformative for that step.

B.3. Attention Standard Curation: Details and Model Comparison

Human seed annotations. Human annotators produced 100 seed attention maps from scratch, spanning all eight benchmarks. For each (question, table) pair, annotators identified the minimal set of cells (by column and row) that a correct reasoning trajectory must attend to in order to answer the question. Disagreements were resolved by majority vote, and inter-annotator agreement (IoU) exceeded 0.90 on a held-out verification subset. The seed set covers a range of question types: single-cell lookup, multi-cell aggregation, cross-column comparison, and multi-hop.

LLM model selection. To identify the strongest LLM for scaling up from the 100 seeds, we sampled 20 WTQ examples from the seed set as a calibration split. Five GPT models were prompted with each (question, table, question type) triple and asked to predict the same cell attention set. Predictions were evaluated against human annotations using Precision, Recall, F1, and IoU:

Table 10. Attention alignment of five GPT models on the 20-example WTQ calibration split drawn from the 100 human seed annotations. GPT-5 is selected for mass-scale generation.

Model	Precision	Recall	F1	IoU
GPT-4.1	0.834	0.674	0.698	0.594
GPT-5.1	0.824	0.676	0.698	0.583
GPT-5.2	0.868	0.776	0.776	0.686
GPT-5.4	0.891	0.836	0.812	0.757
GPT-5	**0.937**	**0.849**	**0.866**	**0.818**

GPT-5 leads on all four metrics and is the only model to exceed F1 = 0.85 and IoU = 0.80, matching human annotator judgment most closely.

Mass-scale generation and quality control. Using the 100 human seeds as few-shot exemplars, GPT-5 was prompted to generate attention standards for all eight benchmarks: WTQ, MMQA, TabMWP, TAT-QA, HiTab, FeTaQA, TabFact, and InfoTabs. Approximately 200 candidates were generated per dataset, stratified by question type where possible. All generated examples were reviewed by human annotators post-generation. Examples failing the minimum IoU threshold of 0.70 were discarded and regenerated, yielding a total of 1,600 human-verified examples. These 1,600 standards form the training set for TABATTN (Equation (1)), fit on an 80/20 split with the inner 20% held out only for early stopping.

Train/evaluation split disjointness. To rule out leakage between the 1,600 attention-standard training data and the QA evaluation in Table 4, all curated examples per benchmark are sampled exclusively from the *training* or *development*

splits of each underlying QA dataset, and none overlap with the held-out test splits used to report downstream reasoning accuracy. We verified this disjointness automatically by hashing every (question, table) pair: zero collisions between the 1,600 attention-standard pool and the union of the eight test splits.

B.4. TABATTN Attention Verifier: Design Justification

TABATTN uses R_{attn} (Equation (1)) as its sole process signal, fit by binary cross-entropy against terminal trajectory correctness on the curated 1,600 attention standards (Section B.3). The ablation in Table 5(b) confirms that attention grounding alone outperforms lexical and semantic content signals, motivating this focused design.

On the validity of the understanding model's cell mask. One might worry about circularity: if the DLM understanding model occasionally hallucinates cell designations, then R_{attn} penalizes a reasoner that correctly deviated from a flawed plan. We bound this concern in two ways. First, the understanding model operates over compact metadata rather than raw cell contents, so its hallucinations are typically schema-level (wrong column name) rather than cell-level, and are detectable by cross-referencing with the schema adherence analysis (Table 8). Second, TABATTN is trained on terminal correctness labels. If the understanding model's cell designations were systematically misleading, the optimizer would suppress the attention signal during training, and the large positive contribution of R_{attn} in Table 5(b) confirms the masks are informative on net.

Cross-dataset transfer. TABATTN trained on WTQ attention standards only transfers to all seven held-out benchmarks without degradation (Section A), validating that the attention-grounding signal generalises across table types and question categories without per-dataset retuning.

C. Theoretical Foundations

This section consolidates the formal results that motivate TABALIGN's state-based reward design. We first show that unguided multi-turn reasoning accumulates variance with trajectory length (Section C.1), then characterise the lexical TABROUGE baseline and prove its Pareto-parsimony together with its monotonicity profile (Sections C.2 and C.3), and finally prove that an explicit step reward acts as a compressed distillation signal that mitigates stochastic drift (Section C.5).

C.1. Variance growth under unguided multi-turn reasoning

Proposition C.1 (Variance growth with interaction length)**.** *Let $Y_S := \ell(\hat{a}(\tau_S), A)$ denote a nonnegative loss after S interaction steps, where $Y_S = 0$ iff the agent produces the correct answer. Assume each additional step introduces stochastic variability without systematic corrective bias: $\mathbb{E}[Y_S \mid Y_{S-1}] = Y_{S-1}$. Then* $\mathrm{Var}(Y_S) \geq \mathrm{Var}(Y_{S-1})$.

Proof. By the law of total variance, $\mathrm{Var}(Y_S) = \mathrm{Var}(\mathbb{E}[Y_S \mid Y_{S-1}]) + \mathbb{E}[\mathrm{Var}(Y_S \mid Y_{S-1})] = \mathrm{Var}(Y_{S-1}) + \mathbb{E}[\mathrm{Var}(Y_S \mid Y_{S-1})] \geq \mathrm{Var}(Y_{S-1})$. □

Implication. Without explicit step-level reward feedback, reasoning trajectories in tool-based TableQA accumulate variance monotonically with trajectory length. State-based rewards (Section 4) break this by introducing a corrective signal that conditions each step on measurable progress, capping variance growth.

C.2. TABROUGE: Lexical Baseline and Pareto-Parsimony

TABROUGE definition. TABROUGE [23] measures how much of the query q is covered by the serialized intermediate table state $\mathrm{Enc}(s_{l+1})$:

$$R_{\mathrm{TABROUGE}}(s_{l+1}; q) = \frac{\mathrm{LCS}(q,\, \mathrm{Enc}(s_{l+1}))}{|\mathrm{Enc}(s_{l+1})|}, \tag{2}$$

where LCS is the longest common subsequence length. The numerator rewards query-relevant lexical coverage, and the denominator penalizes diffuse states retaining irrelevant content. TABROUGE showed a Spearman correlation of 0.521 with downstream answer correctness on WTQ [23].

Despite this correlation, TABROUGE has two structural limitations that motivate the TABATTN design. First, it compares against q, but many correct intermediate states produce values (sums, argmax results, comparisons) that never appear in q, so lexical coverage fails silently on the very steps that matter most. Second, as a surface-form metric it cannot distinguish an intermediate state where the right answer is *present* from one where it was *computed from the right cells*: a table that coincidentally contains the target value in an irrelevant column scores identically to one where the value was derived correctly.

Proposition C.2 (Pareto-parsimony of TABROUGE maximizers)**.** *Given query q and a candidate set $\mathcal{S}$, assume $\mathcal{S}$ is closed under (a) deletion of any substring of* $\mathrm{Enc}(s)$ *disjoint from* $\mathrm{LCS}(q, \mathrm{Enc}(s))$*, and (b) appending of admissible content* Γ *drawn from the source table. Any maximizer $s^* \in \arg\max_{s \in \mathcal{S}} R_{\mathrm{TABROUGE}}(s; q)$ is* Pareto-parsimonious *within $\mathcal{S}$ in two senses. First, removing any substring $\Delta \subset \mathrm{Enc}(s^*)$ disjoint from* $\mathrm{LCS}(q, \mathrm{Enc}(s^*))$ *strictly increases the score, so no removable redundancy exists. Second, appending content* Γ *with marginal density exceeding the current score also strictly increases the score, so s^* does not omit available query-relevant evidence from $\mathcal{S}$. This characterises the maximizer's internal structure, and it does not imply that any maximizer is a globally correct intermediate state, since $\mathcal{S}$ may not contain a state that is both parsimonious and factually correct.*

Proof. Throughout the proof, the constructed states $\tilde{s}$ (substring removal) and s^+ (content appending) lie in $\mathcal{S}$ by the closure assumptions of Proposition C.2, so the strict inequalities below contradict the maximality of s^* within $\mathcal{S}$.

No removable redundancy. Suppose $\exists\, \Delta \subset \mathrm{Enc}(s^*)$ disjoint from $\mathrm{LCS}(q, \mathrm{Enc}(s^*))$. Removing Δ yields $\tilde{s}$ with $R_{\mathrm{TABROUGE}}(\tilde{s}; q) = \frac{\mathrm{LCS}(q, \mathrm{Enc}(s^*))}{|\mathrm{Enc}(s^*)| - |\Delta|} > R_{\mathrm{TABROUGE}}(s^*; q)$, contradicting maximality of s^*.

No available query-relevant evidence is omitted. Suppose content Γ can be appended to form s^+ with marginal density $\delta/|\Gamma| > R_{\mathrm{TABROUGE}}(s^*; q)$, i.e. the fraction of query-relevant tokens in Γ exceeds the current score. Then $R_{\mathrm{TABROUGE}}(s^+; q) = \frac{\mathrm{LCS}(q, \mathrm{Enc}(s^*)) + \delta}{|\mathrm{Enc}(s^*)| + |\Gamma|} > R_{\mathrm{TABROUGE}}(s^*; q)$, since cross-multiplying the strict inequality $\delta/|\Gamma| > \mathrm{LCS}(q, \mathrm{Enc}(s^*))/|\mathrm{Enc}(s^*)|$ gives $|\mathrm{Enc}(s^*)| \cdot \delta > \mathrm{LCS}(q, \mathrm{Enc}(s^*)) \cdot |\Gamma|$, which is exactly the condition for the above strict

inequality. This contradicts maximality of s^*. □

C.3. Monotone TABROUGE score under correct tool execution

Under correct and successful tool execution, each step either removes query-irrelevant content (filter/select) or appends query-derived content (aggregate/compute). Let $L_k = |\mathrm{Enc}(s_k)|$, $c_k = \mathrm{LCS}(q, \mathrm{Enc}(s_k))$, and $\rho_k = c_k/L_k$.

- **Pruning step**: $\Delta L_k < 0$, $\Delta c_k = 0$ (query-relevant tokens preserved). For $c_k > 0$, $\rho_{k+1} = c_k/(L_k + \Delta L_k) > \rho_k$. The boundary case $c_k = 0$ gives $\rho_{k+1} = \rho_k = 0$, so the inequality is non-strict.
- **Derivation step**: $\Delta L_k, \Delta c_k > 0$. The reward improves when $\Delta c_k/\Delta L_k > \rho_k$, i.e. the newly appended tokens are denser in query-relevant content than the current state.

This shows that TABROUGE is monotonically non-decreasing along the restricted subset of trajectories whose every step is either a valid filter/projection or a derivation step whose appended tokens lexically overlap with q. This subset is strictly smaller than the class of trajectories that correctly answer the query, because correct derivations frequently produce values such as sums, argmax outputs, or comparisons that never appear in q. At those steps $\Delta c_k = 0$ while $\Delta L_k > 0$, so TABROUGE strictly decreases at exactly the steps where computed evidence is generated. This residual lexical blind spot motivates the attention-grounded TABATTN reward in Section 4.

C.4. Reward model significance under partial observability

Proposition C.3 (Reward model significance under partial observability)**.** *Let the table-reasoning task be a POMDP where s_t is the hidden table state and $o_t = \phi(s_t)$ is a partial-snapshot observation. The projection induces an information bottleneck: $I(s_t; o_t) \ll H(s_t)$. Assume the scalar step reward $r_t = \mathcal{R}(a_t, s_{t+1}; q)$ is informative about s_{t+1} beyond the partial observation, that is, $I(s_{t+1}; r_t \mid o_{t+1}) > 0$. Then r_t functions as a* compressed table distillation signal *satisfying $H(s_{t+1} \mid o_{t+1}, r_t) < H(s_{t+1} \mid o_{t+1})$, compensating for the limited observability of o_t and stabilizing long-horizon trajectory search.*

C.5. How explicit rewards mitigate stochastic drift

Without a state-based reward, tool-based table transitions follow a stochastic transition kernel $s_{t+1} \sim P(\cdot \mid s_t, a_t, \theta_{a_t})$. Let $s_t^\star$ denote the correct intermediate state at step t, and let $d(\cdot, \cdot)$ be a divergence on encoded states (for example, normalized edit distance over $\mathrm{Enc}(\cdot)$). The per-step divergence $\epsilon_t := d(s_{t+1}, s_{t+1}^\star)$ quantifies the imagination gap between the model's predicted outcome and the realized transformation. Unobserved, these per-step divergences accumulate along the trajectory, and the cumulative divergence $\sum_{l=0}^{S} \epsilon_l$ upper-bounds $d(s_S, s_S^\star)$, which can grow unboundedly with S.

The step reward $r_t = R_{\mathrm{TABROUGE}}(a_t, s_{t+1}; q)$ functions as a compressed distillation signal that maps the hidden global table state to a one-dimensional progress measure. By Proposition C.3, conditioning on r_t reduces the remaining state entropy, allowing the agent to prune divergent trajectories before error compounds across multiple turns.

D. Attention Pilot Extensions

This section reports the four attention-side empirical analyses that extend the pilot study in Section 3: the bidirectional plan-attention signature that motivates the DLM planner (Section D.1), the row-permutation stability stress test (Section D.2), the autoregressive-VLM image-attention analysis that justifies a text-side attention verifier (Section D.3), and extended attention-quality figures (Section D.6).

D.1. Plan-region Attention: Upper-Triangle Fraction

The DLM planning advantage has a directly measurable mechanistic signature: on a shared table-reasoning input, the diffusion model exhibits dense plan-to-plan attention with non-zero mass above the diagonal, whereas the autoregressive baseline has exactly zero forward attention by design (Table 11, with full attention maps in Section D.6). This bidirectional inductive bias is the mechanism underlying the schema-hallucination reduction reported in Section 5.2, together with the trajectory compression and execution stabilization reported in Section E.

Table 11. Plan-region attention upper-triangle fraction. DLM's bidirectionality produces substantial attention above the diagonal, while AR is exactly zero by design.

Model (architecture)	Upper-triangle fraction	Expected
LLaDA-8B-Instruct (DLM, `is_causal=False`)	**39.9%**	≈50%
Qwen3-VL-8B (AR, causal mask)	**0.0%**	0%

Large-scale attention analyses across 1,600 questions and eight benchmarks further corroborate this finding, with their full per-dataset breakdown presented in Section 3.

D.2. Permutation Stability of Table Attention

This subsection details the row-permutation stress test referenced in Section 3, which quantifies whether bidirectional pre-training, in addition to lifting the attention-AUROC *level* (Table 1), also improves *stability* of attention under row reordering, an empirically desirable property given the order-sensitive contextualization imposed by causal AR models on tabular data.

Setup. For each of the four models studied in Section 3, namely Qwen3-VL-8B (causal AR), ModernBERT-base (bidirectional MLM), DiffuLLaMA-7B (causally-pretrained LLaMA adapted via diffusion post-training and bidirectional mask-flipped inference), and LLaDA-8B (natively diffusion-pretrained DLM), we draw the first 50 (table, question) records from each of the eight benchmarks. For every record we generate $K{=}5$ row-permuted views: permutation 0 is the identity, and the other four are seeded from the record id so that all four models see the same row orderings. The human-annotated cell relevance mask is permuted in lockstep with the data rows so that AUROC remains comparable across orderings. Each view is run through the same attention-extraction $\rightarrow$ cell-scoring $\rightarrow$ AUROC pipeline as Table 1, and per record we compute σ_{AUROC} across the five orderings (records with fewer than three valid orderings, e.g. due to truncation at the $2{,}048$-token cap, are dropped). The protocol therefore fixes *everything* except row order, isolating attention stability from the AUROC level.

Results. Table 12 summarises per-model statistics, and Figure 4 shows the per-record σ_{AUROC} distribution. Two observations consistent with the main-text claim emerge. First, both models with native bidirectional pre-training (LLaDA and ModernBERT) yield substantially lower per-record σ_{AUROC} than the causal AR baseline: -40.2% and -62.6% in median, respectively, with p-values from a paired Wilcoxon test below 10^{-22} on $n{=}319$ records that have valid permutation samples for all four models. Second, DiffuLLaMA, which inherits a causally-pretrained LLaMA backbone, applies diffusion post-training, and uses bidirectional mask-flipped inference, shows median σ statistically indistinguishable from AR ($p{=}0.48$). This dissociation isolates the mechanism: *permutation-stable table attention requires the model's pre-training itself to be bidirectional*, and post-hoc diffusion adaptation alone is not sufficient.

Per-(model, dataset) breakdown. Table 13 reports mean σ_{AUROC} for each model on each benchmark. The advantage of LLaDA and ModernBERT over AR, and the AR-vs-DiffuLLaMA null result, are consistent across all eight benchmarks rather than driven by any single dataset.

Table 12. Per-record stability of attention AUROC under five row permutations of each (table, question) record. σ_{AUROC} is computed across the five orderings of the same record, and each entry is the across-record statistic. Paired Wilcoxon tests pair each record across models on the n=319 records that have valid permutation samples in all four models. Lower σ is better.

Model (attention)	median σ	mean σ	p_{90} σ	mean range	paired Wilcoxon p vs. AR
Qwen3-VL-8B (Causal AR)	0.123	0.125	0.238	0.311	—
DiffuLLaMA-7B (DLM, LLaMA)	0.123	0.132	0.248	0.327	0.48
LLaDA-8B (Native DLM)	0.074	0.086	0.166	0.212	1.8×10^{-23}
ModernBERT-base (Bidir. MLM)	0.046	0.055	0.105	0.139	1.1×10^{-41}

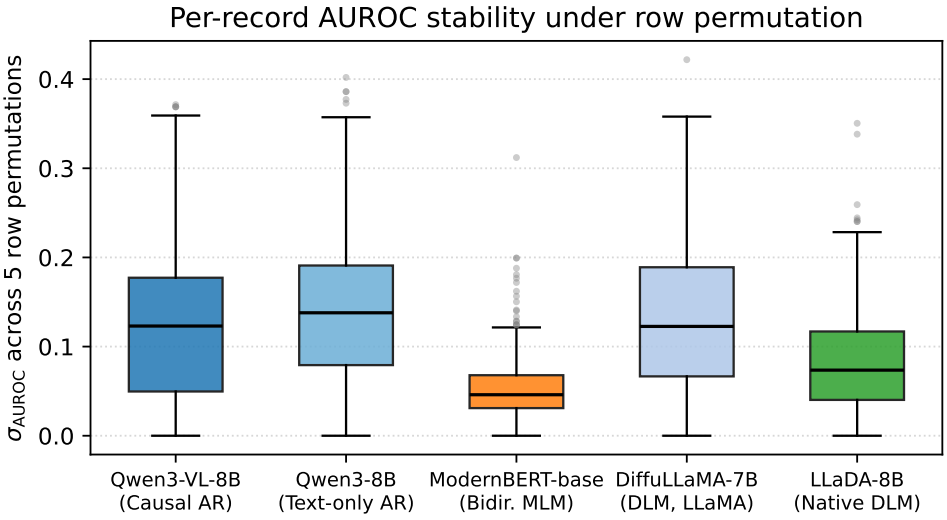


Figure 4. Distribution of per-record σ_{AUROC} across five row permutations of the same (table, question) record, by model. Both natively bidirectional pre-trained models (LLaDA, ModernBERT) cluster at substantially lower σ than the causal AR baseline, while DiffuLLaMA, which is bidirectional only at inference, is indistinguishable from AR.

Relation to the AUROC level result. Mean AUROC (Table 1) and per-record σ_{AUROC} measure independent properties: a model can attend accurately on average yet still shift its attention pattern under row reordering, or attend stably yet inaccurately. The two results combine cleanly: LLaDA achieves both higher mean AUROC than AR and strictly lower σ, ModernBERT trades level for the largest σ reduction observed, and DiffuLLaMA matches AR on σ while underperforming it on level. The headline conclusion, that bidirectional pre-training rather than bidirectional inference alone is the architectural property responsible for table-aligned attention, is therefore reinforced by both axes of evidence.

D.3. LLM Re-Attention to Image Patches in Autoregressive VLMs

This subsection reports the empirical study referenced in Section 4.2 that justifies confining TABATTN to text-side cell attention. The hypothesis under test is architectural: in autoregressive vision-language models, visual information is injected into the language model's residual stream via the vision projector, and the language model's self-attention during decoding is rarely directed back at the image-patch (<|image_pad|>) token positions. If true, a visual mixing term $\lambda_V \cdot a_t^{(c,\text{vis})}$ inside Eq. (1) would carry essentially zero signal regardless of how it is calibrated, making text-side attention the principled supervisory channel for the verifier.

Setup. We load *Qwen3-VL-8B-Instruct* via HuggingFace Transformers in 4-bit NF4 with `attn_implementation="eager"` and register forward hooks on the last 4 LLM self-attention layers (matching the layer-selection convention of TABATTN-T). For each example, the table is rendered with the same matplotlib renderer used by the reasoner's image modality and the prompt follows the standard Qwen3-VL chat template ([*system*, image, question, "Answer:"]). After a single forward pass, we read the row of attention from the *last text token* (the position that would emit the next decoded token) to all sequence positions, average across the 4 hooked layers and across heads, and partition the

Table 13. Mean per-record σ_{AUROC} by (model, dataset). Lower is better. The AR-vs-LLaDA / ModernBERT advantage and the AR-vs-DiffuLLaMA null result hold across all eight benchmarks.

Model	wikitq	tatqa	hitab	tabmwp	mmqa	infotabs	tabfact	fetaqa
Qwen3-VL-8B (AR)	0.106	0.106	0.167	0.221	0.134	0.173	0.091	0.128
DiffuLLaMA-7B (DLM)	0.088	0.073	0.154	0.198	0.158	0.215	0.098	0.115
LLaDA-8B (Native DLM)	0.064	0.040	0.121	0.120	0.089	0.126	0.063	0.092
ModernBERT-base (MLM)	0.042	0.045	0.061	0.069	0.048	0.105	0.046	0.047

resulting attention vector into (i) text positions and (ii) image-pad positions. We report

$$\text{image_attn_frac} \;=\; \frac{\sum_{j\in\mathcal{P}_{\text{img}}} a_j}{\sum_j a_j},$$

where $\mathcal{P}_{\text{img}}$ is the set of image-pad token positions in the input.

Sample. 180 examples drawn evenly from six benchmarks, 30 per dataset (WTQ, TabFact, HiTab, InfoTabs, TabMWP, TAT-QA), reconstructed from the partial-observation pilot set used elsewhere in this paper. Per-example image-pad token counts range from 392 to 1,292 (i.e. the image is genuinely present in the prompt with hundreds of patch tokens), and the attention row sums to 1.0 to within fp16 precision in every case (confirming the softmax is well-formed).

Table 14. Image-patch attention received by the last text token of the prompt across six benchmarks (Qwen3-VL-8B-Instruct, last 4 LLM self-attn layers, mean over heads). *Across all 180 examples, the LLM apportions zero attention mass to image-pad positions, despite each prompt containing hundreds of such tokens.*

Dataset	n	mean	median	max	% $<10^{-2}$	% $<5{\cdot}10^{-2}$
WTQ	30	0.000	0.000	0.000	100.0	100.0
TabFact	30	0.000	0.000	0.000	100.0	100.0
HiTab	30	0.000	0.000	0.000	100.0	100.0
InfoTabs	30	0.000	0.000	0.000	100.0	100.0
TabMWP	30	0.000	0.000	0.000	100.0	100.0
TAT-QA	30	0.000	0.000	0.000	100.0	100.0
Pooled	**180**	**0.000**	**0.000**	**0.000**	**100.0**	**100.0**

Methodology validation. An exactly-zero result invites the natural challenge that the attention pipeline is broken rather than the architecture being silent. To rule this out, we ran a control on a 30-example subset (5 per dataset) that decomposes *the same attention row* from the last text token into four mutually exclusive partitions in a single forward pass: image-pad tokens, the question text span, the trailing "`Answer:`" span, and all remaining text tokens (chat-template / system content). Aggregation is performed in float64 to rule out fp16 underflow at sum time. Table 15 reports the partition means.

Table 15. Decomposition of the attention row from the last text token into mutually exclusive token partitions (Qwen3-VL-8B-Instruct, 30 examples drawn 5-per-dataset across the six benchmarks, float64 aggregation). The same hook that records exactly zero attention to image-pad tokens captures non-zero attention to question and `Answer:` tokens in the same forward pass, ruling out hook failure, wrong query position, or precision underflow as the explanation for the headline result in Table 14.

Token partition	mean	median	min	max
Image-pad tokens	0.000	0.000	0.000	0.000
Question text	0.049	0.040	0.019	0.112
`Answer:` prefix	0.066	0.067	0.051	0.080
Chat-template + system + other	0.885	0.888	0.812	0.924
Total (sanity)	**1.000**	**1.000**	**1.000**	**1.000**

We further verify three structural sanity conditions across the 30 control examples. (i) `pixel_values` is present and non-trivial in every prompt (mean absolute value 0.97, range $[0.95, 1.00]$), confirming the image is genuinely consumed by

the vision encoder. (ii) The count of <|image_pad|> tokens identified per prompt equals exactly $1/4$ of the patch count predicted from the processor's `image_grid_thw` field on *every* example (ratio $= 0.2500$), matching Qwen3-VL's 2×2 spatial patch-merge factor and confirming our token-id mapping is correct. (iii) The attention row sums to 1.0 in float64 in every case (range $[0.99999, 1.00001]$), ruling out an underflow artefact at aggregation time. Together with the partition decomposition, these checks establish that the headline result, zero attention mass on image patches, is a property of the reasoner's decoding behaviour rather than of the measurement: the same hook on the same forward pass measures attention mass exactly where it should be (the `Answer:` source-token's recency window, the question text the reasoner is answering, and the chat-template scaffolding) and exactly zero where the architecture predicts it should be (the image-pad block).

Layer-by-layer extension. The headline experiment in Table 14 reads attention from the reasoner's final 4 LLM self-attention layers, matching TABATTN-T's hook configuration. A natural follow-up question is whether the zero-attention finding is specific to those late layers or whether it persists at earlier layers as well, since one might hypothesise that visual content is consumed via attention in early layers and the late layers simply operate on already-aggregated representations. We test this directly by hooking *all 36* LLM self-attention layers of Qwen3-VL-8B-Instruct and recording the same four-partition decomposition per layer, on the same 30-example control sample. Figure 5 plots the per-layer mean attention mass for each partition.

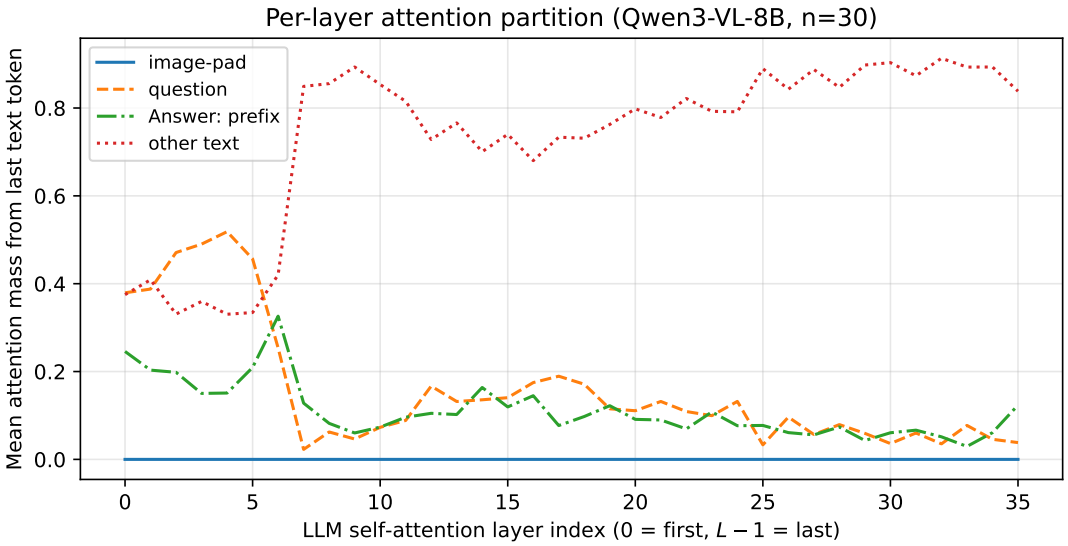


Figure 5. Per-layer mean attention mass from the last text token across all 36 LLM self-attention layers of Qwen3-VL-8B-Instruct (30-example control sample, float64 aggregation). The image-pad partition (solid blue) is exactly zero at *every* layer in *every* example (1,080 of 1,080 measurements at 0.0). The text partitions reveal an incidental two-phase decoding pattern: layers 0 to 6 perform text comprehension (question text receives 38 to 52% of mass), layers 7 to 35 perform aggregation and generation preparation (chat-template / system tokens dominate at 70 to 91%). The visual content never enters this picture as direct attention from the next-token-emit position.

The result is uniform across the entire LLM stack: the next-token-emit query position never allocates non-zero attention mass to image-pad tokens, in any of 36 layers, in any of 30 examples (1,080 of 1,080 measurements at exactly 0.0). The text-partition curves additionally reveal an incidental but informative pattern: between layers 6 and 7 the model sharply transitions from *text comprehension* (heavy attention to question and `Answer:` tokens) to *aggregation/generation preparation* (chat-template / system tokens dominate the attention mass). Visual content does not appear in either phase from the decoding-position query.

Interpretation. TABALIGN's architecture distinguishes *understanding* (the planning stage that benefits from multimodal input, e.g. spatial layout, hierarchical headers, visual cues) from *reasoning* (the stepwise execution stage that TABATTN monitors). The empirical result above shows that this distinction is reflected in the reasoner's internal computation: visual content enters via early-layer, early-token attention from text positions to image-pad positions (or directly via the vision projector's contribution to the residual stream), but by the time each reasoning step reaches its decoding position, that content

has already been laundered into text-token hidden states. The autoregressive language model's reasoning-time computation operates on text-side residual representations only, irrespective of which input modality fed those representations. TABATTN reads attention precisely at the reasoner's decoding position (the position where each reasoning step is being committed), so a visual mixing term $\lambda_V \cdot a_t^{(c,\text{vis})}$ defined as "reasoner attention to cells via image patches" would have no signal to mix in: the reasoning-time query never reads from image patches in any layer. Calibrating λ_V against terminal correctness on such a feature would either drive $\lambda_V \to 0$ or fail to identify a meaningful weight at all. We therefore ground TABATTN on text-side cell attention as a principled match to the reasoner's reasoning-time computational geometry.

Scope and forward pointer. The finding is specific to autoregressive VLMs that route visual information through a residual projector (the dominant present-day design). It does *not* preclude a multimodal mixing variant of TABATTN on architectures that expose a stable visual attention signal: explicit cross-attention VLMs, or AR VLMs augmented with vision-encoder rollout / gradient-based saliency over the rendered table. We discuss this extension in Section 6.

D.4. Mask-Source Falsification: Does R_{attn} Track Grounding or Sharpness?

A reviewer might worry that R_{attn} in Equation (1) measures attention sharpness rather than grounding, because a model that concentrates attention on any small cell subset would receive a high score regardless of whether those cells are the right ones. We falsify this concern with a density-preserving mask perturbation on the 20-question WTQ pilot for which we cache per-cell attention from all four models in Table 1. Holding the cached attention $a_t^{(c)}$ fixed, we replace the human-verified mask $\mathbf{m}^*$ with three density-preserving null masks of equal cell count, namely (i) cells shuffled across the whole table, (ii) cells shuffled within rows preserving column structure, and (iii) cells shuffled within columns preserving row structure, plus a column-permuted control of equal column count. Each null is averaged over 50 random draws per question. If R_{attn} measured only sharpness, all variants would score identically because the number of unmasked cells is preserved.

Table 16 reports R_{attn} averaged across 20 questions per model, alongside paired Wilcoxon p-values comparing the human-verified mask against the strongest density-preserving null (cell-shuffled). Across all four models the human-verified mask scores significantly higher than every density-preserving null at $p < 10^{-2}$, and the advantage is largest for DiffuLLaMA ($3.12\times$) and LLaDA ($1.85\times$). This rules out the sharpness-only interpretation. The DLM advantage tracking the permutation-stability gap in Table 12 indicates the same architectural property, namely that bidirectional pre-training, drives both human-mask alignment and stability under structural perturbation.

Table 16. Mask-source falsification on the 20-question WTQ pilot. R_{attn} is averaged over 20 questions. Density-preserving nulls average 50 random draws per question. The cell-shuffle column is the most adversarial null since it preserves the exact relevant-cell count. The Wilcoxon column reports the paired p-value comparing human-verified vs. cell-shuffled scores per question (n=20). The GT/Null ratio takes the human-verified mean over the cell-shuffled mean.

Model	Human GT	Shuf-cell	Shuf-row	Shuf-col	Wilcoxon p	GT/Null
Qwen3-VL-8B (AR)	0.119	0.079	0.097	0.092	9.5×10^{-6}	$1.50\times$
ModernBERT (MLM)	0.100	0.079	0.099	0.077	8.3×10^{-3}	$1.26\times$
DiffuLLaMA (DLM)	**0.246**	0.079	0.128	0.125	1.9×10^{-5}	$\mathbf{3.12}\times$
LLaDA (DLM)	0.152	0.082	0.111	0.102	4.8×10^{-4}	$1.85\times$

We acknowledge two scope caveats. First, this falsification uses the human-verified mask rather than the DLM-emitted plan mask, so it establishes that R_{attn} tracks the correct cell set and not just attention concentration, leaving the looser plan-vs-human gap as a separate question addressed by the cross-dataset transfer result in Section B.4. Second, the analysis is run on the 20-question pilot with cached cell-level attention rather than on the full 1,600-example training pool, since per-step attention tensors are not retained at TABATTN training time. The consistency of the gap across all four models at n=20 with non-parametric p-values nonetheless lets us reject the sharpness-only interpretation.

D.5. Mask-Source Ablation: Oracle Ceiling and 20% Noise Stress Test

The bottom two rows of Table 5(b), TABATTN + Oracle and TABATTN + 20% Noise, swap the executor-derived cell mask consumed by R_{attn} (Equation (1)) for an alternative mask source while keeping the AR planner (Qwen3-VL-8B), the Qwen3-VL-8B reasoner, and the trained TABATTN verifier fixed. The aim is twofold: bound the headroom of the verifier under a perfect mask (Oracle), and quantify how that headroom degrades when the same perfect mask is partially corrupted (20% Noise).

Mask sources. The Oracle mask for each (table, question) pair is the human-verified attention standard curated in Section B.3, drawn from the dataset-specific JSONL files (e.g., `wikitq_train_200_partial_obs.jsonl` for WTQ, `hitab_test_200_partial_obs.jsonl` for HiTab). The 20% Noise mask is generated per question by independently flipping each cell of the Oracle mask with probability p_{flip}=0.20, preserving the table dimensions while corrupting roughly one in five cells. Both variants leave the rest of the pipeline unchanged: cell-attention extraction is performed by the same HuggingFace `Qwen3-VL-8B-Instruct` attention extractor under 4-bit quantization with a 10 GiB GPU cap, and the only reward component active during scoring is R_{attn} (`-reward-components attn`).

Evaluation protocol. We sweep all six accuracy benchmarks (WTQ, MMQA, InfoTabs, TabFact, TAT-QA, HiTab) under three decoding seeds (0, 17, 42) on a 20-question evaluation slice per seed, giving 60 scored episodes per (dataset, variant). Reasoning is capped at six steps with the same `f_*` tool vocabulary used elsewhere in TABALIGN. Per-dataset accuracy is computed by exact match on `final_answer` against the canonical ground truth, with a 2% relative tolerance on numeric answers, and then averaged across the three seeds. The full runner, the six oracle attention-standard JSONLs, and the aggregator that pools per-seed JSONLs into the values reported in Table 5(b) are released as a self-contained kit at `complex_reward_model/mask_ablation_kit/`.

Sample-size note. The Oracle and 20% Noise rows are computed on a smaller pool (n=60 per dataset across three seeds) than the n=200 single-pass main protocol used for TABATTN (ours) directly above them, so the per-cell binomial standard deviations in Table 9 are the closest reference for run-to-run uncertainty on these two rows. The qualitative ordering Oracle $>$ 20% Noise $>$ executor mask holds across all six datasets and three seeds.

Interpretation. Two observations support the design of TABATTN. First, the +6.4 pp Oracle headroom over the executor-derived mask establishes a finite ceiling: the R_{attn} formulation has measurable room for a stronger mask source, with the gap concentrated on schema-rich tasks (MMQA +17.3 pp, HiTab +8.2 pp) where bidirectional planning also helps most (Table 8). Second, the 20% Noise variant retains 4.4 pp of the 6.4 pp Oracle gain, indicating that the verifier remains useful under materially corrupted mask supervision. Together these results confirm two design points: R_{attn} generalizes beyond its training mask source, and the executor-derived mask used in production sits at a workable operating point relative to the Oracle ceiling.

D.6. Extended Attention Quality Figures

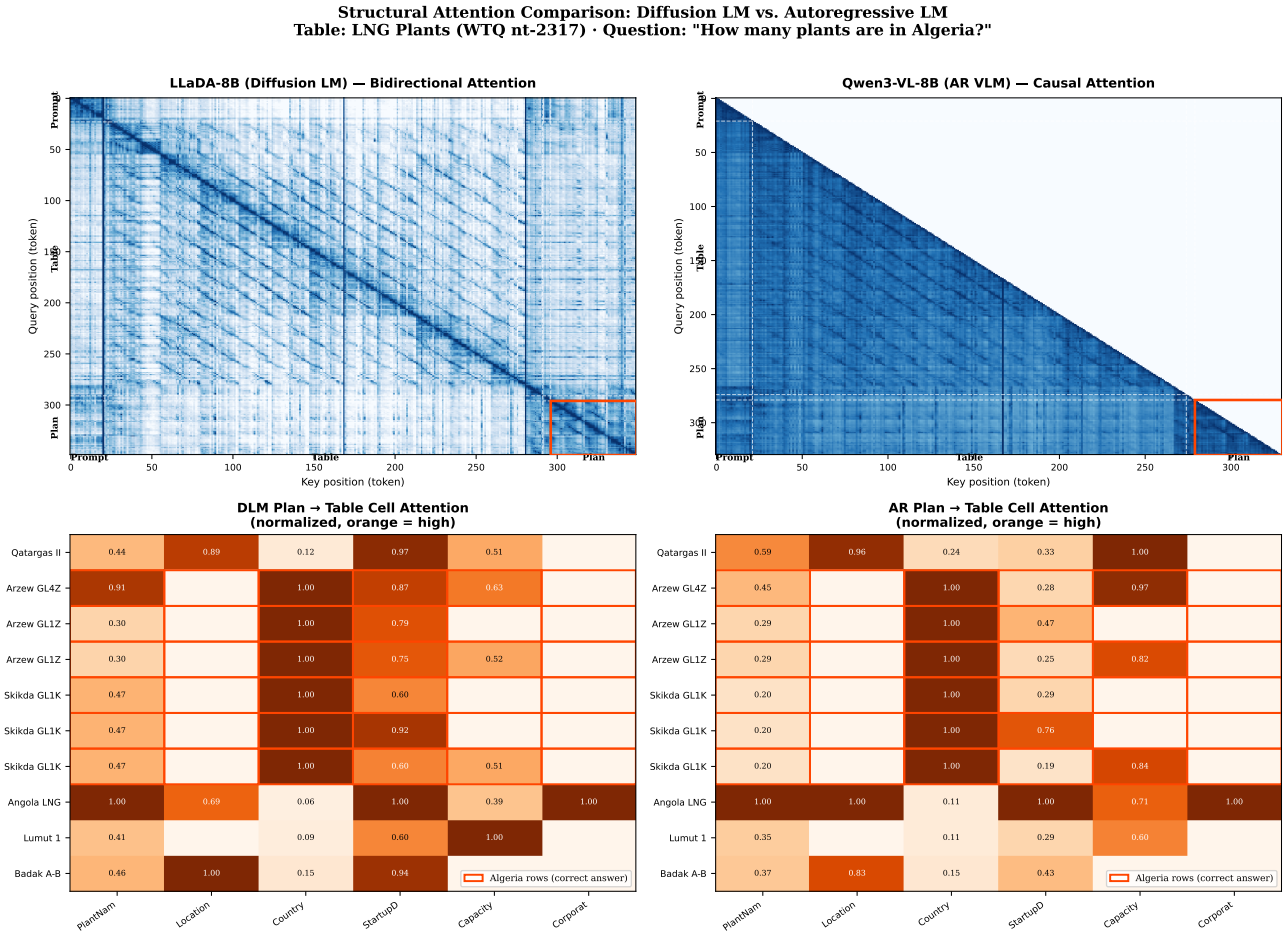


Figure 6. Full self-attention matrices for LLaDA-8B (DLM, left) and Qwen3-VL-8B (AR, right) on the same table-reasoning input. The DLM matrix is dense and bidirectional, while the AR matrix remains strictly causal. The lower panels show that both models can focus on relevant table regions, but only the DLM does so with access to future plan context.

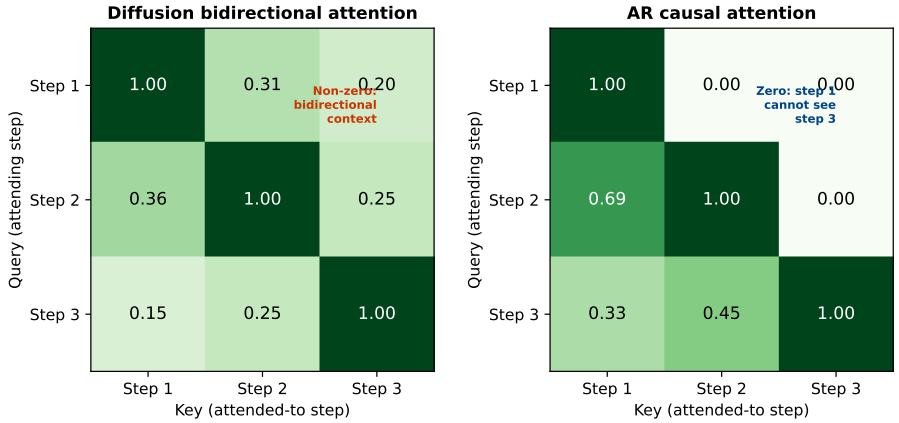


Figure 7. Within-plan step-to-step attention. DLM exhibits non-zero upper-triangular attention (for example Step 1→Step 3), confirming access to future plan positions, while AR upper-triangle entries are exactly zero under causal masking.

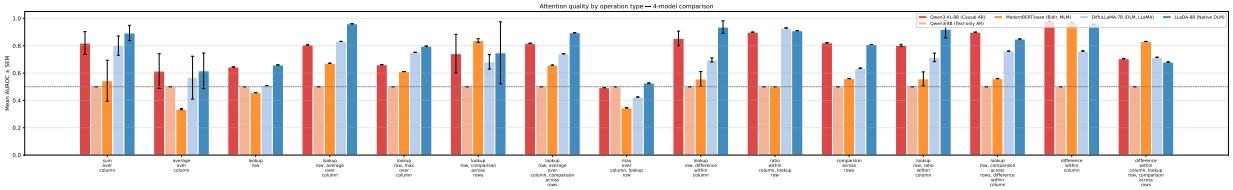


Figure 8. Per-operation breakdown for the four-model attention ablation. The diffusion advantage localizes most clearly to multi-step arithmetic, comparison, and row-grouping operations, rather than simple single-column lookups.

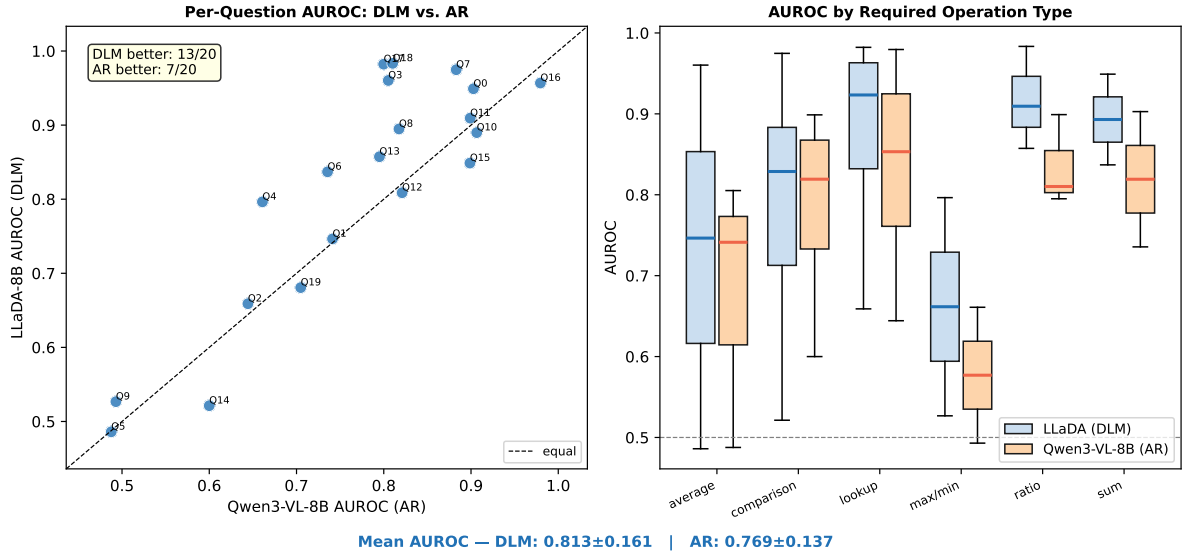


Figure 9. Summary of cell-attention quality against human relevance masks. The diffusion model wins 13 of 20 questions and achieves a higher mean AUROC, with the largest gains concentrated in multi-step comparison and ratio-style operations.

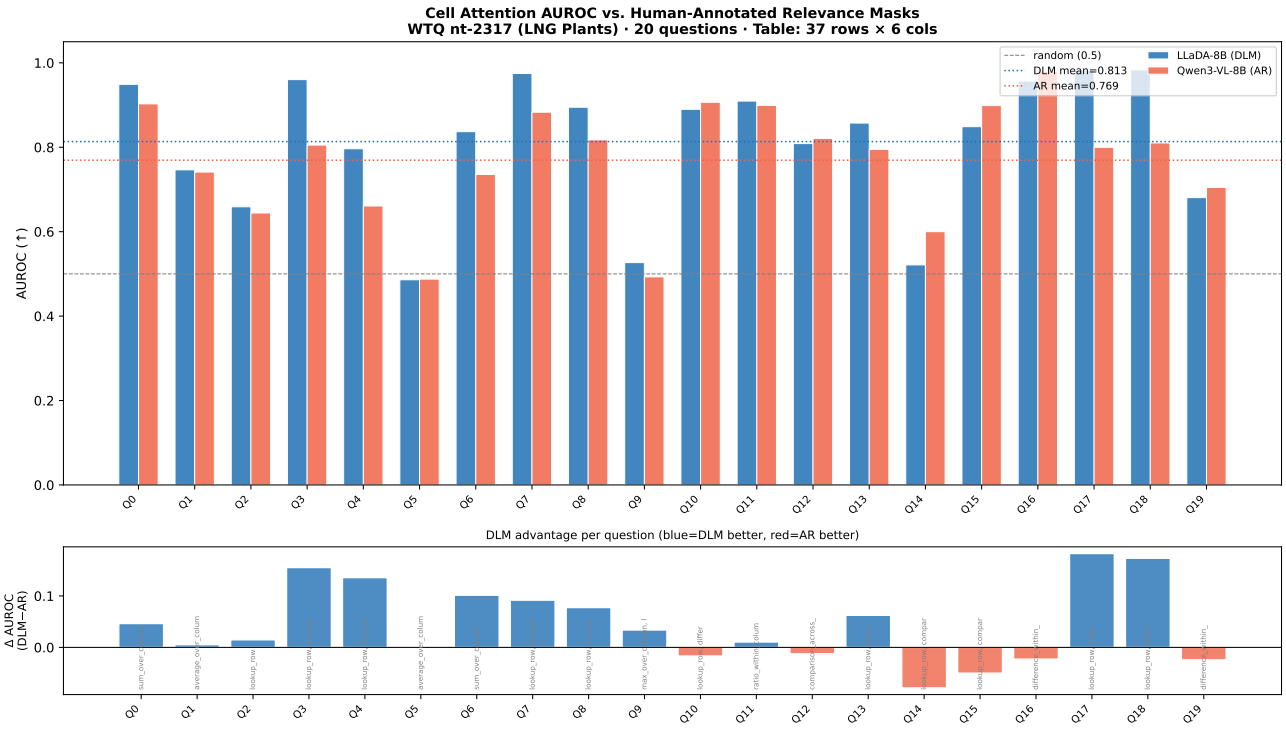


Figure 10. Per-question AUROC for LLaDA (DLM, blue) and Qwen3-VL-8B (AR, orange) on 20 human-annotated questions. DLM's advantage is largest on complex multi-step operations rather than simple single-column scans.

E. Diagnostic Analyses

This section consolidates the diagnostic evidence for *why* diffusion planning helps. We report sensitivity to table complexity (Section E.1), subgoal balance and trajectory compression, execution error modes (Section E.4), inference latency and accuracy-cost trade-offs (Section E.5), failure-mode taxonomies (Section E.6), and the human-grounded attention quality summary (Section E.8).

E.1. Sensitivity to Table Complexity

Binning by table size (Table 17), the DLM advantage is consistent across size bins on WTQ (+8.2 to +8.9 pp) and MMQA (+13.3 to +14.6 pp). Negative gaps on TAT-QA (−5.9 to −9.1 pp) confirm the advantage is tied to structural richness, not table size alone.

Table 17. Accuracy by table size quartile (smallest vs. largest bin). Δ = DLM − AR accuracy.

Dataset	Size bin (cells)	DLM	AR	Δ
WTQ	Small (27–60)	89.8%	81.6%	+8.2 pp
	Large (173–2379)	77.8%	68.9%	+8.9 pp
MMQA	Small (18–48)	68.8%	54.2%	+14.6 pp
	Large (105–242)	53.3%	40.0%	+13.3 pp
TAT-QA	Small (9–21)	52.9%	58.8%	−5.9 pp
	Large (48–126)	50.0%	59.1%	−9.1 pp
TabMWP	Small (4–8)	56.0%	60.0%	−4.0 pp
	Medium (10–14)	88.5%	75.0%	+13.5 pp

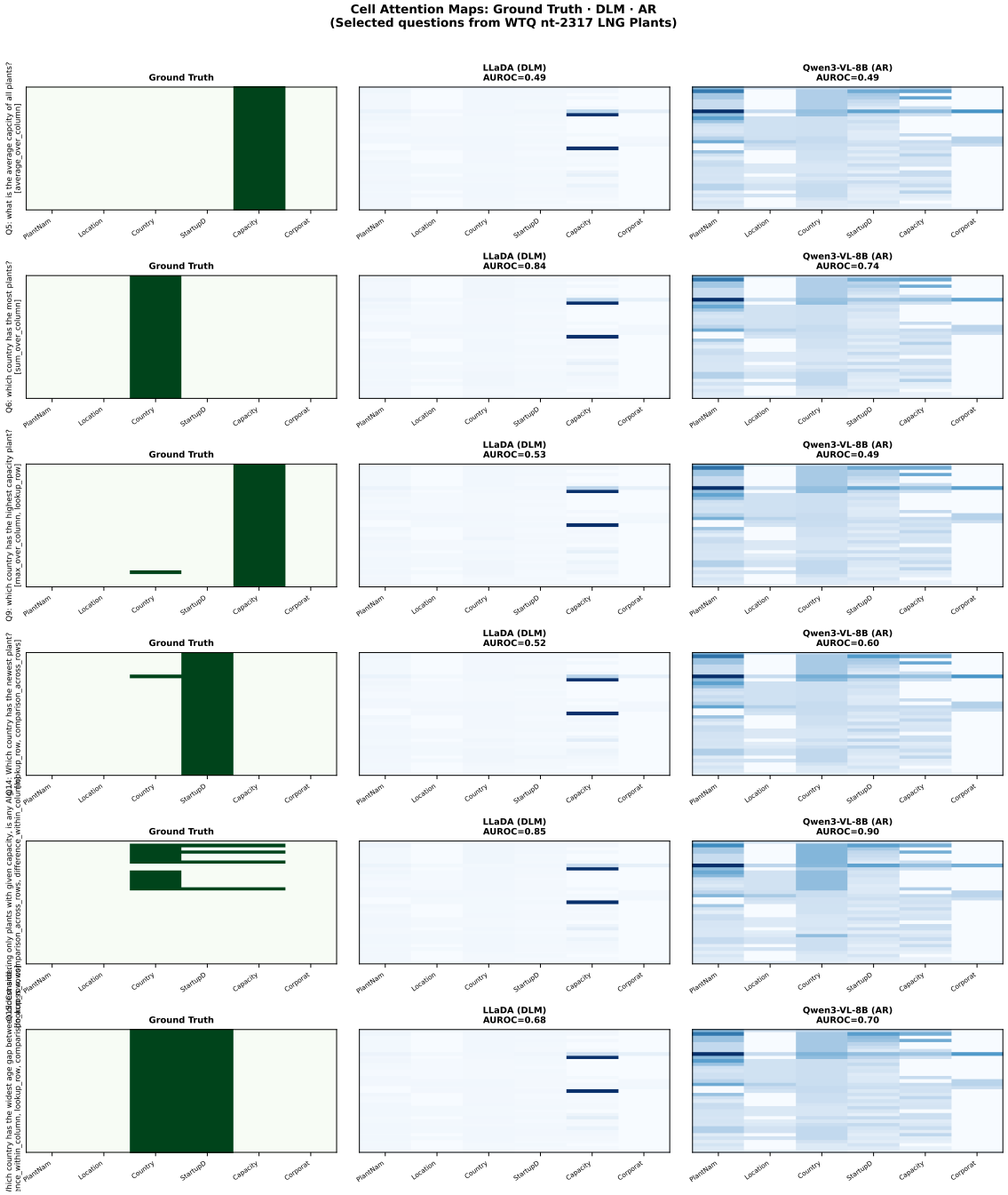


Figure 11. Representative attention maps. DLM tends to form row-level bands that group relevant entities, whereas AR tends to form column gradients with positional decay. This qualitative difference helps explain why AR remains competitive on simple lookup questions but lags on multi-row or multi-column reasoning.

E.2. Subgoal Balance Across Structurally Diverse Tables

DLM planning surfaces the ground-truth answer significantly earlier in the trajectory budget than AR reasoning on six of eight benchmarks (Mann-Whitney $p < 0.05$), with the largest gaps on HiTab ($\Delta\rho = -0.232$, $p < 10^{-10}$) and TAT-QA ($\Delta\rho = -0.164$). The two non-significant datasets (MMQA, TabFact) are shorter-plan regimes with little room for positional imbalance. Full per-dataset statistics are reported in Table 18.

Table 18. Relative answer-surface position ρ by dataset (Mann-Whitney U test). Lower ρ = ground-truth value surfaces earlier in the trajectory budget. Highlighted cells mark statistically significant DLM advantage ($p < 0.05$). n.s. $= p > 0.05$.

Dataset	ρ_{DLM}	ρ_{AR}	Δ	p-value
WTQ	0.171	0.248	-0.077	2.1×10^{-5}
MMQA	0.093	0.119	-0.026	n.s. (0.309)
InfoTabs	0.098	0.169	-0.071	0.025
TabFact	0.070	0.113	-0.043	n.s. (0.623)
TAT-QA	0.121	0.285	-0.164	1.0×10^{-6}
HiTab	0.082	0.314	-0.232	1.3×10^{-11}
FeTaQA	0.121	0.333	-0.212	9.5×10^{-10}
TabMWP	0.125	0.319	-0.194	1.9×10^{-8}

E.3. Trajectory Compression

DLM planning compresses trajectories (Wilcoxon $p < 0.05$ on four of six datasets, largest on HiTab -1.0 steps and TabMWP -0.87 steps), reducing the average matched-pair trajectory length from 2.09 to 1.63 steps (a 22% reduction), with full matched-pair statistics in Table 19.

Table 19. Trajectory compression: matched pairs, steps, accuracy, and Wilcoxon test. † $p < 0.05$, ‡ $p < 0.01$. *Fewer steps with equal-or-higher accuracy is the ideal outcome.*

Dataset	n	Mean steps		Accuracy		p-value
		DLM	AR	DLM	AR	
WTQ	179	**1.89**	2.13	**86.0%**	81.0%	$0.010^{\dagger}$
MMQA	189	**1.46**	1.60	**57.1%**	52.9%	0.051
TabMWP	189	**1.99**	2.86	**67.7%**	63.5%	$< 0.001^{\ddagger}$
TAT-QA	198	**1.58**	2.08	57.1%	**60.1%**	$< 0.001^{\ddagger}$
HiTab	142	**1.38**	2.38	2.1%	3.5%	$< 0.001^{\ddagger}$
InfoTabs	198	1.49	1.50	74.8%	**82.8%**	0.925
Average	1.63	2.09				

E.4. Execution Error Modes

DLM planning reduces runtime failures by 14.3 pp on average and increases the rate of reaching a final-answer state by $+13.2$ pp, at the cost of a modest $+3.6$ pp increase in parse errors (recoverable via fallback to direct answer generation). Full per-dataset breakdown in Table 20 and Section E.6.

E.5. Inference Latency and Accuracy-Cost Analysis

Table 5 decomposes wall-clock time into understanding and reasoning phases. On the planning side, DLM planning via LLaDA-8B 128-step denoising is substantially slower than AR planning (78.0 s/q on average vs. 6.0 s/q), with MMQA the most expensive (95.8 s/q) due to its long, multi-column tables, while shorter-schema benchmarks range from 41 s/q (InfoTabs) to 86 s/q (TabFact). On the execution side, the DLM plan's tighter, lower-hallucination structure partially compensates: DLM downstream reasoning averages 56.8 s/q versus 102.6 s/q for AR planned reasoning, a 44.64% reduction consistent with the -0.46 average step compression in Table 5.

Summing both phases, TABALIGN (134.8 s/q total) requires 24% more wall-clock time than AR Planned Reasoning (108.6 s/q) in exchange for a $+4.69$ pp accuracy gain, and is $4.7\times$ slower than unplanned AR Reasoning (40.5 s/q) for a $+9.58$ pp gain. The TabFact case is the most extreme: DLM planning alone takes 86.4 s/q versus 3.5 s/q for LLM planning

Table 20. Step-level error mode breakdown. Other error captures tool runtime failures, and final answer measures how directly the system reaches termination.

Dataset	Other error ↓		Final answer ↑		Parse error ↓	
	AR	DLM	AR	DLM	AR	DLM
WTQ	25.3%	**13.7%**	38.1%	**46.4%**	5.5%	15.4%
MMQA	25.8%	**14.4%**	53.2%	**63.0%**	3.9%	9.5%
TabFact	17.4%	**6.7%**	64.8%	**68.2%**	0.4%	5.6%
TAT-QA	34.5%	**21.7%**	39.4%	**54.6%**	6.5%	10.5%
InfoTabs	23.2%	**12.5%**	58.1%	**63.7%**	1.0%	1.0%
FeTaQA	24.5%	**13.1%**	41.4%	**59.0%**	4.2%	5.9%
TabMWP	25.9%	**7.6%**	30.6%	**45.8%**	3.6%	4.2%
HiTab	47.0%	**19.9%**	26.3%	**57.0%**	5.7%	7.2%
Avg. Δ	−14.3 pp		+13.2 pp		+3.6 pp	

(24.7× overhead), yet the total DLM pipeline (150.6 s/q) runs only 2.4× longer than the LLM pipeline (62.1 s/q) because DLM execution (64.2 s/q) is faster than LLM execution (58.6 s/q). Notably, the DLM accuracy on TabFact (90.09%) is slightly below the LLM (92.39%), making TabFact the one dataset where the overhead is not compensated by an accuracy gain. This aligns with our analysis in Section 5.2 that TabFact's holistic NLI structure provides little leverage for multi-step planning.

The DLM planning overhead is an inference-time cost amenable to standard DLM serving optimisations such as speculative decoding, batched denoising [46; 52], and flow-matching distillation [45]. We leave accelerated DLM planning as future work.

Parameter footprint at inference. TABALIGN instantiates a two-model pipeline at inference: LLaDA-8B-Instruct as the planner and Qwen3-VL-8B as the reasoner, for an aggregate footprint of approximately 16 B parameters. We position this footprint against the closest competitive baselines in Table 21. TableDART, the strongest dynamic-modality routing baseline, also runs two frozen experts plus a 2-layer MLP gate, totalling roughly 14 to 15 B depending on the image expert variant. TATTOO pairs a 14 B policy reasoner with a separate process reward model, yielding a footprint at or above ours. The single-8B framing applies cleanly to plain VLM baselines such as Qwen3-VL-8B alone, against which TABALIGN reports a +16.35 pp average gain on the six-benchmark accuracy bin (Table 4). The dual-model baselines are the more direct comparison and remain at a comparable parameter scale.

Table 21. Parameter footprint at inference for TABALIGN and the closest competitive baselines. Component sizes are taken from the corresponding model cards. TABALIGN's aggregate footprint is comparable to the dual-expert TableDART variants and below TATTOO.

Configuration	Components	Total (B)
TABALIGN (ours)	LLaDA-8B + Qwen3-VL-8B	16
TableDART (text+Ovis2)	TableGPT2-7B + Ovis2-8B + MLP gate	15
TableDART (text+Qwen2.5-VL)	TableGPT2-7B + Qwen2.5-VL-7B + MLP gate	14
TATTOO	DeepSeek-R1-Distill-Qwen-14B + PRM	≥ 14
TabTrim-8B	VLM reasoner + pruner	~ 8
Qwen3-VL-8B (single)	Qwen3-VL-8B	8

Wall-clock latency is the more meaningful normalisation than parameter count, since the planner and reasoner run sequentially rather than in parallel. Table 5 reports TABALIGN at 134.8 s/q vs. 108.6 s/q for AR Plan+Exec (+24% wall-clock for +4.69 pp accuracy) and 40.5 s/q for unplanned AR Reasoning (4.7× wall-clock for +9.58 pp accuracy).

Stepwise feedback policy and stagnation halt. TABATTN is invoked in-loop after each reasoner action and returns a scalar score together with a short rationale. The score and rationale are appended to the reasoner's prompt for the next step, so the reasoner can revise its plan on the basis of cell-grounding feedback. The pipeline does not perform an explicit re-sample of the previous step. Revision is whatever the reasoner's natural agentic loop produces in response to the in-loop feedback. A *stagnation halt* provides a backstop against unproductive iteration: when the TABATTN scalar fails to improve by more than $\delta_{\text{stag}}=0.02$ for $K_{\text{stag}}=2$ consecutive steps and the table state has not changed (hash-equality check), the pipeline emits the highest-scoring step's answer rather than continuing to spend the remaining tool-step budget. The reasoner runs at

Table 22. Representative error cases for the two attention bottlenecks (real instances from evaluation). Red = erroneous behavior, and **bold** = what the baseline missed.

Category	Model Behavior	Failure & Correction
TK1 Planning-Time Align.	*Q: "When does the tennis event end?"* (TabMWP, Summer Olympics schedule. Cols: Event, Begin, End. 4 relevant cells) AR AUROC = 0.000, with zero attention on relevant cells. Understanding model bypasses Tennis row entirely → ans: (B) 12:30 P.M. ✗ **LLaDA AUROC = 1.000**, perfect: Tennis × End targeted → **ans: (C) 1:40 P.M.** ✓	Answering requires jointly matching "tennis" in the **Event** column and "end" to the **End** column, a cross-column dependency. AR's causal left-to-right processing attends to **Begin** times (seen first) and never reaches the Tennis row's End cell. DLM's bidirectional denoising resolves both column references simultaneously, achieving perfect cell targeting (AUROC gap = +1.000).
TK2 Execution-Time Ground.	*Q: "Total Liabilities & Stockholders' Equity as reported?"* (TAT-QA, ASC 606 financial table) Step 1: `f_select_row([7, 8, 10])` → 60,513 (GT: 62,740) R_{sem}=1.000, content score ⇒ accepted. R_{attn}=0.068: attended "as adjusted" rows, **not** "as reported" rows. TABATTN: ≈0.35 ⇒ flags & resamples ⇒ **62,740** ✓	The step correctly identifies the operation type and output format, so the content reward scores 1.000, yet it **cannot detect** that the reasoner consulted the wrong column variant. $R_{\text{attn}} = 0.068$ reveals near-zero overlap between attended cells and the plan-designated "as reported" cells. TABATTN downgrades the step score to ≈0.35, triggering a resample that retrieves the correct rows.

most $T_{\max}$=6 tool steps, which caps the maximum compute per query independently of the stagnation rule.

E.6. Failure Mode Analyses

Failure mode taxonomy. LLaDA-8B end-to-end reasoning (n=579 wrong episodes) exhibits five recurring failure modes traceable to the fixed-canvas constraint: duplicate action generation (56%), output format and parsing errors (51%), canvas overflow (17%), empty answer (9%), and over-enumeration (7%), with the full taxonomy in Table 6. TABALIGN failures (n=507) are qualitatively different: the dominant mode is plan bypass (67%, where the reasoner jumps to `f_final_answer` on step 1), with secondary modes of hierarchical column navigation errors on HiTab (25%) and entailment-direction flips on classification datasets (91 to 100%). Full per-dataset breakdown in Table 23. The understanding-stage ablation (Table 5), schema hallucination rates (Table 8), and subgoal-balance statistics (Table 18) together confirm that gains are attributable to the DLM plan rather than solely to TABATTN trajectory selection: plan bypass is concentrated in *wrong* episodes, and correct episodes that follow the plan show significantly lower hallucination and earlier answer-surfacing.

Table 23. TABALIGN failure mode rates per dataset (% of wrong episodes, where multiple modes can co-occur). *Plan bypass*: reasoner answers directly without executing plan steps. *Hier. nav.*: runtime error when resolving hierarchical `type|subtype` columns (HiTab-specific). *Agg. confusion*: numeric answer <35% from ground truth but not exact-match. *Entail. flip*: correct label format, wrong entailment class.

Dataset	Acc.	Plan bypass	Hier. nav.	Agg. confusion	Entail. flip
MMQA	71.07%	60%	—	—	—
HiTab	82.48%	72%	25%	2%	—
TabFact	90.09%	59%	—	—	91%
InfoTabs	90.45%	74%	—	—	100%
TAT-QA	91.41%	73%	—	11%	—
WTQ	94.12%	68%	—	—	—
TabMWP	95.13%	59%	—	1%	—
All		67%	9%	3%	8%

The dominant TABALIGN failure is plan bypass (67%): the reasoner produces a direct `f_final_answer` on step 1, ignoring a DLM plan with an average of 5.7 steps. This shortcut is epistemically unreliable and fails for questions requiring multi-step hierarchical column navigation. Plan bypass is concentrated in wrong episodes, and correct episodes that

follow the plan show significantly lower schema hallucination and earlier answer-surfacing, confirming the plan adds genuine value when followed. Two actionable remediation directions follow: (1) plan-compliance enforcement preventing `f_final_answer` before the plan is exhausted, and (2) hierarchical schema bridging preprocessing HiTab column names.

Each failure mode corresponds directly to an architectural property that motivated the plan/execute separation: fixed canvas length, non-causal generation, weaker instruction alignment, and absence of reactive error correction.

E.7. Labelability Experiment: LLM-as-a-Judge Setup

This subsection details the experimental setup behind the pooled labelability numbers in Table 3 and the per-dataset breakdown in Table 25. The goal is to compare step-correctness and cell-relevance under a matched annotation budget by running an identical LLM-as-a-judge pipeline on both sides at each side's natural decision unit, following Krishna et al. [22] and Bologna et al. [7].

Decision unit and pool construction. Each side is evaluated at the granularity at which its supervision target is naturally defined. The step side issues one label per reasoning step in a tool-using trajectory, where each step bundles a reasoning rationale, a tool action, and the resulting tool output. The cell side issues one label per visible table cell within a partial-observation window, asking whether that specific cell is necessary to answer the question. The step pool comprises 1,548 reasoning steps drawn from TABALIGN trajectories on the eight benchmarks, with steps stratified across datasets to match the cell pool's coverage. The cell pool comprises 76,157 per-cell decisions pooled across the same eight benchmarks, derived from the 1,600 partial-observation windows that anchor the human-verified attention standards (Section B.3).

Identical LLM-as-a-judge pipeline. Both pipelines use the same LLM provider (GPT-5 via the OpenAI API), the same JSON-only response format, the same temperature, the same maximum-retry policy, and the same per-record context budget. Prompts are structurally parallel. The cell-relevance prompt presents the question and a markdown-rendered table window, then asks the model to return a list of `[row_index, col_index]` relevant cells with a one-sentence rationale. The step-correctness prompt presents the question, the ground-truth answer, the prior-step summary, and the current step's reasoning text, action, and output, then asks the model to return one of {`yes`, `no`, `unsure`} with a one-sentence rationale. The prompts share the same instruction style, the same JSON schema, and the same single-sentence rationale field, so any agreement gap between the two sides is attributable to the labeling task rather than to differences in prompt engineering. The full prompt texts are reproduced verbatim in our released code as `llm_judge_cell_relevance.py` and `llm_judge_step_correctness.py`.

Human reference labels. The cell side uses the human-verified attention masks underlying the 1,600 standards as gold labels, with majority-vote disagreement resolution and an inter-annotator IoU threshold of 0.90 on a held-out verification subset (Section B.3). The step side uses a smaller human-labeled pool of 360 trajectory steps annotated with 3-way labels in $\{0, 0.5, 1\}$ corresponding to {`no`, `unsure`, `yes`}. The 3-way scheme is mandated by the structure of the task. 46.9% of the human step labels were marked `unsure`, with annotators citing truncated tool outputs, ambiguous step intent, and syntax-only errors whose underlying intent might still be correct as the dominant reasons for abstention. The cell labelers were not given an `unsure` option because the cell-relevance question (*is this cell necessary to answer the question*) admits a confident binary answer once the table window and question are visible, an asymmetry the experiment is designed to expose. We recruited cell annotators and step annotators from the same pool of trained graduate students, and each annotator labeled both sides on a held-out subset to control for annotator-specific behavior.

Agreement metrics and reporting protocol. For both sides we report per-decision agreement (%) and Cohen's κ, computed at the corresponding decision unit. Cell-side κ is computed across all 76,157 binary cell decisions pooled over datasets. Step-side agreement is computed only on the human-confident subset, that is, the 190 steps for which the human label is not `unsure`, since including human-`unsure` steps would inflate apparent disagreement without reflecting genuine labeling difficulty. A step-correctness classifier trained on the full 1,548 LLM-judged step labels reaches held-out F1 = 0.77 and AUROC = 0.88, capped by the noisy supervision signal it inherits, and we report this as a downstream consequence of the upstream labelability ceiling rather than as a separate experiment. Per-dataset agreement and κ for both sides are listed in Table 25, with InfoTabs and TabMWP serving as the two regimes where step labels are easier than cell labels (short-form classification and template-aligned arithmetic respectively).

What the comparison establishes. Holding the LLM, the prompt format, the temperature, and the human reference protocol constant, the only manipulated variable is the decision unit (one binary per step versus one binary per cell). The 19.8 pp pooled agreement gap and the 0.20 κ gap therefore isolate decision-unit difficulty rather than confounding it with prompt design or LLM choice. On six of eight benchmarks the cell side wins, with the two losses on InfoTabs and TabMWP both consistent with their reduced step-label evaluative complexity. The aggregated finding is that cell-relevance is more labelable than step-correctness on average and on most regimes, which is the precondition that makes the curated 1,600 attention standards a reliable training target for the TABATTN verifier in Section 4.

E.8. Human-Grounded Attention Quality

Across 20 human-annotated questions, LLaDA reaches mean AUROC 0.813 vs. 0.769 for Qwen3-VL-8B (13/20 wins, directional evidence).[2] The large-scale four-model ablation and per-dataset AUROC breakdown (Table 24, 1,600 questions, 8 benchmarks) are reported in Section 3 as part of the pilot study motivating TABALIGN, and the simplified summary (Overall and σ_{AUROC} only) appears as Table 1 in the main text. Per-question breakdowns, operation-level heatmaps, and

Table 24. Per-dataset attention quality (AUROC) and permutation stability (σ_{AUROC} over $K{=}5$ row permutations) across eight benchmarks. Masked bidirectional attention improves overall attention quality, and permutation stability if such bidirectionality is achieved via pre-training.

Model	Bidir.	Diff.	WTQ	TAT-QA	HiTab	TabMWP	MMQA	InfoT	TFact	FeTaQA	Overall ↑	σ_{AUROC} ↓
ModernBERT (149M)	✓	✗	0.548	0.526	0.605	0.595	0.566	0.581	0.518	0.545	0.569	**0.046**
Qwen3-VL-8B	✗	✗	0.644	0.583	0.667	0.662	0.636	0.604	0.618	0.616	0.628	0.123
Qwen3-8B	✗	✗	0.639	0.577	0.676	0.658	0.616	0.611	0.609	0.574	0.623	0.138
DiffuLLaMA-7B	✓	✓	0.635	**0.584**	0.632	0.616	0.705	0.705	0.591	0.644	0.639	0.123
LLaDA-8B	✓	✓	**0.673**	0.544	**0.738**	**0.686**	**0.752**	**0.717**	**0.666**	**0.671**	**0.677**	0.074

Table 25. Per-dataset labelability of step-correctness vs cell-relevance under identical LLM-as-a-judge pipelines. Cell-level labels achieve materially higher per-decision agreement and κ across most datasets.

Metric	Method	WTQ	MMQA	InfoT	TFact	TAT	HiTab	TabMWP	FeTaQA	Pooled
Agree (%)	Step	53.3	63.6	91.3	73.1	75.0	57.1	84.0	91.3	73.2
	Cell	**96.4**	**92.5**	80.7	**93.1**	**92.9**	**94.9**	78.1	**93.6**	**93.0**
	Δ	+43.1	+28.9	−10.6	+20.0	+17.9	+37.8	−5.9	+2.3	**+19.8**
κ	Step	0.22	0.36	0.75	0.47	0.50	0.27	0.68	0.70	0.48
	Cell	**0.83**	**0.55**	0.42	**0.70**	**0.69**	**0.62**	0.56	**0.73**	**0.68**
	Δ	+0.61	+0.19	−0.33	+0.23	+0.19	+0.35	−0.12	+0.03	**+0.20**

attention maps are in Figures 8 to 11.

F. Future Work and Outlook

This section consolidates the future-work directions raised throughout the paper. We organize them into three groups: framework-internal extensions that respond to the limitations of the current TABALIGN system (Section F.1), scaling directions for the supervision signal (Section F.2), and broader generalization beyond table reasoning (Section F.3). Each direction preserves the cell-grounding contract introduced in Section 4 and reuses the planner-reasoner-verifier interface without architectural rework.

[2]This 20-question subset (WTQ nt-2317) yields a higher AR AUROC than the large-scale average (Table 1) due to the specific question distribution, with the per-question breakdown in Figure 10.

F.1. Framework-Internal Extensions

Plan-adherence reward shaping. The dominant remaining failure mode of TABALIGN is plan bypass: the AR reasoner emits `f_final_answer` at step 1 and ignores the DLM plan in 67% of error episodes (Table 23), mostly on single-cell lookups where a one-step shortcut is cheap. Plan-following episodes show lower schema hallucination (Table 8) and shorter trajectories (Table 5(a)), so the plan carries useful signal that the current TABATTN reward does not enforce. A natural extension is to add an adherence term to TABATTN that penalizes step-1 termination when the plan contains more than one step, or that rewards trajectories whose tool-call sequence matches the plan's tool sequence. Because adherence and cell-overlap factor across orthogonal axes (which cells, in which order), the two terms can be combined as a probability-simplex mixture analogous to Equation (1) without disturbing the calibration on the existing 1,600 standards.

Accelerated DLM planning. DLM planning is currently 13× slower than AR planning per query on average (78.0 s/q vs. 6.0 s/q, Table 5(a)), partially offset by 44.64% faster downstream execution but leaving TABALIGN at 24% higher total wall-clock time than AR Planned Reasoning. Three serving optimizations are directly compatible with the current framework. First, batched denoising [46; 52] amortizes per-step cost across queries that share a canvas length. Second, flow-matching distillation [45] compresses multi-step denoising into a small number of large steps. Third, speculative decoding adapted to the masked-token setting can pre-fill high-confidence positions before the full denoising sweep. All three preserve the planner-reasoner contract and the cell-mask interface, so accuracy gains carry over without retraining TABATTN.

Free-form answer generation. TABALIGN trails TableLlama-7B in BLEU on FeTaQA (32.63 vs. 38.38), driven by BLEU penalizing correct paraphrases that diverge from the reference surface form rather than by a reasoning gap (78.6% LLM-judge factual correctness, Section A.1). A FeTaQA-style generation head fine-tuned on the cell-grounded reasoner output, or a learned answer-rewriter conditioned on TABATTN's accepted final state, would close most of the BLEU gap while keeping the cell-grounding contract intact. The annotation cost is modest because reference paraphrases already exist for FeTaQA training data.

F.2. Scaling the Supervision Signal

Larger and more diverse attention standards. The 1,600 human-verified attention standards span eight benchmarks (Section B.3), with smaller per-benchmark counts on InfoTabs and TabMWP where step-correctness is the easier label (Table 25). Scaling the standard set, prioritizing benchmarks where cell-relevance and step-correctness diverge, would improve TABATTN's cross-dataset generalization and reduce variance on under-represented regimes. The labelability advantage of cell-relevance over step-correctness (Table 3) makes this scaling cheap relative to a step-correctness annotation effort of comparable size, so the marginal cost per additional decision is low.

Causal grounding beyond attention overlap. The pilot study and permutation-stability analysis (Section D.2) provide empirical evidence that bidirectional pre-training improves cell-relevance alignment. Attention overlap with the plan-designated mask serves as a labelable proxy for grounded reasoning, and a counterfactual-intervention study would convert this proxy claim into a causal one. Concretely, removing or substituting plan-designated cells and measuring the resulting change in answer correctness would quantify how much of the TABALIGN accuracy gain depends on attention concentrating on those cells, and how much follows from the plan's tool-sequence prior alone. Such a study is compatible with the current TABALIGN pipeline through a synthetic-perturbation evaluation harness on top of the existing eight benchmarks.

Cross-lingual and semi-structured tables. The current evaluation scopes to English structured-table benchmarks at 8B-class scale. Two extensions follow naturally from the cell-grounding contract. Cross-lingual benchmarks (Chinese, Japanese, multilingual TableQA) test whether the binary cell mask transfers across tokenizer-level changes in row serialization. Semi-structured settings such as forms, partially merged cells, and prose-embedded tables (closer to TAT-QA, where Table 8 already records the smallest DLM-vs-AR gap) test whether the mask remains a useful supervision target when cell boundaries are fuzzy. Both extensions reuse the existing planner, reasoner, and TABATTN verifier with new data rather than new method components.

F.3. Broader Generalization

TABATTN grounds verification in internal attention weights, making TABALIGN directly applicable to open-weight reasoners and complementary to closed-source API systems whose attention is structurally inaccessible. The cell-grounding contract carries beyond tables. Any structured modality whose evidence localizes to discrete units admits the same labelable

mask-and-verify supervision: code lines for code-reasoning agents, knowledge-graph triples for KG question answering, and structured form fields for document-AI workflows. Continued progress in DLM instruction-tuning and inference acceleration [46; 52; 45] is expected to lift plan quality on these modalities while leaving the framework interface unchanged. We see attention-grounded process supervision over discrete-unit modalities as the natural broader-impact path for the contract introduced in this work.

G. Compute Reporting

We report the compute resources used for this work, complementing the paper checklist and following the spirit of the CVPR 2026 Compute Reporting initiative (`https://cypr.thecvf.com/Conferences/2026/ComputeReporting`). The information below is intended to support reproducibility and is not used in the review process.

Hardware. All experiments were conducted on a single NVIDIA RTX 4090 (24 GB) workstation GPU. No multi-GPU training, multi-node communication, or accelerator beyond a single consumer-class card was used.

Serving stack and quantization. Three model processes share the 24 GB envelope of the single GPU. The Qwen3-VL-8B reasoner is served via **Ollama** (`qwen3-vl:8b`, GGUF Q4 quantization) at roughly 7.7 GB VRAM, and is reached through an OpenAI-compatible local endpoint by the trajectory collector. The Qwen3-VL-8B-Instruct attention extractor used to compute R_{attn} runs concurrently as a HuggingFace process under 4-bit NF4 `bitsandbytes` quantization with double-quant and `attn_implementation="eager"`, capped at 10 GiB VRAM, occupying a further $\sim$3.5 GB at steady state (Sections D.3 and D.5). The LLaDA-8B-Instruct planner is loaded as a separate HuggingFace process with 4-bit `bitsandbytes` weight quantization (default `load_in_4bit=True` in `diffusion_planner.py`), and is invoked sequentially rather than concurrently with the reasoner so that planning compute does not contend with the reasoner-plus-extractor co-residency budget. Steady-state co-resident footprint during reasoning is approximately 11.2 GB / 24 GB.

Foundation-model pretraining. TABALIGN reuses publicly released foundation-model checkpoints and contributes *no* pretraining compute. LLaDA-8B-Instruct [38] and Qwen3-VL-8B [3] are loaded as released and run frozen at inference. The full pretraining cost is borne by the original authors and is not attributable to this work.

TABATTN verifier training. TABATTN consists of a two-parameter logistic calibration on the scalar overlap reward R_{attn} (Equation (1)), fit by binary cross-entropy on the 1,600 curated attention standards with an 80/20 early-stopping split (Sections A and B.3). The training step is a single logistic fit on a CPU-resident scalar dataset and adds negligible GPU time relative to the inference cost. The dominant cost upstream of this fit is the per-step attention extraction needed to compute R_{attn} for each of the 1,600 standards, which reuses the same 4-bit Qwen3-VL-8B inference pipeline used at evaluation time.

Inference cost. Per-query wall-clock latency is the most informative cost measure under our setup, since the planner and reasoner run sequentially rather than in parallel and the policy is one trajectory per query (N=1, no outer best-of-N search). Detailed per-benchmark numbers are reported in Table 5(a) and Section E.5, with summary points reused below for completeness. TABALIGN averages 134.8 s/q across the six accuracy benchmarks (78.0 s/q for DLM planning, 56.8 s/q for AR-reasoner execution under TABATTN feedback). On the same hardware, AR Planned Reasoning averages 108.6 s/q, end-to-end AR Reasoning averages 40.5 s/q, and end-to-end DLM Reasoning averages 21.2 s/q on the five datasets where it does not collapse. Per-benchmark range spans 65.6 s/q (InfoTabs) to 181.6 s/q (MMQA), driven by table width and column count rather than question count.

Compute budget for the reported tables. Each accuracy entry in Tables 4 and 5 is computed over n=200 evaluation questions per dataset under deterministic decoding. The Oracle and 20% Noise rows in Table 5(b) use a smaller n=60 pool across three seeds (Section D.5). Per-cell volatility under this protocol is reported in Table 9. Latency-driven aggregate compute is therefore dominated by the main and ablation runs, with diagnostic analyses (permutation stability, mask-source falsification, latency profiling) adding modest overhead on top.

Failed and exploratory runs. Beyond the experiments reported in this paper, additional compute was spent on early prototyping of the planner-verifier interface, on alternative reward formulations (cosine-similarity, numerical-F1, TABROUGE) before settling on R_{attn}, and on attention-extraction pilots across the four backbones in Table 1. These runs reused the same

single-RTX-4090 environment described above and contributed to the framework design but are not separately enumerated.

H. Declaration of LLM Usage

We disclose all uses of large language models in this work, complementing the checklist entry (item 14).

LLMs as core framework components. Two LLMs serve as the load-bearing modules of TABALIGN. LLaDA-8B-Instruct [38] is the masked diffusion planner that emits the cell-grounded tool-use blueprint (Sections B.2 and 4.1). Qwen3-VL-8B [3] is the multimodal autoregressive reasoner that executes the plan and supplies the per-cell attention scored by TABATTN (Sections 4.2 and D.3). Decoding hyperparameters, canvas length, denoising step count, and tool budgets are documented in Sections A and B.2.

LLMs in supervision and evaluation. GPT-5 is used as the bootstrap annotator that scales 100 human seed cell-relevance maps to the 1,600-example training set for TABATTN, with all generated examples reviewed by human annotators against an IoU ≥ 0.70 acceptance threshold (Section B.3). The model-selection comparison across five GPT variants on the 20-example WTQ calibration split is reported in Table 10. GPT-5-nano is used as the LLM-judge for the FeTaQA semantic-correctness analysis (Section A.1), and an LLM-as-a-judge pipeline supplies the cell-relevance and step-correctness labelability comparison in Tables 3 and 25.

LLMs as evaluation baselines. Several LLMs and VLMs (Llama-2-7B, Table-LLaVA-7B, MiniCPM-V, Qwen2.5-VL, Gemini 2.0 Flash, Qwen3-VL-8B, and others, Section A) are evaluated as baselines or comparison systems. Proprietary models appear for context only since TABATTN requires open attention weights.

LLMs for writing polish. We additionally used LLMs (GPT-5.5 and Claude-Opus-4.7) to polish the writing of this manuscript, including light copy-editing, sentence-level rephrasing, and consistency checks. All technical claims, experimental design, results, and conclusions are the work of the authors. Every LLM-suggested edit was reviewed and accepted, modified, or rejected by the authors. No LLM contributed to the methodology, the experimental design, the empirical results, or the literature analysis.