When LLMs Read Tables Carelessly: Measuring and Reducing Data Referencing Errors

Yuqing Yang[1*], Qi Zhu[2†], Zhen Han[2], Boran Han[2], Zhengyuan Shen[2], Shuai Wang[2], Vassilis N. Ioannidis[2], Huzefa Rangwala[2]

[1]University of Southern California, [2]AWS AI Labs

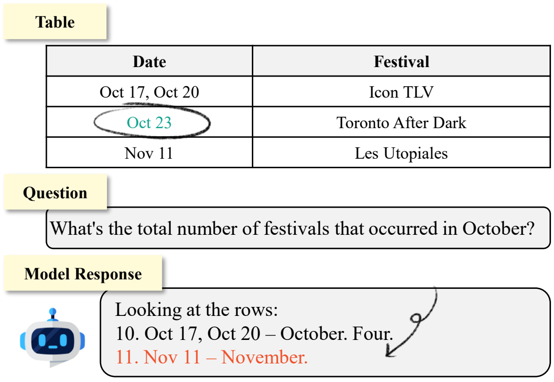
Abstract

While large language models (LLMs) perform well on table tasks, they still make data referencing errors (DREs), *i.e.*, incorrectly citing or omitting table values, despite understanding the table structure. Beyond final-answer accuracy, DREs directly compromise the correctness and reliability of intermediate reasoning steps. Yet prior studies have only offered limited, small-scale analyses. In this work, we present the first systematic evaluation of tabular data referencing errors across different models and tasks. Our results show that DREs occur across all tested models (1.7B to 20B parameters). Furthermore, we demonstrate that incorporating data referencing as a critic significantly improves answer accuracy up to 12.0%, through critic-based filtering and rejection sampling. Finally, we trained a lightweight 4B-parameter critic model that achieves an average F1 score of 78.2% in detecting both in-distribution and out-of-distribution DREs, and effectively assists inference for larger models.

1 Introduction

Tables are one of the most common ways to represent information, providing a structured format for organizing data. They are widely used across real-world domains such as finance (Chen et al., 2021), healthcare (Yan et al., 2025), and scientific reporting (Moosavi et al., 2021; Zhang et al., 2025b), making the ability to effectively perform tasks over tabular data essential. Solving table-related tasks requires several capabilities: understanding tables presented in textual formats, accurately locating and citing relevant values, and reasoning over critical values to derive correct answers. Large Language Models (LLMs) are increasingly applied to these tasks and often achieve strong performance (Yang et al., 2025b; Wu et al., 2025b; Lei et al., 2025), yet they still commit surprisingly basic mistakes even when the table format is correctly parsed, by referencing table content incorrectly, as illustrated in Figure 1.

(a) Incorrect Citation: Model confuses the "Organization" column with the "Award" column.



(b) Omitted Information: Model omits the row of "Oct 23".

Figure 1: Illustration of Tabular DREs.

A primary source of such errors is the dense and similar structure of tables, which makes it hard for models to reliably locate and cite values. For example, answering "Which country had the highest GDP growth between 2020 and 2022?" requires aligning multiple year columns across rows, where, analogous to human oversight, a slip can lead to mistakes. We refer to such failures to faithfully retrieve and cite information from the input as **Data Referencing Errors** (DREs). These errors can degrade response quality and sometimes

*Work done during an internship at Amazon. Email: yyang063@usc.edu.

†Corresponding author. Email: qzhuamzn@amazon.com.

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 16734–16752

final-answer accuracy, yet they are not fully captured by final accuracy metrics alone. Although prior work (Zhang et al., 2025c; Cao, 2025) has observed DREs, analyses remain narrow, typically limited to a single model and a small set of human-annotated cases. In this work, we systematically investigate how prevalent DREs are, how they can be effectively mitigated, and how mitigating them influences final-answer accuracy.

We first categorize tabular DREs into two types: **Incorrect Citation**, involving individual values, and **Omitted Information**, involving entire relevant portions, as illustrated in Figure 1. We then employ LLM-as-a-Judge framework (Zheng et al., 2023) to automatically detect DREs given a table and a generation model's response. Upon evaluation, We find that DREs are ubiquitous across different models (from 1.7B to 20B parameters) and across diverse table-related tasks (including Question Answering, Claim Verification, and Table-to-Text). They are not effectively eliminated by either reasoning models' self-reflection mechanisms (Snell et al., 2024; Muennighoff et al., 2025; Ye et al., 2025) or by prompting-based approaches. For instance, Qwen3-8B (Yang et al., 2025a), even with extended self-reflection, exhibits a 14.04% DRE rate (i.e., the proportion of responses containing DREs) on the WTQ (Pasupat and Liang, 2015) dataset, and still 12.50% when further prompted not to miscite or omit table content.

Incorporating DRE detection as a critic not only improves the quality of intermediate reasoning steps beyond what is captured by final-answer accuracy rewards, but also noticeably enhances overall performance. We explore two approaches. First, **critic-based filtering**, which selects the subset of sampled responses with minimal DREs, yields substantially higher accuracy than using all sampled responses, and can further enhance majority voting when combined. Second, **rejection sampling**, which repeatedly resamples response segments until the critic accepts it, obtains consistent gains and can improve accuracy by up to 11.96%. Notably, DREs are largely avoidable rather than fundamental limitations, yet rejection sampling with the critic offers a more robust way to reduce their occurrence.

Finally, given the high cost and black-box nature of using larger models as critics, we investigate the potential of small-scale LLMs (*i.e.*, Qwen3-4B-Instruct, Yang et al., 2025a) for detecting DREs. To this end, we construct training data from Qwen3-8B responses on the WTQ training set and adopt a two-stage training procedure: supervised fine-tuning for warm-up, followed by RLVR (reinforcement learning with verified reward, Lambert et al., 2024; DeepSeek-AI et al., 2025) to enhance robustness in DRE detection without constraining the Chain-of-Thought (CoT) format. Our experiments show that the trained Critic-4B consistently outperforms the untrained baseline both in-distribution and out-of-distribution, achieving an average improvement of 8.65% F1. Moreover, we demonstrate that this lightweight critic can mitigate DREs across different models and improve final accuracy more effectively than prompting-based methods.[1]

Our work shed light on the overlooked issue of data referencing errors, a unique error pattern that, while particularly common in table-related tasks, also appears in other domains. Although these broader cases are beyond the scope of this paper, we aim to inspire follow-up work to improve both critic and generation models by enhancing their data referencing capabilities.

2 Related Work

Table LLMs Solving tabular tasks using LMs has been a long-standing research topic. Early work relied on table pre-training with specialized architectures, such as TaPas (Herzig et al., 2020) and TaBERT (Yin et al., 2020). With the scaling of general-purpose LLMs, recent methods adapt them to tabular settings through prompting (Ye et al., 2023; Jiang et al., 2023), supervised fine-tuning (Zhang et al., 2024; Su et al., 2024; Zhang et al., 2025a), and reinforcement learning (Yang et al., 2025b; Wu et al., 2025b; Lei et al., 2025). As LLMs become increasingly powerful, particularly with the emergence of reasoning models that demonstrate strong problem-solving capabilities through extended thinking processes (DeepSeek-AI et al., 2025; Yang et al., 2025a; OpenAI, 2025), they can already exhibit strong baseline performance on table tasks without task-specific training, as evidenced by Yang et al. (2025b); Wu et al. (2025b) and our experiments in Table 1. These developments call for moving beyond final-answer accuracy toward more fine-grained limitations.

Evaluation Beyond Accuracy Most evaluation benchmarks emphasize final accuracy for simplicity, overlooking the quality of intermediate reasoning. To address this, prior work has proposed

[1]Our code is available at `https://github.com/ayyyq/table-referencing`.

process-level reward models (PRMs, Lightman et al., 2024; Zhang et al., 2025d), which evaluate reasoning steps rather than only outcomes. Related efforts further decompose evaluation into dimensions: e.g., validity and redundancy in mathematical reasoning (Xia et al., 2025), instruction-following and truthfulness in alignment (Cui et al., 2024), and relevance and completeness in long-form QA (Wu et al., 2023). In table reasoning, however, benchmarks still focus almost exclusively on final correctness (Wu et al., 2025a; Pasupat and Liang, 2015). We introduce data referencing errors as a complementary dimension that captures how reliably models use table values, reflecting both intermediate reasoning quality and final performance.

Existing Work on DREs Table-related tasks, especially Table QA, require models to use table values both completely and accurately. Prior work has recognized this need. For example, Zhang et al. (2025c) analyzed 50 WTQ samples from Distill-Llama-8B (DeepSeek-AI et al., 2025) and found that more than 80% of errors came from incorrect locating and citation. Yet such studies do not systematically characterize DREs. Other work (Wu et al., 2025b; Lei et al., 2025) introduces auxiliary rewards to improve table referencing, but relies on supervised fine-tuning with annotated table regions. For instance, models are trained to generate special tags such as `<|cell content|><|column name|>` when using the specific table values *needed* to answer a question. In contrast, we evaluate models' overall accuracy in referencing *any* table values and analyze how DREs affect performance. Our framework further supports critic-based detection that can be seamlessly integrated into existing LLMs, improving both response quality and final-answer accuracy without requiring special annotations or disrupting reasoning chains (Tang et al., 2025).

3 Characterizing DREs

3.1 Definition and Taxonomy

When answering a table-based question, LLMs are generally required to comprehend the table structure (e.g., distinguish between rows, understand column headers, and interpret each cell's meaning within its row-column context), use table values to support reasoning, and reason over critical ones to derive the correct answer.

Recent LLMs overcome long-standing challenges in tabular data by handling diverse textual formats with large-scale pre-training (Touvron et al., 2023) and enhancing logical and numerical reasoning through targeted post-training (Liu et al., 2025; Wang et al., 2025). However, our analysis reveals that these models, especially smaller ones, still make basic mistakes, which resemble human oversights that could have been avoided with careful attention. As shown in Figure 1, the model confuses columns or overlooks an entire row. In CoT responses, we define **data referencing** as the ability to correctly locate and cite information from inputs. Accordingly, errors or hallucinations in this process constitute data referencing errors (DREs).

While DREs can be found in different domains and modalities (Mirzadeh et al., 2025; Huang et al., 2025), in this work, we focus specifically on table-related tasks. Tables are highly data-intensive and often contain many similar rows and columns (Cao, 2025), which makes models particularly prone to referencing incorrect data. Formally, we categorize tabular DREs based on granularity of referenced content as follows:

- **Incorrect Citation**: The response cites *individual* table content (e.g., values or metadata) that does not match the actual table. This includes citing the wrong value, confusing rows or columns, or fabricating table-based content. As illustrated in Figure 1a, the model mistakenly took "Nikkan Sports Grand Prix (Fall)" as from the "Award" column, whereas the correct value should have been "Best Supporting Actress". This mix-up led to an incorrect final answer.
- **Omitted Information**: The response omits table values that belong to *a required subset of the table*, such as listing all rows or identifying "all teams with more than 5 wins." As shown in Figure 1b, the model correctly listed every row but missed the single row "Oct 23". This suggests that while the model can parse the table format, it still makes avoidable omissions.

In this work, we systematically investigate the occurrence and impact of DREs and propose a plug-in critic module to mitigate them.

3.2 Evaluation via LLM-as-a-Judge

To reduce human effort and enable automatic evaluation of DREs, we adopt LLM-as-a-Judge (Zheng et al., 2023; Wolff and Hulsebos, 2025), leveraging a powerful LLM (*i.e.* Sonnet-3.7, Anthropic, 2025) to detect DREs in model responses. To match human-level annotation quality, we address the following challenges by careful designs:

	Accuracy (%)	DRE Rate (%)	DRE-in-Incorrect (%)	Correct-in-DRE (%)
Qwen3-8B on different datasets				
WTQ	77.14	14.04	32.63	46.89
WTQ + prompting	77.51	12.50	28.76	48.25
WTQ (CSV)	75.94	17.54	37.32	48.82
WTQ (Markdown)	77.26	14.34	32.69	48.15
TableBench	77.48	10.55	30.63	13.43
FinQA	63.21	33.57	39.34	56.88
SciTab	77.53	14.06	21.54	65.57
ToTTo*	14.06	18.45	–	–
Different models on WTQ				
Qwen3-1.7B	57.76	35.52	56.35	32.99
Qwen3-4B	75.69	16.18	35.51	46.66
Qwen3-8B	77.14	14.04	32.63	46.89
Qwen2.5-7B-Instruct	43.32	17.56	23.60	23.85
Table-R1-Zero-7B	76.10	19.29	52.41	35.08
Distill-Qwen-7B	49.47	46.04	66.61	26.90
Distill-Llama-8B	59.78	37.96	60.45	35.96
Llama4-Scout	55.71	46.48	72.77	30.66
gpt-oss-20b	78.38	5.71	16.29	38.31

Table 1: **DRE Evaluation Results** judged by Sonnet-3.7+gt. *: No binary correctness labels for ToTTo.

1. **Long&Verbose Response:** Recent reasoning models often generate lengthy thinking processes (Sui et al., 2025). To cope with this issue, we split the response at each occurrence of reflection tokens (*e.g.* "Wait") and let the judge model evaluate one segment at a time.
2. **Detection Reliability:** Even strong models like Sonnet-3.7 can be swayed by the given response and fail to identify DREs, leading to false negatives (see Figure 7, 8). To counter this, we provide the ground truth to the table-based question in the judge prompt. This helps the judge, especially when the final answer is wrong, to cross-check against the table more carefully and decide whether the error comes from a DRE.

In practice, the judge model is instructed to check whether a model-generated response uses table information accurately by examining the aforementioned two types of DREs—Incorrect Citations and Omitted Information. Manual inspection indicates that Sonnet-3.7 with ground truth (*i.e.* Sonnet-3.7+gt) achieves an accuracy of 92.67% with high consistency. Details and the full judge prompt are provided in Appendix A.

Evaluation Metrics To holistically evaluate the occurrence of DREs in model responses to table questions, we calculate the following metrics:

$$\text{DRE Rate} = \frac{|\text{DRE}|}{|\text{Total}|},$$

where $|\text{DRE}|$ is the number of model responses containing at least one DRE. This metric measures the overall frequency of DREs.

$$\text{Correct-in-DRE Ratio} = \frac{|\text{Correct} \cap \text{DRE}|}{|\text{DRE}|},$$

where $|\text{Correct} \cap \text{DRE}|$ is the number of responses whose final answer is correct despite containing DREs. This metric captures DREs that cannot be detected by evaluating final-answer accuracy alone.

$$\text{DRE-in-Incorrect Ratio} = \frac{|\text{Incorrect} \cap \text{DRE}|}{|\text{Incorrect}|},$$

which provides an approximation of the correlation between DREs and final answer accuracy.

3.3 Prevalence and Analysis

Now, we examine the severity of DREs. We focus on three types of table tasks: **Question Answering**, including WTQ (Pasupat and Liang, 2015), TableBench (Wu et al., 2025a), and FinQA (Chen et al., 2021); **Claim Verification**, represented by SciTab (Lu et al., 2023), where the model is asked to determine whether a given claim is supported by the table; and **Table-to-Text**, represented by ToTTo (Parikh et al., 2020), which requires generating a textual description conditioned on the table.

We evaluate a range of popular LLMs, spanning sizes from 1.7B to 20B and covering different model families: reasoning models that characterize extended thinking processes, such as Qwen3-8B (Yang et al., 2025a); mixture-of-experts (MoE) models such as Llama4-Scout (Meta AI, 2025); and standard LLMs such as Qwen2.5-7B-Instruct (Yang et al., 2024). Following Wu et al. (2025a), we present tables in the JSON format, but we also experiment with CSV and Markdown formats. We further test a prompting-based method that explicitly instructs the model: *Use only the table. Do not omit, miscite, or fabricate information. Ensure all cited values exactly match the table.* Model responses are then evaluated using Sonnet-3.7+gt, and the results are summarized in Table 1. We have the following observations:

(1) Data referencing errors are prevalent across different models, table formats, and table-related tasks. For models, we observe that within a single model family such as Qwen3, data referencing capability improves with model size: larger models tend to produce fewer DREs. However, across different model families, this trend does not necessarily hold, as overall model capability also matters. For example, Llama4-Scout, as a non-reasoning model, shows relatively high rates of DREs (46.48%) despite its size. Additionally, results across different table formats (JSON, CSV, and Markdown) and table-related tasks demonstrate that DREs cannot be attributed to specific formats or tasks, but instead represent a general and widespread challenge.

(2) DREs persist under common mitigation strategies. First, reasoning models including Qwen3 series, Distill series, and gpt-oss-20b are featured by self-reflection (DeepSeek-AI et al., 2025; Snell et al., 2024), yet they still exhibit DREs (5.71%-46.04%); in fact, once the first error is made, the model often repeats it, relying more on its own generation than on the original table (see Appendix Figure 4 for an example). Second, even explicitly prompting the model to focus on data referencing accuracy, *i.e.* WTQ + prompt setting, does not resolve DREs or improve final-answer accuracy. Third, Table-R1-Zero-7B (Yang et al., 2025b) was trained on table-related datasets from Qwen2.5-7B-Instruct using RLVR (Lambert et al., 2024). While this specialized training improves answer accuracy, it does not effectively translate into fewer DREs, highlighting that data referencing is a separate capability that warrants further attention.

(3) DREs may also occur during the reasoning process, even when the final answer is correct. The Correct-in-DRE Ratio captures cases where the response contains DREs but still arrives at the correct final answer. This means that final-answer accuracy alone cannot guarantee the correctness of intermediate steps and the overall quality of the response. Besides, the Correct-in-DRE Ratio visibly varies across tasks. For example, SciTab shows a relatively high ratio (65.57%), because its answers are binary labels (True or False). In such cases, numerical citation errors in reasoning process may not affect the final judgment, as illustrated in Figure 5.

4 Reducing DREs with Critics

We observe that DREs do occur in incorrect cases and can negatively impact final accuracy, as shown quantitatively in Table 1 and qualitatively in Figure 1. Nevertheless, the DRE-in-Incorrect ratio should not be interpreted as indicating that this portion of incorrect answers is directly caused by DREs. This raises an important question: *to what extent do DREs actually harm final accuracy?* In this section, we apply Sonnet-3.7+gt[2] as a high-quality critic to reduce DREs and explore whether this reduction translates into improvements in final-answer accuracy. We focus on three question answering datasets (WTQ, TableBench, FinQA) and three representative models (Qwen3-8B, Distill-Qwen-7B, Llama4-Scout).

4.1 Critic-Based Filtering

Method A common application of a critic model is the Best-of-N (BoN) strategy, where an LLM generates multiple candidate responses and the critic selects the best one as the final output (Snell et al., 2024; Bai et al., 2022; Touvron et al., 2023). While effective in some contexts, this approach assumes that the critic is able to fully judge the correctness of each response (Cobbe et al., 2021a) or assign highly discriminative scores across the set of responses (Lightman et al., 2024; Uesato et al., 2022). Our critic, however, is designed specifically to detect DREs and thus cannot directly determine which single response is best overall. For example, multiple responses may contain no DREs yet still produce different final answers if mistakes occur later in the reasoning stage after retrieving the correct table values.

To address this limitation, we adopt a *critic-based filtering* approach. Specifically, for a generation model (e.g., Qwen3-8B), we sample $N = 8$ responses per question and use the critic to select the subset of responses with the fewest data referencing errors instead of selecting only one "best" response. This design improves the overall quality

[2]Although provided with ground truth answers, Sonnet-3.7 does not directly judge final-answer correctness (see Appendix A). We use Sonnet-3.7+gt to approximate the upper bound of a DRE detection critic.

Dataset	Avg Acc (%)	CF Acc (%)	MV Acc (%)	CF + MV Acc (%)	# Total
Qwen3-8B					
WTQ	64.59	70.44	70.84	**73.49**	1509
TableBench	63.12	67.42	70.17	**71.82**	181
FinQA	54.58	56.48	56.92	**57.54**	325
Distill-Qwen-7B					
WTQ	49.47	61.83	62.05	**65.80**	2851
TableBench	55.06	69.62	67.60	**71.65**	321
FinQA	41.12	46.37	46.82	**48.16**	598
Llama4-Scout					
WTQ	57.02	69.89	64.06	**73.11**	2265
TableBench	50.67	58.93	57.40	**63.23**	223
FinQA	39.53	42.32	44.91	**46.76**	216

Table 2: Critic-based Filtering (CF) Results on the DRE subset. Avg Acc denotes the average accuracy over $N = 8$ sampled responses per question. MV denotes Majority Voting, and CF + MV denotes majority-voting on critic-filtered subset.

of the candidate pool and enables inference-time strategies such as majority voting to operate on a higher-quality set of responses, thereby further improving final accuracy.

Metrics We report the average accuracy of all generated responses versus that of the subset selected by the critic. We also compare these results with majority voting. Our primary focus is the DRE subset, which includes questions for which at least one response contains a data referencing error and at least one does not. This subset emphasizes data-referencing-challenging cases. To illustrate: if all sampled responses are free of referencing errors, then critic-based filtering will naturally show little or no improvement. Conversely, if all responses contain referencing errors, no selection strategy can guarantee correctness. Results on the full evaluation set are also reported in Appendix Table 5 but may underestimate the critic's impact.

Results From Table 2, we can observe that critic-based filtering steadily outperforms the average accuracy of all sampled responses by selecting those with fewer data referencing errors. This indicates that reducing DREs not only improves the quality of intermediate reasoning with fewer hallucinations but also translates into higher final accuracy. More encouragingly, it complements majority voting as an inference-time strategy: applying majority voting within the critic-filtered subset achieves the best performance, consistently surpassing majority voting alone. In some cases like Llama4-Scout on WTQ, even randomly selecting a response from the critic-filtered subset yields higher average accuracy than majority voting.

4.2 Rejection Sampling

Method Another application of the critic model is rejection sampling. In standard LLMs, rejection sampling resembles BoN (Bai et al., 2022; Touvron et al., 2023). In the context of reasoning models, this approach can be inefficient, as the responses of reasoning models are often very long (Chen et al., 2024; Sui et al., 2025) and thus costly to generate when sampling N full completions. Moreover, repeated sampling increases computational expense.

We adapt rejection sampling for reasoning models by working at the segment level. Similar to Section 3.2, for a generation model such as Qwen3-8B, we split a response into segments using the delimiter "Wait". Instead of regenerating an entire response, we selectively resample only the segment (or, when necessary, the entire response) until it passes the critic or reaches a maximum retry limit $N = 8$ is reached. The model then continues with the next segment, repeating this process until the final answer is produced. This design reduces the cost of rejection sampling while preventing error propagation across the reasoning process.

Metrics We report accuracy using rejection sampling on the DRE subset and the full set.

Results As shown in Table 3, rejection sampling with the critic effectively improves final accuracy for both reasoning and non-reasoning models. As expected, the improvement is larger on the DRE subset than on the full set, since data referencing errors are more likely to appear in the DRE subset. It is important to note that the rejection sampling process does not alter the generation model itself. A DRE-free response can be obtained by simply resampling. This highlights that DREs are largely

Dataset	Acc in DRE (%)	Acc in Full (%)
Qwen3-8B		
WTQ	63.88	77.14
+ RS	68.46 (+4.58)	78.94 (+1.80)
TableBench	63.54	77.48
+ RS	69.09 (+5.55)	79.31 (+1.83)
FinQA	53.85	63.21
+ RS	54.46 (+0.61)	63.64 (+0.43)
Distill-Qwen-7B		
WTQ	48.58	49.47
+ RS	57.14 (+8.56)	55.99 (+6.52)
TableBench	53.58	54.77
+ RS	68.85 (+15.27)	66.73 (+11.96)
FinQA	41.14	46.90
+ RS	44.82 (+3.68)	49.52 (+2.62)
Llama4-Scout		
WTQ	56.16	55.71
+ RS	65.39 (+9.23)	61.92 (+6.21)
TableBench	52.47	54.56
+ RS	56.95 (+4.48)	57.20 (+2.64)
FinQA	40.74	58.94
+ RS	43.52 (+2.78)	59.72 (+0.78)

Table 3: Rejection Sampling Results. "Acc in DRE" denotes the results on the DRE subset, which we use as the primary evaluation setting.

avoidable errors rather than fundamental limitations of the model's knowledge or reasoning ability. However, how to reliably reduce the frequency of DREs remains an open problem, and rejection sampling with a DRE detection critic provides a promising and practical solution.

5 Training a Small-Scale Critic

In practice, ground-truth answers are often unavailable to the critic particularly during inference, and Sonnet-3.7 is both a black-box and costly to use. Therefore, in this section, we explore the feasibility of training a smaller-scale LLM (e.g. Qwen3-4B-Instruct) to perform the critic task.

5.1 Small Critic Training

For the critic model, similar to Section 3.2, its task is as follows: given a table, a question based on the table, and a model's response segment, determine whether the response contains DREs (See Appendix Figure 6 for the complete critic prompt). The output is either *True* (contains DREs, treated as positive samples) or *False* (does not contain DREs, including cases where no table values are cited, treated as negative samples). Beyond directly using Qwen3-4B-Instruct, we introduce a two-stage training pipeline to further enhance its ability to detect DREs.

(1) SFT. We begin with supervised fine-tuning (SFT) as a warm-up stage to adapt Qwen3-4B-Instruct more effectively to the critic task. Although it already demonstrates strong instruction-following ability, we observe that training with RL directly, that is, without an SFT warm-up, causes the model to produce malformed outputs such as repeated <judgment></judgment> tags. To solve this, we use judgments from Sonnet-3.7 as distillation data in the first stage. This not only teaches Qwen3-4B-Instruct the expected output format but also transfers potentially useful "critic heuristics" from a stronger model, thereby stabilizing subsequent RL training.

(2) RLVR. In the second stage, we employ reinforcement learning to enhance the critic model's robustness and generalization. Building on SFT foundation, RL enables exploration beyond the limitations of fixed supervision. We apply Reinforcement Learning with Verified Reward (RLVR, Lambert et al., 2024; DeepSeek-AI et al., 2025), which is well suited to our setting: it leverages verified binary labels as reward signals and does not require constraining the CoT. This allows the critic to better detect DREs across diverse tasks.

For training data, we build on Qwen3-8B's responses to the WTQ training set. We use Sonnet-3.7 to label all response segments, yielding a balanced dataset of 2,000 positive and negative samples for SFT, and 5,712 samples for RL training. The critic model trained using this data is called Critic-4B. In addition, we construct synthetic positives by inserting four types of DREs with rule-based heuristics (Appendix C), which reduces reliance on larger models and improves efficiency in both speed and cost. The critic model trained using the synthetic data is called Critic-4B-Synthetic.

5.2 Critic Evaluation

To evaluate a critic's performance on the DRE detection task, we construct a *critic evaluation dataset*. In details, we collect real positive and negative response segments, judged by Sonnet-3.7+gt, from the three models (Qwen3-8B, Distill-Qwen-7B, Llama4-Scout) across the three datasets (WTQ, TableBench, FinQA). This setup allows us to cover both reasoning (Qwen3-8B, Distill-Qwen-7B) and non-reasoning models (Llama4-Scout), as well as a diverse range of table-related question answering tasks spanning general-domain benchmarks and financial reasoning. For each model-dataset pair, we

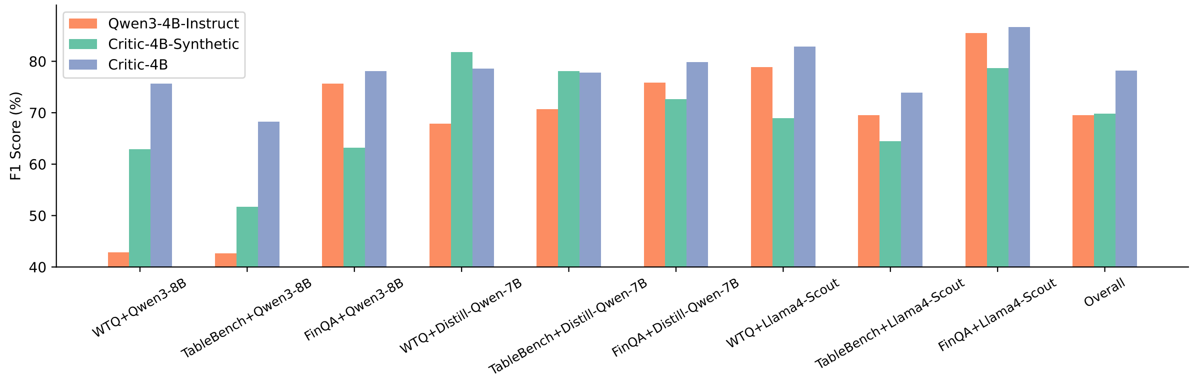


Figure 2: F1 scores across different model-dataset pairs for critic evaluation.

randomly sample 400 response segment with a balanced number of positive and negative examples, resulting in a total of 3,600 samples. We report the standard F1 score for this binary classification task, which balances precision (how often predicted DREs are correct) and recall (how many true DREs are identified).

Results We compare the critic performance of Qwen3-4B-Instruct, Critic-4B-Synthetic, Critic-4B. The evaluation results are presented in Figure 2. We have the following findings:

Critic-4B consistently outperforms the untrained baseline Qwen3-4B-Instruct across all scenarios, achieving 78.16% overall F1 compared to 69.51%. Although it is trained only on the responses of Qwen3-8B on the WTQ training set, Critic-4B generalizes well to the same model's responses on other table-related tasks. This is particularly notable because TableBench and FinQA's tables differ significantly from those in WTQ. In addition, Critic-4B achieves high F1 in identifying DREs in the responses of other models as well.

However, Critic-4B-Synthetic, trained on synthetic data, shows larger gains on in-distribution data, *i.e.* on the same model or the same dataset. Yet, for settings with larger differences, such as FinQA (a different domain) and Llama4-Scout (non-reasoning model), critic performance actually declines. This suggests that the model may have overfit to biases specific to the synthetic data rather than learning to generalize to real-world errors.

5.3 Rejection Sampling

We also examine whether our trained small-scale critic, Critic-4B, can assist inference, using rejection sampling as described in Section 4.2. We report both accuracy and DRE rate (on the full set) with and without rejection sampling.

Dataset	Acc in DRE	Acc in Full	DRE Rate
Qwen3-8B			
WTQ	63.88	77.14	14.04
+ RS	66.37 (+2.49)	78.25 (+1.11)	10.08 (-3.96)
TableBench	63.54	77.48	10.55
+ RS	66.85 (+3.31)	78.50 (+1.02)	9.33 (-1.22)
FinQA	53.85	63.21	33.57
+ RS	54.46 (+0.61)	63.47 (+0.26)	31.21 (-2.36)
Distill-Qwen-7B			
WTQ	48.58	49.47	46.04
+ RS	55.00 (+6.42)	54.01 (+4.54)	30.59 (-15.45)
TableBench	53.58	54.77	46.86
+ RS	63.24 (+9.66)	61.66 (+6.89)	32.45 (-14.41)
FinQA	41.14	46.90	43.94
+ RS	45.82 (+4.68)	50.04 (+3.14)	34.52 (-9.42)
Llama4-Scout			
WTQ	56.16	55.71	46.48
+ RS	60.75 (+4.59)	58.45 (+2.74)	36.90 (-9.58)
TableBench	52.47	54.56	38.54
+ RS	55.16 (+2.69)	56.39 (+1.83)	33.87 (-4.67)
FinQA	40.74	58.94	30.69
+ RS	43.06 (+2.32)	59.46 (+0.52)	28.68 (-2.01)

Table 4: Rejection Sampling Results (%) using Critic-4B.

As shown in Table 4, rejection sampling with Critic-4B consistently achieves higher accuracy compared to the setting without rejection sampling. Although it is less effective than the stronger critic, Sonnet-3.7+gt (refer to Table 3), Critic-4B offers a lightweight and cost-effective alternative. Encouragingly, Critic-4B is smaller than all three generation models and does not benefit from the extended thinking processes as reasoning models like Qwen3-8B and Distill-Qwen-7B, yet it is still able to improve their accuracy. In addition, the DRE rate decreases when using rejection sampling, indicating that the critic not only enhances final-answer accuracy but also improves the overall quality of the model responses by reducing DREs.

6 Conclusions

We show that data referencing errors (DREs) are a pervasive weakness of LLMs on table reasoning tasks, undermining both response quality and final accuracy. By systematically analyzing DREs via LLM-as-a-Judge, we demonstrated their prevalence and propose inference-time strategies and lightweight critics to mitigate them. Our findings establish data referencing as a key evaluation dimension beyond final-answer accuracy for developing more reliable table reasoning systems.

Limitations

Several limitations remain that warrant future study. First, we focus solely on DREs in table-related tasks, as they are a common and non-negligible issue. However, we recognize that DREs also arise in other domains and modalities. For instance, Cobbe et al. (2021b); Mirzadeh et al. (2025) describe the case: *He makes 48 total ice cubes, including 10 giant cubes, 14 small cubes, 12 medium cubes, and some tiny cubes.* Qwen3-8B sometimes mistakenly interprets the order as giant, medium, small, tiny, which leads to errors. We hope future work can generalize to such broader domains.

Second, we did not examine the causes of DREs from an interpretability perspective. In preliminary experiments, we did observe that when the model *prepared* to reference a table value, increasing its attention to the entire table helped reduce subsequent errors, suggesting that DREs are linked to insufficient attention. However, due to resource constraints, we did not scale up attention analyses or steering experiments. Future work could build on this direction.

Acknowledgments

We are grateful to Robin Jia, He Wang, and Zelin He for their insightful feedback and discussions throughout this work. We also thank the anonymous reviewers for their constructive comments.

References

Anthropic. 2025. Claude 3.7 sonnet and claude code. https://www.anthropic.com/news/claude-3-7-sonnet.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional AI: harmlessness from AI feedback. CoRR, abs/2212.08073.

Lang Cao. 2025. Tablemaster: A recipe to advance table understanding with language models. CoRR, abs/2501.19378.

Xingyu Chen, Jiahao Xu, Tian Liang, Zhiwei He, Jianhui Pang, Dian Yu, Linfeng Song, Qiuzhi Liu, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Zhaopeng Tu, Haitao Mi, and Dong Yu. 2024. Do NOT think that much for 2+3=? on the overthinking of o1-like llms. CoRR, abs/2412.21187.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Kenneth Huang, Bryan R. Routledge, and William Yang Wang. 2021. Finqa: A dataset of numerical reasoning over financial data. In Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, pages 3697–3711. Association for Computational Linguistics.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021a. Training verifiers to solve math word problems. CoRR, abs/2110.14168.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021b. Training verifiers to solve math word problems. arXiv preprint arXiv:2110.14168.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. ULTRAFEEDBACK: boosting language models with scaled AI feedback. In Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024. OpenReview.net.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, and 81 others. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. CoRR, abs/2501.12948.

Jonathan Herzig, Pawel Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 4320–4333. Association for Computational Linguistics.

Lei Huang, Xiaocheng Feng, Weitao Ma, Yuchun Fan, Xiachong Feng, Yangfan Ye, Weihong Zhong, Yuxuan Gu, Baoxin Wang, Dayong Wu, Guoping Hu, and Bing Qin. 2025. Improving contextual faithfulness of large language models via retrieval heads-induced optimization. In Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pages 16896–16913. Association for Computational Linguistics.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. 2023. Structgpt: A general framework for large language model to reason over structured data. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, pages 9237–9251. Association for Computational Linguistics.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, and 4 others. 2024. Tülu 3: Pushing frontiers in open language model post-training. CoRR, abs/2411.15124.

Fangyu Lei, Jinxiang Meng, Yiming Huang, Tinghong Chen, Yun Zhang, Shizhu He, Jun Zhao, and Kang Liu. 2025. Reasoning-table: Exploring reinforcement learning for table reasoning. CoRR, abs/2506.01710.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024. OpenReview.net.

Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. 2025. Understanding r1-zero-like training: A critical perspective. CoRR, abs/2503.20783.

Xinyuan Lu, Liangming Pan, Qian Liu, Preslav Nakov, and Min-Yen Kan. 2023. SCITAB: A challenging benchmark for compositional reasoning and claim verification on scientific tables. In Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023, pages 7787–7813. Association for Computational Linguistics.

Meta AI. 2025. Llama 4: Multimodal intelligence. https://ai.meta.com/blog/llama-4-multimodal-intelligence/.

Iman Mirzadeh, Keivan Alizadeh, Hooman Shahrokhi, Oncel Tuzel, Samy Bengio, and Mehrdad Farajtabar. 2025. Gsm-symbolic: Understanding the limitations of mathematical reasoning in large language models. In The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025. OpenReview.net.

Nafise Sadat Moosavi, Andreas Rücklé, Dan Roth, and Iryna Gurevych. 2021. Scigen: a dataset for reasoning-aware text generation from scientific tables. In Proceedings of the Neural Information Processing Systems Track on Datasets and Benchmarks 1, NeurIPS Datasets and Benchmarks 2021, December 2021, virtual.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel J. Candès, and Tatsunori Hashimoto. 2025. s1: Simple test-time scaling. CoRR, abs/2501.19393.

OpenAI. 2025. gpt-oss-120b & gpt-oss-20b model card. Preprint, arXiv:2508.10925.

Ankur P. Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. Totto: A controlled table-to-text generation dataset. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, pages 1173–1186. Association for Computational Linguistics.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers, pages 1470–1480. The Association for Computer Linguistics.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. CoRR, abs/2402.03300.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2024. Hybridflow: A flexible and efficient rlhf framework. arXiv preprint arXiv: 2409.19256.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. CoRR, abs/2408.03314.

Aofeng Su, Aowen Wang, Chao Ye, Chen Zhou, Ga Zhang, Gang Chen, Guangcheng Zhu, Haobo Wang, Haokai Xu, Hao Chen, Haoze Li, Haoxuan Lan, Jiaming Tian, Jing Yuan, Junbo Zhao, Junlin Zhou, Kaizhe Shou, Liangyu Zha, Lin Long, and 14 others. 2024. Tablegpt2: A large multimodal model with tabular data integration. CoRR, abs/2411.02059.

Yang Sui, Yu-Neng Chuang, Guanchu Wang, Jiamu Zhang, Tianyi Zhang, Jiayi Yuan, Hongyi Liu, Andrew Wen, Shaochen Zhong, Na Zou, Hanjie Chen, and Xia Hu. 2025. Stop overthinking: A survey on efficient reasoning for large language models. Trans. Mach. Learn. Res., 2025.

Xiangru Tang, Wanghan Xu, Yujie Wang, Zijie Guo, Daniel Shao, Jiapeng Chen, Cixuan Zhang, Ziyi Wang, Lixin Zhang, Guancheng Wan, Wenlong Zhang, Lei Bai, Zhenfei Yin, Philip Torr, Hanrui Wang, and Di Jin. 2025. Eigen-1: Adaptive multi-agent refinement with monitor-based rag for scientific reasoning. Preprint, arXiv:2509.21193.

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

Jonathan Uesato, Nate Kushman, Ramana Kumar, H. Francis Song, Noah Y. Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. 2022. Solving math word problems with process- and outcome-based feedback. CoRR, abs/2211.14275.

Zengzhi Wang, Fan Zhou, Xuefeng Li, and Pengfei Liu. 2025. Octothinker: Mid-training incentivizes reinforcement learning scaling. CoRR, abs/2506.20512.

Cornelius Wolff and Madelon Hulsebos. 2025. How well do llms reason over tabular data, really? CoRR, abs/2505.07453.

Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xeron Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, Tongliang Li, Zhoujun Li, and Guanglin Niu. 2025a. Tablebench: A comprehensive and complex benchmark for table question answering. In AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA, pages 25497–25506. AAAI Press.

Zeqiu Wu, Yushi Hu, Weijia Shi, Nouha Dziri, Alane Suhr, Prithviraj Ammanabrolu, Noah A. Smith, Mari Ostendorf, and Hannaneh Hajishirzi. 2023. Fine-grained human feedback gives better rewards for language model training. In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.

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

Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2025. Evaluating mathematical reasoning beyond accuracy. In AAAI-25, Sponsored by the Association for the Advancement of Artificial Intelligence, February 25 - March 4, 2025, Philadelphia, PA, USA, pages 27723–27730. AAAI Press.

Jiahuan Yan, Jintai Chen, Chaowen Hu, Bo Zheng, Yaojun Hu, Jimeng Sun, and Jian Wu. 2025. Small models are LLM knowledge triggers for medical tabular prediction. In The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025. OpenReview.net.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 40 others. 2025a. Qwen3 technical report. CoRR, abs/2505.09388.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024. Qwen2.5 technical report. CoRR, abs/2412.15115.

Zheyuan Yang, Lyuhao Chen, Arman Cohan, and Yilun Zhao. 2025b. Table-r1: Inference-time scaling for table reasoning. CoRR, abs/2505.23621.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. 2025. LIMO: less is more for reasoning. CoRR, abs/2502.03387.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. 2023. Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2023, Taipei, Taiwan, July 23-27, 2023, pages 174–184. ACM.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020, pages 8413–8426. Association for Computational Linguistics.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2024. Tablellama: Towards open large generalist models for tables. In Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), NAACL 2024, Mexico City, Mexico, June 16-21, 2024, pages 6024–6044. Association for Computational Linguistics.

Xiaokang Zhang, Sijia Luo, Bohan Zhang, Zeyao Ma, Jing Zhang, Yang Li, Guanlin Li, Zijun Yao, Kangli Xu, Jinchang Zhou, Daniel Zhang-Li, Jifan Yu, Shu Zhao, Juanzi Li, and Jie Tang. 2025a. Tablellm: Enabling tabular data manipulation by llms in real office usage scenarios. In Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pages 10315–10344. Association for Computational Linguistics.

Xuanliang Zhang, Dingzirui Wang, Baoxin Wang, Longxu Dou, Xinyuan Lu, Keyan Xu, Dayong Wu, and Qingfu Zhu. 2025b. SCITAT: A question answering benchmark for scientific tables and text covering diverse reasoning types. In Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pages 3859–3881. Association for Computational Linguistics.

Xuanliang Zhang, Dingzirui Wang, Keyan Xu, Qingfu Zhu, and Wanxiang Che. 2025c. Rot: Enhancing table reasoning with iterative row-wise traversals. CoRR, abs/2505.15110.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025d. The lessons of developing process reward models in mathematical reasoning. In Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025, pages 10495–10516. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. In Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023.

Yaowei Zheng, Richong Zhang, Junhao Zhang, Yanhan Ye, Zheyan Luo, Zhangchi Feng, and Yongqiang Ma. 2024. Llamafactory: Unified efficient fine-tuning of 100+ language models. In Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations), Bangkok, Thailand. Association for Computational Linguistics.

A LLM-as-a-Judge

The complete judge prompt for Sonnet-3.7+gt is shown in Figure 3. Note that we provide ground-truth answers to mitigate false negatives, as explained in Section 3.2. We also explicitly instruct the judge to focus solely on comparing the model response with the table data in order to assess table-referencing accuracy. With this explicit instruction, we find that Sonnet-3.7+gt remains unbiased: it distinguishes reasoning mistakes from genuine DREs rather than assuming that every wrong final answer reflects a DRE. A case of Sonnet-3.7+gt's judgment is shown below:

`While the final answer differs from the reference answer, this appears to be a calculation error rather than a table referencing error. The model accurately extracted and cited all relevant values from the table.`

We randomly sampled 100 instances from the critic evaluation dataset and three annotators at the PhD level independently assessed whether Sonnet-3.7+gt's judgments were correct. Their assessments yielded an average accuracy of 92.67%, indicating near-human reliability.

B Critic-based Filtering

Dataset	Avg Acc (%)	CF Acc (%)	# Total
Qwen3-8B			
WTQ	77.55	79.65	4344
TableBench	76.27	78.05	493
FinQA	64.01	64.74	1147
Distill-Qwen-7B			
WTQ	49.60	58.43	4344
TableBench	55.17	65.87	493
FinQA	47.56	50.68	1147
Llama4-Scout			
WTQ	56.22	62.93	4344
TableBench	53.80	57.54	493
FinQA	57.96	58.48	1147

Table 5: Accuracy comparison between all-sample average and critic-selected subset on the full set.

C Synthetic Positives Construction

We use four strategies to insert DREs given a table and a model's response with correct final answers:

1. **Mix up rows:** Swap the identified value with a value from the same column but a different row.
2. **Mix up columns:** Swap the value with another value from the same row but a different column.

3. **Remove row:** Delete the entire row that contains the used value.
4. **Remove a listed row:** Keep the table unchanged, but if the response enumerates all rows, randomly remove one row from the response and re-index.

We then use Qwen3-8B to perform inference for three times again to see whether the answer changes, and only save the cases where the final answer differs. For each saved case, this indicates that the modified table with the original model response, or the original table with the modified response, do not fully match—*i.e.* there are DREs.

D Training Details

For SFT, we use Llama-Factory (Zheng et al., 2024) with a learning rate of 1e-5, a batch size of 8, 2,000 training examples, and train for 2 epochs. For RLVR, we use verl (Sheng et al., 2024), adopting GRPO (Shao et al., 2024), with a batch size of 256 and 8 rollouts per prompt at a temperature of 1.0. The learning rate is fixed at 1e-6, and we train for 20 epochs. During inference, we apply greedy decoding for the trained critic.

E Generation Details

Except for ToTTO, we use string matching to compare accuracy across different table-related tasks. For ToTTO, following Yang et al. (2025b), we use $(\text{BLEU} + \text{ROUGE-L})/2$. For TableBench, we focus only on the Fact Checking and Numerical Reasoning subsets (493 in total), as the other two subsets, Data Analysis and Visualization, are beyond the tested models' capabilities.

All generation models perform inference with their recommended decoding hyperparameters, as detailed in Table 6. For Llama4-Scout, we use the fp4 quantized version.[3]

Model	Temperature	Top-P	Top-K
Qwen3 Series	0.6	0.95	20
Distill Series	0.6	0.95	-1
Llama4-Scout	0.8	0.95	-1
Qwen2.5-7B-Instruct	0.6	0.95	-1
Table-R1-Zero-7B	0.6	0.96	-1
gpt-oss-20b	1.0	1.0	-1

Table 6: Decoding hyperparamters used for generation models.

[3]https://huggingface.co/nvidia/Llama-4-Scout-17B-16E-Instruct-FP4

F Code of Ethics

All datasets and models we use are public. No ethical, safety, or privacy risks are involved in this study.

The licenses of the datasets and models we use are listed in Table 7.

	License
WTQ	CC-BY-SA-4.0
TableBench	Apache 2.0
FinQA	MIT
SciTab	MIT
ToTTo	Creative Commons Share-Alike 3.0
Qwen3 Series	Apache 2.0
Distill Series	MIT
Llama4-Scout	llama4, nvidia-open-model-license
Qwen2.5-7B-Instruct	Apache 2.0
Table-R1-Zero-7B	Apache 2.0
gpt-oss-20b	Apache 2.0

Table 7: Licenses for datasets and models used in this paper.

This paper used LLMs to polish writing. All original content came from the authors themselves.

Critic Prompt for Sonnet-3.7+gt

Table Referencing Accuracy Evaluation

Task
Your task is to evaluate whether a model-generated response accurately uses information from a given table. You will focus exclusively on table referencing accuracy, not other types of errors.

Input Data
<table>{table}</table>

<question>{question}</question>

<model_response>{model_response}</model_response>

You are also provided with a reference answer, but this does not mean there is table referencing error:
<reference_answer>{answer}</reference_answer>

Evaluation Instructions
Analyze the model response sentence by sentence, focusing only on how accurately it references data from the table. If the response uses any value from the table, perform the following checks:

Check 1: Copied Values Consistency
- Identify any value that is explicitly copied or cited from the table.
- For each value:
    - Check if it exists in the table.
    - Check if it is used in the correct context (e.g., correct entity, row, or column).
- If a value is hallucinated, incorrect, or mismatched, it is a table referencing error.

Check 2: Omission
- When the model response tries to list all rows or items from the table, or items that meet a specific condition (e.g., "all teams with more than 5 wins"),
- Then compare the listed items with the table.
- If any expected value is missing, it is a table referencing error.

Important Guidelines
- Focus ONLY on table referencing accuracy
- Ignore other types of errors (reasoning errors, calculation errors, question misunderstandings, etc.)
- The reference answer is provided for context only—base your judgment solely on comparing the model response with the table data

Output Format
After your analysis, provide your judgment using one of these formats:

1. If you find any table referencing error:
<judgment>Failed [Error Type] Check.</judgment>
Where [Error Type] is either "Copied Values Consistency" or "Omission"

2. If no table referencing errors are found:
<judgment>NA</judgment>

Provide your judgment without additional explanation within the judgment tags.

Now follow the instructions step by step and put your final judgment within <judgment></judgment> tags.

Figure 3: Judge prompt for Sonnet-3.7+gt. Both "Failed Copied Values Consistency Check" and "Failed Omission Check" represents there are DREs.

Failure of Self-Reflection in Reasoning Models

Table

Date	Competition	Location	Country	Event	Placing	Rider	Nationality
31 October 2008	2008–09 World Cup	Manchester	United Kingdom	Sprint	1	Victoria Pendleton	GBR
31 October 2008	2008–09 World Cup	Manchester	United Kingdom	Keirin	2	Jason Kenny	GBR
1 November 2008	2008–09 World Cup	Manchester	United Kingdom	Sprint	1	Jason Kenny	GBR
1 November 2008	2008–09 World Cup	Manchester	United Kingdom	500 m time trial	1	Victoria Pendleton	GBR
2 November 2008	2008–09 World Cup	Manchester	United Kingdom	Team sprint	1	Ross Edgar	GBR
2 November 2008	2008–09 World Cup	Manchester	United Kingdom	Team sprint	1	Jason Kenny	GBR
2 November 2008	2008–09 World Cup	Manchester	United Kingdom	Team sprint	1	Jamie Staff	GBR
2 November 2008	2008–09 World Cup	Manchester	United Kingdom	Keirin	1	Victoria Pendleton	GBR
2 November 2008	5th International Keirin Event	Manchester	United Kingdom	International keirin	2	Ross Edgar	GBR
13 February 2009	2008–09 World Cup	Copenhagen	Denmark	Team sprint	1	Chris Hoy	GBR
13 February 2009	2008–09 World Cup	Copenhagen	Denmark	Team sprint	1	Jason Kenny	GBR
13 February 2009	2008–09 World Cup	Copenhagen	Denmark	Team sprint	1	Jamie Staff	GBR
13 February 2009	2008–09 World Cup	Copenhagen	Denmark	Sprint	1	Victoria Pendleton	GBR
30 October 2009	2009–10 World Cup	Manchester	United Kingdom	Keirin	1	Chris Hoy	GBR
30 October 2009	2009–10 World Cup	Manchester	United Kingdom	Sprint	1	Victoria Pendleton	GBR
30 October 2009	2009–10 World Cup	Manchester	United Kingdom	Sprint	1	Chris Hoy	GBR
30 October 2009	2009–10 World Cup	Manchester	United Kingdom	500 m time trial	2	Victoria Pendleton	GBR
1 November 2009	2009–10 World Cup	Manchester	United Kingdom	Team sprint	1	Ross Edgar	GBR
1 November 2009	2009–10 World Cup	Manchester	United Kingdom	Team sprint	1	Chris Hoy	GBR
1 November 2009	2009–10 World Cup	Manchester	United Kingdom	Team sprint	1	Jamie Staff	GBR

Question
What is the number of 1st place finishes across all events?

Model Response
<think>
Okay, let's tackle this question. The user wants to know the number of 1st place finishes across all events from the provided table.

First, I need to look at the table data. The relevant columns here are "Placing" and "Event". Each row represents a rider's performance in a specific event. The "Placing" column indicates their position, so I need to count how many times "1" appears in that column.

Let me go through each row one by one:

Row 1: 1\\Row 2: 2\\Row 3: 1\\Row 4: 1\\Row 5: 1\\Row 6: 1\\Row 7: 1\\Row 8: 1\\Row 9: 2\\Row 10: 1\\Row 11: 1\\Row 12: 1\\Row 13: 1\\Row 14: 1\\Row 15: 1\\Row 16: 2\\Row 17: 1\\Row 18: 1\\Row 19: 1

So counting the 1s: rows 1,3,4,5,6,7,8,10,11,12,13,14,15,17,18,19. That's 16 instances. Wait, maybe I miscounted earlier. Let me list them:

Row 1: 1\\Row 3: 1\\Row 4: 1\\Row 5: 1\\Row 6: 1\\Row 7: 1\\Row 8: 1\\Row 10: 1\\Row 11: 1\\Row 12: 1\\Row 13: 1\\Row 14: 1\\Row 15: 1\\Row 17: 1\\Row 18: 1\\Row 19: 1

That's 16 entries where placing is 1. Wait, but there are 19 rows total. Let me check again. The rows are from 0 to 18 (assuming 19 rows). Let me check each row's placing:

Row 0: 1\\Row 1: 2\\Row 2: 1\\Row 3: 1\\Row 4: 1\\Row 5: 1\\Row 6: 1\\Row 7: 1\\Row 8: 2\\Row 9: 1\\Row 10: 1\\Row 11: 1\\Row 12: 1\\Row 13: 1\\Row 14: 1\\Row 15: 2\\Row 16: 1\\Row 17: 1\\Row 18: 1

So that's 1 (row0), 1 (row2), 1 (row3), 1 (row4), 1 (row5), 1 (row6), 1 (row7), 1 (row9), 1 (row10), 1 (row11), 1 (row12), 1 (row13), 1 (row14), 1 (row16), 1 (row17), 1 (row18). That's 16 instances. So the answer should be 16.

Wait, but let me check again. Let me count each row's placing:

Row 0: 1\\Row 1: 2\\Row 2: 1\\Row 3: 1\\Row 4: 1\\Row 5: 1\\Row 6: 1\\Row 7: 1\\Row 8: 2\\Row 9: 1\\Row 10: 1\\Row 11: 1\\Row 12: 1\\Row 13: 1\\Row 14: 1\\Row 15: 2\\Row 16: 1\\Row 17: 1\\Row 18: 1

Yes, that's 16 times. So the final answer is 16.
</think>

Final Answer: 16

Figure 4: This is an example of Qwen3-8B on the WTQ test set. The table contains 20 rows in total, but Qwen3-8B identifies only 19, missing the actual 16th row (Placing: 1). Even after repeated checks and attempts with different numbering schemes (starting from 1 or from 0), it consistently reproduced this DRE, ultimately leading to an incorrect answer.

Input

Table

Setting	Metrics	cs-en	de-en	fi-en	lv-en	ru-en	tr-en	zh-en	Avg
Baselines	METEOR++	0.552	0.538	0.720	0.563	0.627	0.626	0.646	0.610
Baselines	RUSE(*)	0.624	0.644	0.750	0.697	0.673	0.716	0.691	0.685
Baselines	BERTScore-F1	0.670	0.686	0.820	0.710	0.729	0.714	0.704	0.719
Sent-Mover	Smd + W2V	0.438	0.505	0.540	0.442	0.514	0.456	0.494	0.484
Sent-Mover	Smd + ELMO + PMeans	0.569	0.558	0.732	0.525	0.581	0.620	0.584	0.595
Sent-Mover	Smd + BERT + PMeans	0.607	0.623	0.770	0.639	0.667	0.641	0.619	0.652
Sent-Mover	Smd + BERT + MNLI + PMeans	0.616	0.643	0.785	0.660	0.664	0.668	0.633	0.667
Word-Mover	Wmd-1 + W2V	0.392	0.463	0.558	0.463	0.456	0.485	0.481	0.471
Word-Mover	Wmd-1 + ELMO + PMeans	0.579	0.588	0.753	0.559	0.617	0.679	0.645	0.631
Word-Mover	Wmd-1 + BERT + PMeans	0.662	0.687	0.823	0.714	0.735	0.734	0.719	0.725
Word-Mover	Wmd-1 + BERT + MNLI + PMeans	0.670	0.708	**0.835**	**0.746**	**0.738**	0.762	**0.744**	**0.743**
Word-Mover	Wmd-2 + BERT + MNLI + PMeans	**0.679**	**0.710**	0.832	0.745	0.736	**0.763**	0.740	**0.743**

Claim

Table 1: In all language pairs, the best correlation is achieved by our word mover metrics that use a BERT pretrained on MNLI as the embedding generator and PMeans to aggregate the embeddings from different BERT layers, i.e., WMD-1/2+BERT+MNLI+PMeans.

Question

Is the above claim true or false?

Model Response

...

6. **tr-en (Turkish-English):**
- Wmd-1: 0.744
- Wmd-2: 0.763
- BERTScore-F1 is 0.704. So Wmd-2 is higher here.

...

Figure 5: This is an example of Qwen3-8B on the SciTab test set. Qwen3-8B misquoted 0.714 as 0.704, but this does not affect the subsequent conclusion that Wmd-2 (0.763) is higher. Therefore, final-answer accuracy cannot fully reflect the presence of a DRE.

Critic Prompt

You are a careful evaluator. Your task is to check whether a model's response uses table information accurately.

<instructions>
Please follow these steps:

1. Review the inputs:
<table>{table}</table>
<question>{question}</question>
<model_response>{model_response}</model_response>

2. Go through the <model_response> sentence by sentence. If the response uses any value from the table, do the following checks:

<check_1>
Copied Values Consistency Check:
- Find any value that is explicitly copied or cited from the table.
- For each value:
    - Check if it exists in the table.
    - Check if it is used in the correct context (e.g., correct entity, row, or column).
- If a value is hallucinated, incorrect, or mismatched, it is an error.
</check_1>

<check_2>
Omission Check:
- If the segment tries to list all rows from the table, or all items that meet a specific condition (e.g., "all teams with more than 5 wins"),
- Then compare the listed items with the table.
- If any expected value is missing, it is an error.
</check_2>

3. If any error is found, stop the evaluation immediately and respond with:
<judgment>True</judgment>

4. If no errors are found in the entire segment, respond with:
<judgment>False</judgment>

Follow the instructions step by step and put your final judgment within <judgment> </judgment> tags.
</instructions>

Figure 6: Critic Prompt for small-scale LLMs. We do not provide ground-truth answers in this prompt.

Judge Prompt Input

Table

Rank	Nation	Gold	Silver	Bronze	Total
1	Soviet Union	*7*	3	6	16
2	Austria	4	3	4	11
3	Finland	3	3	1	7
4	Switzerland	3	2	1	6
5	Sweden	2	4	4	10
6	United States	2	3	2	7
7	Norway	2	1	1	4
8	Italy	1	2	0	3
9	Germany	1	0	1	2
10	Canada	0	1	2	3

Question

Who had the most silver medals?

Model Response

... The question is about silver medals. Let me list all the silver counts:

Soviet Union: 7 (assuming the *7* is 7)
Austria: 3
Finland: 3
Switzerland: 2
Sweden: 4
United States: 3
Norway: 1
Italy: 2
Germany: 0
Canada: 1

So the maximum here is 7, which is the Soviet Union. Therefore, the answer should be Soviet Union.

Figure 7: An input example for the judge prompt. In this case, the generation model correctly cites the "Silver" value, except for the first row, "Soviet Union."

Judge Output

Sonnet-3.7 (without ground truth)
I'll analyze the model's response sentence by sentence, checking for any errors in how it uses data from the table.

The model correctly identifies that Soviet Union has "*7*" in the Silver column (with asterisks), and acknowledges this unusual formatting. It then reasonably interprets this as 7 while being transparent about this assumption.

When listing silver medal counts for all countries, the model states:
- Soviet Union: 7 (with clear explanation about the "*7*" notation)
- Austria: 3 ✓
- Finland: 3 ✓
- Switzerland: 2 ✓
- Sweden: 4 ✓
- United States: 3 ✓
- Norway: 1 ✓
- Italy: 2 ✓
- Germany: 0 ✓
- Canada: 1 ✓

All these values match exactly with the silver medal counts in the table. The model provides a complete list of all countries' silver medal counts without any omissions.

The model's conclusion that Soviet Union has the most silver medals (interpreting "*7*" as 7) is consistent with the data presented in the table.

<judgment>NA</judgment>

Sonnet-3.7 with ground truth
I'll carefully evaluate the model response for table referencing accuracy by checking how it uses values from the provided table.

First, let me examine the actual silver medal counts in the table:
- Soviet Union: 3 silver medals (not 7)
- Austria: 3 silver medals
- Finland: 3 silver medals
- Switzerland: 2 silver medals
- Sweden: 4 silver medals
- United States: 3 silver medals
- Norway: 1 silver medal
- Italy: 2 silver medals
- Germany: 0 silver medals
- Canada: 1 silver medal

The model response makes a critical table referencing error. It incorrectly states that the Soviet Union has 7 silver medals when looking at the row: ["1", "Soviet Union", "*7*", "3", "6", "16"]. The model misinterpreted the data, as the "*7*" appears in the Gold column, not the Silver column. The Soviet Union actually has 3 silver medals according to the table.

The model lists all silver medal counts but assigns the wrong value (7) to the Soviet Union. Based on this incorrect value, it concludes that the Soviet Union has the most silver medals, when in reality Sweden has the most with 4.

<judgment>Failed Copied Values Consistency Check.</judgment>

Figure 8: The judge output of Sonnet-3.7 without and with ground truth. Without a ground-truth answer, Sonnet-3.7 is misled by the model response; with a ground-truth answer, it checks the consistency between the model response and the table more carefully.