CompTab: A Comprehensive Benchmark for Real-World TableQA with Complex Reasoning and Irregular Tables

Zhen Yang[1], Wei Du[1], Jie Wang[1], Wenze Zhou[1], Xiangfeng Meng[1], Zhengyang Wang[1], Suping Sun[1], Ziwei Du[1], Haodong Zou[1], Jie Chen[1], Yongbin Liu[2], Shicheng Tan[3], Jiahao Ying[4], Shu Zhao[1]*

[1]Anhui University [2]University of South China
[3]Tsinghua University [4]Singapore Management University

Abstract

Recent progress in Large Language Model (LLM) based Table Question Answering (TableQA) has demonstrated strong performance on standard benchmarks. However, existing benchmarks mainly focus on well-structured tables and fail to reflect the irregular structures and complex reasoning commonly encountered in real-world scenarios. We propose **CompTab**, a benchmark designed to evaluate TableQA under complex reasoning and irregular table conditions. CompTab covers six representative types, including semantic ambiguity, multi-hop reasoning, transposed tables, merged cells, missing values, and outliers. It is constructed from real-world seed tables across multiple domains using controlled LLM based generation and human verification to ensure realism and diversity. In addition, to improve the generalization of LLMs under complex and irregular table settings, we propose a two-stage training framework that progressively aligns models with textual reasoning and executable decision signals, instantiated as CompTabLLM. Evaluations on 38 representative LLMs and CompTabLLM show clear limitations of existing LLMs under realistic conditions, while the proposed framework improves generalization. CompTab thus provides a challenging benchmark for advancing TableQA in real-world. The dataset will be released here.

1 Introduction

Large Language Models (LLMs) have achieved substantial progress in natural language processing and reasoning, extending from unstructured text to structured and semi-structured data (Wang et al., 2024b; Zhang et al., 2025b; Han et al., 2025; Huang et al., 2024; Wu et al., 2025b; Wang et al., 2024a). As a pervasive carrier of structured information in real-world applications, tables play a central role across domains such as finance, healthcare, education, and industrial analytics. Consequently, Table Question Answering (TableQA) has emerged as a key application scenario for LLMs (Yang et al., 2025b,c; Zhang et al., 2024; Patnaik et al., 2024; Liu et al., 2024; Zeng et al., 2025; Belmehdi et al., 2024). Recent advances in in-context learning and instruction tuning have enabled LLMs to achieve strong performance on several benchmarks (Wang et al., 2025c; Wu et al., 2025d; Ahmad et al., 2025; Wu et al., 2025a; Wang et al., 2025b).

Despite these advances, most existing TableQA benchmarks (Wang et al., 2025c; Qiu et al., 2024; Zhong et al., 2017; Zhang et al., 2023) still rely on well-structured tables with clean layouts, which differ substantially from real-world data. In practice, real tables often exhibit structural irregularities (e.g., transposed layouts and merged cells), content imperfections (e.g., missing values and outliers), and require complex reasoning such as multi-hop queries and ambiguity resolution. Works (Wang et al., 2025c; Li et al., 2025; Qiu et al., 2024) show that even LLMs experience notable performance degradation under such conditions. This gap is further reflected in our empirical observations: while GPT-4o achieves strong performance on regular benchmarks such as WTQ and TabFact (83% and 93%) (Wang et al., 2025a), its accuracy drops to 34% on the irregular and complex tables used in our evaluation. A key reason is that conventional benchmarks tend to focus on isolated challenge settings and fail to provide comprehensive coverage of the diverse structural, content-level, and reasoning difficulties observed in real-world tables.

Recent efforts have begun to construct more realistic TableQA benchmarks by introducing specific forms of structural irregularity or reasoning complexity (He et al., 2024; Wu et al., 2025d; Zhang et al., 2025a; Wu et al., 2025c). While these benchmarks shed light on several important challenge types, the coverage of table phenomena and reasoning demands observed in real-world settings

*Corresponding authors. zhaoshuzs2002@hotmail.com

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 27761–27774
July 2-7, 2026

remains incomplete. *Therefore, there remains a strong need for a benchmark that more comprehensively evaluates LLMs on real-world complex reasoning and irregular tables.*

Motivated by this need, we propose **CompTab**, a comprehensive benchmark for systematically evaluating LLM based TableQA under complex reasoning and irregular table scenarios commonly encountered in real-world applications. In Figure 1, CompTab explicitly targets six representative types: ambiguity, multi-hop reasoning, transposed tables, merged cells, missing values, and outliers, covering diverse structural, content-level, and reasoning difficulties. To ground the benchmark in realistic settings, we collect seed tables from multiple domains and scale data diversity through a controlled LLM-based generation pipeline, coupled with rigorous human verification to ensure structural validity, answerability, and type fidelity.

Beyond benchmarking, we propose a two-stage training framework to improve LLM generalization in such settings. We construct CompTabInstruct, an instruction corpus integrating textual Direct Prompting (DP) reasoning (Liu et al., 2024) and executable Python Agent (PyAgent) traces (Liu et al., 2024). Models are first adapted via supervised fine-tuning on DP trajectories and then aligned with agent-level decisions through direct preference optimization, resulting in a model we refer to as CompTabLLM.

Extensive evaluations on CompTab across a wide range of open-source and closed-source LLMs reveal clear limitations, while CompTabLLM achieves consistent improvements, validating the effectiveness of the proposed framework.

Our contributions are summarized as follows:

- A Realistic and Challenging TableQA Benchmark: We introduce CompTab, a comprehensive benchmark that systematically covers six real-world challenge types, including ambiguity, multi-hop reasoning, transposed tables, merged cells, missing values, and outliers. It enables systematic evaluation of LLM performance on complex and irregular tables.

- Training Framework for Improved Generalization: We propose a framework that improves LLM generalization in challenging TableQA scenarios by combining Direct Prompting supervision and PyAgent-level preference alignment, resulting in CompTabLLM.

- Comprehensive Evaluation: We evaluate 38 representative open-source and closed-source LLMs and CompTabLLM on CompTab. The results reveal clear limitations on irregular and complex tables and validate the proposed training framework.

2 Construction of CompTab

This section describes the data construction pipeline of CompTab. As illustrated in Figure 2, the benchmark is built through three stages: seed collection, controlled table–question generation, and human-verified answer annotation. Each stage is designed to ensure realistic table structures, diverse reasoning patterns, and reliable evaluation signals.

2.1 Formal Definitions of Data Types

Ambiguity. A sample is labeled as Ambiguity if there exist multiple valid interpretation paths $\{\pi_1, \pi_2\}$ over (T, Q) such that:

$$\pi_1 \neq \pi_2, \quad A_{\pi_1} \neq A_{\pi_2}, \tag{1}$$

and both interpretations are semantically consistent with the table and question. In this case, the answer is not uniquely identifiable without external information.

Multi-hop Reasoning. A question is labeled as Multi-hop if solving it requires more than three interdependent reasoning steps. Let a reasoning chain be represented as a sequence of operations:

$$\mathcal{R} = (o_1, o_2, \ldots, o_n), \tag{2}$$

where each operation o_i depends on the output of previous steps. The sample is classified as Multi-hop if $n > 3$. These operations may include filtering, aggregation, comparison, and cross-row or cross-column interactions.

Transposed. A table is Transposed if its schema deviates from the standard row-entity and column-attribute layout, entities appearing as columns.

Merged Cells. Tables containing visually or structurally merged cells, where semantic information spans multiple rows or columns and requires reconstruction of implicit hierarchical structure before correct reasoning.

Missing Values. Tables containing explicit missing entries (e.g., blanks or placeholders), where the question requires reasoning or filtering under incomplete information.

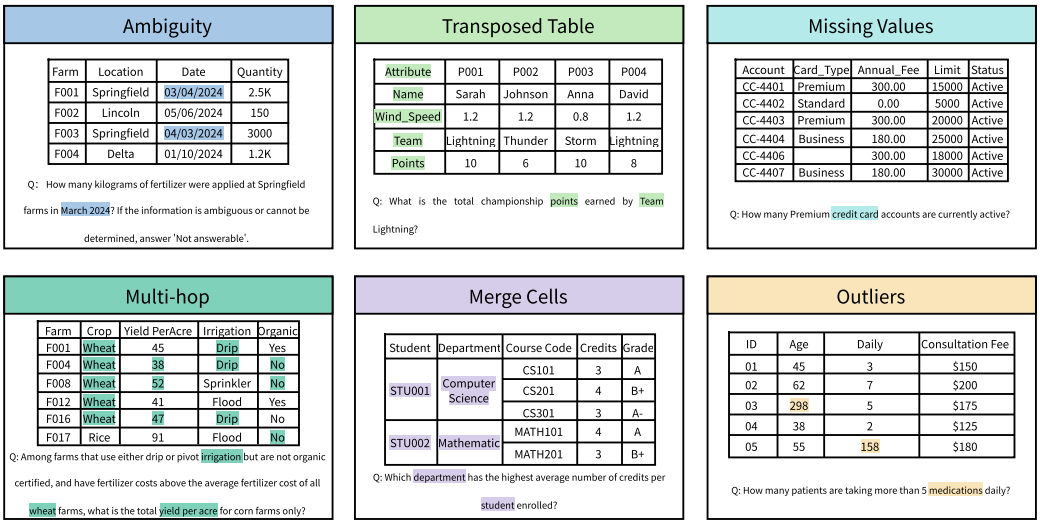


Figure 1: Six Types from realistic TableQA (Ambiguity, Transposed Table, Missing Values, Multi-hop reasoning, Merge Cells, and Outliers) in CompTab.

Outliers. Tables containing values that significantly deviate from the local statistical distribution, where the question involves reasoning based on distributional properties or threshold judgments.

Category Overlap. The six data types are not mutually exclusive, and a sample may exhibit multiple challenges. We adopt a primary-type annotation principle, assigning each sample a dominant category based on its primary source of difficulty. For instance, Multi-hop emphasizes reasoning complexity, while other categories focus on structural or data irregularities. This design is motivated by the observation that most existing benchmarks focus on well-structured tables, where reasoning complexity (e.g., multi-hop) is the dominant difficulty. In contrast, real-world tables often involve irregular structures and imperfect data.

2.2 Seed Collection

To ground the benchmark in realistic scenarios, we curate a collection of seed table–question pairs spanning 12 domains, as illustrated in Figure 4.

The tables are sourced from public reports, web pages, and internal documents, and often exhibit non-standard structures such as missing entries, merged cells, transposed layouts, or extreme values. All tables are anonymized, with personal identifiers and sensitive information removed to ensure privacy compliance. The accompanying questions are derived from real user needs or practical analysis scenarios, involving aggregation, comparison, multi-step reasoning, and semantic ambiguity.

Each seed pair is manually annotated with a single category to align with a targeted complexity or irregularity type, with emphasis on structural validity, semantic clarity, and answer correctness. For each category, we additionally prepare a small set of canonical seed questions to guide the subsequent controlled generation process.

2.3 Table and Question Generation

To scale the dataset while preserving type-specific characteristics, we adopt a controlled LLM-based generation strategy. Starting from the seeds, we design category-specific prompts and employ Claude-sonnet-4-2025051 to generate table–question pairs. Each prompt specifies the target challenge type and includes annotated examples, guiding the model to produce tables and questions with the desired structural properties. Each generation round yields one table with three associated questions.

The generation process follows two constraints. *Domain consistency* requires generated tables to remain within the semantic scope of the original domain, preserving realistic entities and numerical patterns. *Type fidelity* enforces that each table exhibits the defining characteristics of its assigned complexity or irregularity type, such as inferable missing values, identifiable outliers, or plausible matches in ambiguous cases.

For quality control, we randomly sample 10% of the generated data for manual inspection and remove instances with structural errors, unanswer-

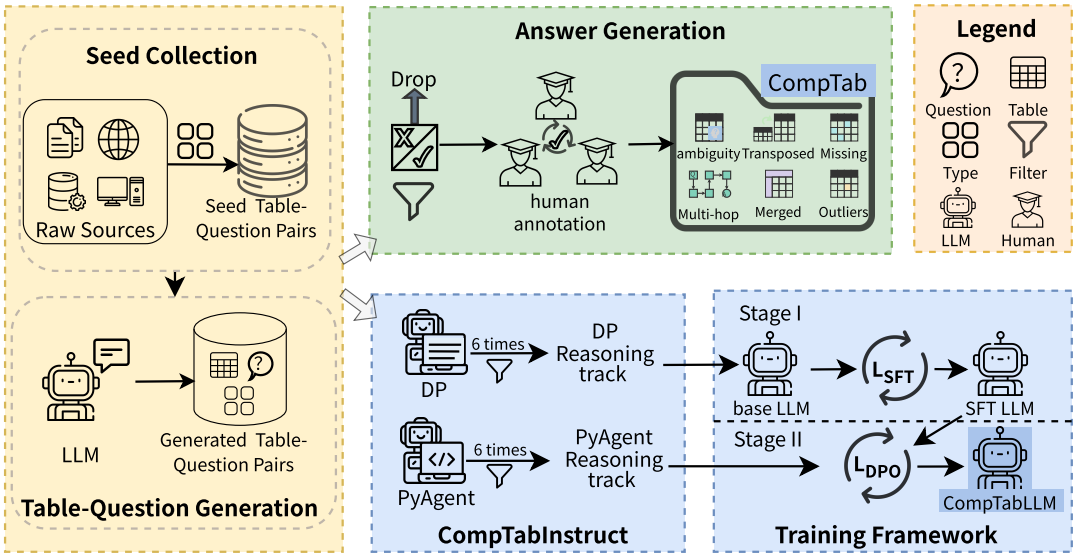


Figure 2: An overview of the CompTab benchmark construction and the CompTabLLM training framework.

able questions, or incorrect type labels. This process results in a curated benchmark that balances diversity, realism, and annotation reliability.

2.4 Answer Generation

To ensure reliable evaluation, we adopt a rigorous expert annotation protocol. Answers are provided by 36 graduate-level annotators with domain expertise. Each question is independently solved by two annotators, and disagreements are resolved through discussion with a third annotator. This process is iterated over multiple rounds until consensus is reached, resulting in a high-confidence ground-truth set.

Overall, five rounds of cross-validation and refinement are conducted to ensure correctness and consistency. By combining LLM-based generation with multi-stage human verification, CompTab scales while maintaining fidelity to real-world complexity and annotation reliability.

2.5 Dataset Statistics

Compare with other datasets. The final dataset contains 614 tables and 1,517 question–answer pairs. As shown in Table 1, most prior benchmarks focus on clean or mildly structured tables (Pasupat and Liang, 2015; Iyyer et al., 2017; Chen et al., 2019; Nan et al., 2022; Chen et al., 2021; Katsis et al., 2022; Zhong et al., 2017; Yu et al., 2018; Li et al., 2023) and cover only a limited subset of real-world types (He et al., 2024; Wu et al., 2025d; Zhang et al., 2025a; Wu et al., 2025c). In contrast, CompTab provides a more comprehensive coverage of six major types of table irregularities, enabling systematic evaluation of LLM performance under diverse structural distortions.

Datasets	AM	TT	MV	MH	MC	OU
WTQ	✗	✗	✗	✗	✗	✗
SQA	✗	✗	✗	✗	✗	✗
TabFact	✗	✗	✗	✗	✗	✗
FeTaQA	✗	✗	✗	✓	✗	✗
FinQA	✗	✗	✗	✓	✗	✗
AIT-QA	✗	✗	✗	✓	✗	✗
WikiSQL	✗	✗	✗	✓	✗	✗
Spider	✗	✗	✗	✓	✗	✗
Bird	✗	✗	✗	✓	✗	✗
Text2Analysis	✗	✗	✗	✓	✗	✓
TableBench	✗	✗	✗	✓	✗	✓
T2R-bench	✗	✓	✗	✓	✓	✗
RealHiTBench	✗	✗	✗	✓	✓	✓
CompTab	✓	✓	✓	✓	✓	✓

Table 1: Comparison with existing datasets in categories. AM denotes Ambiguity, TT denotes Transposed Table, MV denotes Missing Value, MH denotes Multi-hop, MC denotes Merge Cells, and OU denotes Outliers.

Types and Complexity. Figure 3 presents the type distribution and average reasoning steps in CompTab. Although the generation pipeline targets balanced coverage, the final distribution is mildly skewed by quality control, with Ambiguity and Missing Value less frequent due to stricter filtering. The average reasoning steps reflect the intrinsic

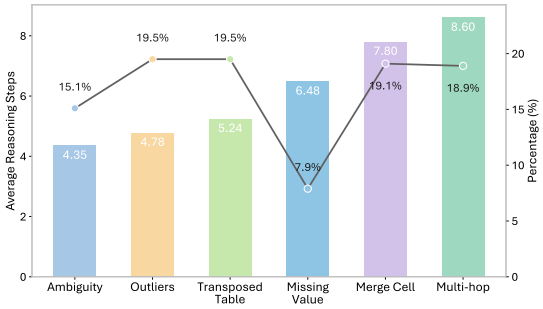


Figure 3: Types and Complexity across CompTab.

complexity of each type, with Multi-hop and Merge Cell involving longer reasoning chains due to multi-step aggregation and structural inference. Overall, CompTab covers diverse table phenomena with varying levels of reasoning complexity.

Domains. Figure 4 presents the domain distribution. Manual inspection filters out samples with domain mismatch or unrealistic numerical patterns. As a result, domains such as healthcare, agriculture, science, transportation, education, and finance each occupy roughly 8% of the dataset, with minor variation. This distribution preserves broad domain coverage while maintaining strict quality standards.

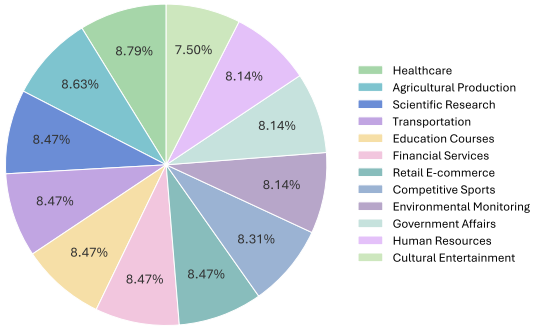


Figure 4: Domain distribution across CompTab.

3 Training Framework and CompTabLLM

To improve LLM generalization under complex and irregular TableQA settings, we propose a two-stage training framework based on CompTabInstruct. We refer to the resulting adapted model as CompTabLLM, as illustrated in Figure 2.

3.1 Instruction Corpus CompTabInstruct

We construct an instruction corpus to provide the two complementary supervision signals required by the framework. The underlying table-question pairs follow the same seed collection and controlled generation pipeline as CompTab, ensuring coverage across the six categories and diverse table structures. Instead of relying on human annotation, we automatically generate instruction trajectories using two reasoning paradigms.

Direct Prompting (DP). DP prompts the LLM to generate a natural-language reasoning trajectory followed by a final answer:

$$\text{DP:} \quad M(T, Q) \rightarrow \{r_1, r_2, \ldots, r_k, A^{DP}\} \quad (3)$$

where T denotes the input table, Q the question, $\{r_i\}$ the intermediate reasoning steps, and A^{DP} the final textual answer.

Python Agent (PyAgent). PyAgent prompts the LLM to generate an executable Python program operating on the table:

$$\text{PyAgent:} \quad M(T, Q) \rightarrow P \xrightarrow{\text{exec}} A^{Py} \quad (4)$$

where P consists of table-access and computation operations (e.g., lookup and aggregation), and A^{Py} is the execution result. The generated code assumes the table is represented as a pandas DataFrame and is executed in a sandboxed environment with a fixed setup to ensure safety and reproducibility.

Together, these paradigms provide complementary supervision for linguistic reasoning and executable table operations.

For each table–question pair (T, Q), both paradigms are executed independently. DP and PyAgent are each run six times, yielding $\{(R_i^{DP}, A_i^{DP})\}_{i=1}^{6}$ and $\{(P_i, A_i^{Py})\}_{i=1}^{6}$, respectively. To reduce noise from stochastic generation and execution errors, we retain only instances for which at least four out of six runs produce consistent outputs. While this filtering does not guarantee perfect correctness, prior work shows that structured reasoning traces can still improve accuracy and generalization even with imperfect answers (Gandhi et al., 2025). The resulting DP and PyAgent together constitute CompTabInstruct, which is used in the two-stage training framework.

3.2 Two-Stage Training Framework

Building on CompTabInstruct, we propose a two-stage training framework to obtain CompTabLLM. The framework progressively aligns LLMs with textual DP reasoning patterns and executable PyAgent decision processes, as illustrated in Figure 2. We instantiate this framework on two

representative base models, Llama3.1-8B and Qwen3-8B, yielding $\text{CompTabLLM}_{\text{Llama3.1-8B}}$ and $\text{CompTabLLM}_{\text{Qwen3-8B}}$. These models are chosen as they represent two widely used and architecturally distinct open-source LLM families.

Stage I: Supervised Fine-Tuning on DP Data. We first perform supervised fine-tuning using the DP subset of CompTabInstruct. The training objective maximizes the likelihood of the generated reasoning trajectory and the final answer:

$$\mathcal{L}_{\text{SFT}} = -\sum_{(T,Q,R^{DP},A)} \log P_\theta(R^{DP}, A \mid T, Q) \tag{5}$$

This stage encourages LLM to acquire stable and interpretable natural-language reasoning behaviors.

Stage II: Direct Preference Optimization on PyAgent Data. While supervised fine-tuning promotes coherent reasoning, it does not explicitly discourage reasoning paths that lead to incorrect computations. To further align the model with verifiable outcomes, we apply Direct Preference Optimization (DPO) on the PyAgent subset. For each question, preferred and rejected traces are constructed based on execution correctness:

$$\begin{aligned} s^+ &= \log P_\theta(R^+_{\text{Py}} \mid T, Q) \\ s^- &= \log P_\theta(R^-_{\text{Py}} \mid T, Q) \\ \mathcal{L}_{\text{DPO}} &= -\mathbb{E}\big[\log \sigma\big(\beta(s^+ - s^-)\big)\big] \end{aligned} \tag{6}$$

where $\sigma(\cdot)$ denotes the sigmoid function and β controls the preference margin. This stage further guides the model toward reasoning trajectories that lead to correct and executable solutions.

4 Experiments

4.1 Evaluation Metrics

Following the classical setting in WikiTableQuestions (Pasupat and Liang, 2015), TabFact(Chen et al., 2019) and prior LLM-based TableQA works (Yang et al., 2025b,c; Zhang et al., 2024), we design the evaluation to locate answers either in table or as the result of a numerical computation. Therefore, we adopt exact match, which is judged correct if and only if it exactly matches the gold answer.

4.2 LLMs

We evaluate 38 LLMs with different sizes, including general/code LLMs, open-source/closed-source LLMs. For open-source LLMs, we evaluate on Qwen2s (Team et al., 2024), Qwen3s (Yang et al., 2025a), QwQ-32B [1], Llama2s (Touvron et al., 2023), Llama3s (Dubey et al., 2024), Llama3.1s, CodeLlamas (Roziere et al., 2023), Deepseek-Coders (Guo et al., 2024), Deepseek LLMs (Bi et al., 2024), Kimi-K2 (Team et al., 2025), MiniMax-M1 (Chen et al., 2025), WizardLM (Xu et al., 2023), Gemma [2], Mistral (Bressand et al., 2023) and StructLMs (Zhuang et al., 2024). For closed-source LLMs, we choose GPTs (Brown et al., 2020; Achiam et al., 2023) (GPT4, GPT5, o1), GLMs (GLM et al., 2024) (GLM 4.6, GLM 4.7) , Claude-4 [3], ERNIE-4.5-Turbo [4] and Geminis (Team et al., 2023) (Gemini 2.5, Gemini 3). Furthermore, we finetune CompTabLLM based on Llama3.1-8B and Qwen3-8B to further explore the TableQA capabilities of LLMs.

4.3 Experimental Setup

All LLMs are evaluated using unified DP and PyAgent templates to ensure fair comparison. Experiments on open-source LLMs are conducted on eight NVIDIA RTX 4090 GPUs, while closed-source LLMs are accessed via their official APIs.

For supervised fine-tuning on CompTabInstruct, we use the Adam optimizer with a batch size of 512 and a maximum sequence length of 4096. Training follows a cosine annealing schedule with an initial learning rate of 3×10^{-5} for 5 epochs. For DPO, we set the learning rate to 1×10^{-5} with $\beta = 0.3$ and train for 5 epochs. These hyperparameters are applied consistently across all LLMs.

4.4 Main Results

Closed-source LLMs. In Table 2, we observe consistent performance gaps between DP and PyAgent, with DP achieving higher accuracy on structurally difficult tables. Closed-source LLMs exhibit the strongest overall performance. Gemini-3-pro-preview attains the highest average accuracy and remains stable across categories, followed closely by GLM4.7, GPT-4, and GPT-5.

Open-source LLMs. Open-source LLMs show substantially larger performance variance. Models such as DeepSeek-V3, DeepSeek-R1, Qwen3-32B, and QwQ-32B achieve the strongest results, while smaller models (e.g., Qwen2-7B and Llama3-8B) perform poorly, highlighting the importance of model scale. PyAgent reasoning is notably un-

[1] huggingface.co/Qwen/QwQ-32B-Preview
[2] ai.google.dev/gemma
[3] www-cdn.anthropic.com
[4] yiyan.baidu.com

Table 2: Exact match (%) of different LLMs on CompTab across six categories.

Model	Ambiguity		Transposed		Outliers		Merge		Missing		Multi-hop		Avg.
	DP	PyAgent	DP	PyAgent	DP	PyAgent	DP	PyAgent	DP	PyAgent	DP	PyAgent	
Human Performance	99.34		92.11		75.43		74.12		94.08		60.34		82.57
Close-source In Context Learning Methods													
GPT-4	32.16	29.59	35.99	30.76	16.14	14.20	42.84	54.15	39.47	22.37	50.62	42.36	34.22
GPT-5	33.36	42.22	38.60	11.30	54.96	18.92	63.82	18.39	44.74	11.92	58.89	14.15	34.27
GPT-o1	36.18	46.05	39.18	9.65	57.51	11.54	63.82	11.47	44.74	9.21	55.12	15.35	33.32
GLM 4.6	42.11	26.97	38.39	11.70	49.58	21.25	62.94	3.82	44.74	7.89	52.36	22.44	32.02
GLM 4.7	27.63	36.84	33.92	28.07	49.86	37.96	62.94	3.33	47.37	22.37	42.13	24.49	34.74
Claude-4	25.66	3.61	36.84	6.64	44.48	8.21	52.65	20.41	42.11	9.91	50.39	9.93	25.90
ERNIE-4.5-Turbo	30.26	29.61	28.36	4.97	28.33	8.78	31.47	0.88	26.32	14.47	30.31	23.23	21.42
Gemini-2.5	49.34	29.27	39.18	12.10	52.41	14.86	62.94	19.75	40.79	11.00	43.31	13.46	32.37
Gemini-3-pro-preview	13.82	21.05	33.92	25.44	50.00	41.64	63.53	16.18	46.05	51.32	42.68	22.05	**35.64**
Open-source In Context Learning Methods													
Qwen2-7B	5.11	2.74	8.50	4.95	4.84	9.56	6.46	0.00	5.18	5.80	2.00	6.03	5.10
Qwen2.5-7B	26.61	19.9	11.83	8.48	3.01	10.97	4.12	0.00	4.63	6.58	0.94	3.32	8.37
Qwen3-8B	22.37	34.87	32.75	7.31	49.01	12.75	56.76	10.71	35.53	6.58	47.24	10.53	27.20
Qwen3-14B	30.87	35.36	38.89	6.43	47.58	13.17	55.88	14.72	39.47	5.88	57.45	10.84	29.71
Qwen3-32B	34.79	36.86	39.77	8.06	55.12	13.88	61.04	14.55	46.05	6.58	57.06	14.15	32.33
QwQ-32B	43.24	38.29	51.17	7.02	55.82	14.58	65.88	14.55	46.05	6.58	49.03	12.89	33.76
Qwen3-Coder-480B	31.58	33.04	44.48	55.00	32.89	23.62	34.87	10.82	14.16	19.41	6.58	8.27	26.23
Llama2-7B	8.55	3.95	0.29	0.00	1.13	0.57	1.18	0.00	0.00	2.63	0.79	0.00	1.59
Llama3-8B	11.95	2.87	2.69	0.60	1.67	2.13	5.29	3.64	4.66	4.88	1.13	0.00	3.59
Llama3.1-8B	11.02	2.87	3.29	0.60	3.57	4.25	5.79	3.64	6.12	3.09	1.15	2.92	4.03
Llama3.1-70B	3.95	15.79	6.14	1.46	4.53	4.25	13.24	3.64	1.32	6.58	1.57	3.15	5.47
CodeLlama-7B	9.21	7.24	1.46	0.00	0.57	3.45	3.24	3.64	1.32	2.63	1.10	0.00	2.82
CodeLlama-13B	15.13	11.18	1.75	1.93	2.55	3.12	5.59	0.00	2.63	2.63	0.00	1.30	3.98
CodeLlama-34B	7.24	14.12	2.34	2.08	1.42	2.13	4.12	0.00	5.26	9.21	0.00	1.57	4.12
Deepseek-Coder-1.3B	8.55	8.70	0.29	0.00	1.70	3.40	3.24	6.18	1.32	1.32	0.79	1.18	3.06
Deepseek-Coder-6.7B	8.55	4.61	2.34	0.58	2.27	1.70	7.35	2.19	3.95	3.95	1.97	1.97	3.45
Deepseek-Coder-33B	17.11	19.74	9.94	3.22	4.53	8.78	13.24	9.41	15.79	9.21	1.97	3.94	9.74
DeepSeek-Coder-V2	6.58	3.95	6.14	3.22	10.48	10.76	15.59	9.41	6.58	6.58	3.54	5.12	7.33
Deepseek V3	35.53	32.24	39.77	14.33	45.04	27.76	60.88	49.41	36.84	25.00	44.49	37.80	**37.42**
Deepseek R1	44.08	46.05	38.30	9.36	53.82	15.30	61.76	6.47	44.74	9.21	62.20	12.20	33.62
Kimi-K2	23.03	24.34	33.33	10.53	44.19	15.30	59.12	17.06	42.11	9.21	46.06	11.02	27.94
MiniMax-M1	27.63	23.03	33.04	9.06	50.71	17.85	60.88	6.18	46.05	11.84	55.12	10.24	29.30
WizardLM-13B	11.18	7.24	1.75	0.58	2.55	0.28	4.12	0.00	5.26	1.32	0.39	0.00	2.89
Gemma-7B	23.03	3.29	1.75	0.29	1.70	1.98	2.35	0.00	1.32	3.95	0.00	1.57	3.44
Gemma2-9B	11.18	8.55	1.46	1.46	0.85	2.83	2.06	0.00	3.95	2.63	0.39	0.79	3.01
Gemma2-27B	8.55	7.24	1.46	1.17	2.27	2.55	6.18	0.00	2.63	0.00	0.00	3.28	2.94
Mistral-7B	1.97	4.61	4.09	1.46	1.70	4.25	13.24	8.24	2.63	6.58	1.57	2.36	4.39
StructLM-13B	3.29	0.00	0.58	0.29	0.00	3.97	1.18	0.00	1.32	7.89	0.00	2.36	1.74
StructLM-34B	0.66	1.32	0.00	0.58	0.00	3.68	0.29	0.00	0.00	3.95	0.00	7.09	1.46
Open-Source Fine-Tuning Methods													
$\text{CompTabLLM}_{\text{Llama3.1-8B}}$	19.74	19.08	29.82	6.73	25.78	13.60	50.59	12.65	30.26	6.58	22.44	7.09	20.36(↑**16.33**)
$\text{CompTabLLM}_{\text{Qwen3-8B}}$	44.74	44.74	32.75	8.19	47.88	16.43	60.29	8.82	40.79	5.26	56.30	18.11	32.03(↑**4.83**)

stable: even capable models suffer large drops on Missing and Transposed tables, primarily due to code generation errors, including syntax and logical mistakes when handling irregular table structures. DeepSeek models exhibit the largest DP–PyAgent divergence: despite strong DP accuracy comparable to closed-source LLMs, their PyAgent performance degrades sharply on several categories. Overall, robust Python-based table reasoning remains a key challenge for open-source LLMs.

CompTabLLM. CompTabLLM consistently improves over its base models across all challenge types, with particularly strong gains on Transposed, Missing, and Multi-hop categories. These results indicate that training on structurally diverse and irregular examples effectively enhances performance and mitigates the fragility of PyAgent execution.

4.5 Ablation Experiment

SFT Bridges the Distribution Gap, DPO Refines Decision Alignment. The ablation results in Table 3 confirm the complementary roles of SFT and

Table 3: Ablation results on CompTab.

Model	Ambiguity		Transposed		Outliers		Merge		Missing		Multi-hop		Avg.
	DP	PyAgent	DP	PyAgent	DP	PyAgent	DP	PyAgent	DP	PyAgent	DP	PyAgent	
CompTabLLM$_{\text{Llama3.1-8B}}$	19.74	19.08	29.82	6.73	25.78	13.60	50.59	12.65	30.26	6.58	22.44	7.09	20.36
w/o SFT	6.58	4.61	0.88	0.00	1.98	2.55	1.47	2.65	2.63	2.63	0.39	1.57	2.33
w/o DPO	15.13	19.08	26.61	4.09	23.80	11.33	49.12	11.47	26.32	7.89	18.50	7.87	18.43
CompTabLLM$_{\text{Qwen3-8B}}$	44.74	44.74	32.75	8.19	47.88	16.43	60.29	8.82	40.79	5.26	56.30	18.11	32.03
w/o SFT	21.71	19.08	14.91	8.07	23.51	12.75	33.24	7.74	14.47	6.58	5.91	5.42	14.45
w/o DPO	42.11	40.79	33.92	7.02	47.88	15.58	61.76	7.65	40.79	5.26	53.54	12.60	30.74

DPO in the two-stage framework. Removing SFT causes substantial performance degradation across all categories, reflecting a pronounced distribution shift introduced by CompTab. In contrast, removing DPO results in smaller but consistent drops, with average accuracy decreasing by 1.3–1.9 points (1.6 points on average) across the two backbones. This pattern aligns with the role of DPO as an agent-level alignment stage that refines decision quality after SFT-based distribution adaptation.

4.6 Fine-Tuning analysis

Figure 5 presents the training dynamics of our two-stage, where the LLM is first SFT on DP data and then optimized with DPO on PyAgent data.

SFT Stage. Across the 3k/6k/10k settings, all curves show rapid loss reduction and stable convergence. Larger SFT datasets lead to smoother trajectories: the 10k run is the most stable, while 3k exhibits mid-stage oscillations, indicating that scaling SFT data improves both fitting and initialization for preference optimization.

DPO Stage. All settings demonstrate consistent loss decreases, with the 3k run dropping fastest due to its smaller size. Reward accuracy quickly approaches 1.0 across all settings, whereas reward margin distinguishes dataset scales more clearly. The 10k setting achieves the highest and most stable margins, followed by 6k and 3k, suggesting that larger preference datasets help the LLM internalize preference intensity rather than merely separating chosen from rejected responses.

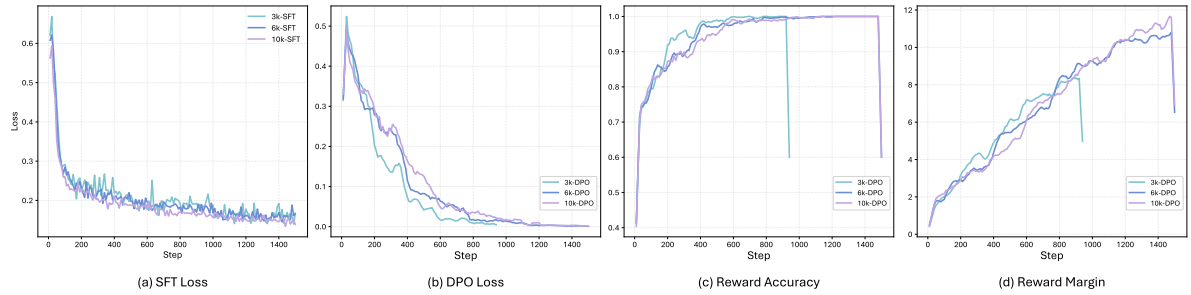
Overall, smaller datasets converge faster, while larger datasets produce smoother curves and stronger reward margins, underscoring the importance of data scale for stable optimization.

4.7 Performance on Question Type

Analysis by Question Type. Figure 6 groups questions by type for the two CompTabLLM variants and reveals several patterns. First, the base capability of each variant strongly influences its behavior across types, indicating that underlying LLM strength remains a primary factor. Second, DP performs better on total/sum and how many questions, as these tasks require global aggregation and benefit from holistic, context-driven reasoning over the entire table. Third, PyAgent achieves higher accuracy on conditional and what questions, since these tasks often reduce to condition filtering or record selection that Python execution handles precisely. Thus, DP favors global reasoning, while PyAgent excels at condition-based selection, and combining the two can further improve overall performance.

5 Related Work

Early TableQA datasets such as WTQ (Pasupat and Liang, 2015), TabFact (Chen et al., 2019) and related work (Zhong et al., 2017; Yu et al., 2018; Li et al., 2023; He et al., 2024; Iyyer et al., 2017; Zhang et al., 2023) have been widely studied, but their reasoning requirements remain relatively simple and do not reflect the complexity of real-world analytical tasks. Subsequent studies such as FeTaQA (Nan et al., 2022), AIT-QA (Katsis et al., 2022), and others (Wang et al., 2025c; Li et al., 2025; Qiu et al., 2024) have taken meaningful steps toward closing this gap by increasing reasoning complexity (Pramanick et al., 2024; Wu et al., 2025a; Zhou et al., 2024; Tang et al., 2024) and better approximating real analytical workflows (Ahmad et al., 2025; Ji et al.; Roychowdhury et al., 2024; Deng et al., 2024). However, most datasets still rely on clean and standardized tables and therefore fail to capture the irregular structures commonly found in practice. Recent benchmarks introduce 2-4 structures (He et al., 2024; Wu et al., 2025d; Zhang et al., 2025a; Wu et al., 2025c), but they do not cover other irregularities such as missing values, value outliers, or ambiguous entries. As a result, the gap between benchmark settings and real-world TableQA remains significant.



(a) SFT Loss (b) DPO Loss (c) Reward Accuracy (d) Reward Margin

Figure 5: Training curves of CompTabLLM, including SFT loss (a), DPO loss (b), reward accuracy (c), and reward margin (d) across different data scales.

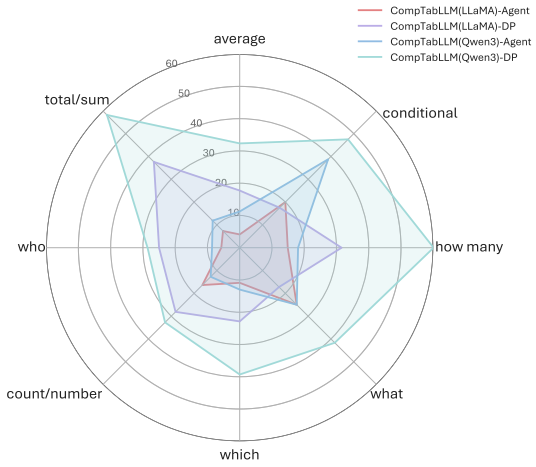


Figure 6: Accuracy of CompTabLLM on question types.

To address this gap, CompTab integrates six representative real-world TableQA types and provides a benchmark that jointly targets complex reasoning and irregular table structures.

6 Conclusion

We propose CompTab, a benchmark for TableQA under complex reasoning and irregular table, covering six representative real-world types. We further construct instruction data with textual and executable reasoning traces and propose a two-stage training framework to improve generalization in realistic scenarios. Evaluations show that LLMs still struggle with structural noise and complex reasoning, while CompTab provides a unified benchmark to support future research on robust TableQA.

7 Limitations

While CompTab provides a comprehensive benchmark for complex and irregular TableQA, several limitations remain. Although it covers six representative types, other challenges such as OCR artifacts, layout misalignment, and multi-table linking are not considered. In addition, our evaluation focuses on single-table, leaving multi-table and hybrid structured–unstructured reasoning for future work.

8 Ethical Considerations

All tables used in CompTab are collected from publicly available sources, authorized industry data, or produced through controlled LLM-based generation. All data are de-identified and manually reviewed to remove sensitive or proprietary information and ensure compliance with copyright and privacy requirements. Instruction data derived from LLM outputs may contain minor biases or inaccuracies; we apply systematic filtering and human verification to mitigate such issues. CompTabLLM is intended solely for academic research on table reasoning and should not be deployed in high-stakes decision-making scenarios. We encourage responsible use of the dataset and models.

9 Acknowledgements

Our work is supported by the National Natural Science Foundation of China (62476003), Anhui Province Excellent Scientific Research and Innovation Team (2024AH010004), Anhui Provincial Natural Science Foundation - Water Science Joint Fund (2408055US006), the University Synergy Innovation Program of Anhui Province (GXXT-2023-050), SMP-Zhipu.AI Large Model Cross-Disciplinary Fund (SMP-Zhipu20240210) and the National Natural Science Foundation of China (62576159). We also acknowledge the support from Zhipu AI-Anhui University Joint Research Center, and the High-Performance Computing Platform of Anhui University.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Mohammad S Ahmad, Zan A Naeem, MichaÃĢl Aupetit, Ahmed Elmagarmid, Mohamed Eltabakh, Xiasong Ma, Mourad Ouzzani, and Chaoyi Ruan. 2025. Hct-qa: A benchmark for question answering on human-centric tables. *arXiv preprint arXiv:2504.20047*.

Chahrazed B. Bachir Belmehdi, Abderrahmane Khiat, and Nabil Keskes. 2024. Predicting an optimal virtual data model for uniform access to large heterogeneous data. *DATA INTELLIGENCE*, 6(2):504–530.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, and 1 others. 2024. Deepseek llm: Scaling open-source language models with longtermism. *arXiv preprint arXiv:2401.02954*.

F Bressand, G Lengyel, G Lample, L Saulnier, and 1 others. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33:1877–1901.

Aili Chen, Aonian Li, Bangwei Gong, Binyang Jiang, Bo Fei, Bo Yang, Boji Shan, Changqing Yu, Chao Wang, Cheng Zhu, and 1 others. 2025. Minimax-m1: Scaling test-time compute efficiently with lightning attention. *arXiv preprint arXiv:2506.13585*.

Wenhu Chen, Hongmin Wang, Jianshu Chen, Yunkai Zhang, Hong Wang, Shiyang Li, Xiyou Zhou, and William Yang Wang. 2019. Tabfact: A large-scale dataset for table-based fact verification. In *International Conference on Learning Representations*.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan R Routledge, and 1 others. 2021. Finqa: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3697–3711.

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

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv e-prints*, pages arXiv–2407.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *arXiv preprint arXiv:2503.01307*.

Team GLM, Aohan Zeng, Bin Xu, Bowen Wang, Chenhui Zhang, Da Yin, Dan Zhang, Diego Rojas, Guanyu Feng, Hanlin Zhao, and 1 others. 2024. Chatglm: A family of large language models from glm-130b to glm-4 all tools. *arXiv preprint arXiv:2406.12793*.

Daya Guo, Qihao Zhu, Dejian Yang, Zhenda Xie, Kai Dong, Wentao Zhang, Guanting Chen, Xiao Bi, Yu Wu, YK Li, and 1 others. 2024. Deepseek-coder: When the large language model meets programming–the rise of code intelligence. *arXiv preprint arXiv:2401.14196*.

Tingxu Han, Zhenting Wang, Chunrong Fang, Shiyu Zhao, Shiqing Ma, and Zhenyu Chen. 2025. Token-budget-aware llm reasoning. In *Findings of the Association for Computational Linguistics: ACL*, pages 24842–24855.

Xinyi He, Mengyu Zhou, Xinrun Xu, Xiaojun Ma, Rui Ding, Lun Du, Yan Gao, Ran Jia, Xu Chen, Shi Han, and 1 others. 2024. Text2analysis: A benchmark of table question answering with advanced data analysis and unclear queries. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18206–18215.

Zixian Huang, Wenhao Zhu, Gong Cheng, Lei Li, and Fei Yuan. 2024. Mindmerger: Efficiently boosting llm reasoning in non-english languages. *Advances in Neural Information Processing Systems*, 37:34161–34187.

Mohit Iyyer, Wen-tau Yih, and Ming-Wei Chang. 2017. Search-based neural structured learning for sequential question answering. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1821–1831.

Xingyu Ji, Aditya Parameswaran, and Madelon Hulsebos. Target: Benchmarking table retrieval for generative tasks. In *NeurIPS 2024 Third Table Representation Learning Workshop*.

Yannis Katsis, Saneem Chemmengath, Vishwajeet Kumar, Samarth Bharadwaj, Mustafa Canim, Michael Glass, Alfio Gliozzo, Feifei Pan, Jaydeep Sen, Karthik Sankaranarayanan, and 1 others. 2022. Aitqa: Question answering dataset over complex tables in the airline industry. In *Proceedings of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies: Industry Track*, pages 305–314.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, and 1 others. 2023. Can llm already serve as a database interface? a big bench for large-scale database grounded text-to-sqls. *Advances in Neural Information Processing Systems*, 36:42330–42357.

Zheng Li, Yang Du, Mao Zheng, and Mingyang Song. 2025. Mimotable: A multi-scale spreadsheet benchmark with meta operations for table reasoning. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 2548–2560.

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

Linyong Nan, Chiachun Hsieh, Ziming Mao, Xi Victoria Lin, Neha Verma, Rui Zhang, Wojciech Kryściński, Hailey Schoelkopf, Riley Kong, Xiangru Tang, and 1 others. 2022. Fetaqa: Free-form table question answering. *Transactions of the Association for Computational Linguistics*, 10:35–49.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480.

Sohan Patnaik, Heril Changwal, Milan Aggarwal, Sumit Bhatia, Yaman Kumar, and Balaji Krishnamurthy. 2024. Cabinet: Content relevance-based noise reduction for table question answering. In *The 12th International Conference on Learning Representations*.

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

Zipeng Qiu, You Peng, Guangxin He, Binhang Yuan, and Chen Wang. 2024. Tqa-bench: Evaluating llms for multi-table question answering with scalable context and symbolic extension. *arXiv preprint arXiv:2411.19504*.

Sujoy Roychowdhury, Sumit Soman, HG Ranjani, Avantika Sharma, Neeraj Gunda, and Sai Krishna Bala. 2024. Evaluation of table representations to answer questions from tables in documents: A case study using 3gpp specifications. *arXiv preprint arXiv:2408.17008*.

Baptiste Roziere, Jonas Gehring, Fabian Gloeckle, Sten Sootla, Itai Gat, Xiaoqing Ellen Tan, Yossi Adi, Jingyu Liu, Romain Sauvestre, Tal Remez, and 1 others. 2023. Code llama: Open foundation models for code. *arXiv preprint arXiv:2308.12950*.

Xiangru Tang, Yiming Zong, Jason Phang, Yilun Zhao, Wangchunshu Zhou, Arman Cohan, and Mark Gerstein. 2024. Struc-bench: Are large language models good at generating complex structured tabular data? In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 12–34.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, and 1 others. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Qwen Team and 1 others. 2024. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2(3).

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Dong Wang, Xiyue Wang, Haoqi Zheng, and 1 others. 2025a. Tablecritic: Refine table reasoning via self-criticism and tool library. In *Proceedings of the Annual Meeting of the Cognitive Science Society*, volume 47.

Jing Wang, Shuo Zhang, and Runzhi Li. 2024a. Gate feature interaction network for relation prediction in knowledge graph. *DATA INTELLIGENCE*, 6(3):749–770.

Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. 2024b. Rethinking the bounds of llm reasoning: Are multi-agent discussions the key? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6106–6131.

Yuxiang Wang, Junhao Gan, and Jianzhong Qi. 2025b. Tabsd: Large free-form table question answering with sql-based table decomposition. *arXiv preprint arXiv:2502.13422*.

Zhensheng Wang, Wenmian Yang, Kun Zhou, Yiquan Zhang, and Weijia Jia. 2025c. Retqa: A large-scale open-domain tabular question answering dataset for real estate sector. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25452–25460.

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

Junde Wu, Jiayuan Zhu, Yuyuan Liu, Min Xu, and Yueming Jin. 2025b. Agentic reasoning: A streamlined framework for enhancing llm reasoning with agentic tools. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 28489–28503.

Pengzuo Wu, Yuhang Yang, Guangcheng Zhu, Chao Ye, Hong Gu, Xu Lu, Ruixuan Xiao, Bowen Bao, Yijing He, Liangyu Zha, and 1 others. 2025c. Realhitbench: A comprehensive realistic hierarchical table benchmark for evaluating llm-based table analysis. *arXiv preprint arXiv:2506.13405*.

Xianjie Wu, Jian Yang, Linzheng Chai, Ge Zhang, Jiaheng Liu, Xeron Du, Di Liang, Daixin Shu, Xianfu Cheng, Tianzhen Sun, and 1 others. 2025d. Tablebench: A comprehensive and complex benchmark for table question answering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25497–25506.

Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

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

Zhen Yang, Ziwei Du, Minghan Zhang, Wei Du, Jie Chen, Fulan Qian, and Shu Zhao. 2025c. Causality meets the table: Debiasing llms for faithful tableqa via front-door intervention. In *Advances in Neural Information Processing Systems*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, and 1 others. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921.

Daojian Zeng, Lin Zhou, Zhiheng Zhang, and Lincheng Jiang. 2025. Autogen: Automated tool learning data generation with domain-specific structured data. *DATA INTELLIGENCE*, 7(4):1108–1128.

Jie Zhang, Changzai Pan, Sishi Xiong, Kaiwen Wei, Yu Zhao, Xiangyu Li, Jiaxin Peng, Xiaoyan Gu, Jian Yang, Wenhan Chang, and 1 others. 2025a. T2r-bench: A benchmark for real world table-to-report task. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 22438–22462.

Jinghan Zhang, Xiting Wang, Weijieying Ren, Lu Jiang, Dongjie Wang, and Kunpeng Liu. 2025b. Ratt: A thought structure for coherent and correct llm reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 26733–26741.

Siyue Zhang, Luu Anh Tuan, and Chen Zhao. 2024. Syntqa: Synergistic table-based question answering via mixture of text-to-sql and e2e tqa. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 2352–2364.

Zhehao Zhang, Xitao Li, Yan Gao, and Jian-Guang Lou. 2023. Crt-qa: A dataset of complex reasoning question answering over tabular data. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2131–2153.

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

Wei Zhou, Mohsen Mesgar, Heike Adel, and Annemarie Friedrich. 2024. Freb-tqa: A fine-grained robustness evaluation benchmark for table question answering. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2479–2497.

Alex Zhuang, Ge Zhang, Tianyu Zheng, Xinrun Du, Junjie Wang, Weiming Ren, Stephen W Huang, Jie Fu, Xiang Yue, and Wenhu Chen. 2024. Structlm: Towards building generalist models for structured knowledge grounding. *arXiv preprint arXiv:2402.16671*.

A Dataset Case

We provide representative cases for each data type to illustrate the diversity and realism of the CompTab dataset.

Title: Farm Machinery Operation Logs

Machine_ID	Operation_Type	Hours_Operated	Fuel_Consumed_Liters	Maintenance_Due	Efficiency_Rating
TRC-001	Plowing	12.5	85.2	No	8.7
TRC-002	Harvesting	18.3	142.8	Yes	9.2
TRC-003	Seeding	9.8	67.4	No	7.9
TRC-004	Plowing	15.2	98.6	Yes	8.1
TRC-005	Irrigation	22.1	156.3	No	9.5
TRC-006	Harvesting	14.7	119.5	Yes	8.8
TRC-007	Fertilizing	11.3	76.9	No	8.4
TRC-008	Plowing	16.8	112.4	No	7.6
TRC-009	Seeding	13.4	89.7	Yes	8.9
TRC-010	Harvesting	19.6	135.2	No	9.1
TRC-011	Irrigation	25.3	178.4	Yes	9.3
TRC-012	Fertilizing	8.9	62.1	No	7.8
TRC-013	Plowing	17.4	125.8	Yes	8.6
TRC-014	Seeding	10.7	71.3	No	8.2
TRC-015	Harvesting	21.2	148.9	Yes	9.4
TRC-016	Irrigation	20.5	143.7	No	8.9
TRC-017	Fertilizing	12.8	84.5	Yes	8.3
TRC-018	Plowing	14.1	96.2	No	7.7

Question: Among machines that either require maintenance OR have efficiency ratings above 8.5 (but not both conditions), and excluding those used for irrigation, what is the average fuel consumption per hour for machines that operated more than 15 hours?

Reasoning:

1) Based on the condition "require maintenance OR have efficiency ratings above 8.5 (but not both conditions)", the machines that qualify are TRC-001, TRC-004, TRC-005, TRC-010, TRC-016, and TRC-017.
2) After applying the filter "excluding those used for irrigation", TRC-005 and TRC-016 are removed.
3) Next, applying "operated more than 15 hours" removes TRC-001 and TRC-017.Therefore, the machines that satisfy all criteria are TRC-004 and TRC-010.
4) For TRC-004, the fuel consumption per hour is calculated as 98.6 / 15.2 ≈ 6.49, and for TRC-010, it is 135.2 / 19.6 ≈ 6.90.The final average fuel consumption per hour is (6.49 + 6.90) / 2 ≈ 6.695.

Gold Answer: 6.695

Figure 7: Multi-hop case from the CompTab dataset.

Title: Credit Card Account Details

Account_ID	CC-4891	CC-7623	CC-2845	CC-9017	Account_ID
Cardholder_Name	Sarah Chen	Marcus Rivera	Elena Volkov	David Kim	Cardholder_Name
Card_Type	Platinum	Gold	Standard	Platinum	Card_Type
Credit_Limit	15000	8500	3000	12000	Credit_Limit
Current_Balance	4250.75	2180.40	890.25	7650.30	Current_Balance
Payment_Due_Date	2024-02-15	2024-02-18	2024-02-12	2024-02-20	Payment_Due_Date
Minimum_Payment	127.52	65.41	26.71	229.51	Minimum_Payment
Interest_Rate	18.99	21.49	24.99	17.99	Interest_Rate
Last_Payment_Amount	500.00	150.00	75.00	800.00	Last_Payment_Amount
Last_Payment_Date	2024-01-18	2024-01-22	2024-01-15	2024-01-25	Last_Payment_Date
Available_Credit	10749.25	6319.60	2109.75	4349.70	Available_Credit
Account_Status	Active	Active	Active	Active	Account_Status
Annual_Fee	195	95	0	195	Annual_Fee
Rewards_Points	8420	3250	1180	12750	Rewards_Points
Cash_Advance_Limit	7500	4250	1500	6000	Cash_Advance_Limit
Account_ID	CC-4891	CC-7623	CC-2845	CC-9017	Account_ID
Cardholder_Name	Sarah Chen	Marcus Rivera	Elena Volkov	David Kim	Cardholder_Name
Card_Type	Platinum	Gold	Standard	Platinum	Card_Type
Credit_Limit	15000	8500	3000	12000	Credit_Limit

Question: Which cardholders have the same annual fee structure?

Reasoning:

1) We need to compare the values in the Annual_Fee row.
2) Grouping the cardholders by identical annual fees yields the following: Annual Fee = 195: Sarah Chen (CC-4891) and David Kim (CC-9017), Annual Fee = 95: Marcus Rivera (CC-7623), Annual Fee = 0: Elena Volkov (CC-2845).
3) Therefore, Sarah Chen and David Kim share the same annual fee structure (both are $195).

Gold Answer: Sarah Chen, David Kim

Figure 8: Transposed case from the CompTab dataset.

Title: Daily Weather Station Readings

Station_ID	Location_s	Temperature_C	Humidity_%	Wind_Speed_kmh	Rainfall_mm
WS001	Downtown	22.5	68	12.3	2.1
WS002	Harbor District	24.1	72	15.8	0.0
WS003	Industrial Zone	21.8	65	9.7	1.5
WS004	Residential Area	23.2	70	11.2	3.2
WS005	Airport	25.6	58	18.4	0.5
WS006	University Campus	22.9	69	13.1	2.8
WS007	City Park	21.5	74	8.9	4.1
WS008	Shopping Mall	24.7	66	14.6	1.2
WS009	Hospital District	23.8	71	10.5	2.7
WS010	Sports Complex	22.1	67	16.2	0.8
WS011	Beach Front	26.3	78	22.1	0.0
WS012	Mountain Base	18.4	82	7.3	5.6
WS013	Business District	24.9	63	13.8	1.9
WS014	Suburban Mall	23.6	69	11.7	2.4
WS015	Train Station	25.1	65	17.9	1.1
WS016	Fire Station	22.7	73	9.4	3.5
WS017	Library	21.9	68	12.6	2.0
WS018	Factory District	456.8	64	14.3	1.7

Question: What is the average temperature recorded across all weather stations?

Reasoning:

1) The temperature recorded at WS018 is 456.8°C, which is clearly an outlier and therefore excluded from the calculation.
2) Sum all valid temperature values:22.5 + 24.1 + 21.8 + 23.2 + 25.6 + 22.9 + 21.5 + 24.7 + 23.8 + 22.1 + 26.3 + 18.4 + 24.9 + 23.6 + 25.1 + 22.7 + 21.9 = 442.7
3) Compute the average:442.7 / 17 = 23.3

Gold Answer: 23.3

Figure 9: Outliers case from the CompTab dataset.

Title: Farmland Weather Monitoring Data

Station_ID	Date	Temperature_C	Rainfall_mm	Wind_Speed	Location
FM001	03/04/2024	22.5	15.2	12	Springfield
FM002	04/03/2024	18.7	8.9	15	Delta
FM003	05/04/2024	25.1	0	8	Franklin
FM004	03/04/2024	19.8	22.7	18	Lincoln
FM005	06/04/2024	21.3	12.5	10	Springfield
FM006	04/03/2024	17.2	31.8	20	Center
FM007	07/04/2024	24.6	5.4	14	Delta
FM008	03/04/2024	20.9	18.3	16	Franklin
FM009	08/04/2024	23.7	2.1	9	Lincoln
FM010	04/03/2024	16.5	28.4	22	Zone A
FM011	09/04/2024	26.2	0.8	11	Springfield
FM012	03/04/2024	21.7	14.6	13	Delta
FM013	10/04/2024	19.4	25.9	19	Franklin
FM014	04/03/2024	18.1	33.2	21	Lincoln
FM015	11/04/2024	22.8	7.3	12	Center
FM016	03/04/2024	20.3	16.7	17	Zone A
FM017	12/04/2024	25.9	3.2	8	Springfield
FM018	04/03/2024	17.8	29.1	23	Delta

Question: What was the average temperature recorded on March 4th, 2024? If the date format is ambiguous and cannot be determined, please answer 'Not answerable'.

Reasoning: It is very difficult to confirm whether '04/03/2024' represents March 4th or April 3rd. The content of the table corresponding to this question is ambiguous and cannot be answered.

Gold Answer: 17

Figure 10: Ambuguity case from the CompTab dataset.

Title: Regional Livestock Production Summary

Farm Region	Livestock Type	Breed	Head Count	Monthly Milk Production (L)	Feed Cost ($)
Northern Valley	Cattle	Holstein	250	18500	3200
		Jersey	180	12600	2400
	Sheep	Merino	400	0	1600
		Romney	320	0	1280
Central Plains	Cattle	Holstein	320	24000	4100
		Angus	280	0	3500
	Goats	Nubian	150	2250	900
		Boer	120	1800	720
Southern Hills	Cattle	Holstein	200	15000	2800
		Hereford	240	0	3000
	Sheep	Dorset	350	0	1400
		Suffolk	290	0	1160
Eastern Coast	Cattle	Jersey	160	11200	2100
		Holstein	190	14250	2600
	Goats	Alpine	110	1650	660
Western Ridge	Cattle	Angus	300	0	3750
	Sheep	Corriedale	380	0	1520
		Leicester	220	0	880

Question: Which farm region has the highest total head count of milk-producing livestock across all breeds?

Reasoning:

1) The milk-producing livestock include Cattle and Goats, while Sheep do not produce milk (their milk production values are 0 in the table).
2) Summing the head counts of milk-producing livestock within each farm region gives the following results: Northern Valley: 250 + 180 = 430, Central Plains: 320 + 150 + 120 = 590, Southern Hills: 200, Eastern Coast: 160 + 190 + 110 = 460, Western Ridge: 0.
3) The highest total is achieved by Central Plains.

Gold Answer: Central Plains

Figure 11: Merged cell case from the CompTab dataset.

Title: E-commerce Product Reviews Dataset

Product_ID	Product_Name	Category	Rating	Review_Count	Price_USD
P001	Wireless Bluetooth Headphones	Electronics	4.5	1250	89.99
P002	Organic Cotton T-Shirt	Clothing	4.2	340	24.99
P003	Smart Fitness Watch	Electronics	4.7	2100	199.99
P004	Yoga Mat Premium	Sports		680	45.99
P005	Coffee Maker Deluxe	Home & Kitchen	4.1	890	129.99
P006	Running Shoes Pro	Sports	4.6	1560	119.99
P007	Wireless Phone Charger	Electronics	4.3	420	29.99
P008	Ceramic Dinner Set		4.4	310	79.99
P009	Bluetooth Speaker Portable	Electronics	4.8	1890	69.99
P010	Memory Foam Pillow	Home & Kitchen	4.0	520	39.99
P011	Gaming Mouse RGB	Electronics	4.5	780	49.99
P012	Stainless Steel Water Bottle	Sports	4.3		19.99
P013	LED Desk Lamp	Home & Kitchen	4.2	290	34.99
P014	Wireless Earbuds Pro	Electronics	4.6	1420	149.99
P015	Bamboo Cutting Board	Home & Kitchen	4.1	180	22.99
P016	Fitness Resistance Bands	Sports	4.4	650	16.99
P017	Smart Home Hub	Electronics	4.2	340	99.99
P018	Organic Face Cream	Beauty	4.3	240	32.99

Question: What is the total number of reviews for all Home & Kitchen products?

Reasoning:

1) Identify the products whose category is "Home & Kitchen": P005, P010, P013, and P015. In addition, P008 can be inferred to belong to "Home & Kitchen" based on its product name *Ceramic Dinner Set*.
2) Calculate the total Review_Count: 890 + 310 + 520 + 290 + 180 = 2190.

Gold Answer: 2190

Figure 12: Missing value case from the CompTab dataset.

B Joint Reasoning

Joint Reasoning via DP and PyAgent. As shown in Figure 13, voting exhibits consistent trends across LLMs. Larger models benefit most from DP-dominant configurations (5,3), while Agent-heavy voting (3,5) leads to sharp drops on Missing and Transposed tables due to execution sensitivity, especially for DeepSeek-R1/V3. Overall, balanced or DP-dominant voting improves performance, whereas over-reliance on Agent outputs amplifies errors under structural noise.

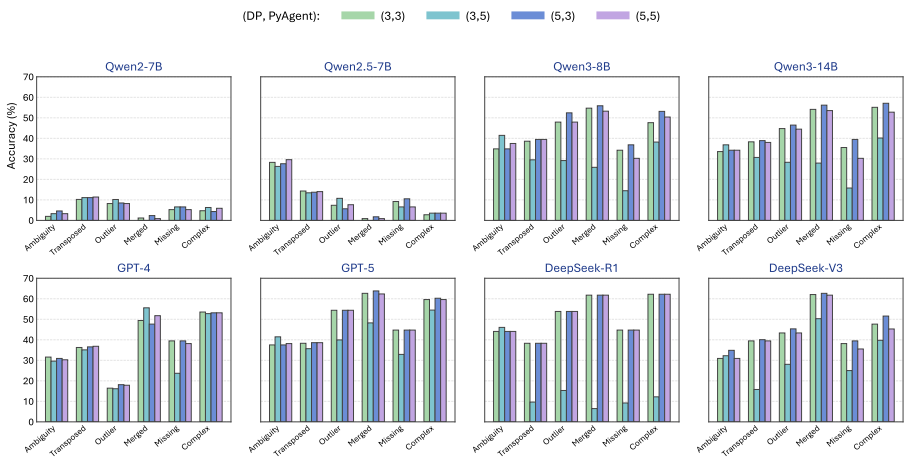


Figure 13: Performance of open-source and closed-source LLMs under different DP–PyAgent voting configurations.