# SubTabQA: Benchmarking Subtable Extraction in LLM-based Table Question Answering

Hanwen Zhang*, Jia Wang*, Peng Fu*(✉), Chuanyu Qin*, Zheng Lin*, Chao Xu‡, and Weiping Wang*
*Institute of Information Engineering, Chinese Academy of Sciences, Beijing, China
†School of Cyber Security, University of Chinese Academy of Sciences, Beijing, China
‡State Grid Jiangsu Electric Power Co., Ltd, Nanjing, China
{zhanghanwen, wangjia, fupeng, qinchuanyu, linzheng}@iie.ac.cn, boboxu1985@163.com, wangweiping@iie.ac.cn

***Abstract*—Table Question Answering (TQA) remains challenging for Large Language Models (LLMs) due to the abundance of noisy information in tables. A prevalent solution is the Decompose-Then-Reason (DTR) paradigm, which first extracts a relevant subtable before reasoning to produce an answer. As a critical prerequisite in DTR, SubTable Extraction (STE) not only simplifies downstream reasoning but also offers a lens into model behavior. However, there is currently a lack of dedicated benchmarks for fine-grained evaluation of STE. To address this, we introduce SubTabQA, the first benchmark dedicated to assessing STE across both flat and hierarchical tables. We further develop TableFocus, a specialized STE model trained via Supervised Fine-Tuning (SFT) and Reinforcement Learning with Verifiable Rewards (RLVR). Comprehensive experiments on SubTabQA reveal that strong TQA performance does not necessarily correlate with robust STE capabilities, highlighting the need for explicit evaluation of the decomposition stage. Notably, TableFocus achieves STE performance comparable to advanced large-scale and closed-source LLMs, demonstrating the effectiveness of our data and training strategy.**

***Index Terms*—Subtable Extraction, Table Question Answering, Large Language Models.**

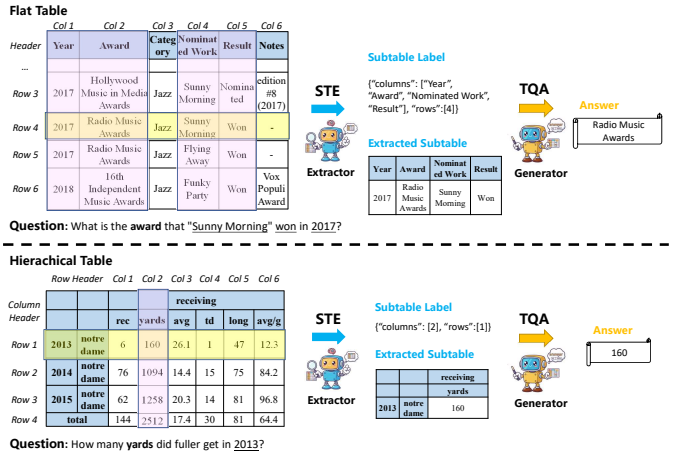


Fig. 1. Two examples from SubTabQA with different table structures, processed by the DTR pipeline: an extractor selects relevant rows (pink) and columns (yellow) to form a subtable, and a generator produces the final answer. Within the tables, light-blue and white cells denote headers and data entries, respectively.

## I. Introduction

Tables, as a ubiquitous two-dimensional form of structured data, pose unique challenges for NLP due to their intrinsic structural complexity and heterogeneous data types [14], [29], [30]. Recently, Large Language Models (LLMs) have made substantial progress on Table Question Answering (TQA), yet still struggle with large or complex tables containing abundant irrelevant information which acts as noise [12], [21]. To mitigate this, the Decompose-Then-Reason (DTR) paradigm is widely adopted [21], which first employs a subtable extractor to identify a question-relevant subtable from the full table, and then performs deeper reasoning with a downstream answer generator operating on the extracted subtable.

The critical subtable extraction step in the Decompose-Then-Reason paradigm faces two key challenges. First, there is no well-established criterion for optimal granularity, since compact subtables reduce noise but may omit relevant information. Second, existing benchmarks focus almost exclusively on final-answer accuracy and largely neglect evaluation of this intermediate step. Without such evaluation, it is difficult to determine whether model failures stem from inaccurate evidence localization or flawed reasoning, hindering deeper insight into model capabilities.

To address the gap above, we first propose a specific and verifiable extraction strategy of the "optimal subtable". Grounded in this unified strategy, we introduce **SubTabQA**, the first benchmark specifically designed to assess SubTable Extraction (STE) capabilities covering both flat and hierarchical tables. Fig. 1 shows two examples from SubTabQA, which supports a joint evaluation of STE and TQA. To facilitate further research, we also present **SubTabInstruct**, an instruction-tuning dataset used to train **TableFocus**, a robust STE baseline model developed through a hybrid strategy of Supervised Fine-Tuning (SFT) and Reinforcement Learning with Verifiable Rewards (RLVR), where RLVR employs a novel composite reward function. Our evaluation of diverse LLMs on SubTabQA shows that strong TQA performance does not ensure accurate subtable extraction, highlighting the need to explicitly evaluate the decomposition stage. Moreover, our specialized TableFocus model significantly outperforms its backbone, reaching performance on par with larger models. Crucially, integrating TableFocus into the DTR pipeline consistently enhances downstream TQA accuracy, highlighting the importance of accurate evidence localization in complex reasoning. Our main contributions are as follows:

1) We formalize subtable extraction as a well-defined task and introduce SubTabQA, the first benchmark dedicated to evaluating subtable extraction for table question answering, comprising 1.3K questions and 811 tables across two structural types.
2) We introduce SubTabInstruct, a large-scale instruction-tuning dataset for STE, and train TableFocus, a dedicated STE model based on an open-source LLM, using a hybrid SFT and RLVR strategy.
3) We conduct a comprehensive evaluation of mainstream LLMs on SubTabQA, revealing that strong TQA performance does not imply accurate subtable extraction. Empirically, our TableFocus based on Qwen3-4B achieves the performance comparable with top large-scale open-source or closed-source models (e.g. GPT-4.1) on STE task. Our code and dataset are available at: https://github.com/Heaven-zhw/SubTabQA-official

## II. Related Work

**LLM-based TQA Methods.** LLMs have significantly advanced table-based reasoning, with techniques like Chain-of-Thought (CoT) enhancing performance by breaking down complex problems [1], [4], [23], [31]. A straightforward approach involves serializing the entire table as input for end-to-end inference, but this method is impractical for large tables due to context length limitations and data redundancy [9], [21]. Consequently, decomposition-based reasoning methods have been increasingly explored, with STE emerging as a critical intermediate step. These methods employ various techniques: Dater [28] employs in-context learning to select relevant rows and columns; TabSQLify [12] converts tables into SQL databases to extract subtables via queries; Chain-of-Table [22] progressively refines the table through a series of chained operations; and PieTa [9] extracts key segments by processing the table in chunks. However, these approaches share critical limitations. They uniformly treat STE as a mere intermediate step to support the final question-answering task, rather than a distinct subtask worthy of in-depth study. This has led to a lack of formal definition and clear evaluation criteria for what constitutes a relevant subtable. Furthermore, existing extraction methods are predominantly designed for standard flat tables, lacking specific adaptations for tables with hierarchical structures.

**TQA Benchmarks.** The benchmarks for TQA have significantly evolved from simple to complex. Early datasets like WikiTQ [13], TabFact [2], and WikiSQL [32] are constructed from flat tables with relatively simple questions generated via templates or manual annotation, while HiTab [3] and AITQA [8] shift focus to hierarchical tables with complex header structures. More recently, researchers have leveraged LLMs to create more diverse and challenging TQA datasets targeting complex question types including data analysis, multi-hop reasoning, and chart generation [24], [26], [33]. Specialized datasets have also emerged to address long-table reasoning [20] and multi-scale spreadsheets [10]. Some datasets even provide CoT annotations to support research into model interpretability [24]. Despite these advances in table diversity and question complexity, existing benchmarks lack explicit annotations of the ground-truth subtable required to derive the answer. This absence has precluded the systematic evaluation of STE as an independent task, which impedes a deeper, process-level understanding of table reasoning. To bridge this critical gap, we introduce SubTabQA, the first benchmark designed for fine-grained evaluation of STE capabilities across diverse table structures and question types.

## III. SubTabQA Benchmark

### A. Subtable Extraction Task in TQA

**Task Definitions.** Given a structured table $T$ and a natural language question $Q$, the objective of TQA is to derive the answer $A$. Rather than producing the final answer directly, STE task aims to identify an **optimal** subtable $T' \subseteq T$, which contains all the information necessary to answer the question while excluding irrelevant content. Followed by the previous study [25], we represent the subtable as a set of rows and columns that can be concatenated to form a structurally complete table. Apart from flat tables, hierarchical tables with multi-level headers are also taken into account in our work. As illustrated in Fig. 1, we treat the table as table matrix and only select rows and columns from the data region, with the corresponding headers automatically included to reconstruct the subtable.

**Subtable Granularity.** Existing TQA methods involving extracting subtables employ inconsistent extraction granularities [9], [11]. To formulate a rigorous evaluation benchmark, it is essential to align the granularity of the "optimal" subtable. In this work, we propose a verifiable subtable extraction strategy based on the principle of decoupling evidence localization from statistical reasoning. Specifically, we exclusively apply explicit and directly verifiable row and column filters based on string matching and constant-based comparisons. When facing complex operations, such as counting, summation, averaging, deduplication, and sorting, we retain all relevant rows and columns involved, delegating the actual computation to the downstream answer generator.

### B. Data Construction

As shown in Fig. 2, the construction of SubTabQA follows a systematic pipeline comprising source data collection, LLM-based subtable annotation, and a meticulous filtering and re-labeling process involving both human and LLM verification.

**Data Collection.** To ensure diversity in table structures, topics, and question types, we collect table–question pairs from six public TQA datasets, followed by subtable annotation and verification. We consider two types of table structures: (1) flat tables, including WikiTQ [13], WikiSQL [32], TableBench [24], and MiMoTable [10], and (2) hierarchical tables, including HiTab [3] and AITQA [8]. We filter out tables with non-standard formats and convert all tables into data matrices, while preserving hierarchical row and column structures for hierarchical tables. To facilitate evaluation, all answers are standardized into a short-form format. Additionally, we sample

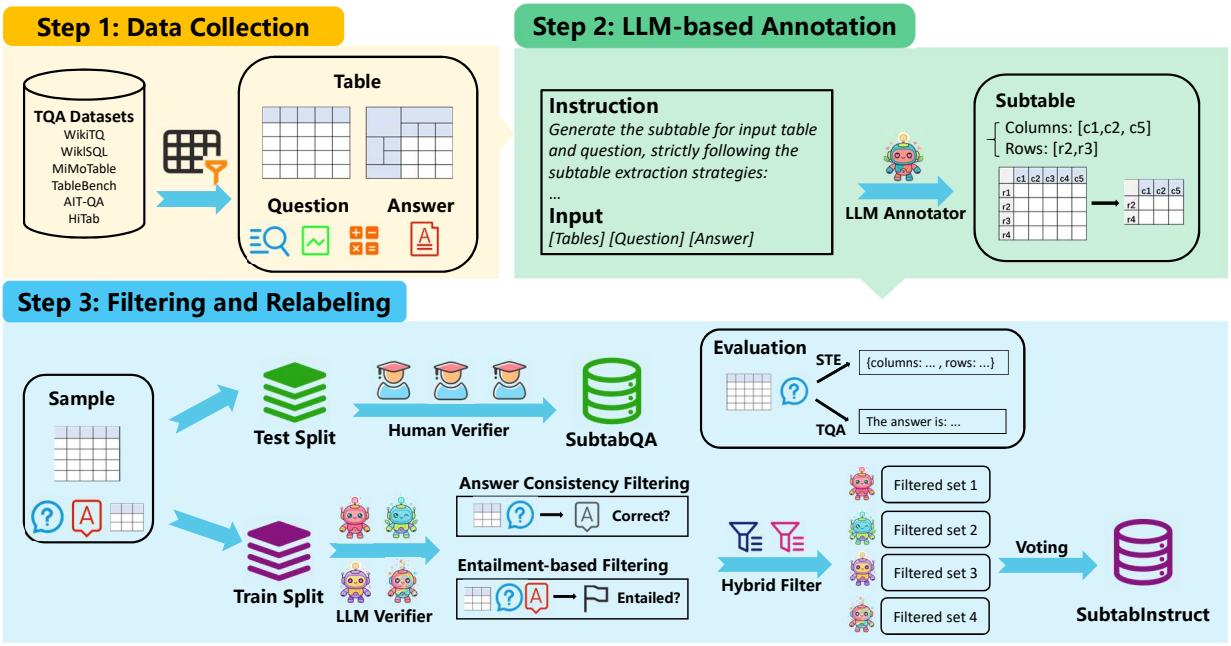


Fig. 2. Overview of the SubTabQA construction pipeline.

data from the training splits of WikiTQ, WikiSQL, and HiTab to construct an instruction-tuning dataset, selecting a limited number of questions per table to ensure broad table coverage.

**LLM-based Annotation.** We leverage Gemini-2.5-Flash to conduct initial annotation. The model is prompted with the subtable extraction strategy, table content, question, and corresponding answer to generate row-column annotations in JSON format. For flat tables, row–column positions are represented by row indices and column names, while for hierarchical tables, they are represented by row and column indices.

**Filtering and Relabeling.** After obtaining initial LLM annotations, we employ different processing strategies for the test and training sets to ensure data quality. (1) **Test Set Processing**: Each example is manually verified by three NLP postgraduates using LabelStudio[1] platform. Cases with non-standard table formats, incorrect answers, unsolvable questions, or lacking any answer-supporting subtable are discarded, and the remaining examples are relabeled with reference to the LLM annotations. Disagreements are resolved through multi-annotator discussion to reach consensus. (2) **Training Set Processing**: To construct the instruction-tuning dataset, we apply a hybrid multi-LLM filtering scheme. Specifically, four LLM verifiers[2] jointly perform two complementary filtering tasks: (i) *answer consistency filtering*, where each verifier answers the question based on the extracted subtable and checks whether the generated answer matches the ground-truth; and (ii) *entailment-based filtering*, where each verifier assesses whether the ground-truth answer is logically entailed by the subtable. Examples that pass both checks by majority voting are retained to form the final training set.

[1]https://labelstud.io/

[2]These LLM verifiers come from three model families: GPT-4.1 (nano), Qwen-3 (30B-A3B-Instruct-2507 and 32B), and Llama-3.3 (70B-Instruct).

### C. Data Statistics

Our SubTabQA benchmark contains 1371 test cases with 879 different tables, which are divided into two subsets: SubTabQA-std for flat tables, and SubTabQA-hier for hierarchical tables. Table I shows the detailed statistics of these two subsets. For SubTabInstruct, the curated corpus consists of 12,269 examples in total, with 4418 from WikiTQ, 3307 from WikiSQL, and 4544 from HiTab.

Additionally, we use GPT-4 to annotate the questions in SubTabQA with question types, categorizing them into seven classes: simple lookup, conditional lookup, sorting, statistics, numerical calculation, multi-hop question, and data analysis.

TABLE I
STATISTICS OF THE SUBTABQA DATASET. SUBTABLE AREA RATIO DENOTES THE PROPORTION OF THE GOLD SUBTABLE'S CELLS TO THOSE IN THE ORIGINAL TABLE. TOKEN LENGTHS ARE COMPUTED USING THE QWEN2.5-7B-INSTRUCT TOKENIZER.

| Statistics | SubTabQA-std | SubTabQA-hier | Total |
|---|---|---|---|
| # Samples | 879 | 492 | 1371 |
| # Tables | 591 | 220 | 811 |
| Avg. Question Length | 20.38 | 19.81 | 20.17 |
| Avg. Cell Length | 4.17 | 4.31 | 4.22 |
| Avg. Table Rows | 17.02 | 14.88 | 16.25 |
| Avg. Table Columns | 8.00 | 5.36 | 7.05 |
| Avg. Table Cells | 137.64 | 76.34 | 115.64 |
| Avg. Row Header Depth | - | 1.80 | - |
| Avg. Column Header Depth | - | 1.90 | - |
| Avg. Subtable Rows | 7.51 | 1.20 | 5.25 |
| Avg. Subtable Columns | 2.31 | 1.10 | 1.88 |
| Avg. Subtable Cells | 15.68 | 1.35 | 10.54 |
| Avg. Subtable Area Ratio (%) | 14.47 | 2.88 | 10.31 |

TABLE II
COMPARISON OF SUBTABQA WITH EXISTING DATASETS. "MD" INDICATES MULTI-DOMAIN, "L&N" INDICATES LOOKUP AND NUMERICAL, "R&A" INDICATES REASONING AND ANALYSIS.

| Dataset | Table | | | Question | | Annotation |
|---|---|---|---|---|---|---|
| | Flat | Hier | MD | L&N | R&A | SubT |
| WikiTQ | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ |
| WikiSQL | ✓ | ✗ | ✗ | ✓ | ✗ | ✗ |
| AITQA | ✗ | ✓ | ✗ | ✓ | ✗ | ✗ |
| HiTab | ✗ | ✓ | ✓ | ✓ | ✗ | ✗ |
| TableBench | ✓ | ✗ | ✓ | ✓ | ✓ | ✗ |
| MiMotable | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| TableEval | ✓ | ✓ | ✓ | ✓ | ✓ | ✗ |
| SubTabQA | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ |

We find that simple lookup, conditional lookup, and numerical calculation comprise the bulk of the dataset, accounting for 40.2%, 24.3%, and 17.8%, respectively, while multi-hop questions and data analysis are also represented as more challenging categories. This shows a broad coverage of question types, ranging from simple retrieval to complex reasoning.

In summary, we provide a concise comparison of SubTabQA with existing TQA benchmarks. As shown in Table II, SubTabQA supplies annotated relevant subtables for TQA and offers broader coverage of table structures, domains, and question types.

### D. Quality of Subtables

To validate the quality of our subtable annotations, we conduct a preliminary experiment on SubTabQA by comparing TQA performance under two settings: answering questions using the full table and using the gold subtable only. Table III reports exact match accuracy across different LLMs and benchmark subsets. Results show that using gold subtables consistently yields substantial performance gains over full tables, with improvements exceeding 10 accuracy points in several cases. These findings indicate that our subtable annotations preserve the essential information for answer inference and highlight subtable extraction as a crucial step for improving downstream TQA performance.

TABLE III
EXACT MATCH COMPARISON BETWEEN QA BASED ON FULL TABLE AND GOLD SUBTABLE.

| Method | SubTabQA-std | | | SubTabQA-hier | | |
|---|---|---|---|---|---|---|
| | Full | Gold | Δ | Full | Gold | Δ |
| Qwen2.5-7B-Instruct [17] | 58.25 | 67.46 | +9.21 | 81.10 | 94.92 | +13.82 |
| Llama-3.1-8B-Instruct [7] | 48.46 | 53.70 | +5.24 | 78.86 | 91.87 | +13.01 |
| Qwen2.5-32B-Instruct [17] | 68.26 | 77.36 | +9.10 | 88.41 | 94.72 | + 6.31 |
| Qwen3-32B [18] | 84.07 | 87.03 | +2.96 | 91.87 | 97.36 | +5.49 |
| Llama-3.3-70B-Instruct [7] | 72.92 | 78.61 | +5.69 | 88.41 | 96.14 | +7.73 |
| TableGPT2-7B [16] | 60.18 | 69.17 | +8.99 | 84.76 | 96.34 | +11.58 |
| Deepseek-V3 [5] | 78.04 | 80.89 | +2.85 | 92.28 | 96.54 | +4.26 |
| GPT-4.1 | 82.82 | 85.55 | +2.73 | 91.87 | 96.75 | +4.88 |
| Gemini-2.5-Flash | 83.28 | 85.10 | +1.82 | 92.68 | 97.36 | +4.68 |

## IV. TABLEFOCUS

To tackle the subtable extraction task, we introduce a specialized model named **TableFocus**. It is initialized from an open-source LLM and optimized through a two-stage training paradigm using our SubTabInstruct dataset.

### A. Supervised Fine-Tuning

The initial SFT stage is designed to teach the model the fundamental mechanics of the task, including instruction comprehension and adherence to a strict output schema. Each instance in the SubTabInstruct dataset consists of a natural language instruction (question and serialized table) as input and a target JSON string encapsulated within `<answer>` tags as the desired output. The training objective is to minimize the standard cross-entropy loss, equivalent to maximizing the log-likelihood of the ground-truth sequence.

### B. Reinforcement Learning with Verifiable Rewards

To further enhance STE performance, we apply reinforcement learning using the GRPO algorithm [15], which optimizes the policy by comparing groups of sampled outputs.

**GRPO Algorithm.** For each input $q$, GRPO samples $G$ candidate JSON outputs $\{o_i\}_{i=1}^{G}$ from the current policy $\pi_{\theta_{\text{old}}}$, and each output receives a reward $R_i$. The group-normalized advantage is computed as:

$$A_i = \frac{R_i - \text{mean}(\{R_j\}_{j=1}^{G})}{\text{std}(\{R_j\}_{j=1}^{G})} \tag{1}$$

The policy is then optimized to maximize rewards while constraining deviation from the reference policy via KL divergence regularization:

$$\max_{\pi_\theta} \mathbb{E}_{o\sim\pi_\theta(q)} \left[ R_{total}(q,o) - \beta \cdot \text{KL}[\pi_\theta(o|q) \| \pi_{\theta_{\text{old}}}(o|q)] \right] \tag{2}$$

where $\beta$ controls the regularization strength, $R_{total}$ is our composite reward function. This approach enables the model to learn from relative quality comparisons among generated subtable results without requiring a separate critic network.

**Composite Reward Function.** To promote effective optimization, we design a novel reward function specifically tailored for the STE task, which comprises three components. First, a **Format Reward** ($R_{\mathbf{F}} \in \{0, \eta\}$) acts as a syntactic gatekeeper, applying a large penalty $\eta$ (e.g., -1.0) if the output is not a valid JSON string enclosed within `<answer>` tags. If the format is correct, a **Validity Reward** ($R_{\mathbf{V}} \in \{0, 1\}$) then assesses semantic grounding by penalizing out-of-bounds indices or hallucinated column headers. Finally, the primary **Task Reward** ($R_{\mathbf{T}}$) is the task accuracy score EM between the extracted and the ground-truth subtable. These components are integrated into a single, comprehensive reward signal, $R_i$. The final reward is formally defined by the following piecewise function, which prioritizes syntactic correctness:

$$R_{total} = \begin{cases} R_{\text{F}} & \text{if } R_{\text{F}} < 0 \\ \alpha \cdot R_{\text{T}} + (1-\alpha) \cdot R_{\text{V}} & \text{otherwise.} \end{cases} \tag{3}$$

TABLE IV
STE PERFORMANCE OF DIFFERENT TYPES OF LLMS ON SUBTABQA. "✓" IN THE "REASONING" COLUMN INDICATES THAT THE MODEL IS A REASONING-ENHANCED LLM.

| Method | Reasoning | SubTabQA-std | | | | SubTabQA-hier | | | |
|---|---|---|---|---|---|---|---|---|---|
| | | EM | IOU | Precision | Recall | EM | IOU | Precision | Recall |
| **General LLMs** | | | | | | | | | |
| Qwen2.5-7B-Instruct | | 21.62 | 48.02 | 71.45 | 52.65 | 41.26 | 49.52 | 51.53 | 59.29 |
| Qwen3-4B-Instruct-2507 | | 51.31 | 70.77 | 90.56 | 72.05 | 59.96 | 63.60 | 65.65 | 67.78 |
| Llama-3.1-8B-Instruct | | 24.00 | 49.90 | 72.01 | 61.40 | 16.46 | 32.27 | 33.04 | 65.64 |
| Qwen2.5-32B-Instruct | | 53.24 | 73.92 | 86.61 | 77.48 | 72.76 | 77.42 | 79.26 | 83.27 |
| DeepSeek-R1-Distill-Qwen-32B | ✓ | 62.57 | 79.07 | 94.05 | 81.57 | 66.87 | 73.82 | 74.94 | 85.37 |
| QWQ-32B | ✓ | 77.13 | 84.91 | 89.42 | 86.60 | 80.49 | 82.42 | 82.84 | 83.98 |
| Qwen3-32B | ✓ | 75.20 | 86.92 | 93.90 | 88.47 | 48.58 | 63.80 | 64.38 | 80.92 |
| Llama-3.3-70B-Instruct | | 57.91 | 72.53 | 85.33 | 75.99 | 58.33 | 68.32 | 69.25 | 83.60 |
| Deepseek-V3 | | 73.49 | 86.14 | 95.45 | 88.59 | 83.13 | 87.05 | 88.04 | **92.76** |
| Deepseek-R1 | ✓ | **84.87** | **92.96** | **96.04** | **95.59** | **89.23** | **90.95** | **91.49** | 92.48 |
| **Tabular LLMs** | | | | | | | | | |
| TableGPT2-7B | | **33.56** | **59.04** | 84.72 | **65.71** | 58.94 | 65.43 | 67.77 | 74.74 |
| Table-R1-SFT-7B | ✓ | 25.03 | 37.40 | 54.15 | 38.38 | 60.77 | 62.78 | 64.43 | 63.93 |
| Table-R1-SFT-8B | ✓ | 24.69 | 42.31 | 69.15 | 44.01 | **77.24** | **81.00** | **84.07** | **82.91** |
| Table-R1-Zero-7B | ✓ | 30.72 | 53.41 | 78.98 | 58.01 | 45.12 | 55.90 | 57.82 | 68.94 |
| Table-R1-Zero-8B | ✓ | 28.78 | 51.52 | **87.43** | 54.22 | 59.15 | 66.46 | 69.58 | 73.91 |
| **Closed-source LLMs** | | | | | | | | | |
| GPT-4.1-nano | | 54.15 | 71.60 | 87.61 | 75.92 | 26.02 | 44.55 | 45.23 | 73.94 |
| GPT-4.1 | | 76.00 | 88.29 | 93.51 | 92.39 | 64.23 | 77.89 | 78.28 | **94.44** |
| Gemini-2.5-Flash | ✓ | **85.55** | **92.47** | **96.69** | **94.44** | **84.55** | **87.13** | **91.32** | 90.69 |
| **Ours** | | | | | | | | | |
| TableFocus-SFT(Qwen3-4B) | | 72.81 | 86.78 | 90.44 | 92.49 | 87.20 | 89.34 | 90.87 | 90.38 |
| TableFocus(Qwen3-4B) | | **73.27** | **87.35** | **90.95** | **93.01** | **87.60** | **89.72** | **90.91** | **91.54** |
| TableFocus-SFT(Llama3.1-8B) | | 68.03 | 83.23 | 86.78 | 90.95 | 87.20 | 89.48 | 91.27 | 90.21 |
| TableFocus(Llama3.1-8B) | | 69.28 | 83.92 | 87.92 | 90.80 | 87.40 | 89.69 | 91.32 | 90.69 |

In our experiments, we set the weighting factor $\alpha = 0.9$ to prioritize task accuracy. By maximizing the expectation of this reward, TableFocus learns to generate outputs that are not only syntactically correct but also highly accurate.

## V. EXPERIMENT

### A. Experimental Settings

**Dataset and Metrics.** We conduct all the experiments on SubTabQA benchmark. We use unified prompts for both STE and TQA tasks to guide LLMs in generating structured outputs. For STE, the prompt explicitly instructs the model to output the extracted subtable as a JSON object. We evaluate the performance using four automated metrics: Exact Match (EM), Intersection over Union (IOU), Precision, and Recall at cell-level. For TQA, LLMs are prompted to generate one or more entities as the answer to each question, and we utilize EM to assess the accuracy of the responses.

**Baselines.** To comprehensively evaluate model performance, we select a diverse set of baseline models from three categories: (1) Open-source generalist LLMs: We adopt a range of models with 4B-681B parameters, such as Qwen2.5 [17], Qwen3-Inst [18], Llama3 [7], and Deepseek-V3 [5]. In particular, some generalist LLMs are enhanced for reasoning through RL-based training or model distillation, including DeepSeek-R1-Distill-Qwen-32B [6], QwQ-32B [19], Qwen3-32B [18], and Deepseek-R1 [6]. (2) Tabular LLMs: This category comprises models that have undergone specialized SFT or RL-based training on table-related tasks, including TableGPT2 [16], Table-R1-SFT [27], and Table-R1-Zero [27]. (3) Closed-source LLMs: We evaluate leading proprietary models, including GPT-4.1-nano, GPT-4.1, and Gemini-2.5-Flash, to provide a broader comparison.

**Implementation Details.** We train TableFocus on SubTabInstruct using two backbones: Qwen3-4B-Instruct-2507 and Llama-3.1-8B-Instruct. All models are trained on four NVIDIA A800 GPUs. We perform SFT for 3 epochs with a learning rate of 1.0e-5 and a maximum sequence length of 4096, followed by GRPO for 3 epochs on a random 20% subset of the training data with a learning rate of 7e-7. By default, all experiments adopt the PIPE table format; for the STE task, we add explicit row/column numbers and convert multi-level hierarchical headers into a flattened representation to encourage valid generation.

### B. Subtable Extraction Results

Table IV presents a comprehensive evaluation of various LLMs on the STE task across two SubTabQA subsets. Overall, reasoning-enhanced models demonstrate superior performance, with Deepseek-R1 achieving the best overall results (84.87% on SubTabQA-std and 89.23% on SubTabQA-hier). Notably, our TableFocus, despite using only 4B/8B parameters,

TABLE V
TQA RESULTS ON THE SUBTABQA DATASET. WE COMPARE TWO MAIN APPROACHES: SINGLE-STAGE TQA BASED ON A FULL OR GOLD TABLE, AND DTR-PIPELINE TQA BASED ON DIFFERENT SUBTABLE EXTRACTORS AND THREE FIXED ANSWER GENERATORS.

| Method | SubTabQA-std | | | SubTabQA-hier | | |
|---|---|---|---|---|---|---|
| | *Answer Generator* | | | *Answer Generator* | | |
| | Llama-3.1-8B-Instruct | Table-R1-Zero-7B | Qwen2.5-32B-Instruct | Llama-3.1-8B-Instruct | Table-R1-Zero-7B | Qwen2.5-32B-Instruct |
| **Single-stage TQA** | | | | | | |
| Full Table | 48.46 | 65.98 | 68.26 | 78.86 | 85.77 | 88.41 |
| Gold Subtable | 53.70 | 75.65 | 77.36 | 91.87 | 94.11 | 94.72 |
| **DTR-pipeline TQA** | | | | | | |
| *Extractor(<10B)* | | | | | | |
| Qwen3-4B-Instruct-2507 | **53.24** | 69.85 | 69.28 | 79.88 | 85.57 | 84.76 |
| Llama-3.1-8B-Instruct | 43.00 | 55.40 | 53.70 | 67.07 | 73.17 | 73.17 |
| TableGPT2-7B | 49.94 | 69.85 | 67.69 | 83.13 | 87.40 | 85.77 |
| TableFocus(Qwen3-4B) | 51.65 | 69.97 | **73.49** | 83.94 | **88.01** | 87.20 |
| TableFocus(Llama3.1-8B) | 52.22 | **70.88** | 72.47 | **84.15** | 86.99 | **88.82** |
| *Extractor(>10B or Closed-source)* | | | | | | |
| Qwen2.5-32B-Instruct | 49.03 | 64.05 | 57.34 | 70.12 | 74.19 | 74.59 |
| QWQ-32B | 51.08 | 66.44 | 59.84 | 81.30 | 85.57 | 85.77 |
| Qwen3-32B | 49.94 | 66.55 | 61.09 | 71.54 | 73.78 | 73.58 |
| Deepseek-V3 | 53.47 | 73.49 | 73.72 | 85.37 | 89.02 | **90.65** |
| Deepseek-R1 | **54.49** | 73.83 | **77.02** | 85.57 | 89.84 | 89.23 |
| GPT-4.1 | 54.38 | 73.49 | 76.11 | **86.38** | **90.65** | 90.04 |
| Gemini-2.5-Flash | 54.95 | **73.95** | 75.88 | 85.57 | 88.01 | **90.65** |

achieves performance comparable to larger-scale models, with TableFocus-4B reaching 87.60% EM on SubTabQA-hier.

**Results across Model Categories.** Among General LLMs, performance strongly correlates with both model scale and reasoning capabilities. Large reasoning models like Deepseek-R1 and QwQ-32B substantially outperform their non-reasoning models of similar scale, demonstrating that STE requires not only table structure understanding but also complex reasoning abilities for accurate information localization. Tabular LLMs, however, show mixed results and generally underperform compared to general LLMs of similar scale. This suggests that current table pre-training approaches may prioritize direct question answering over the conditional localization and instruction-following capabilities required for STE. Closed-source LLMs demonstrate strong competitiveness, with Gemini-2.5-Flash achieving and GPT-4.1 showing robust performance across both subsets. Our TableFocus approach achieves remarkable performance through targeted SFT on SubTabInstruct, with GRPO training providing consistent improvements by reinforcing task-specific rewards and output constraints, enabling these small-scale models to rival their large-scale counterparts.

**Results across Data Subsets.** We observe that most methods achieve higher scores on SubTabQA-hier than on SubTabQA-std, despite the more complex hierarchical table structure. This counterintuitive result may be attributed to the different question distributions: SubTabQA-hier contains more simple lookup-based questions, while SubTabQA-std features diverse reasoning-intensive questions requiring complex operations. Furthermore, the evaluation metrics exhibit distinct patterns across subsets: on SubTabQA-std, most models show higher precision than recall, suggesting conservative predictions that prioritize accuracy; conversely, on SubTabQA-hier, many models achieve more balanced or even higher recall. This metric divergence likely reflects the extraction requirements for different question types—complex questions require preserving all necessary conditions for standalone answering, encouraging conservative selection, while simpler lookup questions allow for more straightforward cell identification. These patterns indicate that both question complexity and table structure jointly influence model strategies in STE.

### C. Table Question Answering Results

To validate the effectiveness of STE for DTR-pipeline TQA, we conduct experiments comparing single-stage TQA with the DTR paradigm using three different answer generators, as shown in Table V. We have the following two main findings:

**(1) The effectiveness of the DTR pipeline is highly dependent on the quality of the extractor.** On SubTabQA-std, some models fail to surpass single-stage TQA based on full table when serving as extractors. In contrast, our TableFocus, despite with only 4B/8B parameters, outperforms the baseline across all of generator configurations. Notably, TableFocus-4B achieves 73.49% EM when paired with Qwen2.5-32B generator, representing 5.23% absolute improvement over the baseline. On SubTabQA-hier, TableFocus similarly demonstrates strong performance, with TableFocus-8B reaching 88.82% EM when combined with Qwen2.5-32B Reader, approaching the results from Deepseek-V3 and GPT-4.1. These results demonstrate that small-scale LLMs specifically optimized for STE can effectively support the DTR paradigm.

**(2) The two TableFocus variants exhibit complementary advantages across data subsets.** TableFocus-4B achieves the highest EM of 73.49% on SubTabQA-std when paired with a strong answer generator. It potentially benefits from the strengths of Qwen3 in multi-step reasoning tasks. Conversely, TableFocus-8B performs slightly better on SubTabQA-hier across two reader configurations, which may be attributed to the strong structured information understanding ability of Llama3.1. More importantly, both TableFocus variants significantly outperform their original base models as extractors, validating the effectiveness of our training methodology.

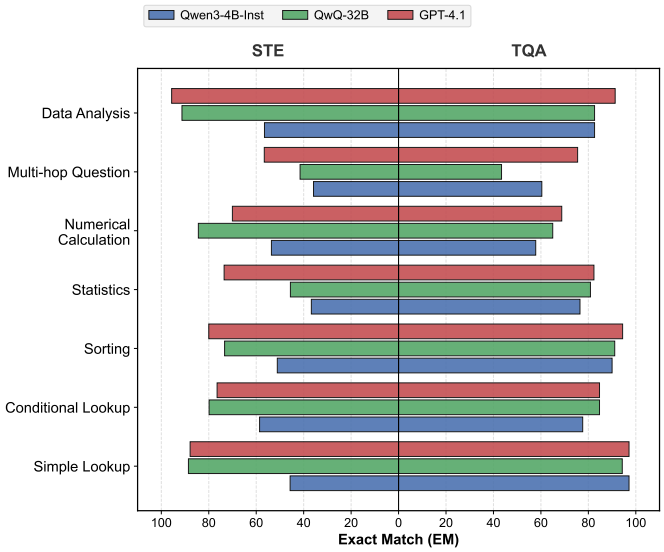


Fig. 3. EM scores of STE and TQA across question types on SubTabQA-std.

### D. Further Analysis

**Performance across Different Question Types.** We analyze the performance of three representative LLMs across different question types. Fig. 3 presents the EM scores for both STE and single-stage TQA on SubTabQA-std, which reveals two key insights as follows: (1) **LLMs show strong STE performance for simple lookup and data analysis questions but struggle with statistics and multi-hop questions.** Advanced models like QwQ-32B and GPT-4.1 excel at the former categories because they involve well-defined subtable boundaries. While simple lookup typically targets specific cells, data analysis requires complete rows/columns. In contrast, statistics and multi-hop questions present ambiguous subtable scopes and require intermediate reasoning during extraction, posing challenges even for strong models. (2) **Strong TQA performance does not guarantee precise STE.** While advanced models maintain balanced STE and TQA capabilities, weaker models exhibit significant disparities. For instance, Qwen3-4B achieves over 90% EM on simple lookup questions in TQA, yet below 50% EM in STE, suggesting a reliance on partial evidence or task-specific shortcuts rather than accurate evidence isolation. This decoupling indicates that models may answer correctly without precisely identifying the minimal relevant information.

**Ablation Study.** We conduct ablation studies to examine the effects of the reward function design and training strategy in TableFocus, with results summarized in Table VI. For reward components, removing valid reward degrades performance on both datasets. Notably, the simultaneous removal of format and valid rewards predominantly affects SubTabQA-std while leaving SubTabQA-hier nearly unchanged. This disparity reflects that cases with larger tables and complex questions rely more heavily on structural constraints to prevent illegal row/column output and cyclic generation. For training phases, removing GRPO results in modest performance drops, whereas skipping SFT causes catastrophic degradation. This substantial gap underscores SFT's critical role as a cold-start phase that establishes task understanding and provides a reasonable initial policy distribution. Overall, the ablation results demonstrate that the reward components address complementary failure modes, and that the two-stage training ensures both foundational capability and refined optimization.

TABLE VI
ABLATION STUDY OF TABLEFOCUS FOR STE TASK. WE ANALYZE THE IMPACT OF REMOVING KEY COMPONENTS FROM OUR REWARD FUNCTION AND TRAINING PHASE.

| Method | SubTabQA-std | | SubTabQA-hier | |
|---|---|---|---|---|
| | EM | IOU | EM | IOU |
| TableFocus(Qwen3-4B) | 73.27 | 87.35 | 87.60 | 89.72 |
| *Ablation on Reward Function* | | | | |
| w/o Valid | 72.58 | 86.82 | 87.40 | 89.59 |
| w/o Format+Valid | 72.47 | 86.42 | 87.60 | 89.72 |
| *Ablation on Training Phase* | | | | |
| w/o GRPO | 72.81 | 86.78 | 87.20 | 89.34 |
| w/o SFT | 58.02 | 77.39 | 72.15 | 75.47 |

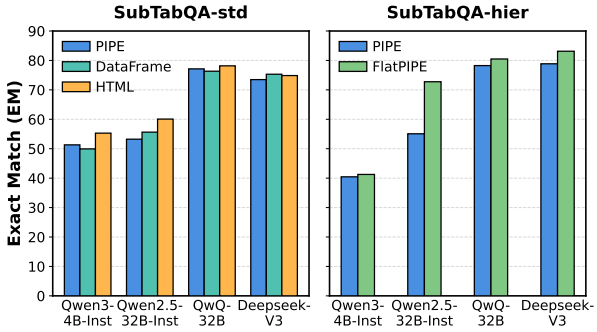


Fig. 4. The EM scores of STE on SubTabQA with different input table format. FlatPIPE denotes PIPE format with flattened multi-level headers.

**Impact of Input Table Format.** Fig. 4 compares different serialization formats for STE. For flat tables, HTML consistently outperforms PIPE and DataFrame across models, while the latter two perform similarly. We nevertheless use PIPE in the main experiments because HTML incurs substantially longer input sequences. For hierarchical tables, flattening multi-level headers (FlatPIPE) consistently improves over standard PIPE, suggesting that reduced structural complexity facilitates more accurate subtable boundary identification.

**Case Study.** To thoroughly validate TableFocus's effectiveness and demonstrate the rationality of "optimal subtable", we selected one example from each of both SubTabQA subsets for comparative analysis against baseline models, as shown in Figure 5. In case (a), while baseline models successfully locate the target values, they neglect essential conditional columns (e.g., Country and Year), resulting in an ambiguous subtable lacking row identifiers. In case (b), the baseline models fail to resolve the semantic mapping between the query keyword and row header, resulting in irrelevant row selection. Conversely, TableFocus accurately extracts subtables preserving both structural integrity and semantic relevance.

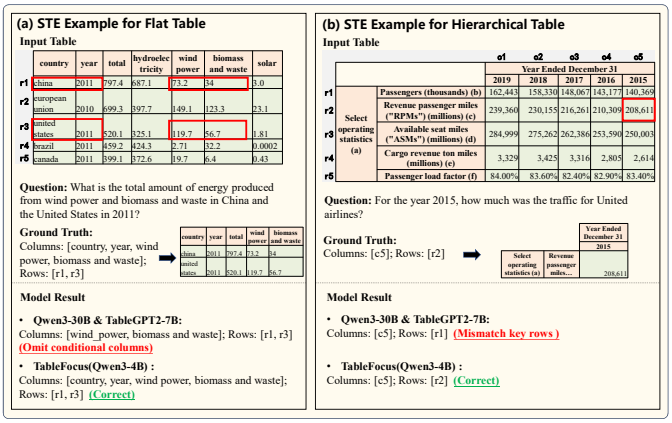


Fig. 5. Case study comparison between two baselines and TableFocus.

## VI. Conclusion

In this work, we address a critical gap in TQA evaluation by introducing SubTabQA, the first benchmark for SubTable Extraction (STE)—a key decomposition step in the Decompose-Then-Reason (DTR) paradigm, which supports both flat and hierarchical tables. To establish a strong baseline, we construct SubTabInstruct, a large instruction-tuning dataset for STE, and train TableFocus, a specialized extractor using a hybrid strategy of supervised fine-tuning and reinforcement learning with verifiable rewards. Our experiments reveal that strong TQA performance does not imply accurate STE, highlighting the necessity of explicitly evaluating the decomposition stage. Remarkably, TableFocus, despite its modest scale, matches the STE accuracy of leading large-scale and closed-source models. Moreover, integrating TableFocus into the DTR pipeline consistently improves downstream TQA performance, empirically confirming that precise evidence localization is foundational to effective table reasoning. Future work includes extending STE to broader table reasoning tasks, exploring program-aided extraction for enhanced robustness, and investigating joint optimization strategies within the DTR framework.

## Acknowledgment

This work was supported by the National Natural Science Foundation of China (Nos. 62472419, 62472420) and the Enterprise Project (No. E4V06811F3).

## References

[1] Chen, W.: Large language models are few(1)-shot table reasoners. In: Findings of ACL: EACL. pp. 1090–1100 (2023)
[2] Chen, W., Wang, H., Chen, J., Zhang, Y., Wang, H., Li, S., Zhou, X., Wang, W.Y.: Tabfact: A large-scale dataset for table-based fact verification. In: ICLR (2020)
[3] Cheng, Z., Dong, H., Wang, Z., Jia, R., Guo, J., Gao, Y., Han, S., Lou, J., Zhang, D.: Hitab: A hierarchical table dataset for question answering and natural language generation. In: Muresan, S., Nakov, P., Villavicencio, A. (eds.) ACL (2022)
[4] Cheng, Z., Xie, T., Shi, P., Li, C., Nadkarni, R., Hu, Y., et al.: Binding language models in symbolic languages. CoRR **abs/2210.02875** (2022)
[5] DeepSeek-AI: Deepseek-v3 technical report (2024), https://arxiv.org/abs/2412.19437
[6] DeepSeek-AI: Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning (2025), https://arxiv.org/abs/2501.12948
[7] Dubey, A., Jauhri, A., Pandey, A., Kadian, A., Al-Dahle, A., Letman, A., et al.: The llama 3 herd of models. CoRR **abs/2407.21783** (2024)
[8] Katsis, Y., Chemmengath, S., Kumar, V., Bharadwaj, S., Canim, M., Glass, M., et al.: Ait-qa: Question answering dataset over complex tables in the airline industry. arXiv preprint arXiv:2106.12944 (2021)
[9] Lee, W., Kim, K., Lee, S., Lee, J., Kim, K.I.: Piece of table: A divide-and-conquer approach for selecting sub-tables in table question answering. CoRR **abs/2412.07629** (2024)
[10] Li, Z., Du, Y., Zheng, M., Song, M.: Mimotable: A multi-scale spreadsheet benchmark with meta operations for table reasoning. In: COLING (2025)
[11] Lin, W., Blloshmi, R., Byrne, B., de Gispert, A., Iglesias, G.: An inner table retriever for robust table question answering. In: ACL. pp. 9909–9926 (2023)
[12] Nahid, M.M.H., Rafiei, D.: Tabsqlify: Enhancing reasoning capabilities of llms through table decomposition. In: NAACL. pp. 5725–5737 (2024)
[13] Pasupat, P., Liang, P.: Compositional semantic parsing on semi-structured tables. In: ACL-IJCNLP. pp. 1470–1480 (2015)
[14] Ruan, Y., Lan, X., Ma, J., Dong, Y., He, K., Feng, M.: Language modeling on tabular data: A survey of foundations, techniques and evolution. arXiv preprint arXiv:2408.10548 (2024)
[15] Shao, Z., Wang, P., Zhu, Q., Xu, R., Song, J., Zhang, M., Li, Y.K., Wu, Y., Guo, D.: Deepseekmath: Pushing the limits of mathematical reasoning in open language models. CoRR **abs/2402.03300** (2024)
[16] Su, A., Wang, A., Ye, C., Zhou, C., Zhang, G., Chen, G., et al.: Tablegpt2: A large multimodal model with tabular data integration. CoRR **abs/2411.02059** (2024)
[17] Team, Q.: Qwen2.5: A party of foundation models (September 2024), https://qwenlm.github.io/blog/qwen2.5/
[18] Team, Q.: Qwen3 technical report (2025), https://arxiv.org/abs/2505.09388
[19] Team, Q.: Qwq-32b: Embracing the power of reinforcement learning (March 2025), https://qwenlm.github.io/blog/qwq-32b/
[20] Wang, L., Zheng, M., Tang, H., Lin, Z., Cao, Y., Wang, J., Cai, X., Wang, W.: Needleinatable: Exploring long-context capability of large language models towards long-structured tables. arXiv preprint arXiv:2504.06560 (2025)
[21] Wang, Y., Gan, J., Qi, J.: Tabsd: Large free-form table question answering with sql-based table decomposition. CoRR **abs/2502.13422** (2025)
[22] Wang, Z., Zhang, H., Li, C., Eisenschlos, J.M., Perot, V., Wang, Z., Miculicich, L., Fujii, Y., Shang, J., Lee, C., Pfister, T.: Chain-of-table: Evolving tables in the reasoning chain for table understanding. CoRR **abs/2401.04398** (2024)
[23] Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E.H., Le, Q.V., Zhou, D.: Chain-of-thought prompting elicits reasoning in large language models. In: NeurIPS (2022)
[24] Wu, X., Yang, J., Chai, L., Zhang, G., Liu, J., Du, X., Liang, D., Shu, D., Cheng, X., Sun, T., Li, T., Li, Z., Niu, G.: Tablebench: A comprehensive and complex benchmark for table question answering. In: AAAI (2025)
[25] Wu, Z., Yang, J., Liu, J., Wu, X., Pan, C., Zhang, J., Zhao, Y., Song, S., Li, Y., Li, Z.: Table-r1: Region-based reinforcement learning for table understanding. CoRR **abs/2505.12415** (2025)
[26] Xing, J., He, Y., Zhou, M., Dong, H., Han, S., Chen, L., Zhang, D., Chaudhuri, S., Jagadish, H.: Mmtu: A massive multi-task table understanding and reasoning benchmark. arXiv preprint arXiv:2506.05587 (2025)
[27] Yang, Z., Chen, L., Cohan, A., Zhao, Y.: Table-r1: Inference-time scaling for table reasoning. CoRR **abs/2505.23621** (2025)
[28] Ye, Y., Hui, B., Yang, M., Li, B., Huang, F., Li, Y.: Large language models are versatile decomposers: Decomposing evidence and questions for table-based reasoning. In: SIGIR (2023)
[29] Zhang, H., Si, Q., Fu, P., Lin, Z., Wang, W.: Are large language models table-based fact-checkers? In: CSCWD. pp. 3086–3091 (2024)
[30] Zhang, X., Wang, D., Dou, L., Zhu, Q., Che, W.: A survey of table reasoning with large language models. CoRR **abs/2402.08259** (2024)
[31] Zhang, Y., Henkel, J., Floratou, A., Cahoon, J., Deep, S., Patel, J.M.: Reactable: Enhancing react for table question answering. Proc. VLDB Endow. **17**(8), 1981–1994 (2024)
[32] Zhong, V., Xiong, C., Socher, R.: Seq2sql: Generating structured queries from natural language using reinforcement learning. CoRR **abs/1709.00103** (2017)
[33] Zhu, J., Wang, J., Yu, B., Wu, X., Li, J., Wang, L., Xu, N.: Tableeval: A real-world benchmark for complex, multilingual, and multi-structured table question answering. CoRR **abs/2506.03949** (2025)