XPageTableRAG: Benchmarking and Modeling Cross-Page Table Question Answering

Anonymous ACL submission

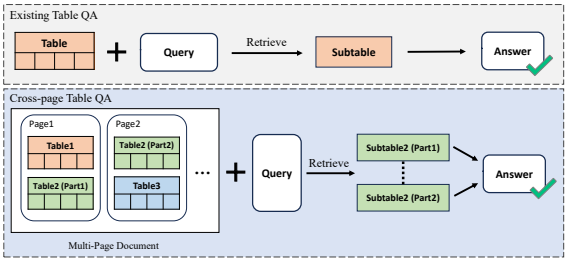
Abstract

Existing table question answering (Table QA) research typically assumes a single complete table as input. However, real-world table QA scenarios often involve multi-page documents. Such documents contain multiple coexisting tables, where individual tables can be fragmented across pages, posing challenges for both retrieval and information integration. In this work, we study question answering in such multi-page document settings and define it as Cross-page table question answering. To facilitate this task, we present CPTQA, a new benchmark that covers diverse table types and varying page spans. We further introduce XPageTableRAG, a framework that reconstructs complete tables via a header-aware table reconstruction module and performs fine-grained retrieval over reconstructed tables. Experimental results show that XPageTableRAG consistently outperforms strong baselines and achieves state-of-the-art performance on CPTQA.

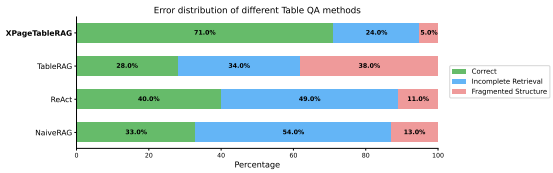
1 Introduction

Table question answering (Table QA) is a fundamental task that aims to answer natural language questions over structured tables. Retrieval-augmented generation (RAG) has become a common paradigm for retrieving relevant table fragments and performing reasoning over structured data (Zhou et al., 2025; Pan et al., 2022; Gao et al., 2023b). Recent extensions further improve reasoning and structured interaction mechanisms (Yao et al., 2022; Yu et al., 2025).

Despite these advances, existing Table QA methods typically assume a single complete table as input (Pasupat and Liang, 2015; Chen et al., 2020). However, real-world Table QA scenarios are often more complex and commonly involve multi-page documents rather than isolated single tables. Such documents contain multiple coexisting tables, and individual tables are frequently fragmented across pages. We define this setting as Cross-page table question answering, as illustrated in Figure 1(a).



(a)



(b)

Figure 1: (a) Comparison between existing Table QA and Cross-page table question answering. Top: Existing Table QA assumes a single complete table as input, where methods retrieve relevant subtables for reasoning and answer generation. Bottom: Cross-page table question answering considers multi-page documents containing multiple coexisting tables, where a table may be fragmented across pages and each query corresponds to a single target table. Methods retrieve relevant subtables from fragmented table segments across multiple pages for reasoning. (b) Error distribution of different Table QA paradigms under the Cross-page table question answering setting, evaluated on 100 multi-page queries from CPTQA.

Due to the mismatch between the standard single-table assumption and real-world document structure, existing Table QA methods exhibit significant performance degradation under Cross-page table question answering settings. As shown in Figure 1(b), chunk-based methods such as NaiveRAG and ReAct often suffer from incomplete cross-page retrieval, as multiple tables coexist within a document and relevant evidence is distributed across

pages, increasing retrieval difficulty. Structure-aware methods such as TableRAG rely on SQL-based execution over reconstructed tables, making them sensitive to header inconsistencies and fragmented table structures, which may lead to schema misalignment and execution errors.

These issues stem from two key challenges in Cross-page table question answering: coarse-grained retrieval fails to capture all relevant evidence across pages, while fragmented table structures hinder reliable reasoning over incomplete or inconsistent schemas (Wang et al., 2022; Smock et al., 2025; Salaheldin Kasem et al., 2024; Deng et al., 2022).

In addition, existing benchmarks mainly focus on single-table QA (Pasupat and Liang, 2015; Chen et al.), hybrid QA over tables and text (Chen et al., 2020; Yu et al., 2025), or multi-table QA over multiple independent and complete tables, where reasoning involves aligning or combining information across tables (Pal et al., 2023; Wu et al., 2025), but overlook this multi-page document setting.

To address this gap, we present CPTQA, the first benchmark for Cross-page table question answering. CPTQA is designed to evaluate both table retrieval and reasoning abilities under realistic multi-page document settings. It covers diverse table types, including single-page tables, multi-page tables with headers repeated on each page segment, and multi-page tables where headers appear only in the first page segment. Queries span both single-page and multi-page settings and require different reasoning types such as lookup, comparison, aggregation, and multi-hop reasoning. In addition, queries require different numbers of pages to answer, reflecting varying evidence scope. These characteristics enable comprehensive evaluation under varying structural complexity and reasoning difficulty.

We further introduce XPageTableRAG, a unified framework that reconstructs complete tables via a header-aware reconstruction module and performs fine-grained retrieval over reconstructed tables. We evaluate XPageTableRAG against strong baselines on CPTQA, and experiments show that it consistently achieves state-of-the-art performance. We will release the code and data upon acceptance. [1]

Our contributions are summarized as follows:

- We introduce CPTQA, the first benchmark for Cross-page table question answering, designed for evaluating table retrieval and reasoning in realistic multi-page document settings. It covers diverse table types and queries with varying reasoning types and evidence scopes across single-page and multi-page settings.

- We conduct extensive experiments on CPTQA, revealing two key limitations of existing Table QA methods: coarse-grained retrieval fails to capture all relevant cross-page evidence, while fragmented table structures hinder reliable reasoning over incomplete or inconsistent schemas.

- We propose XPageTableRAG, a unified framework that reconstructs fragmented tables and supports fine-grained retrieval for effective reasoning over cross-page tables, achieving state-of-the-art performance on CPTQA.

2 Related Work

2.1 Retrieval-Augmented Generation

RAG has become a prominent paradigm for reducing hallucinations and improving the accuracy of LLM-generated answers by leveraging relevant external knowledge (Izacard and Grave, 2021). RAG retrieves relevant text chunks and conditions a language model on these chunks for answer generation (Lewis et al., 2020; Guu et al., 2020). In table QA, RAG typically linearizes tables into text chunks for retrieval (Finardi et al., 2024).

Recent methods introduce more advanced retrieval and reasoning strategies. Multi-step and iterative retrieval methods, including MAIN-RAG (Chang et al., 2025), FAIR-RAG (Asl et al., 2025), and ReAct (Yao et al., 2022), improve the quality of retrieved information and progressively refine answers. In addition, query reformulation and expansion techniques (Wang et al., 2023), as well as zero-shot dense retrieval enhancements without relevance labels (Gao et al., 2023a), further improve retrieval effectiveness.

Despite these advancements, existing table QA frameworks still rely on coarse-grained retrieval over linearized table representations and assume complete tables, limiting their ability to handle fragmented tables and multi-table scenarios in real-world documents.

[1] Anonymous code repository: https://anonymous.4open.science/r/XPageTableRAG-D6C7

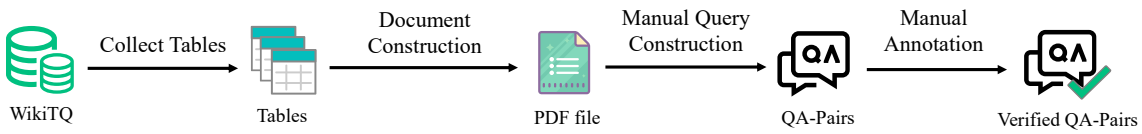


Figure 2: Data collection pipeline for CPTQA.

2.2 Table QA Benchmarks

Existing benchmarks for table question answering can be broadly categorized into three groups.

Single-table QA benchmarks focus on reasoning over a single structured table. Representative datasets such as WikiTQ (Pasupat and Liang, 2015) and TabFact (Chen et al.) evaluate compositional reasoning and fact verification on single tables. FETA-QA (Nan et al., 2022) further extends this setting by requiring free-form answer generation grounded in table content.

Hybrid QA benchmarks combine tables with unstructured text, requiring models to jointly retrieve and reason over both modalities. HybridQA (Chen et al., 2020) requires integrating evidence from tables and associated textual passages to answer questions. HeteQA (Yu et al., 2025) extends the hybrid QA setting with more complex questions, increasing multi-hop reasoning difficulty across heterogeneous sources.

Multi-table QA benchmarks focus on reasoning over multiple independent and complete tables. For instance, MultiTabQA (Pal et al., 2023) requires aggregating information across distinct tables, posing challenges in cross-table retrieval and fusion; MMQA (Wu et al., 2025) further extends this to multi-hop reasoning across tables, elevating the complexity of cross-table inference.

However, existing benchmarks typically assume complete and independently available tables, and do not consider multi-page document settings where relevant information may be distributed across multiple tables and fragmented table segments. As a result, this setting remains underexplored in existing evaluation frameworks.

3 Task Definition

Given a question q and a multi-page document D containing a set of tables $T = \{T_1, T_2, \ldots, T_m\}$, where each table may appear on a single page or span multiple pages. Each question is associated with a single relevant table $T^* \in T$, and the goal is to generate an answer A.

4 Benchmark Construction

To advance research on Cross-page table question answering, we construct CPTQA, a new benchmark for this task.

4.1 Data Collection

The data collection process is illustrated in Figure 2.

Document Construction We construct the document based on tables from the WikiTQ dataset (Pasupat and Liang, 2015). We select and process suitable tables to better reflect Cross-page table question answering scenarios. We adopt PDF due to its prevalence and native support for page-based layouts. The processed tables are rendered into PDFs with realistic layouts, where tables may be split across pages according to layout constraints, producing cross-page table structures within the document.

To better reflect how tables appear in real-world multi-page documents, we define the *table types*, which categorize tables based on page organization and header distribution:

- Single-page: the table appears on a single page as one segment.
- Multi-page (repeated headers): the table spans multiple pages, with headers repeated in each segment.
- Multi-page (non-repeated headers): the table spans multiple pages, where only the first segment contains the header.

Figure 3 illustrates the two multi-page variants.

Query Construction We manually construct queries for the collected tables. Each query is associated with a single table and may require evidence from one or multiple pages of that table. All queries and answers are manually verified to ensure correctness and consistency.

During annotation, each query inherits the table type of its corresponding table and is further

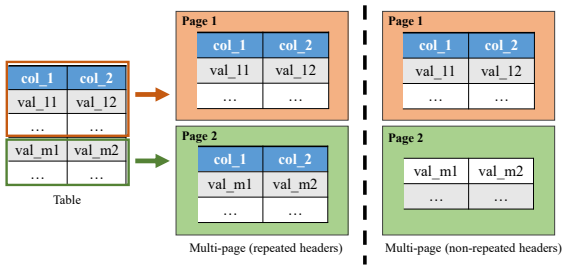


Figure 3: Illustration of the two multi-page table types in CPTQA. Left: a table. Middle: multi-page (repeated headers), where each segment repeats the header. Right: multi-page (non-repeated headers), where only the first segment contains the header.

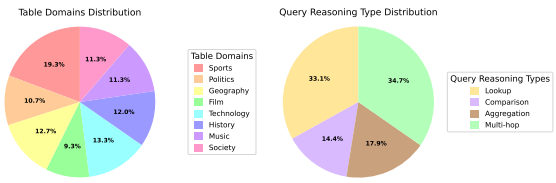


Figure 4: Distributions of table domains and query reasoning types in CPTQA.

labeled with the *referenced page count*, i.e., the number of pages whose information is required to answer the query. We group referenced page counts into three categories: 1 page, 2 pages, and 3 or more pages.

4.2 Benchmark Statistics

CPTQA contains 150 tables rendered into 212 PDF pages, resulting in 332 table segments under the defined table types. We construct 750 queries (5 per table), enabling systematic evaluation of both retrieval and reasoning under realistic multi-page document settings.

Figure 4 shows the distributions of table domains and query reasoning types in CPTQA, covering diverse real-world topics such as sports, politics, and technology, as well as multiple reasoning types including lookup, comparison, aggregation, and multi-hop reasoning. Figure 5 further illustrates the distribution of queries across table types and referenced page counts, demonstrating balanced coverage across structural settings and reasoning difficulty.

5 Method

5.1 Overview

We propose XPageTableRAG, a novel framework for Cross-page table question answering. As shown in Figure 6, it adopts a two-stage pipeline balancing

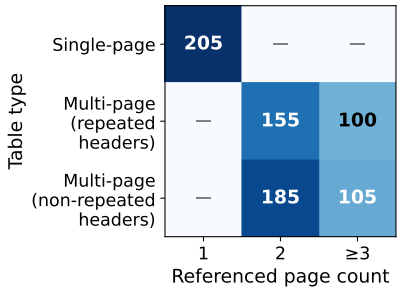


Figure 5: Distribution of queries in CPTQA across table types and referenced page counts.

Algorithm 1 Header consistency validation for table reconstruction.

Require: h_{prev}: previous segment header, h_{curr}: current segment header, $has_{\text{prev}}, has_{\text{curr}}$: header presence indicators

Ensure: Boolean indicating whether the two headers are semantically consistent

1: **if** not has_{curr} **then return** True
2: **else if** not has_{prev} **then return** False
3: **end if**
4: **if** $\textsc{Normalize}(h_{\text{prev}}) == \textsc{Normalize}(h_{\text{curr}})$ **then return** True
5: **end ifreturn** $\textsc{LLMJudge}(h_{\text{prev}}, h_{\text{curr}})$

efficiency and accuracy, with an offline stage for cross-page table processing, and an online stage for inference and answer generation.

5.2 Cross-Page Table Processing

The offline stage reconstructs complete tables and performs fine-grained embedding for efficient retrieval.

Header-aware Table Reconstruction Given table segments extracted from document pages, we reconstruct complete tables by grouping segments with consistent headers. We first use an LLM to determine whether each segment contains a header. We then determine whether adjacent segments belong to the same table by comparing their headers, using rule-based matching for exact consistency and LLM-based judgment for semantic similarity when needed. The detailed decision process is summarized in Algorithm 1. Segments deemed consistent are merged into the same table.

Fine-grained Embedding Benefiting from the reconstructed table structure, we construct fine-grained embeddings at the cell level to represent each table. Each cell is represented as a tex-

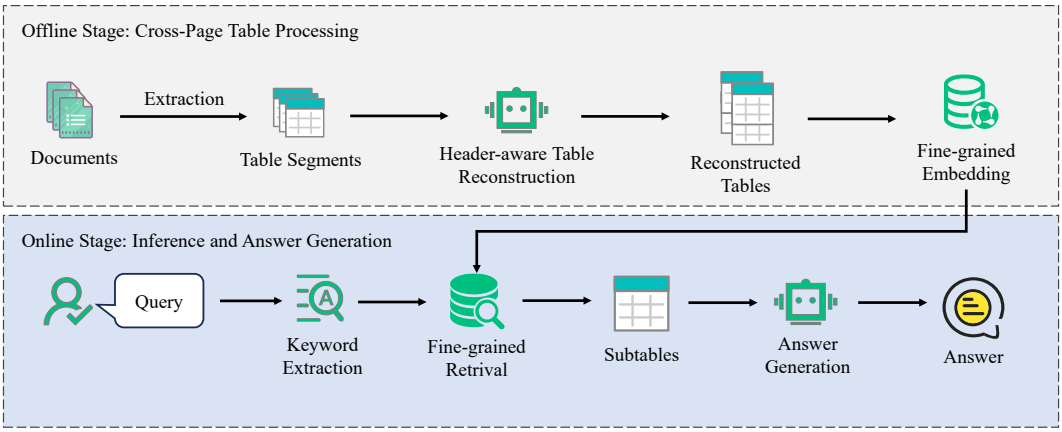


Figure 6: Overall architecture of XPageTableRAG.

tual string combining its column name and value, in the form *"Column: {column_name} | Value: {cell_value}"*, and encoded using an embedding model.

5.3 Inference and Answer Generation

The online stage retrieves relevant subtables and generates answers. A concrete example illustrating this process is provided in Appendix A.

Keyword Extraction Given an input query, we use an LLM to extract keywords that capture its core semantic intent. Extraction is guided to favor table-related elements (e.g., column names and cell values), enabling better alignment with table content.

Fine-grained Retrieval Given the extracted keywords, we perform fine-grained retrieval by matching keyword embeddings against cell-level embeddings.

For each cell embedding e, we compute its relevance score by taking the maximum similarity over all keywords:

$$s(e) = \max_{w \in \mathcal{W}} \text{sim}(w, e),$$

where $\mathcal{W}$ denotes the set of extracted keywords, and w denotes an individual keyword. $\text{sim}(\cdot, \cdot)$ is the cosine similarity.

We then select the top-k highest-scoring cells and retrieve their corresponding rows. These rows are aggregated into subtables based on their originating tables.

Answer Generation We linearize the retrieved subtables into a structured textual format and condition the LLM on this representation together with the input query to generate the final answer.

6 Experiments

6.1 Evaluation Setup

Dataset We conduct experiments on CPTQA, the first benchmark for Cross-page table question answering, covers diverse table types and queries with varying reasoning types and evidence scopes across single-page and multi-page settings.

Baselines We compare our method with several baselines: (1) Direct, which directly uses an LLM to answer questions without retrieval; (2) NaiveRAG, which linearizes tables into text chunks and retrieves relevant chunks based on embedding similarity for answer generation; (3) ReAct (Yao et al., 2022), which performs iterative reasoning and actions with external knowledge sources; and (4) TableRAG (Yu et al., 2025), an SQL-based RAG framework for table-centric reasoning.

Implementation Details All methods use MinerU2.5 (Niu et al., 2025) to parse documents and extract tables consistently. For dense retrieval, we adopt bge-large-en-v1.5 (Xiao et al., 2023) with cosine similarity. For LLM inference, we evaluate three backbone models: GPT-4o (Hurst et al., 2024), DeepSeek-V3 (Liu et al., 2024), and Qwen2.5-72B (Yang et al., 2025), covering both proprietary and open-source models. We report all results averaged over 3 runs with temperature set to 0 to ensure stability under potential randomness in retrieval and system components. All methods use task-specific prompts, with our prompts provided in Appendix C. Method-specific settings follow each method's standard configuration. For our method, we set the number of retrieved evidence units to $k = 15$. Additional implementation details

for all baselines are provided in Appendix B.

Evaluation Metrics Due to the open-ended nature of cross-page table question answering, exact-match metrics are insufficient, as they fail to capture semantically correct but lexically different answers (Kamalloo et al., 2023). We instead adopt an LLM-based evaluation protocol: the query, ground-truth answer, and model prediction are fed into Qwen3-Max (Team, 2025), which outputs a binary correctness judgment. The evaluation prompt is provided in Appendix D.

6.2 Main Results

Table 1 reports the main results on CPTQA. (1) Direct performs poorly across all settings due to the absence of retrieval, as pre-trained knowledge alone is insufficient to answer most questions grounded in document tables. (2) NaiveRAG substantially improves performance on single-page tables but degrades on multi-page settings, with larger drops observed in more challenging cases (e.g., ≥3-page tables and those without repeated headers). This indicates that coarse-grained retrieval over linearized tables struggles to capture complete evidence when tables are fragmented across pages, especially when header information is missing. (3) ReAct generally improves over NaiveRAG on more challenging multi-page settings, particularly for larger page spans (≥3 pages), but shows less consistent performance on simpler settings such as single-page or 2-page cases. This suggests that iterative retrieval and reasoning can help integrate dispersed information, but may introduce noise or unnecessary complexity when the answer can be obtained from a more localized context. (4) TableRAG underperforms compared to other RAG-based methods, especially on tables without repeated headers and those spanning more pages, highlighting the limitations of SQL-based modeling for document tables. In particular, fragmented structures and missing headers hinder reliable schema construction, making it difficult to formulate and execute effective SQL queries. (5) XPageTableRAG consistently outperforms or remains competitive with strong baselines across different backbone LLMs. The gains are especially pronounced on multi-page settings, including cases with missing headers and larger page spans, demonstrating its effectiveness in reconstructing complete tables and enabling fine-grained retrieval over fragmented table content.

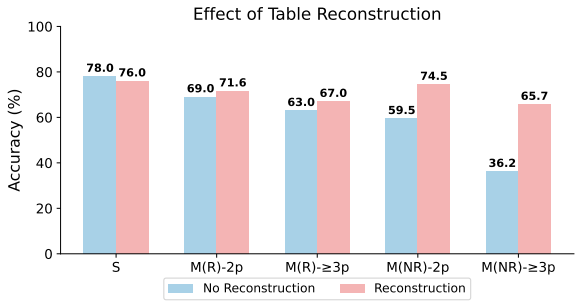


Figure 7: Effect of table reconstruction on different table settings. "S", "M(R)", and "M(NR)" denote single-page tables, multi-page tables with repeated headers, and multi-page tables without repeated headers, respectively. "2p" and "≥3p" indicate tables spanning 2 pages and 3 or more pages.

6.3 Ablation Study

We conduct ablation studies on table reconstruction and retrieval granularity using DeepSeek-V3 as the backbone model.

Effect of Table Reconstruction To evaluate the importance of table reconstruction, we compare our full method with a variant without reconstruction.

Figure 7 shows that both methods achieve comparable performance on single-page tables, indicating that reconstruction is unnecessary in this setting. In contrast, on multi-page tables, reconstruction consistently improves performance, with larger gains observed as the number of pages increases. The improvement is especially pronounced for tables without repeated headers, where the absence of explicit structural cues makes it difficult to recover table continuity. Without reconstruction, the model struggles to retrieve complete evidence, leading to significant performance degradation. By reconstructing fragmented table segments into complete tables, our method enables more accurate evidence retrieval and reasoning.

Overall, these results demonstrate that table reconstruction is crucial for handling fragmented tables, particularly in settings with missing headers and larger page spans.

Effect of Retrieval Granularity We evaluate the impact of retrieval granularity by comparing coarse-grained and fine-grained retrieval. In coarse-grained retrieval, tables are partitioned into fixed-size chunks of three rows following standard table RAG practices.

As shown in Figure 8, fine-grained retrieval consistently outperforms coarse-grained retrieval

Method	Backbone	Single	Multi (R)		Multi (NR)		Overall
			2p	≥3p	2p	≥3p	
Direct	GPT-4o	21.9	16.7	20.0	8.4	15.3	14.4
	DeepSeek-V3	16.9	7.6	10.9	5.3	11.8	11.2
	Qwen2.5-72B	13.8	6.7	7.3	4.2	8.2	8.8
NaiveRAG	GPT-4o	80.0	51.4	38.2	67.4	25.9	57.8
	DeepSeek-V3	80.0	48.6	38.2	64.2	28.2	57.0
	Qwen2.5-72B	76.9	49.5	36.4	65.3	28.2	56.2
ReAct	GPT-4o	75.6	49.5	45.5	65.2	36.5	58.2
	DeepSeek-V3	75.6	50.5	45.4	65.3	38.8	58.8
	Qwen2.5-72B	72.5	46.7	43.6	61.1	37.8	56.0
TableRAG	GPT-4o	75.0	33.3	25.5	16.8	18.8	40.2
	DeepSeek-V3	47.5	20.0	9.1	20.0	12.9	26.4
	Qwen2.5-72B	63.8	15.2	18.2	9.5	8.2	28.8
XPageTableRAG	GPT-4o	75.1	72.2	68.0	74.0	64.7	71.8
	DeepSeek-V3	76.0	71.6	67.0	74.5	65.7	72.1
	Qwen2.5-72B	76.5	67.7	62.0	71.3	54.2	68.4

Table 1: Main results on CPTQA across different methods and backbone LLMs. "R" and "NR" denote multi-page tables with repeated and non-repeated headers, respectively. "2p" and "≥3p" indicate tables spanning 2 pages and 3 or more pages.

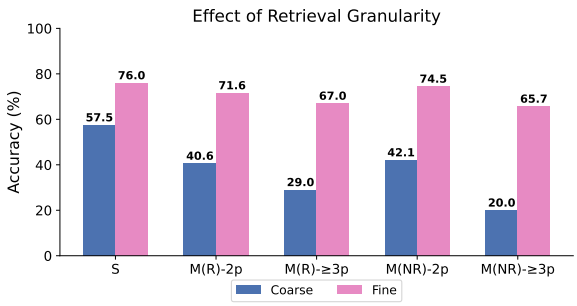


Figure 8: Effect of retrieval granularity on different table settings. "S" denotes single-page tables. "M(R)" and "M(NR)" denote multi-page tables with repeated and non-repeated headers, respectively. "2p" and "≥3p" indicate tables spanning 2 pages and 3 or more pages.

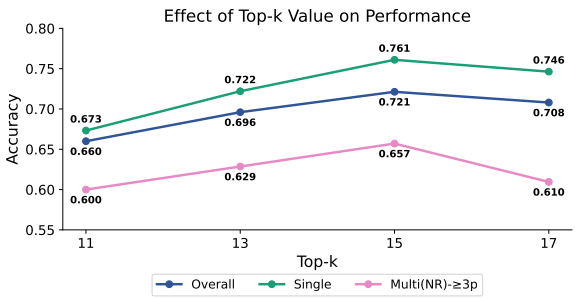


Figure 9: Effect of Top-k value on overall and per-setting performance. We report results on single-page tables, overall accuracy, and the most challenging multi-page setting with non-repeated headers (NR) and page spans of three or more pages (≥3p).

across all settings. The performance gap is relatively small on single-page tables, but becomes significantly larger on multi-page tables, especially in cases with larger page spans and missing headers. This suggests that coarse-grained chunking often fails to precisely locate relevant evidence when tables are fragmented across pages. These results highlight that fine-grained retrieval is crucial for accurately identifying relevant table content in cross-page table question answering.

7 Analysis

All analysis experiments are conducted using DeepSeek-V3 as the backbone model.

7.1 Hyperparameter Analysis

Figure 9 illustrates the impact of different Top-k values on final performance across different settings. Among all multi-page settings, we use the configuration with non-repeated headers and page spans of three or more pages as the representative case, since it exhibits the highest difficulty and the largest performance variance across different Top-k values. This makes it the most informative setting for analyzing retrieval sensitivity.

We observe that performance improves consistently when increasing Top-k from 11 to 15 across all settings, indicating that a larger retrieval budget improves coverage of relevant table evidence. However, when Top-k increases to 17, performance slightly decreases, suggesting that excessive retrieved content introduces noise and negatively af-

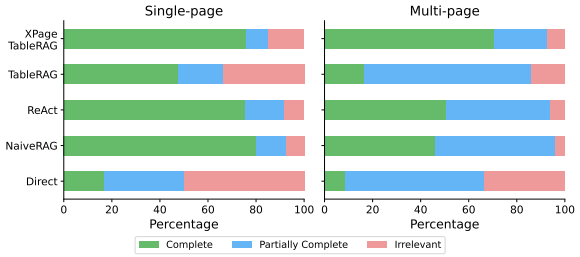


Figure 10: Answer completeness distribution on single-page and multi-page tables across different methods.

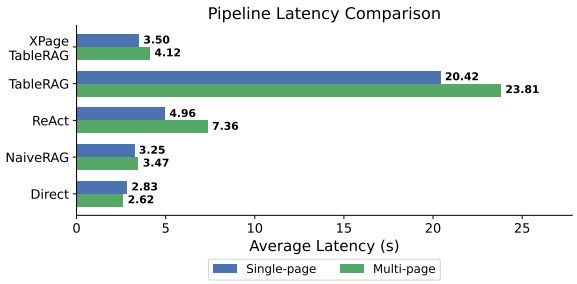


Figure 11: Average inference latency of different methods on single-page and multi-page tables.

fects reasoning. Single-page tables show relatively stable performance across different Top-k values, as relevant evidence is typically concentrated and easy to retrieve. In contrast, multi-page tables are more sensitive to the retrieval budget. This is particularly evident in the representative setting, where information is scattered across page segments, making retrieval coverage more dependent on Top-k. We thus adopt Top-k=15 as the default setting in all main experiments, as it achieves the best trade-off between retrieval coverage and noise.

7.2 Answer Completeness Analysis

Figure 10 presents the answer completeness distribution across different methods on single-page and multi-page tables.

We categorize answers into complete, partially complete, and irrelevant. Irrelevant answers indicate failures in retrieving the correct table content. Partially complete answers indicate that relevant information is not fully retrieved or not fully utilized during answer generation.

For single-page tables, all retrieval-based methods significantly reduce irrelevant answers compared to Direct, while maintaining a high proportion of complete answers. NaiveRAG and ReAct perform strongly in this setting, suggesting that coarse-grained retrieval is sufficient when table structures are not fragmented. For multi-page tables, differences between methods become more pronounced. NaiveRAG and ReAct produce more partially complete answers, indicating that the retrieved information is either incomplete or not fully utilized. TableRAG shows higher irrelevant and partially complete rates, reflecting the difficulty of SQL-based modeling under fragmented table structures. In contrast, XPageTableRAG consistently achieves the best or comparable performance in complete and irrelevant rates across different settings, and significantly reduces partially complete answers, demonstrating stronger robustness in cross-page table scenarios.

7.3 Latency Analysis

Figure 11 reports the average inference latency of different methods under single-page and multi-page settings.

Direct and NaiveRAG achieve the lowest latency due to the absence of retrieval or only lightweight embedding-based retrieval overhead. ReAct incurs higher latency because of iterative reasoning and multi-step retrieval, while TableRAG is the slowest owing to iterative SQL construction and execution over complex document structures. In contrast, XPageTableRAG maintains moderate and stable latency across both settings, demonstrating that fine-grained retrieval and cross-page aggregation can be achieved without substantial additional inference cost.

8 Conclusion

Cross-page table question answering remains challenging due to fragmented table structures, multiple coexisting tables in multi-page documents, and the difficulty of retrieving complete evidence across pages. Existing methods mainly assume complete tables and rely on coarse-grained retrieval, which limits their effectiveness in such settings. Moreover, existing benchmarks lack evaluation for realistic Cross-page Table Question Answering scenarios. To address these issues, we introduce CPTQA, the first benchmark for Cross-page Table Question Answering, and propose XPageTableRAG, a unified framework that reconstructs fragmented tables and enables fine-grained retrieval over reconstructed structures. Experimental results demonstrate that XPageTableRAG achieves strong performance across diverse cross-page table question answering settings.

9 Limitations

Despite its effectiveness, XPageTableRAG has several limitations. First, the offline table reconstruction and indexing process introduces additional computational overhead, which may affect scalability to very large document collections. Second, although XPageTableRAG performs well on CPTQA, its generalization to domains with different table layouts or languages remains untested and may require further adaptation. Moreover, CPTQA is constructed from curated Wikipedia tables and may not fully reflect the diversity and noise of real-world documents. Future work includes improving the efficiency and robustness of XPageTableRAG, as well as expanding CPTQA to more diverse and complex cross-page table question answering settings.

References

Mohammad Aghajani Asl, Majid Asgari-Bidhendi, and Behrooz Minaei-Bidgoli. 2025. Fair-rag: Faithful adaptive iterative refinement for retrieval-augmented generation. *arXiv preprint arXiv:2510.22344*.

Chia-Yuan Chang, Zhimeng Jiang, Vineeth Rakesh, Menghai Pan, Chin-Chia Michael Yeh, Guanchu Wang, Mingzhi Hu, Zhichao Xu, Yan Zheng, Mahashweta Das, and 1 others. 2025. Main-rag: Multi-agent filtering retrieval-augmented generation. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2607–2622.

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

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Yang Wang. 2020. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1026–1036.

Xiang Deng, Huan Sun, Alyssa Lees, You Wu, and Cong Yu. 2022. Turl: Table understanding through representation learning. *ACM SIGMOD Record*, 51(1):33–40.

Paulo Finardi, Leonardo Avila, Rodrigo Castaldoni, Pedro Gengo, Celio Larcher, Marcos Piau, Pablo Costa, and Vinicius Caridá. 2024. The chronicles of rag: The retriever, the chunk and the generator. *arXiv preprint arXiv:2401.07883*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2023a. Precise zero-shot dense retrieval without relevance labels. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1762–1777.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yixin Dai, Jiawei Sun, Haofen Wang, Haofen Wang, and 1 others. 2023b. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2(1):32.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. 2020. Retrieval augmented language model pre-training. In *International conference on machine learning*, pages 3929–3938. PMLR.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, and 1 others. 2024. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*.

Gautier Izacard and Edouard Grave. 2021. Leveraging passage retrieval with generative models for open domain question answering. In *Proceedings of the 16th conference of the european chapter of the association for computational linguistics: main volume*, pages 874–880.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st annual meeting of the Association for Computational Linguistics (volume 1: long papers)*, pages 5591–5606.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, and 1 others. 2024. Deepseek-v3 technical report. *arXiv preprint arXiv:2412.19437*.

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.

Junbo Niu, Zheng Liu, Zhuangcheng Gu, Bin Wang, Linke Ouyang, Zhiyuan Zhao, Tao Chu, Tianyao He, Fan Wu, Qintong Zhang, and 1 others. 2025. Mineru2. 5: A decoupled vision-language model for efficient high-resolution document parsing. In *The 64th Annual Meeting of the Association for Computational Linguistics–Industry Track*.

Vaishali Pal, Andrew Yates, Evangelos Kanoulas, and Maarten de Rijke. 2023. Multitabqa: Generating tabular answers for multi-table question answering. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 6322–6334.

Feifei Pan, Mustafa Canim, Michael Glass, Alfio Gliozzo, and James Hendler. 2022. End-to-end table question answering via retrieval-augmented generation. *arXiv preprint arXiv:2203.16714*.

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

Mahmoud Salaheldin Kasem, Abdelrahman Abdallah, Alexander Berendeyev, Ebrahem Elkady, Mohamed Mahmoud, Mahmoud Abdalla, Mohamed Hamada, Sebastiano Vascon, Daniyar Nurseitov, and Islam Taj-Eddin. 2024. Deep learning for table detection and structure recognition: A survey. *ACM Computing Surveys*, 56(12):1–41.

Brandon Smock, Valerie Faucon-Morin, Max Sokolov, Libin Liang, Tayyibah Khanam, and Maury Courtland. 2025. Pubtables-v2: A new large-scale dataset for full-page and multi-page table extraction. *arXiv preprint arXiv:2512.10888*.

Qwen Team. 2025. Qwen3-max: Just scale it.

Liang Wang, Nan Yang, and Furu Wei. 2023. Query2doc: Query expansion with large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9414–9423.

Yingyao Wang, Junwei Bao, Chaoqun Duan, Youzheng Wu, Xiaodong He, and Tiejun Zhao. 2022. MuGER2: Multi-granularity evidence retrieval and reasoning for hybrid question answering. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6687–6697, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jian Wu, Linyi Yang, Dongyuan Li, Yuliang Ji, Manabu Okumura, and Yue Zhang. 2025. Mmqa: Evaluating llms with multi-table multi-hop complex questions. In *The thirteenth international conference on learning representations*.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding. *Preprint*, arXiv:2309.07597.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 23 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.

Xiaohan Yu, Pu Jian, and Chong Chen. 2025. Tablerag: A retrieval augmented generation framework for heterogeneous document reasoning. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14074–14093.

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

A Example of XPageTableRAG on CPTQA

We present a concrete example to illustrate the inference process of XPageTableRAG on CPTQA, as shown in Table 2.

B Implementation Details

This section provides detailed information on the implementation of the baselines.

B.1 NaiveRAG

Following the standard RAG paradigm, we implement NaiveRAG by first linearizing parsed HTML tables into text and then splitting them into three-row table chunks. Relevant chunks are retrieved based on embedding similarity, with top-5 chunks used for answer generation.

B.2 ReAct

We implement the ReAct framework based on the description in the original paper and with reference to publicly available code. The preprocessing of table chunks is the same as in NaiveRAG. The framework employs an iterative reasoning loop—including thinking, acting, and observing steps—to answer user queries, with a maximum of 5 iterations.

B.3 TableRAG

We use the official TableRAG implementation. To adapt the input format, parsed HTML tables are converted into Excel tables before being fed into TableRAG. All settings follow the default configuration of the original implementation.

Step	Details
Query	What method was used in the most recent executions in Kentucky, Mississippi, and Vermont?
Keyword Extraction	[method, most recent executions, Kentucky, Mississippi, Vermont]
Fine-grained Retrieval	Retrieve top-k high-scoring cells based on similarity between keyword embeddings and cell-level embeddings. Key matches include Kentucky, Mississippi, and Vermont, which are among the top-k retrieved cells. The corresponding rows of these cells are retrieved from the table.
Subtable Construction	(Partial subtable constructed from retrieved rows) Columns: State | Last execution date | Name | Crime | Method Kentucky | 21 November 2008 | Louis Jones, Jr. | aggravated murder | lethal injection Mississippi | 20 June 2012 | Gary Carl Simmons, Jr. | aggravated murder | lethal injection Vermont | 8 December 1954 | Donald DeMag | murder | electric chair
Final Answer Generation	The model aggregates retrieved evidence and determines the execution methods: Kentucky: lethal injection, Mississippi: lethal injection, Vermont: electric chair.

Table 2: Example of XPageTableRAG inference process on a CPTQA query.

C Prompts

This section provides all prompts used in our framework for reproducibility.

C.1 Header Detection Prompt

Input: An HTML table chunk
Output: A boolean indicating whether the table contains a header

```
You are a table-understanding expert.
Please determine whether the following
HTML table contains a header.

[HTML Table]
{html_chunk}

[Output Format]
First, give a brief justification.
Then, provide the final answer in JSON
format exactly as below:
{
    "has_header": true/false
}
```

C.2 Header Matching Prompt

Input: Two HTML table headers
Output: A boolean indicating whether the headers are consistent

```
You are a table understanding assistant.
Please determine whether the following
HTML table headers output by OCR models
are the same.

[Header 1]
{header1_html}

[Header 2]
{header2_html}

[Rule]
- If the table header structure and
field names are consistent, and only OCR
-related differences exist (such as
spelling errors, case differences, or
extra spaces), consider is_same=true.
- If the table header structure or field
 names are clearly different, consider
is_same=false.

[Output Format]
First, give a brief justification.
Then, provide the final answer in JSON
format exactly as below:
{
    "is_same": true/false
}
```

C.3 Keyword Extraction Prompt

Input: A natural language text query
Output: A list of extracted keywords

```
You are an expert in extracting
information related to tables.
Your task is to extract the keywords
related to the table from the following
text.

[Text]
{text}

[Extraction Rules]
1. Extract keywords that are likely to
appear in a table, such as column names,
 cell values, or important table-related
 terms.
2. All extracted keywords MUST be copied
 verbatim from the original text. Do NOT
 normalize, paraphrase, or infer any
keywords.

[Output Format]
{
    "keywords": ["keyword1", ...]
}
```

C.4 Answer Generation Prompt

Input: A question and a reconstructed table
Output: The final answer

```
You are an expert at answering questions
 using table data.

[Question]
```

```
{question}

[Table]
{table_text}

[Output Format]
First, give a brief justification.
Then, provide the final answer in JSON
format exactly as below.
- "answer" field MUST fully satisfy ALL
parts of the question.
- "answer" field must be a single string

{
    "answer": ""
}
```

D Evaluation Details

This section describes the LLM-based evaluation protocol used in our experiments. The prompt provided to the evaluator model is shown below.

```
You are an expert evaluator for question
 answering.
Your task is to judge whether the
predicted answer correctly and
completely answers the question, based
on comparison with the ground-truth
answer.

[Question]
{question}

[Ground Truth Answer]
{gt_answer}

[Predicted Answer]
{pred_answer}

[Output Format]
{
    "reason": "<briefly explain why the
    predicted answer is correct or
    incorrect>",
    "is_correct": true/false
}
```

E Artifact Licensing and Usage

E.1 Licensing Terms

All scientific artifacts used in this work are publicly available under permissive open-source licenses that permit non-commercial academic research use. The CPTQA benchmark dataset and XPageTableRAG framework code will be released under the MIT license upon acceptance, which allows free use, modification, and distribution for both academic and commercial purposes.

E.2 Intended Use Compliance

All existing artifacts (WikiTQ dataset, MinerU2.5 document parser, bge-large-en-v1.5 embedding model, and all backbone LLMs) are employed exclusively for academic research purposes, which is fully consistent with their intended use as specified by their creators.

The intended use of our created artifacts (CPTQA dataset and XPageTableRAG code) is to facilitate research in cross-page table question answering. The CPTQA dataset is a derivative work of the WikiTQ dataset, which is licensed for non-commercial research use, and our derivative work will adhere to the same access conditions.