Large Language Models in Real-World Table Task Workflows: A Survey

Anonymous EMNLP submission

Abstract

Tables are widely used across various fields such as finance, healthcare, and public administration, playing an indispensable role in modern society. Despite their importance, the structured nature of tabular data, like permutation invariance, adds complexity to its processing. Large Language Models (LLMs) offer new opportunities, but their performance remains suboptimal due to the unique characteristics of tables. Rapidly improving LLMs' ability to process tables is unattainable in the short term. Therefore, we believe that table tasks should be broken down into many interrelated subtasks to enhance performance. So, we define workflows for handling table tasks, refine existing methods based on these workflows, and compare potentially effective methods, such as LLM-based agents, for implementing all workflows, thus providing assistance for future development.

1 Introduction

Tables are common in our daily lives and widely used in fields such as finance (Hwang et al., 2023a), healthcare (Shi et al., 2024), public administration (Musumeci et al., 2024), and chemistry (Do et al., 2023). They play an indispensable role in modern society. However, their structured nature like permutation invariance adds complexity to understanding, processing, and utilization. As data volume and complexity increase, the challenges in table processing grow. Unlike text and images, tabular data have received less attention in machine learning (van Breugel and van der Schaar, 2024).

Large Language Models (LLMs) like GPT-4 (Achiam et al., 2023) present novel opportunities for handling tabular data. By converting tabular data into text or utilizing Multimodal Large Language Models (MLLMs) (OpenAI, 2023) for processing image-formatted tables, LLMs can effectively execute certain table-related tasks. However, despite relatively clean datasets like BIRD (Li et al., 2024b), LLMs often fall short due to inherent challenges such as permutation invariance.

Focusing solely on specific table tasks limits the exploitation of LLMs' inherent capabilities, hindering the development of sufficiently practical systems for real-world applications. Rapidly enhancing LLMs' capacity to handle tables to a level suitable for real-world tasks is unrealistic in the current scenario. A potential approach involves decomposing the processing flow of table tasks and implementing comprehensive workflows to address them more effectively.

Hence, it is imperative to establish a holistic workflow tailored for real-world table task scenarios. This entails grasping the prerequisites for table tasks, pinpointing overlooked or unattended areas through a thorough review of existing methodologies, and delineating pathways for future enhancements in table processing.

LLM-based agents, powered by large language models, can exhibit some autonomous behavior and complete diverse tasks (Wang et al., 2024a). Although there are various shortcomings in current LLM-based agents handling table tasks, such as SheetAgent (Chen et al., 2024), we believe they have the potential to accomplish entire workflows. Therefore, this paper compares several existing LLM-based agents for table tasks, analyzing which parts of the workflow they can complete and which remain unaddressed. This analysis aims to guide the future development of LLM-based agents for table tasks.

The contributions of this paper is as follows.

1. We define the workflow of the entire table application, including characteristics and issues of tables, table reading, table preprocessing, user query understanding, table retrieval, table reasoning, table manipulation and tabular data safety. We also take into account the various challenges that table tasks face across various domains.

2. We summarize existing methods across different workflows and analyze areas that remain unexplored or minimally addressed. We also propose potential research directions, such as adapting methods from other fields.

3. We compare existing LLM-based agents on tables, analyze their module compositions, and evaluate the gaps between them and real-world table task requirements.

2 Characteristics and Categories of Tabular Data

Table information, or structured information, includes both textual data and structural details. That is, a table is a two-dimensional data structure composed of rows and columns with schema information. This structural aspect gives tabular data characteristics that general textual information lacks. For LLMs, this presents several challenges:

(1) Permutation Invariance. Tabular data remains unchanged if rows and columns are swapped. The model's output should be consistent even when the input table's rows or columns are exchanged (Zhu et al., 2022). **(2) Heterogeneity.** Tabular data usually contains both numerical and categorical features (Borisov et al., 2022a). **(3) Sparsity.** Tabular data is often sparse and imbalanced, resulting in long-tail distributions in training samples (Sauber-Cole and Khoshgoftaar, 2022). **(4) Data Quality Issues.** Tabular data often has missing features, noise, and class imbalance.

Apart from common characteristics, different types of tables also have unique features. Based on their structural information, tables can generally be categorized as follows:

(1) Database Tables, Primarily Relational Database Tables. These tables have a uniform structure and often store large amounts of data. Multiple tables within the same database are deeply connected through foreign keys. They contain complete schema information and adhere to database design paradigms. In relational databases, tables can be manipulated using SQL (Silberschatz et al., 2011). **(2) Text-Serialized Tables.** These tables, which can be in CSV or TSV formats, are similar to those in relational databases but lack explicitly defined schema information. **(3) Excel Spreadsheets or Sheets.** These are similar to CSV and TSV formats but more versatile, as a file can contain multiple sheets. They may include merged cells, highlighting, and other visual formats (Chen et al., 2024). **(4) Tables Found in Documents.** These tables, commonly encountered in web pages or PDFs, vary in forms. They often necessitate OCR or similar technologies for recognition (Shafait and Smith, 2010). **(5) Other Types of Broadly Customized Hierarchical Tables.** These tables, like invoices, menus, or receipts, exhibit diverse formats and intricate hierarchical relationships with logical mappings (Cheng et al., 2021).

In summary, to enable LLMs to better process tabular data and complete table-related tasks, a series of optimizations tailored to the characteristics of tabular data is necessary.

3 Introduction to Workflow

In this chapter, we will provide a detailed exploration of the workflow for table-related tasks in real-world scenarios and introduce the different characteristics of table tasks in various domains. The workflow diagram can be seen in Figure 1.

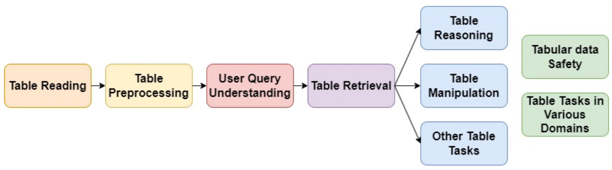

Figure 1: The workflow diagram.

3.1 Table Reading

There are two primary carriers for machine-readable tables: text-based Carriers and image-based Carriers. Our main focus is on the former, while discussions related to the latter are included in Appendix A.

Tables need to be serialized into text to be input into LLMs and serialization formats include: **(1) Text-based Table Serialization.** Formats like Markdown, CSV, TSV, HTML, XML, and LaTeX (Singha et al., 2023a; Sui et al., 2024) are commonly used for representing tables. **(2) Encoder-based Formats.** Research utilizes encoders to process tables and input encoded information into LLMs. This approach involves pre-training the Table Encoder and aligning the encoded information with LLMs (Jin et al., 2024).

3.2 Table Preprocessing

Table preprocessing involves traditional methods like filling missing data, detecting anomalies, and normalizing data. Additionally, column name cleaning restores original semantics to aid tables with abbreviated or desensitized column names (Zhang et al., 2023b). We also considered some other possible Table Preprocessing scenarios, but due to the lack of sufficient research, we have included them in Appendix B.

3.3 User Query Understanding

Tasks involving table processing require both tabular data and user queries. However, user queries often pose challenges due to ambiguity. To tackle these challenges, advanced intent detection within LLMs are necessary (Liu et al., 2019). When faced with vague questions, models must demonstrate robustness and the ability to follow instructions (Zhou et al., 2023). In instances of excessively ambiguous queries, models might need to prompt users to clarify their requirements (Wu, 2023).

3.4 Table Retrieval

Using large tables in LLMs can lead to long contexts, increased costs, and reduced performance (Kaddour et al., 2023). To address this, Retrieval-Augmented Generation (RAG) (Gao et al., 2023b) can be applied within tables to extract key information, avoiding the need to input large tables directly. Of course, RAG can also be used for table tasks such as query-answer retrieval, but this is not our main focus. Relevant content is included in Appendix C. A common method for RAG-in-Table is schema link, which selects the necessary tables, columns, and key values from the schema for a given task (Pourreza and Rafiei, 2024a). However, schema link mainly suits relational databases. Other methods aim to filter essential information from diverse table types (Kong et al., 2024).

3.5 Table Reasoning

Various papers offer different definitions of table reasoning (Zhang et al., 2024e; Zhao et al., 2022b). Here, our focus lies on LLMs' ability to comprehend tables autonomously, without external aids. Tasks in table reasoning typically include Table Question Answering (TableQA) (Pasupat and Liang, 2015), Table Fact Verification (Chen et al., 2019), Table-to-Text (Nan et al., 2022a), and Table Interpretation (Zhang et al., 2023d).

3.6 Table Manipulation

Table manipulation involves modifying table information to fulfill user needs, which includes changing content, schema, or the table's structure (Chen et al., 2024). One method is to directly use LLMs for generating modified tables (Li et al., 2023c). However, due to the limitations of LLMs, primarily their context window constraints and difficulties in handling tables, they are often used to generate code or invoke tools (Schick et al., 2024). Text-to-SQL is a significant area of research, generating SQL queries based on input tables and user queries (Yu et al., 2018). SQL has limitations, such as visualization, prompting the use of Python or domain-specific languages (DSLs) for table manipulation (Lai et al., 2023; Zha et al., 2023).

3.7 Other Table Tasks

These tasks include table prediction and table generation.

Table Prediction. In table prediction tasks, predictions are made using given tabular data. This falls under predictive tasks in machine learning. Handling the heterogeneity and continuity of numerical features is typically required for these tasks (Yan et al., 2024).

Table Generation. Table generation tasks are broadly divided into two types. Firstly, summary-form table generation involves summarizing input information into a table, serving as a summary of the input (Prasad et al., 2024). Secondly, synthetic data generation involves creating or augmenting data to supplement training data when it's insufficient. This usually requires generating high-quality data closely resembling real-world data (Borisov et al., 2022b).

3.8 Tabular data Safety

LLMs must resist adversarial attacks and avoid generating biased, discriminatory, or privacy-violating content. They should also ensure data security by not producing code that threatens it. Additionally, when dealing with table tasks, LLMs should prioritize data integrity, reliability, confidentiality, and traceability (Yao et al., 2024).

3.9 Table Tasks in Various Domains

Table tasks in different domains show distinct characteristics. In the financial domain, tasks often involve complex mathematical computations (Chen et al., 2021b). In the medical domain, tables often have numerous columns (potentially tens of thousands or more) and sparse data (Margeloiu et al., 2023). Moreover, each domain has its specific terminologies. This requires special handling when dealing with table tasks in different fields.

4 Methods of Different Workflows

This section outlines methods within each workflow, addressing problems and research gaps. Its aim is to help build comprehensive table processing systems, enhancing their capabilities within workflows.

4.1 Table reading

Various serialization methods exist for tables, with Markdown format being the most common (Fang et al., 2024). Markdown offers readability and requires fewer tokens. Other methods include JSON (Sui et al., 2024), DFLoader (Singha et al., 2023b), Attribute-Value Pairs (Wang et al., 2023c), HTML (Sui et al., 2023b), Latex (Jaitly et al., 2023), and converting tables into natural language (Yu et al., 2023; Gong et al., 2020). HTML format has been found beneficial for GPT models due to their pre-training on web data (Sui et al., 2023b, 2024). Latex has also shown promise (Jaitly et al., 2023). HTML and Latex may retain more structural information, aiding LLMs' understanding, but require more tokens. Adding identifiers and highlighting key information in serialization impact LLMs' understanding of tables (Deng et al., 2024).

Insufficient research exists on table serialization, partly because LLMs do not consider serialization method during pre-training. This may affect their ability to understand tables.

For Encoder-based formats, HGT transforms tables into heterogeneous graphs (HG) (Jin et al., 2024), while DictLLM treats tables as Key-Value structured data (Guo et al., 2024). They then enter the converted form into the Encoder. TableGPT also introduces a Table Encoder.

Encoder-based formats convert tabular data into abstract representations, handling various tables, reducing token counts, capturing key information, and addressing permutation invariance. Table encoders can enhance RAG and other LLMs' performance (Herzig et al., 2020; Yin et al., 2020; Deng et al., 2022; Wang et al., 2021b; Wang and Sun, 2022; Ye et al., 2023), yet their application and research in LLMs remain limited. More research is needed to explore their effectiveness in handling table tasks, aiding LLMs in processing such tasks better.

4.2 Table Preprocessing

Traditional Table Preprocessing. DAAgent (Hu et al., 2024) treats table preprocessing as a task, while Table-GPT (Li et al., 2023c) incorporates related tasks like data imputation. LLMs with robust table capabilities could potentially streamline preprocessing and enhance missing value prediction. However, direct LLM use for table processing is costly; integrating LLMs with Automated Machine Learning (AutoML) (He et al., 2021) appears more feasible.

Column Name Cleaning. It originally seen as a classification task (Ammar et al., 2011; Veyseh et al., 2020), categorizing abbreviations into fixed options. NameGuess (Zhang et al., 2023b) treats it as a generative task, reconstructing full names using LLMs' knowledge. However, NameGuess only provides partial information from abbreviations, thus facing difficulties in handling omitted details in column names. Improving LLMs' table understanding and integrating external knowledge could enhance Column name cleaning. It could also serve as pre-training to enhance LLMs' table comprehension and synergize with table generation tasks.

4.3 User Query Understanding

Intention clarification is a common method to deal with ambiguity (Mu et al., 2023). It can involve direct questioning or asking the user for clarification. Agents like SheetAgent (Chen et al., 2024) and TableGPT (Zha et al., 2023) have Intent Detection capabilities. Multi-turn Text-to-SQL datasets, such as CoSQL (Yu et al., 2019), require agents to seek clarification from users when necessary.

A significant challenge is defining and understanding ambiguity in table contexts. Human annotators only agree 62Papicchio et al. developed a pipeline to evaluate a model's performance in resolving input ambiguity (Papicchio et al.), yet other forms of ambiguity remain unaddressed. Huang et al. found that documentation meant for humans can assist GPT-4 in Text-to-SQL tasks (Huang et al., 2023). Bhaskar et al. propose using a top-k approach to generate possible candidates for users (Bhaskar et al., 2023). Advanced LLMs like GPT-4 can identify and introduce ambiguity in user queries (Floratou et al., 2024), indicating potential for future research.

4.4 Table Retrieval

The simplest schema link approach involves using LLMs directly to output schema link results based on schema information and user queries (DTS-SQL

(Pourreza and Rafiei, 2024b), DIN-SQL (Pourreza and Rafiei, 2024a), MAC-SQL (Wang et al., 2023a), etc.).

For large databases with many tables, some approaches use a two-stage method: first selecting tables, then performing column-level schema link within them. Examples include CRUSH4SQL (Kothyari et al., 2023) and MURRE (Zhang et al., 2024d).

In handling even larger databases where schema information can't fit into LLMs at once, Blar-SQL (Domínguez et al., 2024) suggests schema chunking, dividing schema information into chunks that fit within the context window and then merging the results.

Besides schema link, similar efforts focus on RAG-in-Table tasks. For example, OPENTAB (Kong et al., 2024) conducts column selection via SQL first, then proceeds with row and column selection. ReAcTable's (Zhang et al., 2023g) primary operation involves selecting rows and columns. PURPLE (Ren et al., 2024) represents schema as a graph and uses the Steiner (Hwang and Richards, 1992) tree problem to prune the schema.

But the schema chunking method in Blar-SQL (Domínguez et al., 2024) naturally causes a performance loss. To boost schema link performance, exploring methods like PURPLE, which rely less on LLMs' capabilities, such as invoking tools, may be necessary.

Incorrect RAG-in-Table directly affects subsequent table tasks' effectiveness. However, there's a lack of evaluation work on RAG-in-Table. Many testing methods, like SLSQL (Lei et al., 2020), often rely on simple metrics like Precision, Recall, F1. To cater to LLMs' needs in schema link, DFIN-SQL (Volvovsky et al., 2024) proposes a new metric, Schema Link Accuracy Metric, yet relevant datasets or benchmarks are still lacking.

4.5 Table Reasoning

There are numerous datasets for tasks like TableQA (Pasupat and Liang, 2015; Nan et al., 2022b; Herzig et al., 2021a; Chen et al., 2020b,a) and Table Fact Verification (Chen et al., 2019). Considerable research leverages LLMs for table reasoning. The simplest approach uses prompt engineering. For example, ToolWriter (Gemmell and Dalton, 2023) directly generates answers for straightforward TableQA tasks using LLMs. Sui et al. employ self-augmented prompting (Sui et al., 2024), where LLMs first generate an understanding of the table and then use it to generate answers. Some methods also enhance LLMs' capabilities using Chain-of-Thought (CoT) (Wei et al., 2022), as demonstrated in research (Liu et al., 2023c) by Liu et al.

Since LLMs are typically not specifically trained on tabular data, one crucial way to improve table reasoning is through fine-tuning. Many models are fine-tuned for table tasks or structured data tasks. Examples include TableLlama (Zhang et al., 2023e), UnifiedSKG (Xie et al., 2022) and TabFMs (Zhang et al., 2023a). Microsoft developed a specialized fine-tuning method called table-tuning (Li et al., 2023c). It includes tasks like Table Summarization and Row-to-Row Transformation.

However, research on table reasoning directly using LLMs to generate answers is relatively scarce. LLMs have limited capabilities and are highly sensitive to the format of table input. Liu et al. (Liu et al., 2023c) found that transposing tables or rearranging rows and columns greatly affects LLM performance. Due to the inherent limitations of LLM architectures, these challenges are currently difficult to address.

4.6 Table Manipulation

There is limited research on directly modifying tables. For instance, Microsoft's Table-GPT (Li et al., 2023c) handles tasks related to outputting modified tables. In contrast, code-based methods are commonly used. The commonly used code for table manipulation includes SQL, Python, and DSL. A simple comparison between them is shown in Table 1.

SQL. Various studies explore Text-to-SQL, utilizing datasets like Spider (Yu et al., 2018), BIRD (Li et al., 2024b) and WikiSQL (Zhong et al., 2017). Methods like C3 (Dong et al., 2023) employ prompt engineering for Text-to-SQL in zero-shot mode, while Nan et al. (Nan et al., 2023) utilize few-shot learning. QDecomp+InterCOL (Tai et al., 2023) introduces CoT to tackle Text-to-SQL challenges. Approaches such as TableLLaMA (Zhang et al., 2023e), UnifiedSKG (Xie et al., 2022), and RESDSQL (Li et al., 2023a) are based on fine-tuning.

Effective Text-to-SQL methods often split the task into multiple stages. One common approach involves Schema Link followed by SQL generation, as seen in Blar-SQL (Domínguez et al., 2024) and DTS-SQL (Pourreza and Rafiei, 2024b). An-

Language	SQL	Python	DSL
Example	SELECT * FROM Person WHERE Age > 18;	result = df[df['Age'] > 18]	`{"commands": "SelectCondition", "commands_args": {"columns": ['Age'], "condition": "Age>18"}}`

Table 1: Comparison of three languages commonly used for Table Manipulation: SQL, Python, and DSL. Among them, DSL is mainly based on TableGPT (Zha et al., 2023).

other method first generates the SQL structure, then its content, demonstrated by ZeroNL2SQL (Gu et al., 2023b) and SC-Prompt (Gu et al., 2023a). SGU-SQL (Zhang et al., 2024c) enhances linkage between user queries and databases, then uses syntax trees to guide SQL generation. PET-SQL (Li et al., 2024d) generates preliminary SQL, performs schema link based on it, then finalizes the SQL. ReBoostSQL (Sui et al., 2023a) and DIN-SQL (Pourreza and Rafiei, 2024a) primarily address query rewriting, schema link, SQL generation, and self-correction. DFIN-SQL ((Volvovsky et al., 2024)), an enhancement of DIN-SQL, focuses on producing concise table descriptions. PURPLE (Ren et al., 2024) focuses on schema pruning, skeleton prediction, demonstration selection, and database adaptation to accommodate SQL rule variations among different databases.

However, Text-to-SQL still faces several issues: **(1)** SQL syntax remains limited, mostly revolving around SELECT statements. ALTER and UPDATE queries are notably scarce. In the BIRD (Li et al., 2024b) dataset, among 12,751 reference SQL queries analyzed, only 36 instances of left join or right join were found, with the majority being inner join. **(2)** Evaluation of SQL queries is inadequate. Key metrics like Exact Matching and Execution Accuracy are commonly used, as in Spider (Yu et al., 2018), yet additional metrics such as Valid Efficiency Score are seldom employed, except in datasets like BIRD. **(3)** Certain Chinese Text-to-SQL datasets, like CSpider (Min et al., 2019), are translations from English counterparts. This often results in disparities between table and column names in the database and their representations in questions. Moreover, column names are sometimes implied within the question's semantics (LYU et al., 2022).

Python. There are few datasets specifically designed for Python-based table tasks. Datasets like HumanEval (Chen et al., 2021a), MBPP (Austin et al., 2021), and DS-1000 (Lai et al., 2023) include problems with pandas but often lack complexity and table input. Datasets like SOFSET, which focus on real-world StackOverflow problems with table inputs, are scarce.

However, some studies show Python may not always be advantageous for table manipulation. For example, Liu et al. found Direct Prompting outperformed Python (Liu et al., 2023d), and ReBoostSQL found Text-to-Python less effective than Text-to-SQL (Sui et al., 2023a). Still, Python has strengths in visualizing data and processing non-database tables.

DSL. Some studies investigate the effectiveness of DSL for table tasks. For instance, ReBoostSQL (Sui et al., 2023a) designs encapsulated functions and demonstrates superior performance compared to SQL generation methods on advanced models like GPT-4. CHAIN-OF-TABLE (Wang et al., 2024b) employs custom atomic operations to process tables step-by-step following the CoT approach. TableGPT (Zha et al., 2023) defines a DSL for table manipulation.

To improve LLMs' ability with tabular data, beyond adapting LLMs to existing tools, optimizing tools for LLM compatibility is crucial. Recent studies indicate human-readable data and code representations might not be ideal for LLMs. Recent studies propose AI-oriented languages like SimPy (Sun et al., 2024), a Python variant, and SUQL (Liu et al., 2023b), a modified SQL with free-text querying capabilities. These AI-oriented languages aid LLMs in managing table tasks. Additionally, proposals recommend designing databases with LLM interaction in mind, integrating views to boost LLM understanding of database data (Nascimento et al.).

4.7 Other Table Tasks

Table Prediction. Numerous studies explore using Large Language Models (LLMs) for table prediction (Yan et al., 2024; Slack and Singh, 2023; Manikandan et al., 2023). LLMs face challenges due to their limited mathematical capabilities (Plevris et al., 2023), particularly in representing numerical values, and struggle with tasks like Extreme Multi-label Classification (Plevris et al., 2023; D'Oosterlinck et al., 2024). On the other hand, tree-based models like XGBoost maintain advantages in table prediction (Grinsztajn et al., 2022). Alternatively, there are several avenues to explore, as summarized here:

(1) Utilizing LLMs as General Predictors. Aky"urek et al. show that Transformers fit linear regression models via in-context learning (Akyürek et al., 2022). Google develops Universal Regressors — OmniPred (Song et al., 2024) — enhancing tokens to better represent data and training extensively across prediction tasks, highlighting LLMs' potential. **(2) Employing LLMs as Feature Extractors.** Do et al. convert tabular data into text, using LLMs to embed these texts as features for XGBoost training (Do et al., 2023). LLMs can aid in feature selection, necessitating further research. **(3) Integrating LLMs into AutoML Frameworks.** LLMs in tooling invoke other ML methods via APIs. AutoML-GPT (Zhang et al., 2023c) and MYCRUNCHGPT (Kumar et al., 2023b) focus on AutoML and Scientific Machine Learning tasks, respectively. JarviX (Liu et al., 2023a) employs LLMs for automated guidance and high-precision data analysis on table datasets, integrating AutoML for predictive modeling. Optimization opportunities exist in AutoML's complexity.

Table Generation. LLMs can generate summary tables effectively, as demonstrated by Prasad et al. (Kim et al., 2024a) with LLMs. ChartAssistant (Meng et al., 2024) utilizes table generation from charts as a pre-training task to improve LLMs' comprehension of visual data. Incorporating summary table generation into pre-training tasks can enhance LLMs' ability to understand and produce structured information, warranting further exploration. Methods for generating synthetic data tables with decoder-only architecture LLMs are limited. CLLM (Seedat et al., 2023) uses GPT-4 to generate tabular data, then filters the output. Kim et al. employ grouped CSV format prompts to expand tables (Kim et al., 2024a). Gretel Navigator [1] is an online tool for creating, editing, and augmenting tabular data. It contributes to the synthetic_text_to_sql (Meyer et al., 2024) dataset. Currently, evaluating the quality of generated tables remains immature, reflecting a lack of understanding of what constitutes good tabular data.

4.8 Tabular Data Safety

There is significant research on adversarial attacks on LLMs (Kumar et al., 2023a; Xhonneux et al., 2024). However, there is less research focused on tabular data. The primary concerns regarding the security of tabular data are privacy and data safety. TableGPT (Zha et al., 2023) emphasizes private deployment to address these concerns. The security of tabular data is also linked to the traceability of data processing. One approach is using DSL to limit LLM functionalities, preventing them from generating malicious code. Examples include CHAIN-OF-TABLE (Wang et al., 2024b) and ReAcTable (Zhang et al., 2023g). Another method involves post-hoc mitigation. For example, DataLore (Lou et al., 2024) finds methods to transform Table A into its enhanced form A'.

However, requiring traceability in table processing and designing an effective DSL together can limit LLM capabilities. Balancing performance and security is a significant challenge due to these constraints. Existing table datasets also overlook security concerns, necessitating research and development of relevant benchmarks.

4.9 Table Tasks in Various Domains

Research in the Finance Domain has been extensive. Datasets in TableQA include TAT-QA (Zhu et al., 2021), FinQA (Chen et al., 2021b), ConvFinQA (Chen et al., 2022), and Multihiertt (Zhao et al., 2022a). Benchmarks like KnowledgeMATH (Zhao et al., 2023) for TableQA and BULL (Zhang et al., 2024a) for Text-to-SQL focus on financial knowledge and computational abilities. Hwang et al. (Hwang et al., 2023b) augmented FinQA using LLMs. They trained models that outperform others in TableQA within finance. FinSQL (Zhang et al., 2024a) enhanced Text-to-SQL in finance through prompt engineering, schema link, fine-tuning, and output calibration. RAVEN (Theuma and Shareghi, 2024) improved Tabular Data Analysis in Finance through tool usage and fine-tuning. This enhanced performance in both TableQA and Text-to-SQL.

In the medical domain, DictLLM (Guo et al., 2024) aids in handling structured data like medical laboratory reports. EHRAgent (Shi et al., 2024) deals with Electronic Health Records (EHRs). In the power domain, Sun et al. (Sun et al., 2023)

[1] https://gretel.ai/navigator

boosted LLMs' Text-to-SQL capabilities through secondary pre-training and instruction fine-tuning. In the materials domain, Do et al. (Do et al., 2023) serialize tables into text, then use LLMs for embedding before employing xgboost for table prediction.

Research on optimizing table tasks for different domains remains insufficient. More domains should leverage LLMs to address a wider array of real-world issues.

5 Comparison of Existing LLM-based Agents' Capabilities

In this chapter, we compare various LLM-based agents. We summarize their workflow capabilities and assess the status of each part of their implementation.

SheetAgent (Chen et al., 2024) is a framework with multi-agents. It employs iterative task reasoning and reflection to manipulate spreadsheets precisely and autonomously. SheetCopilot (Li et al., 2024a) devises atomic operations as abstractions of spreadsheet functions. They also developed a task planning framework for interaction with large language models. Data-Copilot (Zhang et al., 2023f) is a code-centric data analysis agent. It executes queries, processes, and visualizes data based on human requests. ReAcTable (Zhang et al., 2023g) specializes in TableQA problems. It uses SQL and Python code in its processing, incorporating features like voting. DAAgent (Hu et al., 2024) performs table tasks using Python, referencing the ReAct (Yao et al., 2022) framework. DB-GPT (Xue et al., 2023) understands natural language queries and generates accurate SQL queries. It includes a Python library for developer convenience. Data Formulator (Wang et al., 2023b) assists in visualizing tabular data and automates table preprocessing for visualization pipelines. TableGPT (Zha et al., 2023) utilizes a Table Encoder for comprehensive table understanding. It handles various operations such as question answering, data manipulation, visualization, analysis report generation, and automated prediction. EHRAgent (Shi et al., 2024) focuses on EHRs and autonomously executes complex clinical tasks, integrating medical knowledge during processing.

A comparison of these agents based on workflow is available in Table 2 in Appendix D. It can be seen that currently, there is no LLM-based agent capable of executing the complete workflow.

6 Conclusion

This paper outlined the workflow for real-world table tasks, highlighting critical steps such as table reading, preprocessing, user query understanding, table retrieval, reasoning, and manipulation, along with safety considerations. Existing methods address specific tasks but lack integration across the entire process. Further research is necessary to fill the gaps and enhance comprehensive automation of the workflow.

Future efforts should prioritize the development of solutions capable of fully automating the entire process, empowering LLM-based table processing systems, particularly agents, to effectively manage all aspects of table processing. Addressing these challenges will advance LLMs' capabilities and their utility in real-world table tasks, ultimately enhancing efficiency and effectiveness across various domains.

Limitations

This paper represents the first attempt, to our knowledge, to comprehensively review table workflow definitions. However, our current definitions have several issues and do not fully address real-world table processing requirements. The current definitions of table tasks do not cover all existing tasks, and the relationships between different tasks are not clearly explained. Many workflow definitions are vague and need refinement. We aim to standardize these table tasks more formally in the future.

This paper focuses on the workflow process and does not extensively cover techniques related to LLMs, such as prompt engineering, Chain-of-Thought, and fine-tuning. Nevertheless, these techniques are widely used in the methods discussed in our paper.

It's important to note that not all table processing workflows require every step outlined in our paper. As LLMs evolve, particularly those with improved table processing capabilities, many workflow steps may become less crucial.

Ethics Statement

We, the authors of this review article, collectively present the culmination of our extensive literature review and synthesis. We affirm that our research process has been conducted with integrity and adherence to ethical principles.

All sources cited in this review have been obtained from reputable scholarly databases and publications, and we have taken care to accurately represent the findings of each study. We have made every effort to provide a balanced and objective analysis of the literature, while acknowledging any biases that may have influenced the selection and interpretation of sources.

No human or animal subjects were involved in the research process, as this review is based solely on existing published literature. Therefore, ethical approval from institutional review boards was not required.

We acknowledge the importance of proper citation and attribution of sources, and have diligently cited all relevant works in accordance with academic standards. Any potential conflicts of interest have been disclosed.

In conclusion, this review article has been conducted with the utmost professionalism and commitment to ethical conduct, as a result of collaborative efforts among us, the authors, aiming to provide readers with a comprehensive and reliable overview of the topic at hand.

References

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

Ekin Akyürek, Dale Schuurmans, Jacob Andreas, Tengyu Ma, and Denny Zhou. 2022. What learning algorithm is in-context learning? investigations with linear models. *arXiv preprint arXiv:2211.15661*.

Waleed Ammar, Kareem Darwish, Ali El Kahki, and Khaled Hafez. 2011. Ice-tea: in-context expansion and translation of english abbreviations. In *Computational Linguistics and Intelligent Text Processing: 12th International Conference, CICLing 2011, Tokyo, Japan, February 20-26, 2011. Proceedings, Part II 12*, pages 41–54. Springer.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, et al. 2021. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Adithya Bhaskar, Tushar Tomar, Ashutosh Sathe, and Sunita Sarawagi. 2023. Benchmarking and Improving Text-to-SQL Generation under Ambiguity. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7053–7074, Singapore. Association for Computational Linguistics.

Vadim Borisov, Tobias Leemann, Kathrin Seßler, Johannes Haug, Martin Pawelczyk, and Gjergji Kasneci. 2022a. Deep neural networks and tabular data: A survey. *IEEE Transactions on Neural Networks and Learning Systems*.

Vadim Borisov, Kathrin Seßler, Tobias Leemann, Martin Pawelczyk, and Gjergji Kasneci. 2022b. Language models are realistic tabular data generators. *arXiv preprint arXiv:2210.06280*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021a. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Wenhu Chen, Ming-Wei Chang, Eva Schlinger, William Wang, and William W Cohen. 2020a. Open question answering over tables and text. *arXiv preprint arXiv:2010.10439*.

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. *arXiv preprint arXiv:1909.02164*.

Wenhu Chen, Hanwen Zha, Zhiyu Chen, Wenhan Xiong, Hong Wang, and William Wang. 2020b. Hybridqa: A dataset of multi-hop question answering over tabular and textual data. *arXiv preprint arXiv:2004.07347*.

Yibin Chen, Yifu Yuan, Zeyu Zhang, Yan Zheng, Jinyi Liu, Fei Ni, and Jianye Hao. 2024. Sheetagent: A generalist agent for spreadsheet reasoning and manipulation via large language models. *arXiv preprint arXiv:2403.03636*.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, et al. 2021b. Finqa: A dataset of numerical reasoning over financial data. *arXiv preprint arXiv:2109.00122*.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. Convfinqa: Exploring the chain of numerical reasoning in conversational finance question answering. *arXiv preprint arXiv:2210.03849*.

Zhoujun Cheng, Haoyu Dong, Zhiruo Wang, Ran Jia, Jiaqi Guo, Yan Gao, Shi Han, Jian-Guang Lou, and Dongmei Zhang. 2021. Hitab: A hierarchical table dataset for question answering and natural language generation. *arXiv preprint arXiv:2108.06712*.

Naihao Deng, Zhenjie Sun, Ruiqi He, Aman Sikka, Yulong Chen, Lin Ma, Yue Zhang, and Rada Mihalcea. 2024. Tables as images? exploring the strengths and limitations of llms on multimodal representations of tabular data. *arXiv preprint arXiv:2402.12424*.

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.

Tuyen Do, Bichar Dip Shrestha Gurung, Shiva Aryal, Anup Khanal, Sandeep Chataut, Venkataramana Gadhamshetty, Carol Lushbough, and Etienne Z Gnimpieba. 2023. Utilizing xgboost for the prediction of material corrosion rates from embedded tabular data using large language model. In *2023 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 4497–4499. IEEE.

José Manuel Domínguez, Benjamín Errázuriz, and Patricio Daher. 2024. Blar-sql: Faster, stronger, smaller nl2sql. *arXiv preprint arXiv:2401.02997*.

Xuemei Dong, Chao Zhang, Yuhang Ge, Yuren Mao, Yunjun Gao, Jinshu Lin, Dongfang Lou, et al. 2023. C3: Zero-shot text-to-sql with chatgpt. *arXiv preprint arXiv:2307.07306*.

Karel D'Oosterlinck, Omar Khattab, François Remy, Thomas Demeester, Chris Develder, and Christopher Potts. 2024. In-context learning for extreme multi-label classification. *arXiv preprint arXiv:2401.12178*.

Xi Fang, Weijie Xu, Fiona Anting Tan, Jiani Zhang, Ziqing Hu, Yanjun Qi, Scott Nickleach, Diego Socolinsky, Srinivasan Sengamedu, and Christos Faloutsos. 2024. Large language models (llms) on tabular data: Predic-tion, generation, and understanding-a survey. *arXiv preprint arXiv:2402.17944*.

Avrilia Floratou, Fotis Psallidas, Fuheng Zhao, Shaleen Deep, Gunther Hagleither, Wangda Tan, Joyce Cahoon, Rana Alotaibi, Jordan Henkel, Abhik Singla, Alex Van Grootel, Brandon Chow, Kai Deng, Katherine Lin, Marcos Campos, K. V. Emani, Vivek Pandit, Victor Shnayder, Wenjing Wang, and Carlo Curino. 2024. NL2SQL is a solved problem... Not! In *Conference on Innovative Data Systems Research*.

Dawei Gao, Haibin Wang, Yaliang Li, Xiuyu Sun, Yichen Qian, Bolin Ding, and Jingren Zhou. 2023a. Text-to-sql empowered by large language models: A benchmark evaluation. *arXiv preprint arXiv:2308.15363*.

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

Carlos Gemmell and Jeffrey Dalton. 2023. Generate, transform, answer: Question specific tool synthesis for tabular data. *arXiv preprint arXiv:2303.10138*.

Heng Gong, Yawei Sun, Xiaocheng Feng, Bing Qin, Wei Bi, Xiaojiang Liu, and Ting Liu. 2020. Tablegpt: Few-shot table-to-text generation with table structure reconstruction and content matching. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1978–1988.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. 2022. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in neural information processing systems*, 35:507–520.

Zihui Gu, Ju Fan, Nan Tang, Lei Cao, Bowen Jia, Sam Madden, and Xiaoyong Du. 2023a. Few-shot text-to-sql translation using structure and content prompt learning. *Proceedings of the ACM on Management of Data*, 1(2):1–28.

Zihui Gu, Ju Fan, Nan Tang, Songyue Zhang, Yuxin Zhang, Zui Chen, Lei Cao, Guoliang Li, Sam Madden, and Xiaoyong Du. 2023b. Interleaving pre-trained language models and large language models for zero-shot nl2sql generation. *arXiv preprint arXiv:2306.08891*.

YiQiu Guo, Yuchen Yang, Ya Zhang, Yu Wang, and Yanfeng Wang. 2024. Dictllm: Harnessing key-value data structures with large language models for enhanced medical diagnostics. *arXiv preprint arXiv:2402.11481*.

Xin He, Kaiyong Zhao, and Xiaowen Chu. 2021. Automl: A survey of the state-of-the-art. *Knowledge-based systems*, 212:106622.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Martin Eisenschlos. 2021a. Open domain question answering over tables via dense retrieval. *arXiv preprint arXiv:2103.12011*.

Jonathan Herzig, Thomas Müller, Syrine Krichene, and Julian Martin Eisenschlos. 2021b. Open domain question answering over tables via dense retrieval. *arXiv preprint arXiv:2103.12011*.

Jonathan Herzig, Paweł Krzysztof Nowak, Thomas Müller, Francesco Piccinno, and Julian Martin Eisenschlos. 2020. Tapas: Weakly supervised table parsing via pre-training. *arXiv preprint arXiv:2004.02349*.

Anwen Hu, Yaya Shi, Haiyang Xu, Jiabo Ye, Qinghao Ye, Ming Yan, Chenliang Li, Qi Qian, Ji Zhang, and Fei Huang. 2023. mplug-paperowl: Scientific diagram analysis with the multimodal large language model. *arXiv preprint arXiv:2311.18248*.

Xueyu Hu, Ziyu Zhao, Shuang Wei, Ziwei Chai, Guoyin Wang, Xuwu Wang, Jing Su, Jingjing Xu, Ming Zhu, Yao Cheng, et al. 2024. Infiagent-dabench: Evaluating agents on data analysis tasks. *arXiv preprint arXiv:2401.05507*.

Zezhou Huang, Pavan Kalyan Damalapati, and Eugene Wu. 2023. Data Ambiguity Strikes Back: How Documentation Improves GPT's Text-to-SQL.

Madelon Hulsebos, Kevin Hu, Michiel Bakker, Emanuel Zgraggen, Arvind Satyanarayan, Tim Kraska, Çagatay Demiralp, and César Hidalgo. 2019. Sherlock: A deep learning approach to semantic data type detection. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1500–1508.

Frank K Hwang and Dana S Richards. 1992. Steiner tree problems. *Networks*, 22(1):55–89.

Yechan Hwang, Jinsu Lim, Young-Jun Lee, and Ho-Jin Choi. 2023a. Augmentation for context in financial numerical reasoning over textual and tabular data with large-scale language model. In *NeurIPS 2023 Second Table Representation Learning Workshop*.

Yechan Hwang, Jinsu Lim, Young-Jun Lee, and Ho-Jin Choi. 2023b. Augmentation for context in financial numerical reasoning over textual and tabular data with large-scale language model. In *NeurIPS 2023 Second Table Representation Learning Workshop*.

Hiroshi Iida, Dung Thai, Varun Manjunatha, and Mohit Iyyer. 2021. Tabbie: Pretrained representations of tabular data. *arXiv preprint arXiv:2105.02584*.

Sukriti Jaitly, Tanay Shah, Ashish Shugani, and Razik Singh Grewal. 2023. Towards better serialization of tabular data for few-shot classification. *arXiv preprint arXiv:2312.12464*.

Rihui Jin, Yu Li, Guilin Qi, Nan Hu, Yuan-Fang Li, Jiaoyan Chen, Jianan Wang, Yongrui Chen, and Dehai Min. 2024. Hgt: Leveraging heterogeneous graph-enhanced large language models for few-shot complex table understanding. *arXiv preprint arXiv:2403.19723*.

Jean Kaddour, Joshua Harris, Maximilian Mozes, Herbie Bradley, Roberta Raileanu, and Robert McHardy. 2023. Challenges and applications of large language models. *arXiv preprint arXiv:2307.10169*.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.

Mahmoud Kasem, Abdelrahman Abdallah, Alexander Berendeyev, Ebrahem Elkady, Mohamed Mahmoud, Mahmoud Abdalla, Mohamed Hamada, Sebastiano Vascon, Daniyar Nurseitov, and Islam Taj-Eddin. 2022. Deep learning for table detection and structure recognition: A survey. *ACM Computing Surveys*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 39–48.

Jinhee Kim, Taesung Kim, and Jaegul Choo. 2024a. Group-wise prompting for synthetic tabular data generation using large language models. *arXiv preprint arXiv:2404.12404*.

Yoonsik Kim, Moonbin Yim, and Ka Yeon Song. 2024b. Tablevqa-bench: A visual question answering benchmark on multiple table domains. *arXiv preprint arXiv:2404.19205*.

Kezhi Kong, Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Chuan Lei, Christos Faloutsos, Huzefa Rangwala, and George Karypis. 2024. Opentab: Advancing large language models as open-domain table reasoners. *arXiv preprint arXiv:2402.14361*.

Mayank Kothyari, Dhruva Dhingra, Sunita Sarawagi, and Soumen Chakrabarti. 2023. Crush4sql: Collective retrieval using schema hallucination for text2sql. *arXiv preprint arXiv:2311.01173*.

Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Soheil Feizi, and Hima Lakkaraju. 2023a. Certifying llm safety against adversarial prompting. *arXiv preprint arXiv:2309.02705*.

Varun Kumar, Leonard Gleyzer, Adar Kahana, Khemraj Shukla, and George Em Karniadakis. 2023b. Mycrunchgpt: A llm assisted framework for scientific machine learning. *Journal of Machine Learning for Modeling and Computing*, 4(4).

Yuhang Lai, Chengxi Li, Yiming Wang, Tianyi Zhang, Ruiqi Zhong, Luke Zettlemoyer, Wen-tau Yih, Daniel Fried, Sida Wang, and Tao Yu. 2023. Ds-1000: A natural and reliable benchmark for data science code generation. In *International Conference on Machine Learning*, pages 18319–18345. PMLR.

Wenqiang Lei, Weixin Wang, Zhixin Ma, Tian Gan, Wei Lu, Min-Yen Kan, and Tat-Seng Chua. 2020. Re-examining the role of schema linking in text-to-sql. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6943–6954.

Haoyang Li, Jing Zhang, Cuiping Li, and Hong Chen. 2023a. Resdsql: Decoupling schema linking and skeleton parsing for text-to-sql. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 13067–13075.

Hongxin Li, Jingran Su, Yuntao Chen, Qing Li, and ZHAO-XIANG ZHANG. 2024a. Sheetcopilot: Bringing software productivity to the next level through large language models. *Advances in Neural Information Processing Systems*, 36.

Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, et al. 2024b. 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.

Min Li, Liping Zhang, Mingle Zhou, and Delong Han. 2023b. Uttsr: A novel non-structured text table recognition model powered by deep learning technology. *Applied Sciences*, 13(13):7556.

Peng Li, Yeye He, Dror Yashar, Weiwei Cui, Song Ge, Haidong Zhang, Danielle Rifinski Fainman, Dongmei Zhang, and Surajit Chaudhuri. 2023c. Table-gpt: Table-tuned gpt for diverse table tasks. *arXiv preprint arXiv:2310.09263*.

Tianle Li, Ge Zhang, Quy Duc Do, Xiang Yue, and Wenhu Chen. 2024c. Long-context llms struggle with long in-context learning. *arXiv preprint arXiv:2404.02060*.

Zhishuai Li, Xiang Wang, Jingjing Zhao, Sun Yang, Guoqing Du, Xiaoru Hu, Bin Zhang, Yuxiao Ye, Ziyue Li, Rui Zhao, et al. 2024d. Pet-sql: A prompt-enhanced two-stage text-to-sql framework with cross-consistency. *arXiv preprint arXiv:2403.09732*.

Weihong Lin, Zheng Sun, Chixiang Ma, Mingze Li, Jiawei Wang, Lei Sun, and Qiang Huo. 2022. Tsrformer: Table structure recognition with transformers. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 6473–6482.

Weizhe Lin, Rexhina Blloshmi, Bill Byrne, Adrià de Gispert, and Gonzalo Iglesias. 2023. Li-rage: Late interaction retrieval augmented generation with explicit signals for open-domain table question answering. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1557–1566.

Jiao Liu, Yanling Li, and Min Lin. 2019. Review of intent detection methods in the human-machine dialogue system. In *Journal of physics: conference series*, volume 1267, page 012059. IOP Publishing.

Shang-Ching Liu, ShengKun Wang, Tsungyao Chang, Wenqi Lin, Chung-Wei Hsiung, Yi-Chen Hsieh, Yu-Ping Cheng, Sian-Hong Luo, and Jianwei Zhang. 2023a. Jarvix: A llm no code platform for tabular data analysis and optimization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 622–630.

Shicheng Liu, Jialiang Xu, Wesley Tjangnaka, Sina J Semnani, Chen Jie Yu, Gui Dávid, and Monica S Lam. 2023b. Suql: Conversational search over structured and unstructured data with large language models. *arXiv preprint arXiv:2311.09818*.

Tianyang Liu, Fei Wang, and Muhao Chen. 2023c. Rethinking tabular data understanding with large language models. *arXiv preprint arXiv:2312.16702*.

Tianyang Liu, Fei Wang, and Muhao Chen. 2023d. Rethinking tabular data understanding with large language models. *arXiv preprint arXiv:2312.16702*.

Yuze Lou, Chuan Lei, Xiao Qin, Zichen Wang, Christos Faloutsos, Rishita Anubhai, and Huzefa Rangwala. 2024. Datalore: Can a large language model find all lost scrolls in a data repository?

Haoyu Lu, Wen Liu, Bo Zhang, Bingxuan Wang, Kai Dong, Bo Liu, Jingxiang Sun, Tongzheng Ren, Zhuoshu Li, Yaofeng Sun, et al. 2024. Deepseek-vl: towards real-world vision-language understanding. *arXiv preprint arXiv:2403.05525*.

Jianqing LYU, Xianbing WANG, Gang CHEN, Hua ZHANG, and Minggang WANG. 2022. Chinese text-to-sql model for industrial production. *Journal of Computer Applications*, 42(10):2996.

Hariharan Manikandan, Yiding Jiang, and J Zico Kolter. 2023. Language models are weak learners. *Advances in Neural Information Processing Systems*, 36:50907–50931.

Andrei Margeloiu, Nikola Simidjievski, Pietro Lio, and Mateja Jamnik. 2023. Weight predictor network with feature selection for small sample tabular biomedical data. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9081–9089.

Fanqing Meng, Wenqi Shao, Quanfeng Lu, Peng Gao, Kaipeng Zhang, Yu Qiao, and Ping Luo. 2024. Chartassisstant: A universal chart multimodal language model via chart-to-table pre-training and multitask instruction tuning. *arXiv preprint arXiv:2401.02384*.

Yev Meyer, Marjan Emadi, Dhruv Nathawani, Lipika Ramaswamy, Kendrick Boyd, Maarten Van Segbroeck, Matthew Grossman, Piotr Mlocek, and Drew Newberry. 2024. Synthetic-Text-To-SQL: A synthetic dataset for training language models to generate sql queries from natural language prompts.

Qingkai Min, Yuefeng Shi, and Yue Zhang. 2019. A pilot study for chinese sql semantic parsing. *arXiv preprint arXiv:1909.13293*.

Fangwen Mu, Lin Shi, Song Wang, Zhuohao Yu, Binquan Zhang, Chenxue Wang, Shichao Liu, and Qing Wang. 2023. ClarifyGPT: Empowering LLM-based Code Generation with Intention Clarification.

Emanuele Musumeci, Michele Brienza, Vincenzo Suriani, Daniele Nardi, and Domenico Daniele Bloisi. 2024. Llm based multi-agent generation of semi-structured documents from semantic templates in the public administration domain. In *International Conference on Human-Computer Interaction*, pages 98–117. Springer.

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

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

Linyong Nan, Yilun Zhao, Weijin Zou, Narutatsu Ri, Jaesung Tae, Ellen Zhang, Arman Cohan, and Dragomir Radev. 2023. Enhancing text-to-sql capabilities of large language models: A study on prompt design strategies. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 14935–14956.

Eduardo R Nascimento, Yenier T Izquierdo, Grettel M Garcıa, Gustavo MC Coelho, Lucas Feijó, Melissa Lemos, Luiz AP Paes Leme, and Marco A Casanova. My database user is a large language model.

OpenAI. 2023. Gpt-4v(ision) system card. OpenAI.

Simone Papicchio, Paolo Papotti, and Luca Cagliero. Evaluating Ambiguous Questions in Semantic Parsing.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. *arXiv preprint arXiv:1508.00305*.

Vagelis Plevris, George Papazafeiropoulos, and Alejandro Jiménez Rios. 2023. Chatbots put to the test in math and logic problems: A comparison and assessment of chatgpt-3.5, chatgpt-4, and google bard. *AI*, 4(4):949–969.

Mohammadreza Pourreza and Davood Rafiei. 2024a. Din-sql: Decomposed in-context learning of text-to-sql with self-correction. *Advances in Neural Information Processing Systems*, 36.

Mohammadreza Pourreza and Davood Rafiei. 2024b. Dts-sql: Decomposed text-to-sql with small large language models. *arXiv preprint arXiv:2402.01117*.

Deepak Prasad, Mayur Pimpude, and Alankar Alankar. 2024. Towards development of automated knowledge maps and databases for materials engineering using large language models. *arXiv preprint arXiv:2402.11323*.

Devashish Prasad, Ayan Gadpal, Kshitij Kapadni, Manish Visave, and Kavita Sultanpure. 2020. Cascadetabnet: An approach for end to end table detection and structure recognition from image-based documents. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pages 572–573.

Tonghui Ren, Yuankai Fan, Zhenying He, Ren Huang, Jiaqi Dai, Can Huang, Yinan Jing, Kai Zhang, Yifan Yang, and X Sean Wang. 2024. Purple: Making a large language model a better sql writer. *arXiv preprint arXiv:2403.20014*.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Rick Sauber-Cole and Taghi M Khoshgoftaar. 2022. The use of generative adversarial networks to alleviate class imbalance in tabular data: a survey. *Journal of Big Data*, 9(1):98.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

Nabeel Seedat, Nicolas Huynh, Boris van Breugel, and Mihaela van der Schaar. 2023. Curated llm: Synergy of llms and data curation for tabular augmentation in ultra low-data regimes. *arXiv preprint arXiv:2312.12112*.

Faisal Shafait and Ray Smith. 2010. Table detection in heterogeneous documents. In *Proceedings of the 9th IAPR International Workshop on Document Analysis Systems*, pages 65–72.

Wenqi Shi, Ran Xu, Yuchen Zhuang, Yue Yu, Jieyu Zhang, Hang Wu, Yuanda Zhu, Joyce C Ho, Carl Yang, and May Dongmei Wang. 2024. Ehragent: Code empowers large language models for few-shot complex tabular reasoning on electronic health records. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Abraham Silberschatz, Henry F Korth, and Shashank Sudarshan. 2011. Database system concepts.

Ananya Singha, José Cambronero, Sumit Gulwani, Vu Le, and Chris Parnin. 2023a. Tabular representation, noisy operators, and impacts on table structure understanding tasks in llms. *arXiv preprint arXiv:2310.10358*.

Ananya Singha, José Cambronero, Sumit Gulwani, Vu Le, and Chris Parnin. 2023b. Tabular representation, noisy operators, and impacts on table structure understanding tasks in llms. *arXiv preprint arXiv:2310.10358*.

Dylan Slack and Sameer Singh. 2023. Tablet: Learning from instructions for tabular data. *arXiv preprint arXiv:2304.13188*.

Xingyou Song, Oscar Li, Chansoo Lee, Daiyi Peng, Sagi Perel, Yutian Chen, et al. 2024. Omnipred: Language models as universal regressors. *arXiv preprint arXiv:2402.14547*.

Yoshihiko Suhara, Jinfeng Li, Yuliang Li, Dan Zhang, Çağatay Demiralp, Chen Chen, and Wang-Chiew Tan. 2022. Annotating columns with pre-trained language models. In *Proceedings of the 2022 International Conference on Management of Data*, pages 1493–1503.

Guanghu Sui, Zhishuai Li, Ziyue Li, Sun Yang, Jingqing Ruan, Hangyu Mao, and Rui Zhao. 2023a. Reboost large language model-based text-to-sql, text-to-python, and text-to-function–with real applications in traffic domain. *arXiv preprint arXiv:2310.18752*.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2023b. Evaluating and enhancing structural understanding capabilities of large language models on tables via input designs. *arXiv preprint arXiv:2305.13062*.

Yuan Sui, Mengyu Zhou, Mingjie Zhou, Shi Han, and Dongmei Zhang. 2024. Table meets llm: Can large language models understand structured table data? a benchmark and empirical study. In *Proceedings* of the 17th ACM International Conference on Web Search and Data Mining*, pages 645–654.

Gang Sun, Ran Shen, Liangfeng Jin, Yifan Wang, Shiyu Xu, Jinpeng Chen, and Weihao Jiang. 2023. Instruction tuning text-to-sql with large language models in the power grid domain. In *Proceedings of the 2023 4th International Conference on Control, Robotics and Intelligent System*, pages 59–63.

Zhensu Sun, Xiaoning Du, Zhou Yang, Li Li, and David Lo. 2024. Ai coders are among us: Rethinking programming language grammar towards efficient code generation. *arXiv preprint arXiv:2404.16333*.

Chang-You Tai, Ziru Chen, Tianshu Zhang, Xiang Deng, and Huan Sun. 2023. Exploring chain-of-thought style prompting for text-to-sql. *arXiv preprint arXiv:2305.14215*.

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

Adrian Theuma and Ehsan Shareghi. 2024. Equipping language models with tool use capability for tabular data analysis in finance. *arXiv preprint arXiv:2401.15328*.

Boris van Breugel and Mihaela van der Schaar. 2024. Why tabular foundation models should be a research priority. *arXiv preprint arXiv:2405.01147*.

Amir Pouran Ben Veyseh, Franck Dernoncourt, Quan Hung Tran, and Thien Huu Nguyen. 2020. What does this acronym mean? introducing a new dataset for acronym identification and disambiguation. *arXiv preprint arXiv:2010.14678*.

Shai Volvovsky, Marco Marcassa, and Mustafa Panbiharwala. 2024. Dfin-sql: Integrating focused schema with din-sql for superior accuracy in large-scale databases. *arXiv preprint arXiv:2403.00872*.

Bing Wang, Changyu Ren, Jian Yang, Xinnian Liang, Jiaqi Bai, Qian-Wen Zhang, Zhao Yan, and Zhoujun Li. 2023a. Mac-sql: Multi-agent collaboration for text-to-sql. *arXiv preprint arXiv:2312.11242*.

Chenglong Wang, John Thompson, and Bongshin Lee. 2023b. Data formulator: Ai-powered concept-driven visualization authoring. *IEEE Transactions on Visualization and Computer Graphics*.

Daheng Wang, Prashant Shiralkar, Colin Lockard, Binxuan Huang, Xin Luna Dong, and Meng Jiang. 2021a. Tcn: Table convolutional network for web table interpretation. In *Proceedings of the Web Conference 2021*, pages 4020–4032.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, et al. 2024a. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):1–26.

Zhiruo Wang, Haoyu Dong, Ran Jia, Jia Li, Zhiyi Fu, Shi Han, and Dongmei Zhang. 2021b. Tuta: Tree-based transformers for generally structured table pre-training. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, pages 1780–1790.

Zhiruo Wang, Zhengbao Jiang, Eric Nyberg, and Graham Neubig. 2022. Table retrieval may not necessitate table-specific model design. *arXiv preprint arXiv:2205.09843*.

Zifeng Wang, Chufan Gao, Cao Xiao, and Jimeng Sun. 2023c. Meditab: Scaling medical tabular data predictors via data consolidation, enrichment, and refinement.

Zifeng Wang and Jimeng Sun. 2022. Transtab: Learning transferable tabular transformers across tables. *Advances in Neural Information Processing Systems*, 35:2902–2915.

Zilong Wang, Hao Zhang, Chun-Liang Li, Julian Martin Eisenschlos, Vincent Perot, Zifeng Wang, Lesly Miculicich, Yasuhisa Fujii, Jingbo Shang, Chen-Yu Lee, et al. 2024b. Chain-of-table: Evolving tables in the reasoning chain for table understanding. *arXiv preprint arXiv:2401.04398*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Jie JW Wu. 2023. Does asking clarifying questions increases confidence in generated code? on the communication skills of large language models. *arXiv preprint arXiv:2308.13507*.

Sophie Xhonneux, Alessandro Sordoni, Stephan Günnemann, Gauthier Gidel, and Leo Schwinn. 2024. Efficient adversarial training in llms with continuous attacks. *arXiv preprint arXiv:2405.15589*.

Tianbao Xie, Chen Henry Wu, Peng Shi, Ruiqi Zhong, Torsten Scholak, Michihiro Yasunaga, Chien-Sheng Wu, Ming Zhong, Pengcheng Yin, Sida I Wang, et al. 2022. Unifiedskg: Unifying and multi-tasking structured knowledge grounding with text-to-text language models. *arXiv preprint arXiv:2201.05966*.

Siqiao Xue, Caigao Jiang, Wenhui Shi, Fangyin Cheng, Keting Chen, Hongjun Yang, Zhiping Zhang, Jianshan He, Hongyang Zhang, Ganglin Wei, et al. 2023. Db-gpt: Empowering database interactions with private large language models. *arXiv preprint arXiv:2312.17449*.

Jiahuan Yan, Bo Zheng, Hongxia Xu, Yiheng Zhu, Danny Chen, Jimeng Sun, Jian Wu, and Jintai Chen. 2024. Making pre-trained language models great on tabular prediction. *arXiv preprint arXiv:2403.01841*.

Hang Yang, Jing Guo, Jianchuan Qi, Jinliang Xie, Si Zhang, Siqi Yang, Nan Li, and Ming Xu. 2024. A method for parsing and vectorization of semi-structured data used in retrieval augmented generation. *arXiv preprint arXiv:2405.03989*.

Zhengyuan Yang, Linjie Li, Kevin Lin, Jianfeng Wang, Chung-Ching Lin, Zicheng Liu, and Lijuan Wang. 2023. The dawn of lmms: Preliminary explorations with gpt-4v (ision). *arXiv preprint arXiv:2309.17421*, 9(1):1.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. 2024. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, page 100211.

Chao Ye, Guoshan Lu, Haobo Wang, Liyao Li, Sai Wu, Gang Chen, and Junbo Zhao. 2023. Ct-bert: learning better tabular representations through cross-table pre-training. *arXiv preprint arXiv:2307.04308*.

Pengcheng Yin, Graham Neubig, Wen-tau Yih, and Sebastian Riedel. 2020. Tabert: Pretraining for joint understanding of textual and tabular data. *arXiv preprint arXiv:2005.08314*.

Bowen Yu, Cheng Fu, Haiyang Yu, Fei Huang, and Yongbin Li. 2023. Unified language representation for question answering over text, tables, and images. *arXiv preprint arXiv:2306.16762*.

Tao Yu, Rui Zhang, He Yang Er, Suyi Li, Eric Xue, Bo Pang, Xi Victoria Lin, Yi Chern Tan, Tianze Shi, Zihan Li, et al. 2019. Cosql: A conversational text-to-sql challenge towards cross-domain natural language interfaces to databases. *arXiv preprint arXiv:1909.05378*.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, et al. 2018. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-sql task. *arXiv preprint arXiv:1809.08887*.

Liangyu Zha, Junlin Zhou, Liyao Li, Rui Wang, Qingyi Huang, Saisai Yang, Jing Yuan, Changbao Su, Xiang Li, Aofeng Su, et al. 2023. Tablegpt: Towards unifying tables, nature language and commands into one gpt. *arXiv preprint arXiv:2307.08674*.

Chao Zhang, Yuren Mao, Yijiang Fan, Yu Mi, Yunjun Gao, Lu Chen, Dongfang Lou, and Jinshu Lin. 2024a. Finsql: Model-agnostic llms-based text-to-sql framework for financial analysis. *arXiv preprint arXiv:2401.10506*.

Dan Zhang, Yoshihiko Suhara, Jinfeng Li, Madelon Hulsebos, Çağatay Demiralp, and Wang-Chiew Tan. 2019. Sato: Contextual semantic type detection in tables. *arXiv preprint arXiv:1911.06311*.

Ge Zhang, Xinrun Du, Bei Chen, Yiming Liang, Tongxu Luo, Tianyu Zheng, Kang Zhu, Yuyang Cheng, Chunpu Xu, Shuyue Guo, et al. 2024b. Cmmmu: A chinese massive multi-discipline multimodal understanding benchmark. *arXiv preprint arXiv:2401.11944*.

Han Zhang, Xumeng Wen, Shun Zheng, Wei Xu, and Jiang Bian. 2023a. Towards foundation models for learning on tabular data. *arXiv preprint arXiv:2310.07338*.

Jiani Zhang, Zhengyuan Shen, Balasubramaniam Srinivasan, Shen Wang, Huzefa Rangwala, and George Karypis. 2023b. Nameguess: Column name expansion for tabular data. *arXiv preprint arXiv:2310.13196*.

Qinggang Zhang, Junnan Dong, Hao Chen, Wentao Li, Feiran Huang, and Xiao Huang. 2024c. Structure guided large language model for sql generation. *arXiv preprint arXiv:2402.13284*.

Shujian Zhang, Chengyue Gong, Lemeng Wu, Xingchao Liu, and Mingyuan Zhou. 2023c. Automl-gpt: Automatic machine learning with gpt. *arXiv preprint arXiv:2305.02499*.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2023d. Tablellama: Towards open large generalist models for tables. *arXiv preprint arXiv:2311.09206*.

Tianshu Zhang, Xiang Yue, Yifei Li, and Huan Sun. 2023e. Tablellama: Towards open large generalist models for tables. *arXiv preprint arXiv:2311.09206*.

Wenqi Zhang, Yongliang Shen, Weiming Lu, and Yueting Zhuang. 2023f. Data-copilot: Bridging billions of data and humans with autonomous workflow. *arXiv preprint arXiv:2306.07209*.

Xuanliang Zhang, Dingzirui Wang, Longxu Dou, Qingfu Zhu, and Wanxiang Che. 2024d. Multi-hop table retrieval for open-domain text-to-sql. *arXiv preprint arXiv:2402.10666*.

Xuanliang Zhang, Dingzirui Wang, Longxu Dou, Qingfu Zhu, and Wanxiang Che. 2024e. A survey of table reasoning with large language models. *arXiv preprint arXiv:2402.08259*.

Yunjia Zhang, Jordan Henkel, Avrilia Floratou, Joyce Cahoon, Shaleen Deep, and Jignesh M Patel. 2023g. Reactable: Enhancing react for table question answering. *arXiv preprint arXiv:2310.00815*.

Weichao Zhao, Hao Feng, Qi Liu, Jingqun Tang, Shu Wei, Binghong Wu, Lei Liao, Yongjie Ye, Hao Liu, Houqiang Li, et al. 2024. Tabpedia: Towards comprehensive visual table understanding with concept synergy. *arXiv preprint arXiv:2406.01326*.

Yilun Zhao, Yunxiang Li, Chenying Li, and Rui Zhang. 2022a. Multihiertt: Numerical reasoning over multi hierarchical tabular and textual data. *arXiv preprint arXiv:2206.01347*.

Yilun Zhao, Hongjun Liu, Yitao Long, Rui Zhang, Chen Zhao, and Arman Cohan. 2023. Knowledgemath: Knowledge-intensive math word problem solving in finance domains. *arXiv preprint arXiv:2311.09797*.

Yilun Zhao, Linyong Nan, Zhenting Qi, Rui Zhang, and Dragomir Radev. 2022b. Reastap: Injecting table reasoning skills during pre-training via synthetic reasoning examples. *arXiv preprint arXiv:2210.12374*.

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

Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models. *arXiv preprint arXiv:2311.07911*.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. Tat-qa: A question answering benchmark on a hybrid of tabular and textual content in finance. *arXiv preprint arXiv:2105.07624*.

Yujin Zhu, Zilong Zhao, Robert Birke, and Lydia Y Chen. 2022. Permutation-invariant tabular data synthesis. In *2022 IEEE International Conference on Big Data (Big Data)*, pages 5855–5864. IEEE.

A Table of Image-based Carriers

These tables offer flexibility for depicting complex structures (Zhao et al., 2024). However, they pose usability challenges. Two main approaches exist. One approach is table recognition, which employs techniques like OCR (Lin et al., 2022) to convert image tables into text. The other approach involves integrating images of tables as separate modalities into multimodal models (Kim et al., 2024b). However, this form of tables is currently difficult to modify using predefined operations, making it not the primary focus of our research. Nonetheless, it remains a promising area for future exploration.

Utilizing OCR (Lin et al., 2022; Prasad et al., 2020) and similar technologies (Li et al., 2023b) for table recognition within images and their conversion into textual formats is a straightforward approach (Kasem et al., 2022). However, our focus lies primarily on directly inputting image-formatted tables into MLLMs. Since the emergence of MLLMs such as GPT-4V (OpenAI, 2023) and Gemini (Team et al., 2023), several investigations have aimed to evaluate MLLMs' comprehension of table or chart images (Yang et al., 2023). Some MLLMs, like DeepSeek-VL (Lu et al., 2024), have even incorporated image-formatted tabular data during their pre-training phases. Additionally, multimodal datasets like CMMU (Zhang et al., 2024b) and M-Paper (Hu et al., 2023) contain image-based tabular data, while TableVQA-Bench (Kim et al., 2024b) is specifically designed for image-based table tasks.

Concerning image-formatted tables, their visual attributes are crucial. Deng et al. (Deng et al., 2024) found that applying different colors to individual rows enhances table reasoning for MLLMs, though the performance improvement for strong LLMs is marginal. Their study confirmed highlighting improves table reasoning. Other studies, such as TableVQA-Bench, distinguished between Table, Cell, Border, and Text attributes during table rendering but did not investigate the potential impact of these visual attributes on MLLMs' table comprehension.

MLLMs handle Text-based and Image-based tables similarly, Deng et al. (Deng et al., 2024) found. Both have strengths and weaknesses, but Text-based outperforms Image-based significantly in TableVQA-Bench's three tasks. Due to task and input method differences, further investigation into these methods' superiority is needed.

Research on image-formatted tables is nascent, lacking models optimized for such tables. Current benchmarks focus on computer-rendered tables, neglecting handwritten ones.

B Schema Construction

Another possible direction for table preprocessing is schema construction. Schema information is the structural information of the table. However, in many real-world scenarios, a lot of Schema information is missing, such as relational database tables stored in CSV files. For such data, if we can restore or reconstruct the table's schema information, it will be beneficial for LLMs to understand the table information. More diverse methods can be used to handle table data. An example of this would be generating SQL code for relational database tables that are stored in CSV files.

There have been some studies on column type detection (Hulsebos et al., 2019; Zhang et al., 2019; Suhara et al., 2022) and column relationship identification (Wang et al., 2021a; Iida et al., 2021), but they still differ from the requirements of schema construction. For instance, column type detection often involves classification tasks, making it challenging to handle variable types like VARCHAR(48). Relationship identification often doesn't involve recognizing primary keys or foreign keys. Schema construction itself is a complex issue, and due to the limitations of contextual windows, it's not feasible for LLMs to handle it entirely. Combining LLMs with tools might be a viable approach.

C Table RAG

In addition to RAG-in-Table, there is also Table RAG, which utilizes RAG technology to retrieve relevant question-answer pairs related to tables. However, table retrieval faces distinct challenges compared to conventional text tasks with RAG. These challenges include determining how to encode table information, identifying similar tables, and coordinating the encoding of table information with that of textual information.

In addition to retrieving relevant data for current table tasks through RAG, another approach is to apply RAG internally within the tables themselves, essentially treating the tables as retrieval knowledge bases. This allows for the retrieval of specific table contents based on input text information. The reason for conducting retrieval within the tables is to avoid inputting excessively large tables directly into LLMs, which can lead to excessively long contexts. This, in turn, increases inference costs, overlooks crucial information, and results in performance degradation (Kaddour et al., 2023). Even purportedly long-context supporting models may not fully address all challenges associated with lengthy contexts (Li et al., 2024c).

Many tasks involving the use of LLMs to handle tables often mention RAG. However, they often do not vectorize the tables themselves, only vectorizing user queries or SQL, such as DAILSQL (Gao et al., 2023a), DB-GPT (Xue et al., 2023), etc. Yang et al. (Yang et al., 2024) adopted a simple approach of converting all information into .docx documents and then vectorizing them. OPENTAB (Kong et al., 2024) employs BM25 (Robertson et al., 2009) for table RAG, while LIRAGE (Lin et al., 2023) uses ColBERT (Khattab and Zaharia, 2020) to simultaneously encode queries and tables for table RAG.

Wang et al. (Wang et al., 2022) tested two table vectorization methods — DPR (Karpukhin et al., 2020), and DTR (Herzig et al., 2021b) — and found that the structural information of tables might not be crucial in table retrieval. What matters most is the textual information within the tables. Hence, there might not be a need to design a specialized table vectorization model; employing a text vectorization model directly could suffice. However, this research only delves into the performance of table retrieval itself, without verifying the impact of different vectorization methods on the downstream tasks' performance with LLMs. We believe that more research is needed in the realm of table RAG.

D Comparison of LLM-based Agents for Table Tasks Based on Workflow

We have compiled the performance comparison of various agents into Table 2.

Our primary reference for summarizing these agents was their respective papers, without considering further improvements made to these systems when applied in real-world scenarios. Additionally, since these agents often do not adhere to the workflows defined in this paper, we mainly examined whether each step was explicitly mentioned in the literature. If a relevant technique was not mentioned in the paper, we represented it in the table with "-". The design of comparison metrics primarily follows the workflows defined in this paper and does not encompass all aspects of these agents' capabilities. We have endeavored to summarize the abilities of these agents to the best of our ability, but there may be areas where our summary is incomplete. We will further refine this table in the future.

Workflow	SheetAgent (Chen et al., 2024)	SheetCopilot (Li et al., 2024a)	Data-Copilot (Zhang et al., 2023f)	ReAcTable (Zhang et al., 2023g)	DAAgent (Hu et al., 2024)	DB-GPT (Xue et al., 2023)	Data Formulator (Wang et al., 2023b)	TableGPT (Zha et al., 2023)	EHRAgent (Shi et al., 2024)
Handled Spreadsheet Types	Sheet	Sheet	Sheet, database	Database	Sheet	Sheet, database	Sheet	Sheet, database	Database
Table Preprocessing Methods	-	-	-	-	Feature engineering, outlier detection, comprehensive data preprocessing	-	✓	-	-
Methods for Handling User Queries	Identification of unclear requirements	-	Query rewriting	-	-	Query rewriting	Supports user-designed concepts, some requiring programming, with multi-round interaction capability	Intent detection, vague input rejection	-
Table Retrieval	-	-	-	✓	-	-	-	-	-
Output Language Types	NL (natural language), SQL, Python	DSL	NL, code, interface invocation, tool-usage	NL, SQL, Python	NL, Python	NL, SQL, tool-usage	NL, Python	NL, DSL	NL (used for plan generation), Python, tool-usage
Table Reasoning Capability	-	-	-	✓	-	-	-	✓	-
Table Manipulation Types	Worksheet management, value processing, content summary, chart design, format adjustment	Cell value modification, formatting, worksheet management, formulas and functions, charts and pivot tables, etc.	Data visualization	Query	Summary statistics, correlation analysis, distribution analysis	SQL-based data analysis, data visualization, etc.	Reshaping, derivation (creating new variables), data visualization	Question answering, data manipulation, insert, delete, query, modify operations, data visualization, analysis report generation, etc.	SQL-based data analysis
Other Spreadsheet Task Abilities	-	-	-	-	Using machine learning algorithms for table prediction	Plan to add table prediction in the future	-	Table prediction	-
Tabular Data Safety	-	-	-	-	-	Protecting data privacy and security	-	privacy protection	-
Domain Adaptation	General	General	General	General	General	General	General	General, customizable fine-tuning for different domains	Medical

Table 2: Comparison of LLM-based agents for table tasks based on workflow.