A Dataset-Centric Survey of LLM-Agents for Data Science

Chuxuan Hu* *chuxuan3@illinois.edu*
chuxuan3

Dwip Dalal* *dwip2@illinois.edu*
dwip2

Xiaona Zhou* *xiaonaz2@illinois.edu*
xiaonaz2

Abstract

Large Language Models (LLMs) are revolutionizing data science workflows through automation and enhanced analytical capabilities, yet the effectiveness of these LLM-based agents varies significantly across different dataset structures and domain contexts. In this dataset-centric survey, we systematically examine how LLM-based data agents adapt to structured, semi-structured, and unstructured data, emphasizing their design considerations and operational capabilities. We introduce a hierarchical taxonomy linking agent functionality—ranging from data collection, storage, and preprocessing to analytical tasks like modeling, evaluation, interpretation, decision-making, and visualization—to specific dataset modalities and application domains. By conducting a detailed comparative analysis of over 50 recent LLM-based data agents, we reveal critical insights into how dataset characteristics influence agent architectures, planning strategies, multi-agent interactions, self-correction mechanisms, and specialized tool integration. Furthermore, we identify prominent gaps in current benchmark frameworks, highlighting the need for more comprehensive, standardized evaluation methods to assess robustness and generalizability. Finally, we outline future research directions that stress adaptive dataset-aware agent design, advanced multi-agent collaboration, domain-specific customization, and enhanced interpretability and real-time responsiveness, aiming to build more robust, adaptable, and transparent data science automation tools.

1 Introduction

The rapid advancements in Large Language Models (LLMs) have led to the emergence of LLM-based data agents, which promise to transform data analysis workflows by automating complex and time-consuming tasks. These agents have demonstrated capabilities in data exploration, preprocessing, visualization, model training, and even full pipeline orchestration. As a result, they are increasingly used to support analysts and domain experts across diverse fields, including finance, healthcare, education, and engineering. However, despite growing adoption and continuous innovation, there remains a critical gap in our understanding of how the design and performance of these agents vary depending on the characteristics of the datasets they interact with.

While a number of surveys have explored the design and functionality of LLM-based systems (Sun et al., 2024b; Lu et al., 2025), most center their analysis on internal mechanisms, such as prompting strategies, tool integration, reasoning capabilities, and interaction modes—without systematically considering how agents are shaped by the structure or domain of the data they are meant to analyze. Yet, dataset characteristics such as structure (e.g., tabular vs. document-based), domain specificity, noise levels, and scale can significantly impact the requirements and success of LLM agents. For instance, agents designed for relational databases

[*] Equal contribution

may require schema reasoning and SQL generation, while agents built for financial forecasting or medical research must accommodate domain-specific terminology, regulatory constraints, and temporal patterns. These distinctions are rarely made explicit in the literature, making it difficult to assess how well current agents generalize across use cases.

This survey addresses this overlooked dimension by adopting a dataset-centric perspective. Rather than grouping LLM agents solely by architectural or functional traits, we categorize them based on the datasets they are evaluated on and the types of data they are designed to handle. We construct a taxonomy that reflects both the data modality (structured, semi-structured, unstructured) and the domain context (e.g., scientific computing, enterprise analytics, healthcare), drawing on publicly available benchmarks and self-collected datasets cited in recent literature. This allows us to analyze the interplay between dataset properties and key agent design decisions, such as whether an agent adopts end-to-end automation versus modular planning, supports multi-agent collaboration, or integrates specialized tools for code generation, visualization, or database querying.

To guide this analysis, we introduce a two-layer framework. The first layer distinguishes between types of input data—structured (e.g., CSV, relational databases), semi-structured (e.g., JSON, spreadsheets), and unstructured (e.g., free-form documents or mixed text-visual inputs). The second layer maps agents to the domain-specific contexts in which they are evaluated, providing insight into how task requirements in areas like finance, biomedical analytics, or software engineering influence agent behavior. This structured view reveals both the strengths and blind spots of existing LLM-based agents when applied to real-world data workflows.

The scope of this survey is intentionally defined to enable a focused yet comprehensive analysis. We limit our coverage to LLM-based agents explicitly designed for data science tasks, such as data wrangling, analysis, and modeling, and that have been evaluated on identifiable datasets. This includes agents that operate in natural language environments (e.g., system prompts or conversational interfaces), as well as those embedded in programming notebooks, spreadsheets, or orchestration platforms. We exclude general-purpose LLM applications, such as open-domain dialogue systems or instruction-tuned models not tied to specific data analysis objectives. This scope ensures a grounded review that reflects both the practical utility and technical limitations of existing systems. By centering the survey on datasets and their associated demands, we aim to expose underexplored design considerations, inform benchmark development, and ultimately improve the generalizability and reliability of LLM-based data agents.

Contributions. This survey presents a structured synthesis of recent LLM-based data agents, emphasizing how they support end-to-end data science workflows across diverse dataset types and domains. Our main contributions are as follows:

- We organize agents using a hierarchical taxonomy (Figure 2) that reflects their roles across data management (collection, storage, preprocessing) and data analysis (modeling, evaluation, interpretation, decision making).
- We analyze four core analytical capabilities—*data modeling*, *model evaluation*, *data interpretation*, and *decision making*—with case studies highlighting how agents adapt to structured and unstructured data in different application contexts.
- We review upstream data management processes, showing how agents automate data collection, construct task-specific storage, and perform semantic preprocessing to support downstream tasks.
- We identify how dataset structure and domain specificity shape design choices, evaluation strategies, and agent behavior, and outline challenges in benchmarking and generalization.
- We provide a comprehensive comparison of over 25 recent agents (Table 1), covering planning, tool use, self-correction, interaction style, and evaluation datasets.

In the following sections, we summarize existing LLM-based agents designed for data analysis, highlighting their evaluation datasets, methodologies, and frameworks. While these agents have demonstrated success

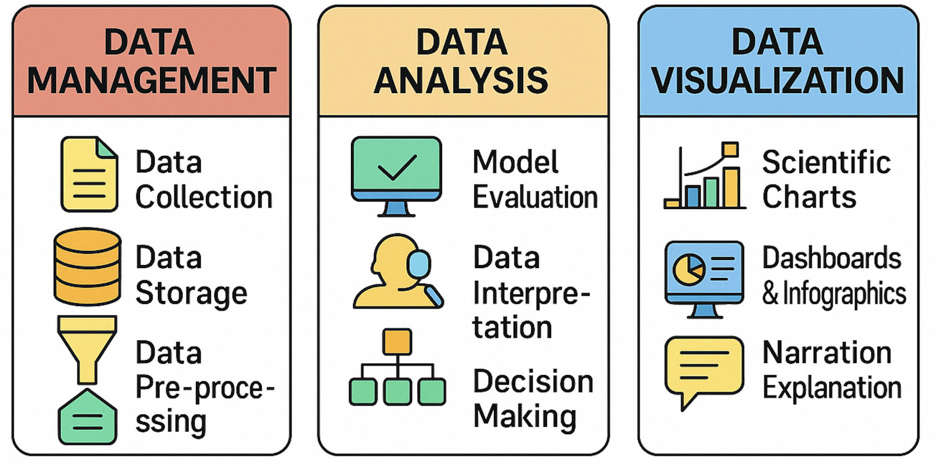

Figure 1: This figure presents a structured taxonomy of the data ecosystem, segmented into three core modules: Data Management, Data Analysis, and Data Visualization. Each module comprises semantically aligned sub-processes, illustrated with distinct, color-coded icons. Data Management includes foundational tasks like collection, storage, and pre-processing. Data Analysis captures the computational core, spanning modeling, evaluation, and interpretability. Data Visualization focuses on insight communication through charts, dashboards, and narratives. The taxonomy serves as both a conceptual framework and a practical workflow guide for building intelligent, data-centric systems.

in automating data science workflows, their effectiveness is often dataset-dependent, and no standardized framework exists for evaluating them across diverse datasets. Our survey addresses this by analyzing LLM-based data agents from a dataset-aware perspective, exposing key limitations in current benchmarks and methodologies.

2 Data Management

Data management consists of several critical stages that together form the foundation of modern data-centric systems. These stages include data collection, data storage, and data preprocessing, where preprocessing further encompasses data labeling and data cleaning. Each stage plays a vital role in transforming raw, unstructured inputs into high-quality, structured data that can support downstream analytics and decision-making. With the growing complexity and scale of real-world tasks, AI agents have increasingly been employed to automate, accelerate, and enhance each stage of this pipeline.

2.1 Data Collection

Automating the collection of data has been widely studied and applied across various domains. These systems range from simple web crawlers and scrapers to advanced AI-powered agents capable of navigating complex web interfaces, formulating search queries, and extracting task-specific information.

Numerous models for open web information retrieval have been proposed (Etzioni et al., 2004; 2011; Kamp et al., 2023), enabling systems to identify and retrieve relevant information in response to natural language queries. These models are foundational for applications such as question answering, fact verification, and report generation, where up-to-date and context-specific data is essential. As web content continues to grow rapidly in both scale and diversity, developing effective retrieval methods that can locate accurate, timely, and domain-relevant information remains a central focus in the field.

Agent Name	Type	UI	Planning	Self-Correcting	Multi-Agent	Tool Integration	Dataset(s)
Data-Copilot Zhang et al. (2023)	End-to-End	System	Linear	✓	✗	Python, C++, Matlab	Own(financial)
Spider2-V Cao et al. (2024)	End-to-End	OS-Based	-	✓	✗	BigQuery, dbt, Airbyte, etc	494 tasks from data warehousing to orchestration
Data Interpreter Hong et al. (2024)	End-to-End	CLI	Hierarchical	✓	✗	Python	InfiAgent-DABench, MATH, ML-Benchmark, own(Open-ended task benchmark)
LAMBDA Sun et al. (2024a)	Conversational	System	Basic I/O	✓	✓	Python	AIDS Clinical Trials Group Study 175, NHANES, Breast Cancer Wisconsin, Wine, Concrete Compressive Strength, Combined Cycle Power Plant, Abalone, Airfoil Self-Noise
Data Formulator 2 Wang et al. (2024a)	Conversational	System	Basic I/O	✓	✗	Vega-Lite	CO2 and electricity
Jupybara Wang et al. (2025)	Conversational	IDE-based	Linear	✓	✓	Jupyter Notebook	gender pay gap in Ireland
TableLLM Zhang et al. (2024b)	Conversational	Web-based	Linear	✓	✗	Python, SQL	WikiTableQuestion (WikiTQ), TAT-QA, FeTaQA and OTTQA
DS-Agent Guo et al. (2024)	End-to-end	CLI	Linear	✓	✓		30 representative data science tasks
AutoKaggle Li et al. (2024)	End-to-end	CLI	Linear	✓	✓	Python	eight Kaggle competitions, similar to DS-agent
CoddLLM Zhang et al. (2025)	Conversational	System	Linear	✗	✗	SQL	AnalyticsMMLU, Table Selection, Text-to-SQL
GPT4-As-Data-Analyst Cheng et al. (2023b)	End-to-End	System	Basic I/O	✗	✗	Python, SQL	NvBench (Natural Language to Visualization Tasks)
LEAP Hu et al. (2025)	Conversational	CLI	Linear	✓	✓	Python, SQL	QUIET-ML (self-collected social science research questions)
StructGPT Jiang et al. (2023)	End-to-End	System	Linear	✗	✗	SQL	WebQuestionsSP, MetaQA, WikiSQL, WTQ, TabFact
TAP4LLM Sui et al. (2024)	End-to-End	System	Hierarchical	✗	✗	SQL	SQA, HybridQA, TabFact, ToTTo, Spider
SheetCopilot Li et al. (2023)	End-to-End	IDE-based	Hierarchical	✓	✗	Web APIs	Adapted tasks from `https://superuser.com/`
Binder Cheng et al. (2023c)	End-to-End	System	Hierarchical	✗	✗	SQL, Web APIs	WIKITQ, TABFACT
TroVE Wang et al. (2024b)	End-to-End	System	Hierarchical	✗	✗	SQL, Python	TabMWP, WTQ, HiTab, GQA
TAG Biswal et al. (2024)	End-to-End	CLI	Basic I/O	✗	✗	SQL	BIRD
Dater Ye et al. (2023)	End-to-End	System	Hierarchical	✗	✗	SQL	TabFact, WikiTableQuestion, FetaQA
WaitGPT Xie et al. (2024)	Conversational	Web-based	Linear	✓	✗	Python, Web APIs	Corporate Compensation Insights; Flight Price Prediction; Synthesized dataset
InsightPilot Ma et al. (2023)	End-to-End	System	Hierarchical	✗	✗	LLM, QuickInsight, MetaInsight, XInsight	Student performance; Car sales
JarviX Liu et al. (2023)	End-to-End	Web-based	Hierarchical	✗	✗	LLM, AutoML, Whisper, Postgres, Elastic Search	Solar cell manufacturing; LCD factory data; Open-source tabular datasets
LLMDB Zhou et al. (2024)	End-to-End	System	Hierarchical	✓	✗	LLMs, vector databases, domain-specific models, LLM agent	Query rewrite; database diagnosis; data analytics
MatPlotAgent Yang et al. (2024b)	Conversational	Web-based	Linear	✓	✗	Python, Matplotlib, multi-modal LLMs	MatPlotBench; various scientific datasets
HuggingGPT Shen et al. (2024)	Conversational	Web-based	Hierarchical	✗	✓	ChatGPT, Hugging Face expert models	Multi-modal AI tasks (language, vision, speech)
ChatGPT as Data Scientist Hassan et al. (2023)	Conversational	System	Hierarchical	✗	✓	ChatGPT, Scikit-Learn	User-provided datasets
AutoML-Agent Trirat et al. (2024)	End-to-End	System	Hierarchical	✓	✓	LLMs, retrieval-augmented planning, multi-stage verification, plan decomposition	Seven downstream tasks; fourteen datasets
UFO Zhang et al. (2024a)	End-to-End	OS-Based	Hierarchical	✓	✓	GPT-Vision, pywinauto, Windows UI Automation	WindowsBench
DocETL Shankar et al. (2024)	End-to-End	CLI	Hierarchical	✓	✗	YAML, LLMs	Four unstructured document analysis tasks (e.g., police records, legal contracts)

Table 1: Comparison of LLM-based data agents for data science and analysis grouped by evaluation dataset.

In parallel, a wide range of web search tools (OpenAI, 2024b) and autonomous agents (He et al., 2024; Google, 2024; OpenAI, 2024a; Yang et al., 2023) have emerged to support complex information-seeking tasks. These tools form the core infrastructure that enables AI systems to interact with and retrieve information from the open web in a scalable and task-aware manner. Modern web search tools not only interface with APIs or search engines, but also incorporate capabilities such as dynamic query rewriting, result filtering, multi-hop reasoning, and evidence consolidation. They serve as the foundation upon which more sophisticated agents are built, allowing those agents to operate effectively in real-world scenarios where relevant information may be distributed across multiple sources, hidden behind interaction-heavy interfaces, or continually evolving.

For example, the Gemini Deep Research platform (Google, 2024) is capable of browsing the web to gather information and generate structured research reports, leveraging deep integration with Google's search ecosystem. It combines real-time web exploration with summarization and structured organization of content, allowing users to receive in-depth research briefings with citations and source tracking. Gemini Deep Research is designed to support a wide range of analytical workflows, from policy monitoring and academic literature reviews to competitive intelligence. It adapts its behavior based on task specifications, retrieving up-to-date content and synthesizing it into a format that resembles professional-grade research output. The system benefits from Google's infrastructure for indexing, ranking, and understanding web documents, giving it access to comprehensive and well-ranked sources while maintaining relevance to the task context.

Similarly, the recently released OpenAI Research Agent (OpenAI, 2024a) demonstrates multi-step planning and adaptive web search behavior, enabling it to retrieve, synthesize, and write complete reports in response to user-defined analytical tasks. The agent can decompose high-level objectives into a series of subgoals, each linked to targeted web searches, followed by content distillation, evidence alignment, and structured output generation. It supports reasoning over diverse document types, including news articles, blog posts, academic publications, and government websites. The OpenAI Research Agent can also engage in follow-up querying based on partial information and reconcile conflicting claims by returning to the source documents. Its design reflects a growing emphasis on grounded decision-making, as it explicitly cites supporting evidence and explains how retrieved content relates to the query. This makes it particularly suited for research support, claim validation, and high-stakes knowledge synthesis.

WebVoyager (He et al., 2024) is another notable example, which takes a more interaction-oriented approach, navigating through full web interfaces and emulating human-like browsing behaviors to extract task-relevant content. Unlike systems that rely primarily on API-based search or text retrieval, WebVoyager directly interacts with HTML-based environments, simulating cursor clicks, text inputs, and page transitions. It learns exploration policies through reinforcement learning and imitation learning, enabling it to handle diverse web layouts and dynamic content. This design makes it suitable for tasks that require interacting with forms, clicking through multi-layered menus, or extracting data from sources that do not expose structured APIs. WebVoyager also incorporates reward functions aligned with task success metrics, such as accuracy of extracted information or efficiency of navigation, which guide its training and refinement. Its human-like interaction capabilities open the door to more flexible and generalizable web agents that can operate across arbitrary websites without prior customization.

This process of retrieving data from the open web has become a critical foundation for enabling automated decision making, knowledge discovery, and research assistance in dynamic, real-world contexts.

2.2 Data Storage

Automating the storage of information plays a central role in enabling end-to-end intelligent systems, particularly in tasks that require multi-step reasoning or long-term context retention. One effective approach to this is the construction of task-driven databases, where structured storage is dynamically created and updated based on the specific information needs of a given task.

A widely adopted framework that supports this capability is Retrieval-Augmented Generation (RAG) (Lewis et al., 2021), which integrates knowledge retrieval with language generation to improve response quality. A knowledge base (Wang et al., 2023b; Liška et al., 2022; Kasai et al., 2024) is a critical component of RAG systems, providing a structured repository for storing and accessing previously retrieved content. While many RAG models retrieve unstructured text passages from large corpora, the integration of a knowledge base offers a mechanism for organizing this information into structured formats that support precise querying. This approach improves interpretability and supports richer multi-turn interactions by enabling consistent use of previously retrieved information.

A concrete example of this integration is TAG (Biswal et al., 2024), which extends the RAG framework by incorporating NL2SQL techniques to enable dynamic construction of sub-structured tables directly from large relational databases. Given a natural language query, TAG identifies relevant portions of the original dataset and generates a tailored table that captures the subset of fields and rows most pertinent to the user's question. These sub-structured views are not generic summaries but are semantically aligned with the query intent, providing a precise, declarative interface that bridges human language and machine-readable data. By translating user intent into SQL-compatible forms, TAG enables systems to interact with databases in a flexible and interpretable manner, making it possible to execute complex queries without requiring predefined schemas or manual intervention. The resulting tables can be directly used for downstream reasoning, aggregation, and reporting, streamlining the end-to-end workflow from query to insight. This architecture supports a wide range of applications, from enterprise analytics to automated report generation, where the ability to adaptively structure and interpret original data is key to delivering accurate and actionable results.

2.3 Data Pre-processing

This step typically involves converting unstructured data into structured tables by automatically assigning labels and cleaning the data to ensure its usability for downstream analysis.

For example, in the context of data labeling, LEAP (Hu et al., 2025) is an end-to-end system designed to support social scientists in answering natural language queries that require semantic understanding of unstructured data, such as Tweets. It iteratively applies machine learning functions to annotate the data with relevant semantic labels, such as sentiment or emotion, and transforms it into structured tabular form. LEAP also addresses key challenges such as selecting the appropriate ML models and handling vague or underspecified queries. By filtering out ambiguous questions and enabling programmatic integration of both built-in and user-defined ML functions, it allows domain experts to perform sophisticated semantic analyses without writing any code, making the data preparation process both efficient and accessible.

In the context of data cleaning, this step focuses on filtering out irrelevant information, resolving inconsistencies, and reformatting the data to meet the requirements of subsequent processing stages. The Data Interpreter (Hong et al., 2024) exemplifies this process by using a combination of hierarchical graph modeling and programmable node generation to break down complex data workflows into manageable subcomponents. Through iterative refinement and verification, it ensures the correctness of each intermediate step, which improves both code robustness and overall workflow reliability. Its design allows the system to dynamically adapt to changes in task structure and data state, making it particularly effective for real-world data science problems that involve multiple interconnected and evolving subgoals.

Together, these systems highlight how recent advances in LLM-based agents are enhancing the automation of early-stage data workflows, particularly in scenarios that demand both semantic understanding and rigorous preprocessing of real-world data.

3 Data Analysis

LLM-based data agents are increasingly deployed to automate and support core components of the data analysis workflow. This section examines how agents operate across four key analytical capabilities: *data modeling*, *model evaluation*, *data interpretation*, and *decision making*. These categories reflect the major functional roles that agents assume in real-world data science tasks, from understanding schema relationships to assessing model performance and communicating insights. Our analysis is grounded in representative case studies of recent agents, such as AutoKaggle (Li et al., 2024), CoddLLM (Zhang et al., 2025), Data Interpreter (Hong et al., 2024), GPT4-As-Data-Analyst (Cheng et al., 2023b), and Jupybara (Wang et al., 2025), and emphasizes how different design choices respond to challenges posed by dataset structure, domain specificity, and task complexity. Through this lens, we highlight emerging trends, comparative strengths, and common limitations in current agent designs.

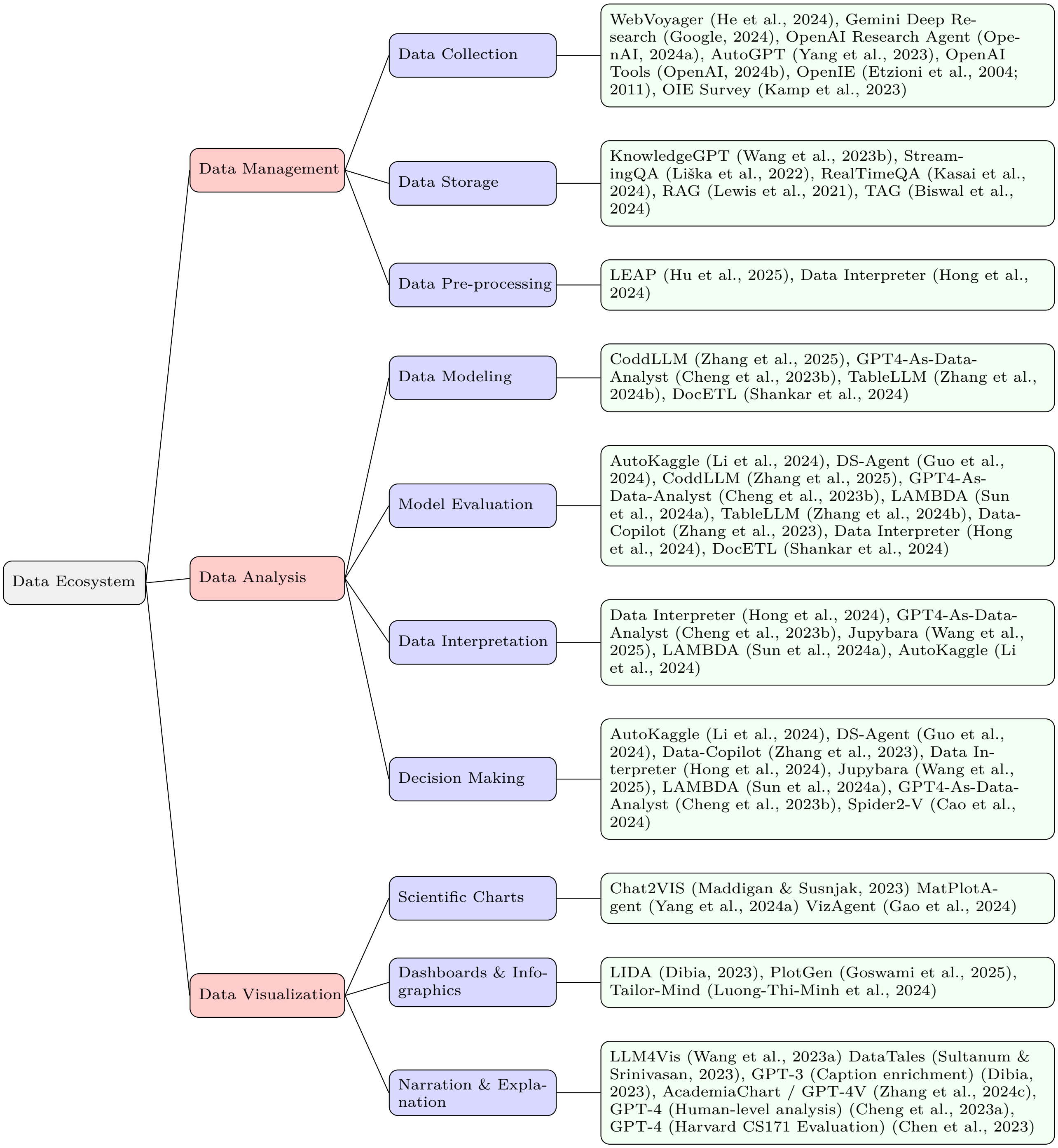
Data Ecosystem
Data Management
Data Collection
WebVoyager (He et al., 2024), Gemini Deep Research (Google, 2024), OpenAI Research Agent (OpenAI, 2024a), AutoGPT (Yang et al., 2023), OpenAI Tools (OpenAI, 2024b), OpenIE (Etzioni et al., 2004; 2011), OIE Survey (Kamp et al., 2023)
Data Storage
KnowledgeGPT (Wang et al., 2023b), StreamingQA (Liška et al., 2022), RealTimeQA (Kasai et al., 2024), RAG (Lewis et al., 2021), TAG (Biswal et al., 2024)
Data Pre-processing
LEAP (Hu et al., 2025), Data Interpreter (Hong et al., 2024)
Data Analysis
Data Modeling
CoddLLM (Zhang et al., 2025), GPT4-As-Data-Analyst (Cheng et al., 2023b), TableLLM (Zhang et al., 2024b), DocETL (Shankar et al., 2024)
Model Evaluation
AutoKaggle (Li et al., 2024), DS-Agent (Guo et al., 2024), CoddLLM (Zhang et al., 2025), GPT4-As-Data-Analyst (Cheng et al., 2023b), LAMBDA (Sun et al., 2024a), TableLLM (Zhang et al., 2024b), Data-Copilot (Zhang et al., 2023), Data Interpreter (Hong et al., 2024), DocETL (Shankar et al., 2024)
Data Interpretation
Data Interpreter (Hong et al., 2024), GPT4-As-Data-Analyst (Cheng et al., 2023b), Jupybara (Wang et al., 2025), LAMBDA (Sun et al., 2024a), AutoKaggle (Li et al., 2024)
Decision Making
AutoKaggle (Li et al., 2024), DS-Agent (Guo et al., 2024), Data-Copilot (Zhang et al., 2023), Data Interpreter (Hong et al., 2024), Jupybara (Wang et al., 2025), LAMBDA (Sun et al., 2024a), GPT4-As-Data-Analyst (Cheng et al., 2023b), Spider2-V (Cao et al., 2024)
Data Visualization
Scientific Charts
Chat2VIS (Maddigan & Susnjak, 2023) MatPlotAgent (Yang et al., 2024a) VizAgent (Gao et al., 2024)
Dashboards & Infographics
LIDA (Dibia, 2023), PlotGen (Goswami et al., 2025), Tailor-Mind (Luong-Thi-Minh et al., 2024)
Narration & Explanation
LLM4Vis (Wang et al., 2023a) DataTales (Sultanum & Srinivasan, 2023), GPT-3 (Caption enrichment) (Dibia, 2023), AcademiaChart / GPT-4V (Zhang et al., 2024c), GPT-4 (Human-level analysis) (Cheng et al., 2023a), GPT-4 (Harvard CS171 Evaluation) (Chen et al., 2023)

Figure 2: Taxonomy of tools in the data ecosystem categorized by function. Each functional node has a child node containing corresponding systems.

3.1 Data Modeling

Among LLM-based agents for data analysis, CoddLLM (Zhang et al., 2025) stands out for its explicit focus on data modeling capabilities. It addresses a major limitation in prior work—most models operate on single or pairs of tables—by training on tasks that require understanding inter-table relationships and complex data schemas. This includes a Text-to-Schema task, where the model must design a database schema from a textual scenario, and a table selection benchmark (WikiPage-TS) that demands comprehension of multi-table structures. The training corpus also includes content centered around database modeling and representation, supporting the agent's ability to capture how business entities are represented in complex datasets.

While not designed explicitly for data modeling, GPT4-As-Data-Analyst (Cheng et al., 2023b) requires GPT-4 to navigate existing database schemas to extract and analyze relevant data. This implicitly reflects an understanding of relational models, as the agent must identify which tables are relevant, how they connect, and what fields to extract or join to generate insights. Similarly, TableLLM (Zhang et al., 2024b) contributes to the field by supporting advanced table manipulation tasks such as querying, updating, merging, and charting. Each of these operations assumes a coherent understanding of table structure and data organization, aligning with the principles of data modeling at a tabular level. DocETL (Shankar et al., 2024) also engages with document-centric data modeling, requiring agents to reason over unstructured sources such as contracts or policy documents. Through agentic query rewriting and structured output extraction, it builds implicit models of complex documents to support information retrieval and downstream processing.

3.2 Model Evaluation

Model evaluation is a central capability of LLM-based data agents, serving not only to assess the performance of trained models but also to inform workflow revisions, agent collaboration, and strategic decisions. While agents vary in their architectural and procedural designs, their evaluation mechanisms fall broadly into three categories: (1) dedicated evaluator modules, (2) iterative feedback-based pipelines, and (3) interpretive or critique-oriented systems.

The most modular evaluation structures are seen in AutoKaggle (Li et al., 2024) and LAMBDA (Sun et al., 2024a), where evaluation is delegated to specialized agents. In AutoKaggle, the Developer agent performs model building, validation, and prediction across multiple machine learning tools, selecting the best model (e.g., a random forest achieving a validation score of 0.8379). Simultaneously, the Reviewer agent aggregates scores and feedback from multiple agents into a unified report. These mechanisms support both detailed validation and high-level metrics such as the Average Normalized Performance Score and Comprehensive Score. LAMBDA adopts a similar architecture in which a Programmer builds and trains models while an Inspector assesses their correctness and reliability. Ablation studies demonstrate that removing the Inspector agent reduces execution pass rates, underscoring its role in evaluation.

In contrast, agents such as DS-Agent (Guo et al., 2024) and Data Interpreter (Hong et al., 2024) treat model evaluation as an integral part of an iterative feedback loop. DS-Agent employs a case-based reasoning framework that builds, evaluates, and refines models using metrics like AUROC, MAE, RMSE, and MCRMSE. Its Logger module summarizes model performance and progress, guiding further revisions. Data Interpreter, which automates end-to-end workflows, includes model evaluation as a core step and applies standard metrics on ML-Benchmark datasets such as Breast Cancer Wisconsin and Wine Recognition. Evaluation feedback informs subsequent actions within the pipeline, reinforcing an adaptive execution style.

Other agents embed evaluation within interpretive or critique-driven workflows. Jupybara (Wang et al., 2025), for instance, uses a multi-agent setup where Critics assess generated responses—including model training and evaluation code—for correctness, clarity, and strategic alignment. These critiques guide a Refiner agent in improving results. Similarly, GPT4-As-Data-Analyst (Cheng et al., 2023b) is assessed using both automatic and human evaluation criteria such as correctness, alignment, and complexity. In comparative studies, its performance is benchmarked against that of junior and senior data analysts, simulating realistic human-centered evaluation settings.

Spider2-V (Cao et al., 2024) offers a flexible evaluation interface through its evaluator dictionary, allowing custom metric functions for each task. These may include traditional model performance measures or domain-specific criteria for assessing generated outputs in interactive workflows.

In summary, while model evaluation is universally critical, the strategies employed by LLM agents vary. AutoKaggle and LAMBDA exemplify modular evaluation through specialized agents. DS-Agent and Data Interpreter integrate evaluation directly into their execution pipelines for continuous feedback. Jupybara and GPT4-As-Data-Analyst emphasize interpretation and critique to guide refinement, and Spider2-V highlights extensibility in evaluation logic. Together, these systems illustrate the diverse yet foundational role of model evaluation in enabling reliable, intelligent data agents.

3.3 Data Interpretation

Data interpretation allows LLM-agents not only to process and analyze data but also to translate analytical outputs into meaningful, human-understandable insights. While all the agents operate within data-centric workflows, their interpretive strategies can be broadly grouped into three categories: integrated interpretation within procedural workflows, dedicated interpretive agents in multi-agent systems, and summarization components for stakeholder communication.

Agents such as Data Interpreter (Hong et al., 2024) and GPT4-As-Data-Analyst (Cheng et al., 2023b) embed interpretive reasoning throughout the stages of the analysis pipeline. Data Interpreter performs tasks like outlier detection, correlation analysis, and visualization in an end-to-end setting, requiring the agent to continuously evaluate intermediate results to guide subsequent decisions. Its 25% performance boost on InfiAgent-DABench suggests that this interpretive loop is crucial for effective decision-making across tasks. Similarly, GPT-4 demonstrates interpretive reasoning aligned with human analyst expectations. It is evaluated not only on output correctness and fluency, but also on its ability to recognize complex trends, make comparisons, and deliver insights in a concise and structured format. These agents exemplify a procedural model of interpretation, where understanding emerges from step-by-step engagement with data transformations.

In contrast, systems like Jupybara (Wang et al., 2025) and LAMBDA (Sun et al., 2024a) emphasize agent-level interpretability, embedding dedicated roles for critique and summarization. Jupybara's Interpretation & Summary Critic reviews outputs from other agents to produce narratives that are semantically precise and pragmatically relevant. This decentralized interpretive process mimics human review cycles and is credited with producing more digestible and informative analysis, according to user studies. LAMBDA, though oriented around code execution, outputs final natural language responses that summarize results for users. These summaries are informed by the Inspector agent's assessment of execution correctness and the Programmer's encoded logic, together ensuring that final insights are accurate and relevant. Both systems prioritize clarity and readability in the final presentation of results, supporting interpretation through dedicated reasoning components.

AutoKaggle (Li et al., 2024) provides a distinct model where interpretation is embedded in its Summarizer agent, whose role is to distill complex, multi-phase workflows into coherent narratives. This includes selecting relevant images, designing phase-aligned questions, and answering them to form structured reports. The Summarizer interprets outputs from other agents to bridge the gap between raw analytical results and stakeholder-facing communication. While AutoKaggle's Developer and Planner modules focus on task execution, the Summarizer ensures that the analytical outcomes are contextually framed and readily consumable.

Together, these agent systems reflect a broader evolution in LLM-based data science workflows—from mechanical execution to reflective reasoning and communication. Whether integrated within procedural workflows, structured as standalone interpretive components, or dedicated to summarization, these systems prioritize the transformation of raw analysis into insights that are intelligible, actionable, and aligned with user needs.

3.4 Decision-Making

LLM-based data agents demonstrate a wide range of decision-making capabilities, which can be broadly grouped into three categories: structured planners and evaluators, case-based and feedback-driven systems, and multi-agent deliberative frameworks. These categories capture how agents analyze situations, select actions, and adapt over time within complex data science workflows.

Structured planners and evaluators such as AutoKaggle (Li et al., 2024), Data-Copilot (Zhang et al., 2023), and Data Interpreter (Hong et al., 2024) rely on predefined or dynamically constructed pipelines where decisions are made about task sequencing, tool invocation, and evaluation. In AutoKaggle, the Planner structures task roadmaps by organizing actions based on context and past results, while the Reviewer aggregates performance feedback and recommendations into coherent summaries. Similarly, Data-Copilot makes decisions by classifying task and operation types, then loading the appropriate modules for further processing. Data Interpreter also engages in sequential decision-making by selecting tools based on dependencies and choosing when and how to apply techniques like outlier detection or correlation analysis.

In contrast, agents like DS-Agent (Guo et al., 2024) and GPT4-As-Data-Analyst (Cheng et al., 2023b) embody more adaptive, feedback-oriented decision processes. DS-Agent is grounded in a case-based reasoning paradigm, retrieving and re-ranking prior cases based on similarity and utility feedback to guide experiment planning. It iteratively refines its decision-making across retrieval, reuse, and ranking steps informed by past performance. GPT-4, acting as a data analyst, demonstrates contextual decision-making by synthesizing variables, proposing new evaluation metrics, and adapting analysis strategies based on implicit reasoning over the data and background knowledge.

A third class involves multi-agent deliberation and critique, as seen in Jupybara (Wang et al., 2025) and LAMBDA (Sun et al., 2024a). These systems decentralize decision-making across cooperating agents that assume specialized roles. In Jupybara, critics evaluate and challenge initial analytical outputs, with the Refiner agent synthesizing these critiques into improved responses, providing rationales for each accepted or rejected suggestion. This layered feedback mechanism simulates peer review and iterative refinement. LAMBDA takes a more procedural view, with a Programmer generating code and an Inspector deciding how and when to intervene for debugging. Together, they make iterative decisions that align with real-world coding and analysis workflows.

Agents like Spider2-V (Cao et al., 2024) further expand the space by embedding decision-making into real-time interactive environments. In this setup, decisions correspond to fine-grained physical actions (e.g., clicking, typing) guided by observation and goal-directed reasoning. While this is distinct from structured planning or high-level critique, it still reflects dynamic, context-sensitive decision-making.

Across these systems, we observe shared goals but differing mechanisms. Structured planners tend to optimize execution efficiency, feedback-driven systems adapt over time to improve performance, and deliberative multi-agent architectures emphasize interpretability and flexibility. These contrasting approaches highlight the growing sophistication of LLM-based agents in navigating and managing decision-making within end-to-end data workflows.

4 Data Visualization

Large-language-model agents are beginning to own the entire visual-analytics loop—from raw tables to polished stories. We survey recent work (2022-2024) by grouping it according to the type of visualization output the agent targets.

4.1 Exploratory & Scientific Chart

The most active line of research equips code-synthesis LLMs with agentic scaffolds that translate natural-language analytics intents into single-view plots. Early prompt-only systems such as Chat2VIS show that a few carefully crafted exemplars let GPT-3, Codex, or ChatGPT emit correct Vega-Lite or Matplotlib scripts for underspecified queries, outperforming bespoke NL2Vis models while slashing engineering cost (Maddigan & Susnjak, 2023). Building on this, MatPlotAgent introduces an execution–inspection–repair loop: a code LLM drafts Python, renders the figure, and a vision-enabled GPT-4V judges the bitmap; any mismatch triggers automatic debugging. On the new MatPlotBench corpus of 100 scientific tasks, the self-correcting agent raises success rates of several base LLMs by up to 30 percentage points (Yang et al., 2024a). VizAgent generalizes the pattern—interpreting a dataset, eliciting user intent, generating alternative charts in Matplotlib, Seaborn, or Plotly, and ranking them by LLM-based heuristics—thereby demonstrating cross-library versatility and pinpointing library-specific failure modes (Gao et al., 2024). Complementary evaluation work confirms that even zero-/few-shot GPT-3.5 already surpasses purpose-built NL2Vis baselines on NVBench, though schema grounding and multi-table queries remain brittle .

4.2 Dashboards, Infographics & Multi-View Composition

When the goal is a compound visualization—dashboards, multi-panel reports, or stylized infographics—researchers orchestrate several LLM calls into modular pipelines.

LIDA exemplifies this trend. Its four-stage workflow (data Summarizer → Goal Explorer → VisGenerator → Infographer) lets GPT-4 alternate between reasoning over data semantics, emitting grammar-agnostic code, and post-processing with image-generation models to yield publication-ready infographics, a hybrid GUI / chat UI supports iterative refinement (Dibia, 2023). In the scientific domain, PlotGen extends the concept with multiple collaborating agents—planner, coder, and critic—each prompting the next to converge on high-fidelity plots for research datasets (Goswami et al., 2025). Domain-specific dashboards are also emerging: Tailor-Mind fine-tunes an LLM on educational Q-&-A data, couples it with a knowledge-map visualization, and acts as a personalized tutor inside the dashboard; user studies reveal significant gains in self-regulated learning engagement (Luong-Thi-Minh et al., 2024). Together, these systems hint at a future where LLM agents dynamically compose multi-view, interactive artefacts rather than isolated charts.

4.3 Narrative, Explanation & Visual Storytelling

LLMs excel at language; recent work leverages this strength to augment visuals with prose and to evaluate visualization choices. LLM4Vis reframes chart-type recommendation as a two-step ChatGPT dialogue that not only selects an optimal chart but also produces a human-like rationale; few-shot bootstrapping improves rationale coherence while achieving state-of-the-art accuracy on VizML (Wang et al., 2023a). DataTales goes further, attaching an LLM to Tableau dashboards to draft full paragraphs that weave insights into a cohesive story; professionals in an IEEE VIS study valued the reduced "blank-page" effort, yet demanded better tone control (Sultanum & Srinivasan, 2023). Caption-level support is feasible too—GPT-3 can rewrite terse figure captions into engaging narratives when prompted with context, illustrating post-hoc enrichment of static visuals (Dibia, 2023).

On the evaluative side, AcademiaChart probes vision-language models' ability to read charts: GPT-4V can reproduce underlying Python code from 2500 scientific figures, especially when guided by chain-of-thought prompts, whereas open-source VLMs lag markedly (Zhang et al., 2024c). Broader assessments show GPT-4 approaching human analysts in end-to-end tasks—including SQL, plotting, and insight narration (Cheng et al., 2023a)—and even scoring a "B" on Harvard's CS171 coursework (Chen et al., 2023). These studies signal that LLMs are maturing from code generators into communicative partners that both craft and critique visual stories.

Across charting, dashboard composition, and storytelling, LLM agents increasingly interleave natural-language reasoning with code execution and visual feedback. While closed-source models like GPT-4/4V set the bar, emerging agentic scaffolds (iterative debugging, multi-agent collaboration, fine-tuned domain tutors) suggest pathways to reliable, transparent visualization assistance—even with less capable models. Open challenges remain in robust schema linkage, perceptual fidelity checks for complex layouts, and human-in-the-loop control of narrative tone, marking fertile ground for future research in LLM-centric data science.

5 Future directions

To further enhance the capabilities and generalizability of LLM-based data agents, several promising research directions merit attention. One key avenue involves dataset-aware adaptation, where agents dynamically adjust their internal strategies in response to the specific structure, complexity, and domain of a dataset. This calls for the integration of meta-learning, adaptive prompting mechanisms, and dataset-conditioned modeling techniques, allowing agents to better generalize and effectively tackle a wide range of previously unseen data types. Alongside this, the establishment of standardized and comprehensive benchmarking frameworks is crucial. These benchmarks should systematically evaluate performance across datasets of varying complexities and domains using well-defined metrics, thereby improving reproducibility and enabling rigorous comparisons between agent architectures.

Model	Structured	Semi-structured	Unstructured	Domain	Specific Domains
Data-Copilot	✓	✗	✓	Single	Finance
Spider2-V	✓	✗	✓	Multiple	Economics, Education, Engineering
Data Interpreter	✓	✗	✓	Multiple	Mathematics, Machine Learning, Open-ended Tasks
LAMBDA	✓	✓	✗	Multiple	Healthcare, Biology/Agriculture, Engineering/Materials Science, Education
Data Formulator 2	✓	✓	✗	Multiple	Environmental Science, Energy, Finance
Jupybara	✓	✗	✓	Multiple	Social Sciences, Finance, Healthcare
TableLLM	✓	✓	✓	Multiple	Question Answering, General Knowledge, SQL
DS-Agent	✓	✓	✓	Multiple	Agriculture, Machine Learning, Time Series, Finance, Healthcare, Question Answering
AutoKaggle	✓	✓	✗	Multiple	House Prices, Education, Healthcare, Manufacturing
CoddLLM	✓	✗	✓	Multiple	Database Management, Sports, Web Data, General Knowledge
GPT4-As-Data-Analyst	✓	✓	✓	Multiple	Hospitality, Education, E-commerce, Politics
LEAP	✓	✗	✓	Multiple	Social Sciences, Psychology, Legal Services, Misinformation
StructGPT	✓	✓	✗	Multiple	Question Answering, Database Management, Semantic Parsing
TAP4LLM	✓	✓	✓	Multiple	Question Answering, General Knowledge, SQL
SheetCopilot	✗	✓	✗	Single	Finance
Binder	✓	✗	✗	Single	Question Answering
TroVE	✓	✓	✓	Multiple	Math, Question Answering
TAG	✓	✗	✗	Multiple	Database Management, Semantic Reasoning
Dater	✓	✗	✗	Single	Question Answering
WaitGPT	✓	✗	✗	Multiple	Finance, Economics, Education
InsightPilot	✓	✗	✗	Multiple	Finance, Education
JarviX	✓	✗	✗	Single	Material Science
MatPlotAgent	✓	✗	✗	Multiple	Scientific data from Matplotlib Gallery and OriginLab GraphGallery
HuggingGPT	✗	✗	✓	Multiple	NLP, CV, Audio, Video
ChatGPT as Data Scientist	✓	✗	✗	Single	Education
AutoML-Agent	✓	✓	✓	Multiple	Biology, Economics, Education, Weather, Electricity, Academics
UFO	✓	✓	✓	Single	Software Applications
DocETL	✗	✓	✓	General	Police records, legal contracts, medical reports, financial disclosures

Table 2: Model Capabilities and Domain Applicability

Another important direction lies in advancing multi-agent collaboration. Future systems should mimic human-like team workflows, incorporating structured coordination strategies, intelligent task division, and feedback mechanisms among agents. Such enhancements can simulate the dynamics of collaborative data science teams, leading to more efficient and contextually aware problem-solving. Additionally, domain-specific customization is essential for deploying LLM-based agents in specialized fields such as healthcare, finance, or scientific research. Customization can be achieved through fine-tuning with domain-specific corpora, injecting structured knowledge, and integrating specialized tools that align with the particular challenges and terminology of the domain.

Finally, as these agents become increasingly sophisticated, interpretability and real-time interactivity become vital. It is critical to ensure transparency in decision-making by incorporating methods for rationale generation, intermediate reasoning steps, and outputs that are easily understandable by end-users, particularly in sensitive or high-stakes settings. Moreover, agents must evolve to support interactive and real-time decision-making, improving their responsiveness and allowing them to adapt fluidly to user inputs and live data streams. Such real-time capabilities will significantly broaden their applicability and bring them closer to the performance and adaptability of human analysts.

6 Conclusion

This survey has presented a comprehensive dataset-centric perspective on LLM-based data agents, critically analyzing their capabilities and limitations across various stages of data science workflows. We introduced a hierarchical taxonomy categorizing these agents based on their proficiency in managing structured, semi-structured, and unstructured datasets, and their adaptability across diverse domain contexts. By reviewing key design strategies—such as modular planning, multi-agent collaboration, self-correction mechanisms, and specialized tool integrations—we have highlighted how dataset-specific characteristics shape agent design, performance, and evaluation.

Despite significant advances, our analysis identifies clear gaps in benchmark diversity, robustness in generalization, and adaptability to complex dataset characteristics. Addressing these challenges through standardized evaluation frameworks, dynamic dataset-aware agent designs, domain-specific customization, and enhanced interpretability will be essential for future development. As LLM-based data agents continue to mature, pursuing these directions will ensure that they become increasingly reliable, flexible, and valuable tools in real-world data science workflows, capable of effectively supporting analysts and domain experts across varied and complex analytical tasks.

References

Asim Biswal, Liana Patel, Siddarth Jha, Amog Kamsetty, Shu Liu, Joseph E. Gonzalez, Carlos Guestrin, and Matei Zaharia. Text2sql is not enough: Unifying ai and databases with tag, 2024. URL `https://arxiv.org/abs/2408.14717`.

Ruisheng Cao, Fangyu Lei, Haoyuan Wu, Jixuan Chen, Yeqiao Fu, Hongcheng Gao, Xinzhuang Xiong, Hanchong Zhang, Yuchen Mao, Wenjing Hu, Tianbao Xie, Hongshen Xu, Danyang Zhang, Sida Wang, Ruoxi Sun, Pengcheng Yin, Caiming Xiong, Ansong Ni, Qian Liu, Victor Zhong, Lu Chen, Kai Yu, and Tao Yu. Spider2-v: How far are multimodal agents from automating data science and engineering workflows?, 2024. URL `https://arxiv.org/abs/2407.10956`.

Zhutian Chen, Chenyang Zhang, Qianwen Wang, Jakob Troidl, Simon Warchol, Johanna Beyer, Nils Gehlenborg, and Hanspeter Pfister. Beyond generating code: Evaluating gpt on a data visualization course. In *2023 IEEE VIS Workshop on Visualization Education, Literacy, and Activities (EduVis)*, pp. 16–21. IEEE, 2023.

Liying Cheng, Xingxuan Li, and Lidong Bing. Is GPT-4 a good data analyst? In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 9496–9514, Singapore, December 2023a. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-emnlp.637. URL `https://aclanthology.org/2023.findings-emnlp.637/`.

Liying Cheng, Xingxuan Li, and Lidong Bing. Is gpt-4 a good data analyst?, 2023b. URL `https://arxiv.org/abs/2305.15038`.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. Binding language models in symbolic languages, 2023c. URL `https://arxiv.org/abs/2210.02875`.

Victor Dibia. LIDA: A tool for automatic generation of grammar-agnostic visualizations and infographics using large language models. In Danushka Bollegala, Ruihong Huang, and Alan Ritter (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 3: System Demonstrations)*, pp. 113–126, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-demo.11. URL `https://aclanthology.org/2023.acl-demo.11/`.

Oren Etzioni, Michael Cafarella, Doug Downey, Stanley Kok, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Web-scale information extraction in knowitall: (preliminary results). In *Proceedings of the 13th International Conference on World Wide Web*, WWW '04, pp. 100–110, New York, NY, USA, 2004. Association for Computing Machinery. ISBN 158113844X. doi: 10.1145/988672.988687. URL `https://doi.org/10.1145/988672.988687`.

Oren Etzioni, Anthony Fader, Janara Christensen, Stephen Soderland, and Mausam Mausam. Open information extraction: the second generation. In *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence - Volume Volume One*, IJCAI'11, pp. 3–10. AAAI Press, 2011. ISBN 9781577355137.

Lin Gao, Jing Lu, Zekai Shao, Ziyue Lin, Shengbin Yue, Chiokit Ieong, Yi Sun, Rory James Zauner, Zhongyu Wei, and Siming Chen. Fine-tuned large language model for visualization system: A study on self-regulated learning in education. *IEEE Transactions on Visualization and Computer Graphics*, 2024.

Google. Gemini deep research. `https://gemini.google/overview/deep-research/`, 2024.

Kanika Goswami, Puneet Mathur, Ryan Rossi, and Franck Dernoncourt. Plotgen: Multi-agent llm-based scientific data visualization via multimodal feedback. *arXiv preprint arXiv:2502.00988*, 2025.

Siyuan Guo, Cheng Deng, Ying Wen, Hechang Chen, Yi Chang, and Jun Wang. Ds-agent: Automated data science by empowering large language models with case-based reasoning. *arXiv preprint arXiv:2402.17453*, 2024.

Md Mahadi Hassan, Alex Knipper, and Shubhra Kanti Karmaker Santu. Chatgpt as your personal data scientist. 2023.

Hongliang He, Wenlin Yao, Kaixin Ma, Wenhao Yu, Yong Dai, Hongming Zhang, Zhenzhong Lan, and Dong Yu. Webvoyager: Building an end-to-end web agent with large multimodal models, 2024. URL https://arxiv.org/abs/2401.13919.

Sirui Hong, Yizhang Lin, Bangbang Liu, Binhao Wu, Danyang Li, Jiaqi Chen, Jiayi Zhang, Jinlin Wang, Lingyao Zhang, Mingchen Zhuge, et al. Data interpreter: An llm agent for data science. *arXiv preprint arXiv:2402.18679*, 2024.

Chuxuan Hu, Austin Peters, and Daniel Kang. Leap: Llm-powered end-to-end automatic library for processing social science queries on unstructured data. *arXiv preprint arXiv:2501.03892*, 2025.

Jinhao Jiang, Kun Zhou, Zican Dong, Keming Ye, Xin Zhao, and Ji-Rong Wen. StructGPT: A general framework for large language model to reason over structured data. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 9237–9251, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.574. URL https://aclanthology.org/2023.emnlp-main.574/.

Serafina Kamp, Morteza Fayazi, Zineb Benameur-El, Shuyan Yu, and Ronald Dreslinski. Open information extraction: A review of baseline techniques, approaches, and applications, 2023. URL https://arxiv.org/abs/2310.11644.

Jungo Kasai, Keisuke Sakaguchi, Yoichi Takahashi, Ronan Le Bras, Akari Asai, Xinyan Yu, Dragomir Radev, Noah A. Smith, Yejin Choi, and Kentaro Inui. Realtime qa: What's the answer right now?, 2024. URL https://arxiv.org/abs/2207.13332.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks, 2021. URL https://arxiv.org/abs/2005.11401.

Hongxin Li, Jingran Su, Yuntao Chen, Qing Li, and Zhaoxiang Zhang. Sheetcopilot: Bringing software productivity to the next level through large language models, 2023. URL https://arxiv.org/abs/2305.19308.

Ziming Li, Qianbo Zang, David Ma, Jiawei Guo, Tianyu Zheng, Xinyao Niu, Xiang Yue, Yue Wang, Jian Yang, Jiaheng Liu, et al. Autokaggle: A multi-agent framework for autonomous data science competitions. *arXiv preprint arXiv:2410.20424*, 2024.

Shang-Ching Liu, ShengKun Wang, Tsungyao Chang, Wenqi Lin, Chung-Wei Hsiung, Yi-Chen Hsieh, Yu-Ping Cheng, Sian-Hong Luo, and Jianwei Zhang. 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*, pp. 622–630, 2023.

Adam Liška, Tomáš Kočiský, Elena Gribovskaya, Tayfun Terzi, Eren Sezener, Devang Agrawal, Cyprien de Masson d'Autume, Tim Scholtes, Manzil Zaheer, Susannah Young, Ellen Gilsenan-McMahon, Sophia Austin, Phil Blunsom, and Angeliki Lazaridou. Streamingqa: A benchmark for adaptation to new knowledge over time in question answering models, 2022. URL https://arxiv.org/abs/2205.11388.

Weizheng Lu, Jing Zhang, Ju Fan, Zihao Fu, Yueguo Chen, and Xiaoyong Du. Large language model for table processing: A survey. *Frontiers of Computer Science*, 19(2):192350, 2025.

Hue Luong-Thi-Minh, Vinh Nguyen-The, and Truong Quach Xuan. Vizagent: Towards an intelligent and versatile data visualization framework powered by large language models. In *International Conference on Advances in Information and Communication Technology*, pp. 89–97. Springer, 2024.

Pingchuan Ma, Rui Ding, Shuai Wang, Shi Han, and Dongmei Zhang. Insightpilot: An llm-empowered automated data exploration system. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pp. 346–352, 2023.

Paula Maddigan and Teo Susnjak. Chat2vis: Fine-tuning data visualisations using multilingual natural language text and pre-trained large language models. *arXiv preprint arXiv:2303.14292*, 2023.

OpenAI. Openai agents python: Research bot example. `https://github.com/openai/openai-agents-python/tree/main/examples/research_bot`, 2024a. Accessed: 2025-04-08.

OpenAI. Tools and web search | openai platform, 2024b. URL `https://platform.openai.com/docs/guides/tools-web-search?api-mode=chat`. Accessed: 2025-04-08.

Shreya Shankar, Tristan Chambers, Tarak Shah, Aditya G Parameswaran, and Eugene Wu. Docetl: Agentic query rewriting and evaluation for complex document processing. *arXiv preprint arXiv:2410.12189*, 2024.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. In *Advances in Neural Information Processing Systems*, 2024.

Yuan Sui, Jiaru Zou, Mengyu Zhou, Xinyi He, Lun Du, Shi Han, and Dongmei Zhang. Tap4llm: Table provider on sampling, augmenting, and packing semi-structured data for large language model reasoning, 2024. URL `https://arxiv.org/abs/2312.09039`.

Nicole Sultanum and Arjun Srinivasan. Datatales: Investigating the use of large language models for authoring data-driven articles. In *2023 IEEE Visualization and Visual Analytics (VIS)*, pp. 231–235. IEEE, 2023.

Maojun Sun, Ruijian Han, Binyan Jiang, Houduo Qi, Defeng Sun, Yancheng Yuan, and Jian Huang. Lambda: A large model based data agent. *arXiv preprint arXiv:2407.17535*, 2024a.

Maojun Sun, Ruijian Han, Binyan Jiang, Houduo Qi, Defeng Sun, Yancheng Yuan, and Jian Huang. A survey on large language model-based agents for statistics and data science. *arXiv preprint arXiv:2412.14222*, 2024b.

Patara Trirat, Wonyong Jeong, and Sung Ju Hwang. Automl-agent: A multi-agent llm framework for full-pipeline automl. `https://arxiv.org/abs/2410.02958`, 2024. arXiv preprint arXiv:2410.02958.

Chenglong Wang, Bongshin Lee, Steven Drucker, Dan Marshall, and Jianfeng Gao. Data formulator 2: Iteratively creating rich visualizations with ai. *arXiv preprint arXiv:2408.16119*, 2024a.

Huichen Will Wang, Larry Birnbaum, and Vidya Setlur. Jupybara: Operationalizing a design space for actionable data analysis and storytelling with llms. *arXiv preprint arXiv:2501.16661*, 2025.

Lei Wang, Songheng Zhang, Yun Wang, Ee-Peng Lim, and Yong Wang. Llm4vis: Explainable visualization recommendation using chatgpt. *arXiv preprint arXiv:2310.07652*, 2023a.

Xintao Wang, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyu He, Zhouhong Gu, Yanghua Xiao, and Wei Wang. Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases, 2023b. URL `https://arxiv.org/abs/2308.11761`.

Zhiruo Wang, Daniel Fried, and Graham Neubig. Trove: Inducing verifiable and efficient toolboxes for solving programmatic tasks, 2024b. URL `https://arxiv.org/abs/2401.12869`.

Liwenhan Xie, Chengbo Zheng, Haijun Xia, Huamin Qu, and Chen Zhu-Tian. Waitgpt: Monitoring and steering conversational llm agent in data analysis with on-the-fly code visualization. In *Proceedings of the 37th Annual ACM Symposium on User Interface Software and Technology (UIST)*, pp. 1–14, Pittsburgh, PA, USA, 2024. doi: 10.1145/3654777.3676374.

Hui Yang, Sifu Yue, and Yunzhong He. Auto-gpt for online decision making: Benchmarks and additional opinions, 2023. URL https://arxiv.org/abs/2306.02224.

Zhiyu Yang, Zihan Zhou, Shuo Wang, Xin Cong, Xu Han, Yukun Yan, Zhenghao Liu, Zhixing Tan, Pengyuan Liu, Dong Yu, Zhiyuan Liu, Xiaodong Shi, and Maosong Sun. MatPlotAgent: Method and evaluation for LLM-based agentic scientific data visualization. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11789–11804, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.701. URL https://aclanthology.org/2024.findings-acl.701/.

Zhiyu Yang, Zihan Zhou, Shuo Wang, Xin Cong, Xu Han, Yukun Yan, Zhenghao Liu, Zhixing Tan, Pengyuan Liu, Dong Yu, Zhiyuan Liu, Xiaodong Shi, and Maosong Sun. Matplotagent: Method and evaluation for llm-based agentic scientific data visualization. *arXiv preprint arXiv:2402.11453*, 2024b.

Yunhu Ye, Binyuan Hui, Min Yang, Binhua Li, Fei Huang, and Yongbin Li. Large language models are versatile decomposers: Decompose evidence and questions for table-based reasoning, 2023. URL https://arxiv.org/abs/2301.13808.

Chaoyun Zhang, Liqun Li, Shilin He, Xu Zhang, Bo Qiao, Si Qin, Minghua Ma, Yu Kang, Qingwei Lin, Saravan Rajmohan, and Qi Zhang. Ufo: A ui-focused agent for windows os interaction. 2024a.

Jiani Zhang, Hengrui Zhang, Rishav Chakravarti, Yiqun Hu, Patrick Ng, Asterios Katsifodimos, Huzefa Rangwala, George Karypis, and Alon Halevy. Coddllm: Empowering large language models for data analytics. *arXiv preprint arXiv:2502.00329*, 2025.

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

Xiaokang Zhang, Jing Zhang, Zeyao Ma, Yang Li, Bohan Zhang, Guanlin Li, Zijun Yao, Kangli Xu, Jinchang Zhou, Daniel Zhang-Li, et al. Tablellm: Enabling tabular data manipulation by llms in real office usage scenarios. *arXiv preprint arXiv:2403.19318*, 2024b.

Zhehao Zhang, Weicheng Ma, and Soroush Vosoughi. Is GPT-4V (ision) all you need for automating academic data visualization? exploring vision-language models' capability in reproducing academic charts. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 8271–8288, Miami, Florida, USA, November 2024c. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.485. URL https://aclanthology.org/2024.findings-emnlp.485/.

Xuanhe Zhou, Xinyang Zhao, and Guoliang Li. Llm-enhanced data management. *arXiv preprint arXiv:2402.02643*, 2024.