A Survey of Data Agents: Emerging Paradigm or Overstated Hype?

Yizhang Zhu, Liangwei Wang, Chenyu Yang, Xiaotian Lin, Boyan Li, Wei Zhou, Xinyu Liu, Zhangyang Peng, Tianqi Luo, Yu Li, Chengliang Chai, Chong Chen, Shimin Di, Ju Fan, Ji Sun, Nan Tang, Fugee Tsung, Jiannan Wang, Chenglin Wu, Yanwei Xu, Shaolei Zhang, Yong Zhang, Xuanhe Zhou, Guoliang Li* and Yuyu Luo*

Awesome Data Agents: https://github.com/HKUSTDial/awesome-data-agents

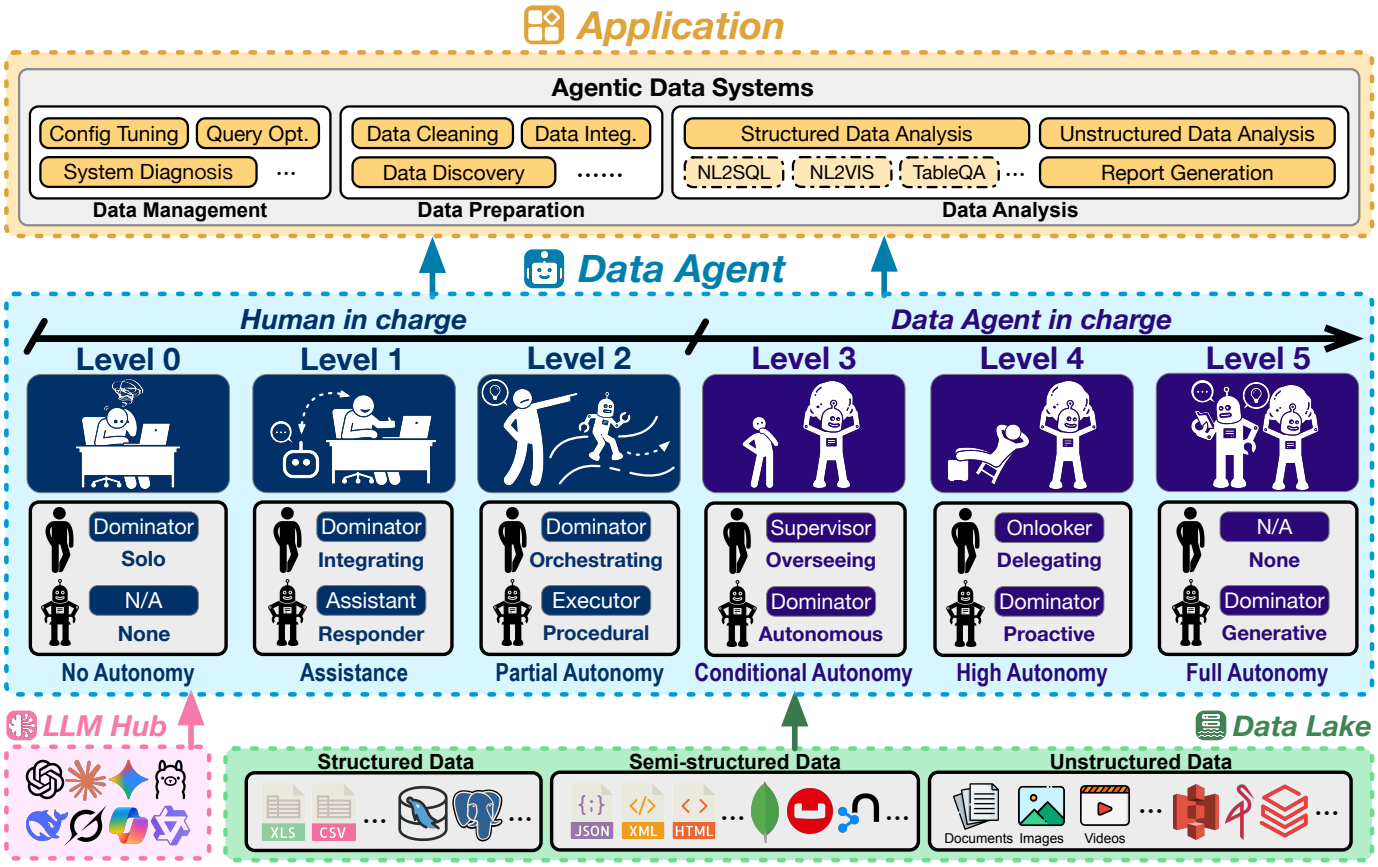

Fig. 1: An Overview of Data Agents.

***Abstract*—The rapid advancement of large language models (LLMs) has spurred the emergence of data agents—autonomous systems designed to orchestrate Data + AI ecosystems for tackling complex data-related tasks. However, the term "data agent" currently suffers from terminological ambiguity and inconsistent adoption, conflating simple query responders with sophisticated autonomous architectures. This terminological ambiguity fosters mismatched user expectations, accountability challenges, and barriers to industry growth. Inspired by the SAE J3016 standard for driving automation, this survey introduces the first systematic hierarchical taxonomy for data agents, comprising six levels that delineate and trace progressive shifts in autonomy, from manual operations (L0) to a vision of generative, fully autonomous data agents (L5), thereby clarifying capability boundaries and responsibility allocation. Through this lens, we offer a structured review of existing research arranged by increasing autonomy, encompassing specialized data agents for data management, preparation, and analysis, alongside emerging efforts toward versatile, comprehensive systems with enhanced autonomy. We further analyze critical evolutionary leaps and technical gaps for advancing data agents, especially the ongoing L2-to-L3 transition, where data agents evolve from procedural execution to autonomous orchestration. Finally, we conclude with a forward-looking roadmap, envisioning the advent of proactive, generative data agents.**

***Index Terms*—Data Agents, Autonomy, Data Management, Data Preparation, Data Analysis, Data Lake**

*Corresponding authors: Guoliang Li (liguoliang@tsinghua.edu.cn) and Yuyu Luo (yuyuluo@hkust-gz.edu.cn).

Yizhang Zhu, Liangwei Wang, Chenyu Yang, Xiaotian Lin, Boyan Li, Xinyu Liu, Zhangyang Peng, Tianqi Luo, Nan Tang, Fugee Tsung and Yuyu Luo are with The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China.

Wei Zhou and Xuanhe Zhou are with Shanghai Jiao Tong University, Shanghai, China. Yu Li, Ju Fan and Shaolei Zhang are with Renmin University of China, Beijing, China. Chengliang Chai is with Beijing Institute of Technology, Beijing, China. Shimin Di is with Southeast University, Nanjing, China. Ji Sun, Jiannan Wang, Yong Zhang and Guoliang Li are with Tsinghua University, Beijing, China. Chong Chen and Yanwei Xu are with Huawei. Chenglin Wu is with DeepWisdom.

I. INTRODUCTION

THE way humans interact with data is undergoing a revolutionary transformation. Traditionally, handling data-related tasks from management, preparation, and analysis has been a demanding endeavor, requiring specialized expertise, extensive manual effort, and solid technical proficiency [1]–[10]. Therefore, a long-standing aspiration in data science and analytics has been to develop an intelligent agent capable of autonomously managing, preparing, and analyzing data to deliver trustworthy insights with minimal human intervention [11].

The advent of large language models (LLMs) and LLM agents is bringing us closer to this vision [12]–[14]. Leveraging their remarkable capabilities in comprehension and reasoning [15]–[17], LLM agents have evolved beyond simple question answering. Recent advances in LLM-based agents demonstrate not only enhanced reasoning ability but also emerging capacities for environmental perception and interaction, memory retention, problem decomposition, strategic planning, and external tool invocation [18]. For instance, early frameworks like ReAct [19] integrate reasoning with action, enabling LLMs to create and adjust high-level plans while interacting with external environments. Building on this, CoALA [20] introduces a memory system to augment decision-making, while AFlow [21] focuses on the automated generation and optimization of agentic workflows. Tool-using in LLM agents is also a significant topic. ToolQA [22] explores the use of external tools like database loaders and code sandboxes, and more recent efforts like ReTool [23] employ reinforcement learning to further enhance the tool invocation capabilities of LLM agents.

A. The Dawn of Data Agents

Within this trend, *Data Agents* are introduced to address the distinctive challenges of data-intensive environments. A data agent is defined as a comprehensive, LLM-powered architecture that orchestrates the Data + AI ecosystem to autonomously perform a wide range of data-related tasks [24], [25]. As illustrated in Figure 1, the data agents serve as a central intelligence layer that bridges user-facing applications with the underlying data infrastructure by producing output such as optimized database configurations, prepared data, data insights, visualization charts, or analytical reports. Formally, we can define a data agent $\mathcal{A}$ that operates on raw data $\mathcal{D}$ within an environment $\mathcal{E}$ (which can include DBMS, code interpreters, APIs, etc.), utilizing LLMs $\mathcal{M}$, ultimately producing an output $\mathcal{O}$ to tackle the data-related task $\mathcal{T}$. This can be abstractly represented as:

$$\mathcal{A} : (\mathcal{T}, \mathcal{D}, \mathcal{E}, \mathcal{M}) \rightarrow \mathcal{O}.$$

Different from general LLM agents such as those designed for mathematical reasoning or open-ended conversation, data agents are built to navigate vast, heterogeneous data lakes that are too large and complex for holistic processing. Table I compares data agents with general LLM agents. Unlike general reasoning or generation tasks, where problems are self-contained and well-defined in finite prompts, data lakes comprise large-scale, heterogeneous sources that vary in format and structure, making it infeasible to ingest all data into a context window. Consequently, data agents must excel in actively exploring and interacting with the data environment, sampling subsets, probing schemas, and refining queries to uncover insights on demand without exhaustive data consumption. Furthermore, instead of working with static, readily available data, data agents constantly handle dynamic and potentially noisy data, presenting challenges in properly managing and preparing data to support efficient and effective data-related tasks.

Addressing these challenges demands specialized capabilities, including: (i) perception, monitoring, and interactive exploration of vast, heterogeneous data lakes through techniques like sampling, querying, and environmental feedback to enable reasoning that aligns with user intent without full data ingestion; (ii) robust invocation of specialized data toolkit, such as SQL equivalence checker, DBMS utilities, code interpreters, or visualization libraries; (iii) adaptive resolution and specialized knowledge to dynamically address noise, inconsistencies, scalability constraints, and real-time updates in data environments, ensuring effectiveness in tasks like data cleaning or integration where errors can cascade and compromise downstream insights.

The emergence of data agents marks a critical step toward realizing the aspiration of democratizing data-related tasks [12], [26], [27]. This progress is evident across the data lifecycle which outlines three interconnected phases: (1) Data Management, encompassing (i) Configuration Tuning for optimizing system parameters (*e.g.,* database knobs) to boost performance; (ii) Query Optimization involving SQL query rewriting and efficient execution plan selection; and (iii) System Diagnosis for detecting, analyzing, and resolving system anomalies or faults. (2) Data Preparation, involving (i) Data Cleaning to detect and fix errors, inconsistencies, or missing values in raw data; (ii) Data Integration to merge heterogeneous sources while handling schema and entity conflicts; and (iii) Data Discovery to identify relevant datasets, metadata, or patterns in expansive data lakes. (3) Data Analysis, covering (i) Structured Data Analysis for reasoning over tabular/relational data (*e.g.,* via TableQA, NL2SQL, or NL2VIS); (ii) Unstructured Data Analysis to extract insights from documents, images, etc.; and (iii) Report Generation to compile findings into coherent narratives.

Recent studies seek to alleviate the labor-intensive nature of these traditional tasks through the development of advanced data agents. For instance, to mitigate the expertise gap in database maintenance, GaussMaster [28] introduced a multi-agent copilot for services like index advising. To tackle the time-consuming aspects of data preparation, AutoPrep [29] is introduced with enhanced reasoning and tool invocation to perform natural language question-aware data preparation. Furthermore, recognizing that data analysis frequently requires solid domain knowledge and skills, Alpha-SQL [30] and nvAgent [31] facilitate seamless interaction with databases and the curation of data visualizations through natural language interfaces. To support more extensive and versatile data operations, recent frameworks such as iDataLake [32] have been

TABLE I: Comparison between General LLM Agents and Data Agents

Aspect	General LLM Agents	Data Agents
Primary Focus	Task and Content Centric: *Completing defined tasks or generating content.*	Data-Lifecycle Centric: *Data management, preparation, and analysis.*
Problem Scope	Self-contained and Static: *Acts on explicit instructions and a finite prompt.*	Exploratory and Dynamic: *Actively explores and navigatse vast, dynamic data lakes.*
Input Data	Small-Scale and Ready-to-Use: *Typically receives manageable, clean inputs.*	Large-Scale and "Raw": *Designed to handle heterogeneous, dynamic, and noisy raw data.*
Tool Invocation	General-Purpose Toolkit: *Web search, calculators, OCR, image generators, etc.*	Specialized Data Toolkit: *DB loaders, SQL equivalence checker, visualization libraries, etc.*
Primary Output	Generative Artifacts: *Human-consumable product: dialogues, reasoning, images, etc.*	Data Products and Insights: *Config, processed data, insights, visualizations, analytical report, etc.*
Error Consequnce	Localized: *Typically affects limited to only the direct output.*	Cascading: *Errors can cascade, affecting downstream insights.*

proposed to encompass data linking, pipeline orchestration, and execution for heterogeneous data analytics over data lakes.

B. The Terminological Ambiguity of Data Agents

Despite significant advancements and promise, the term "Data Agent" is applied inconsistently across research and industry, resulting in considerable terminological ambiguity. This ambiguity conflates systems of profoundly different autonomy, reliability, and complexity under a single, imprecisely defined umbrella term. For instance, recent research endeavors focus on developing sophisticated agentic data systems to autonomously interact with data lakes, invoke external tools (such as search engines, code interpreters, and database connectors), orchestrate and optimize tailored pipelines for highly comprehensive and complex data-related tasks with minimal human intervention. In contrast, the "data agent" term is also applied to more rudimentary, narrowly scoped systems, which are often limited to providing atomic task assistance without self-optimization through environmental perception and interaction.

Such terminological ambiguity introduces challenges for user trust, governance, and the healthy progression of the field, introducing interconnected risks.

- **User-Side Risk.** The ambiguity creates expectation mismatches. When users are unable to understand accurately or are unaware of the scope and limitations of data agents' capabilities, they can form a flawed understanding of their functionality. This leads to a notable gap between users' expectations and the system's actual performance. Consequently, users may either reject valid outputs or, more perilously, place undue reliance on erroneous ones.
- **Governance Risk.** This expectation mismatch further exposes a fundamental challenge of accountability. When a data agent is operated beyond its capabilities and potentially results in consequences such as data leakage, privacy violations, erroneous analytical reports, or regulatory noncompliance, the lines of responsibility become indistinct. This creates complex legal and ethical considerations: does the onus of responsibility lie with the human operator, who may have misused the system due to an incomplete understanding of its capability boundaries, or with the data agent vendor, for providing a flawed system? This indistinction highlights a critical challenge for effective governance.
- **Industry-Side Risk.** The combination of user misunderstanding and ambiguous accountability creates barriers to industry development. Without a shared taxonomy to distinguish data agents by autonomy and responsibility, it can hinder objective system comparison and foster an environment susceptible to blurring and overstated claims, which further exacerbates the initial challenges of mismatched user expectations and accountability indistinction. When system failures occur without clear accountability, market confidence can be eroded, ultimately slowing the adoption of emerging technologies. Addressing these challenges requires establishing a clear, common language for classifying data agents.

C. A Hierarchical Taxonomy for Data Agents

The challenges currently facing data agents are not unprecedented. A similar situation was previously encountered by the driving-automation community, which contended with the widespread and often imprecise use of the term "self-driving". This single label conflated fundamentally different capabilities, ranging from driver assistance to efforts towards higher autonomy. To tackle this, the Society of Automotive Engineers (SAE) introduced the J3016 standard [33], a six-level taxonomy. By explicitly allocating responsibility between human and machine across Levels 0 to 5, SAE J3016 has established a common language for users, manufacturers, engineers, regulators, and insurers to assess capabilities, limitations, and liability [34]. Such a shared framework has proven effective in grounding public discourse, clarifying accountability, and guiding engineering roadmaps.

The precedent set by the driving automation field offers an illuminative path forward. To resolve the confusion surrounding data agents, we advocate for and structure this survey around a hierarchical taxonomy for data agents, analogous to the SAE standard, that maps the progressive transitions of control and responsibility in data-related tasks from the human to the data agent as autonomy increases. It reflects how human role shifts from a hands-on operator to a supervisor, a passive onlooker, and eventually to complete disengagement, while the data agent evolves from an assistance tool into a fully autonomous and accountable data scientist.

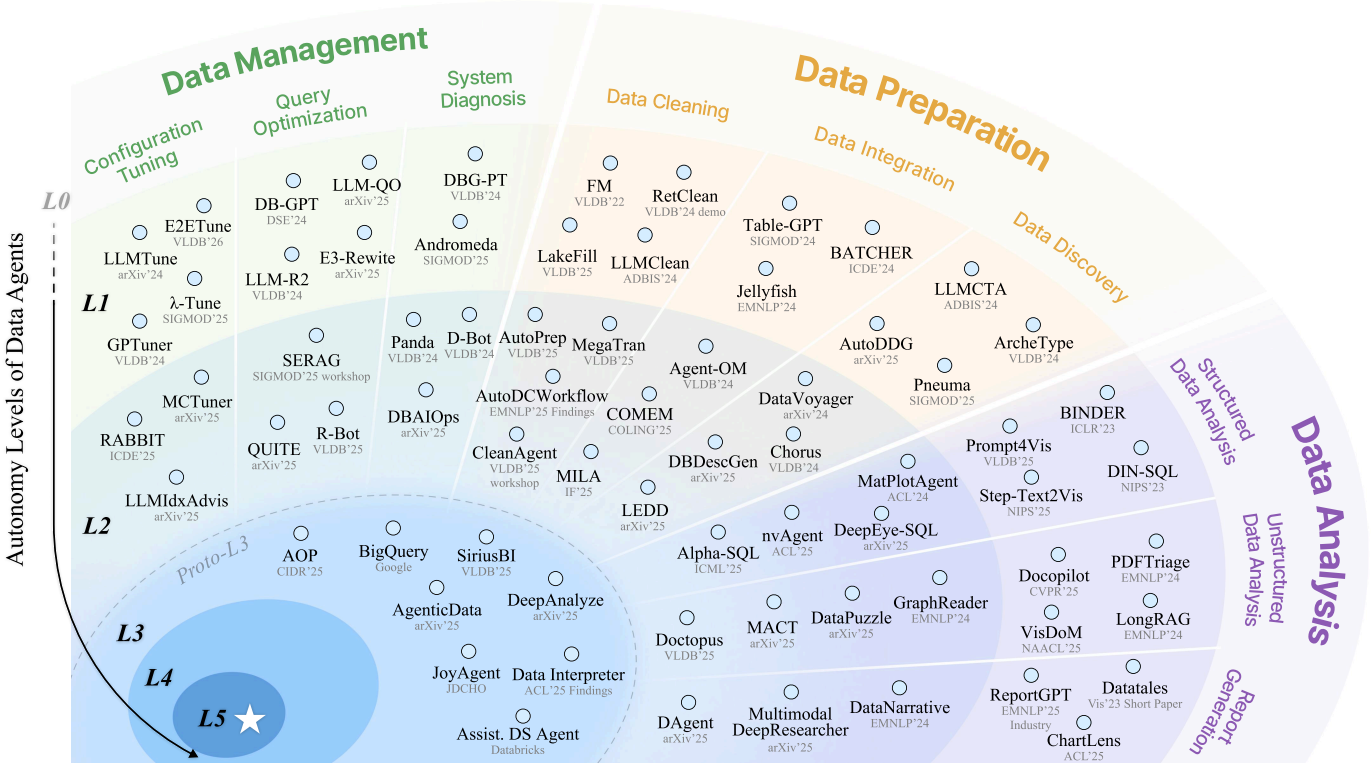

Fig. 2: Representative Data Agents Across Different Levels.

Concretely, as illustrated in Figure 1, we define six levels of data agent autonomy. L0 corresponds to purely manual work performed entirely by humans. L1 introduces preliminary, stateless assistance for isolated tasks. L2 advances to partial automation, where the data agent can perceive and interact with the environment to execute task-specific procedures operating within human-orchestrated pipelines. L3 marks conditional autonomy: data agents autonomously orchestrate and optimize pipelines to address diverse and comprehensive data-related tasks under human supervision. L4 represents high autonomy, where proactive data agents achieve sustained self-governance, requiring no supervision. Finally, L5 envisions fully autonomous data agents that are capable of knowledge creation and paradigm innovation, functioning as expert data scientists.

By introducing such a hierarchical taxonomy for data agents to investigate and discuss existing work, we aim to clarify capability boundaries and lines of accountability at each level, thereby helping manage user expectations, steer research, and support effective governance. A detailed elaboration of each level is provided in Section II. Figure 2 illustrates representative data agents across levels and tasks examined in this paper.

D. Comparison and Our Contributions

Differences from Existing Surveys Our survey distinguishes itself from existing surveys [11], [26], [27], [35]–[40] and tutorials [12]–[14], [41] in following aspects.

- We propose a novel hierarchical taxonomy (L0–L5) for data agents, classifying them by progressive autonomy levels. Building on this, we construct a structured review through its lens as demonstrated in Figure 2, whereas prior surveys primarily categorize by agentic architecture [27], [37], [39] or application scenarios [11], [26], lacking a unified framework for tracing their autonomy progression in data-related tasks.
- We investigate data agents across a comprehensive data-related tasks, spanning management, preparation, and analysis for data varying in structures and formats within heterogeneous data lakes. In contrast, some earlier surveys often focus on subdomains or individual stages [13], [35], [36], [40], [41], lacking a holistic perspective across diverse data-related tasks.
- We emphasize the cutting-edge advances in data agents, covering both state-of-the-art specialized systems tailored for specific tasks, and emerging efforts toward versatile and comprehensive data agents with self-orchestrating pipelines, which are underrepresented or absent in existing surveys and tutorials [11], [12], [14], [26].
- We highlight and discuss evolutionary leaps and technical gaps in data agent advancement, particularly the ongoing progress toward self-orchestrating and versatile data agents, revealing challenges such as limited pipeline orchestration, reliance on predefined operators, incomplete data lifecycle coverage, and deficiencies in strategic reasoning that hinder the current leap. This evolutionary perspective identifies key research bottlenecks and opportunities overlooked by prior works [11], [26], [27], [35], [37]–[39].
- We provide a forward-looking roadmap, outlining future directions such as autonomous problem discovery in data

lakes, trade-offs over the data lifecycle, long-horizon and holistic views across data management, preparation, and analysis to achieve highly autonomous data agents with proactive self-governance; ultimately, we envision fully autonomous and generative data agents.

Contributions. Building on these distinctions, our primary contributions are:

- *Novel Hierarchical Taxonomy:* The first systematic hierarchical taxonomy (L0–L5) for data agents, establishing a clear framework to compare existing data agents, delineate capability boundaries, and clarify accountability, thus enabling practitioners to align expectations and intervention with autonomy levels.
- *Structured and Systematic Review:* A structured review of data agents tracing autonomy progression in data-related tasks, mapping the state-of-the-art, and identifying underexplored areas.
- *Analysis of Evolutionary Leaps and Gaps:* In-depth analysis of evolutionary leaps and current challenges hindering the development of autonomous data agents.
- *Forward-Looking Roadmap and Vision:* A research roadmap detailing promising future directions and visions toward proactive and ultimately generative data agents.

E. Survey Organization.

In the remainder of this survey, Section II provides an exposition of our proposed L0–L5 taxonomy for data agents, and elaborates on the evolutionary leaps between autonomy levels. Following this, Sections III, IV, and V systematically survey existing works that fall into L0/L1, L2, and the emerging efforts toward L3, respectively. In Section VI, we look to the future, discussing the vision and challenges associated with achieving L4 and L5 data agents. Finally, Section VII concludes this survey.

II. Levels of Data Agents

A. The L0–L5 Hierarchy

As discussed in Section I-B, the inconsistent use of the term "data agent" leads to ambiguity regarding their capabilities and responsibility boundaries. To bring clarity, inspired by the SAE J3016 standard for driving automation [33], we propose a hierarchical framework with levels from 0 (L0) to 5 (L5), categorizing data agents based on their degree of autonomy.

As illustrated in Figure 1, the introduced hierarchical taxonomy is structured around the progressive transfer of dominance and responsibility in data-related tasks from the human to the data agent. When the data agent's autonomy level increases, the human's role transfers from a hands-on dominator to a supervisor, then to a passive onlooker, and finally to complete disengagement, while the data agent evolves from an auxiliary query responder into a fully autonomous, generative data scientist. Below, we elaborate on each level of autonomy for data agents.

- **L0: No Autonomy.** L0 is defined as having no data agent involvement in data tasks, with all tasks in data management, preparation, and analysis being entirely human-driven. For example, every step from formulating SQL queries to interact with databases, coding scripts for data cleaning, to conducting analyses, creating visualizations, and interpreting results is done by human analysts. At L0, humans are solo practitioners, while data agents are uninvolved yet.
- **L1: Assistance.** L1 data agents operate within a stateless, prompt-response framework, offering preliminary assistance for data-related tasks by responding to user queries. Emerging from the initial wave of LLMs, they exhibit significant advancements in understanding and reasoning, enabling them to assist by generating code snippets or providing advice based on embedded knowledge upon users' queries. At this stage, L1 data agents function primarily as nascent intelligent assistants, while humans retain task dominance and responsibility for interacting with the environment, as well as integrating, verifying, and optimizing data agents' outputs.
- **L2: Partial Autonomy.** At L2, data agents gain the ability to perceive and interact with their environment, including data lakes, code interpreters, APIs, and other resources. In addition, L2 data agents can possess memory, invoke external tools, and adaptively optimize their actions based on environmental feedback, enabling partial autonomy in task-specific procedures. At this level, they evolve from simple responders to procedural executors operating within human-orchestrated pipelines, where humans remain responsible for managing the overall workflow and still retain dominance over data-related tasks.
- **L3: Conditional Autonomy.** L3 data agents are expected to autonomously orchestrate tailored data pipelines for a wide range of diverse and comprehensive data-related tasks under supervision, extending beyond human-defined workflows or specific tasks. This level marks a critical transition in which the data agent assumes a dominant role in data-related tasks, while humans act as supervisors overseeing the data agents' operation.
- **L4: High Autonomy.** At L4, data agents are expected to achieve a high level of autonomy and reliability, eliminating the need for human supervision and explicit task instructions. They can proactively identify issues worthy of investigation through continuous monitoring and exploration of data lakes, and selectively orchestrate pipelines to tackle self-discovered problems. At this level, data agents take initiative in their operations while humans fully delegate responsibility, becoming onlookers.
- **L5: Full Autonomy.** At the ultimate level of L5, beyond applying existing methods, data agents are envisioned to be capable of inventing novel solutions and pioneering new paradigms. In doing so, they advance the state of the art in data management, preparation, and analysis, making any form of human involvement unnecessary.

We organize the literature review according to the increasing levels of data agent autonomy in the following sections. More detailed explanations and examples are provided in the corresponding sections (Sections III, IV, V, and VI).

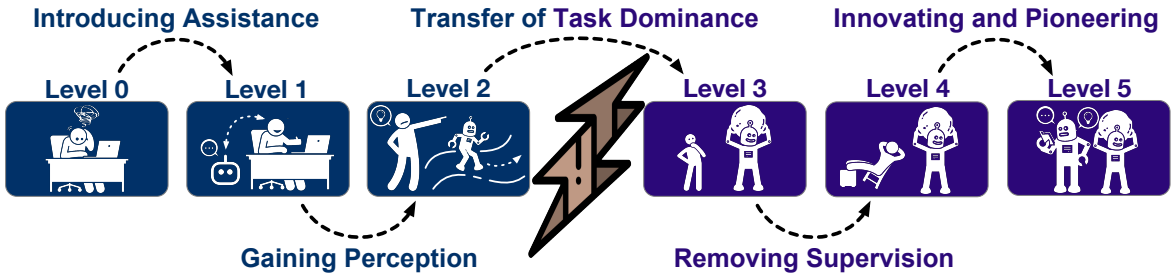

Fig. 3: Evolutionary Leaps Between Data Agent Levels.

B. The Evolutionary Leaps Between Levels

The progression through the levels reflects evolutionary leaps in data agent intelligence and autonomy, accompanied by a corresponding shift in the roles of humans and agents. Understanding these transitions is essential for grasping the technical challenges and paradigm shifts at each stage. In the following, as illustrated in Figure 3, we discuss the evolutionary leaps between consecutive levels.

• *From L0 to L1: The Advent of Assisted Intelligence.* This initial leap introduces intelligent data agents as preliminary assistants in manual workflows, shifting humans from solo practitioners to users of query-responsive assistants that enhance efficiency and reduce workload.

• *From L1 to L2: Gaining Perception.* Data agents evolve from stateless responders to procedural executors, gaining environmental perception via connection to data or code environments. They handle specific procedures autonomously and adapt actions via environmental feedback or memory, but remain within human-orchestrated pipelines, with humans retaining overall control.

• *From L2 to L3: Transfer of Task Dominance.* The transition from L2 to L3 marks a revolutionary leap where data agents evolve from procedural executors into versatile dominators capable of managing diverse and comprehensive data-related tasks. This leap involves a shift in task dominance: while L2 data agents operate within human-designed pipelines for specific subtasks, L3 data agents interpret high-level user intentions to autonomously orchestrate end-to-end pipelines across diverse tasks. Conditional autonomy emerges under human supervision, transferring primary responsibility to data agents and necessitating new accountability frameworks.

• *From L3 to L4: Removing Supervision.* Advancing to L4 presents the transition from supervised to unsupervised operation and from reactive to proactive engagement. Through this leap, data agents achieve unsupervised reliability, continuously monitoring Data + AI ecosystems to autonomously identify and tackle valuable tasks, shifting humans from cautious supervisors to fully delegating onlookers.

• *From L4 to L5: Innovating and Pioneering.* The final leap enables generative capabilities, where data agents innovate novel methodologies and pioneer new paradigms to advance the frontiers of data management, preparation and analysis, beyond applying existing techniques. As L5 data agents are envisioned to demonstrate unmatched expertise and creativity, this final leap eliminates human involvement.

The progression from L0 to L5 outlines an evolutionary trajectory for data agents, characterized by a systematic shift in responsibility and a progressive enhancement of data agent autonomy. By introducing this shared vocabulary, our taxonomy provides a structured framework for analyzing the current landscape, assessing developments, and identifying the critical challenges at the frontier of data agent research. Building on this foundation, we next review literature on data agents, organized and guided by this hierarchical taxonomy.

III. L0–L1: From Manual Labor to Preliminary Assistance

The first evolutionary leap from L0 to L1 marks the advent of assisted intelligence. This transition introduces intelligent data agents as preliminary assistance into a previously manual workflow, shifting the human's role from a solo practitioner to the user of an intelligent assistant. In this section, we first review the traditional manual processes involved in data-related tasks, followed by the initial advancement represented by the introduction of L1 data agents in data-related tasks.

A. L0: Manual Labor in Early Ages

At L0, there is no data agent involvement. In the traditional approach, all data management, preparation, and analysis tasks are performed entirely by humans without any intelligent assistance. Formally, the human $\mathcal{H}$ is responsible for the entire process, orchestrating ($\pi_{\mathcal{H}}$) pipeline P and executing ($\epsilon_{\mathcal{H}}$), while the data agent $\mathcal{A}$ is uninvolved:

$$\mathcal{H}: \pi_{\mathcal{H}}(\mathcal{T}, \mathcal{D}, \mathcal{E}) \rightarrow P; \quad \epsilon_{\mathcal{H}}(P, \mathcal{D}, \mathcal{E}) \rightarrow \mathcal{O}$$
$$\mathcal{A}: \emptyset$$

For instance, in conventional data management, operators are required to closely monitor databases, write and optimize queries, manually tune configurable knobs in DBMS (*e.g.,* shared buffers and maximum connections) based on their expertise, and diagnose and resolve issues drawing on their experience [42], [43]. Similarly, data preparation typically involves manually writing scripts to clean the data, addressing numerous minor issues during data transformation, and often integrating data from heterogeneous sources to support downstream analysis [4]. During data analysis, analysts may spend considerable time and effort writing SQL queries to extract and analyze relevant data [44], curating visualizations with tools like Vega-Lite or Matplotlib, and meticulously producing insightful interpretations [45], [46].

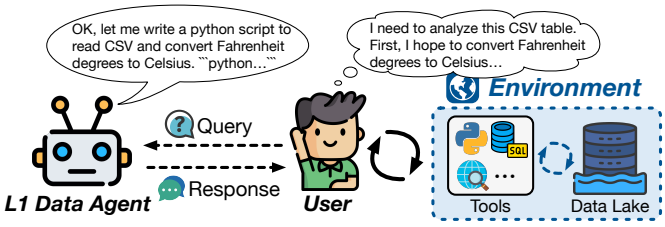

Fig. 4: L1 Data Agents (Assistance).

Such a traditional mode of work is labor-intensive and time-consuming, often requiring extensive domain knowledge and strong technical skills, which also creates high barriers to entry for non-experts. Efforts have consistently been made to alleviate these challenges, and this field has changed significantly with the advent of LLMs with powerful reasoning and comprehension abilities, starting to provide preliminary assistance in data-related tasks.

B. L1: Preliminary Assistance

Aligning with the early wave of LLM assistants, as outlined in Section II, L1 data agents operate on a prompt-response basis, assisting with data-related tasks by providing organized answers or generating code snippets to handle specific operations upon user queries. In this survey, we also consider them nascent and underdeveloped data agents, which are yet stateless and lack capabilities in environmental perception and interaction.

As illustrated in Figure 4, an L1 data agent functions as an intelligent assistant within a predominantly human-driven workflow. For example, in response to a user request, it may generate code snippets to convert Fahrenheit to Celsius for a given CSV file or clarify the parameters of an API. Formally, the human $\mathcal{H}$ still remains responsible for both pipeline orchestration $\pi_{\mathcal{H}}$ and execution $\epsilon_{\mathcal{H}}$, while data agent $\mathcal{A}$ can provide response r to response human's query q to assist in this process. The human leverages r in addressing data-related tasks:

$$\mathcal{H} : \pi_{\mathcal{H}}(\mathcal{T}, \mathcal{D}, \mathcal{E}) \rightarrow P; \quad \epsilon_{\mathcal{H}}(P, \mathcal{D}, \mathcal{E}, r) \rightarrow \mathcal{O}$$
$$\mathcal{A} : (q, \mathcal{M}) \rightarrow r$$

While this enhances efficiency on trivial and routine tasks, at this stage, L1 data agents simply generate immediate and stateless output in response to user queries; they lack the ability to perceive or interact with the environment, limiting their capacity for adaptive interaction or optimization based on feedback. Users remain responsible for integrating the generated snippets into their environment, executing and validating the code, and determining whether further refinement is necessary based on environmental feedback. The absence of environmental perception confines L1 data agents to isolated, one-off interactions, leaving end-to-end execution and verification under human control. Next, we will discuss L1 data agents designed to handle tasks in data management, preparation, and analysis, which are summarized and compared in Table II.

C. L1 Assistance in Data Management

Data management involves overseeing and optimizing database systems to ensure efficient and reliable operation. This section presents the application of L1 data agents in data management across three key tasks: (1) Configuration Tuning, which aims to identifying effective system settings such as database knobs and indexes; (2) Query Optimization, where SQL queries are improved through logical rewrites and physical plan selection to enhance performance; and (3) System Diagnosis, focusing on detecting and diagnosing system faults and anomalies.

Configuration Tuning. Configuration tuning is targeted to select and adjust system settings like database knobs and indexes to optimize overall system performance and efficiency [11].

In the initial surge of LLM advancements, some configuration tuning approaches have explored using LLMs as assistants to identify improved configurations, typically corresponding to L1 data agents. Specifically, these methods involve customizing prompts with diverse system information to extract effective tuning recommendations. Further techniques also leverage the extensive domain knowledge to enhance the tuning process.

Guiding Traditional Optimizers with LLMs. Earlier practices combine LLM guidance with traditional optimization modules to refine configurations further. For example, LLM-Tune [47] inputs workload information into an LLM, relying on its understanding and reasoning capabilities to generate a high-quality initial configuration recommendation for specific workloads. This recommendation is subsequently refined by a base optimizer such as Bayesian Optimization (BO). LLM-Tune also employs a data generation procedure to synthesize $< \texttt{workload}, \texttt{configuration} >$ pairs to fine-tune the backbone model, further improving performance. Similarly, GPTuner [49] develops an LLM-driven pipeline to collect and refine domain knowledge, employing a prompt ensemble algorithm to unify a structured view of the refined knowledge for enhanced knob configuration. It also deploys a coarse-to-fine BO framework within a knowledge-based configuration recommender to efficiently explore the search space. Building on this paradigm, LATuner [48] improves upon this by constructing structured prompts that include target tuning task details, relevant system information, and hardware environment to generate promising knobs candidates to enable a zero-shot warm-start configuration recommendation. It further employs a hybrid sampler combining LLMs and Gaussian Process (GP), along with an adaptive surrogate model based on multi-armed bandits to predict performance to select optimal configurations, effectively balancing tuning cost and efficiency.

End-to-End LLM-Based Tuning. Some studies aim to eliminate traditional optimizers to overcome challenges such as lengthy and ineffective iterations, reconfiguration overhead, and limited transferability to unseen workloads. These systems instead rely directly on domain knowledge and LLM reasoning to recommend suitable configurations. For example, λ-Tune [50] and E2ETune [51] generate multiple configuration candidates based on workload features and use performance prediction guided by cost models to select the best option.

TABLE II: Comparison of L1 Data Agents. ICL: In-Context Learning; RAG: Retrieval-Augmented Generation; SFT: Supervised Fine-Tuning; RL: Reinforcement Learning. Data complexity dimensions include Multi-source (Multis.), Heterogeneous (Hete.), and Multimodal (Multim.) data support.

	Task	Data Agent	Years	ICL	RAG	SFT	RL	Data Complexity		
								Multis.	Hete.	Multim.
Data Management	Configuration Tuning	LLMTune [47]	2024	Zero-shot	-	✓	-	-	✓	-
		LATuner [48]	2024	Few-shot	-	-	-	-	-	-
		GPTuner [49]	2024	Zero-shot	-	-	-	✓	✓	-
		λ-Tune [50]	2025	Zero-shot	✓	-	-	-	-	-
		E2ETune [51]	2025	Few-shot	-	✓	-	-	-	-
	Query Optimization	DB-GPT [52]	2024	Few-shot	✓	✓	-	✓	✓	-
		LLM-R^2 [53]	2024	Few-shot	✓	-	-	-	✓	-
		GenRewriter [54]	2024	Zero-shot	-	-	-	-	-	-
		LITHE [55]	2025	Zero-shot	-	-	-	-	-	-
		LLM-QO [56]	2025	Few-shot	-	✓	-	-	-	-
		LLMOpt [57]	2025	Zero-shot	-	-	-	-	✓	-
		E3-Rewite [58]	2025	Few-shot	✓	✓	GRPO	-	✓	-
	System Diagnosis	DBG-PT [59]	2024	Zero-shot	-	-	-	-	✓	-
		Andromeda [60]	2025	Few-shot	✓	-	-	✓	✓	-
Data Preparation	Data Cleaning	FM [61]	2022	Zero-shot	-	-	-	-	-	-
		RetClean [62]	2024	Zero-shot	✓	✓	-	✓	✓	-
		LakeFill [63]	2025	Zero-shot	✓	✓	-	✓	✓	-
		UniDM [64]	2024	Zero-shot	✓	-	-	✓	✓	-
		LLMClean [65]	2024	Few-shot	-	-	-	✓	✓	-
	Data Integration	Table-GPT [66]	2024	CoT	-	✓	-	-	-	-
		BATCHER [67]	2024	Batch Prompting	-	-	-	-	-	-
		Jellyfish [68]	2024	Few-shot, CoT	-	✓	-	-	-	-
	Data Discovery	ArcheType [69]	2024	Zero-shot	-	✓	-	✓	✓	-
		RACOON [70]	2024	Few-shot, CoT	✓	-	-	✓	✓	-
		Cocoon [71]	2024	Few-shot, CoT	-	-	-	✓	✓	-
		AutoDDG [72]	2025	Zero-shot	-	-	-	✓	✓	-
		Pneuma [73]	2025	Zero-shot	✓	-	-	✓	✓	-
		LLMCTA [74]	2025	Few-shot, CoT-SC	-	✓	-	✓	✓	-
		Columbo [75]	2025	Few-shot, CoT	-	-	-	✓	✓	-
Data Analysis	TableQA	Dater [76]	2023	Few-shot, CoT	-	-	-	-	-	-
		BINDER [77]	2023	Few-shot	-	-	-	-	✓	✓
		Sui et al. [78]	2024	Few-shot	-	-	-	-	-	-
		TableLlama [79]	2024	Zero-shot	-	✓	-	-	-	-
	NL2SQL	DIN-SQL [80]	2023	Few-shot	-	-	-	-	-	-
		DAIL-SQL [81]	2023	Few-shot	✓	-	-	✓	✓	-
		ACT-SQL [82]	2023	CoT	-	-	-	-	-	-
		SuperSQL [83]	2024	Few-shot	✓	-	-	✓	✓	-
	NL2VIS	Chat2VIS [84]	2023	Few-shot	-	-	-	-	-	-
		NL4DV-LLM [85]	2024	Iterative Prompt	-	-	-	-	-	-
		Prompt4Vis [86]	2025	Few-shot	-	-	-	✓	✓	-
		Step-Text2Vis [87]	2025	Few-shot	-	✓	Step-DPO	-	-	-
	Unstructured Data Analysis	LongRAG [88]	2024	Zero-shot	✓	✓	-	-	-	-
		RAPTOR [89]	2024	Zero-shot	✓	-	-	-	-	-
		PDFTriage [90]	2024	Zero-shot	✓	-	-	-	✓	-
		DSE [91]	2024	Zero-shot	✓	✓	-	✓	✓	✓
		VisDoM [92]	2025	Zero-shot	✓	-	-	✓	✓	✓
		Docopilot [93]	2025	Zero-shot	-	✓	-	✓	✓	✓
	Report Generation	Datatales [94]	2023	Few-shot	-	-	-	-	-	-
		Choe et al. [95]	2024	Zreo-shot, CoT	-	-	-	-	-	-
		ReportGPT [96]	2025	Few-shot	-	-	-	-	-	-
		InterChat [97]	2025	Few-shot, CoT	-	-	-	-	-	-
		VizTA [98]	2025	Few-shot	-	-	-	-	-	-
		ChartLens [99]	2025	Few-shot	-	-	-	-	✓	✓

Limitations. These systems help advance configuration tuning, reducing reliance on human operation and costly traditional optimization iteration. However, at this level, they lack the capability to directly perceive the live database environment or interactively adjust configurations in response to dynamic system states and feedback. This limitation significantly restricts their adaptability and effectiveness in real-world tuning scenarios where continuous monitoring and iterative refinement are critical.

Query Optimization. Query optimization seeks to improve the efficiency of SQL execution through logical and physical optimization techniques. Traditional logical optimization relies on predefined rewrite rules or learning-based methods to determine the order of applying these rules, while physical optimization uses heuristic algorithms based on statistical data or learning-based approaches that leverage features of query plans. However, these methods often neglect external SQL optimization knowledge, which limits their effectiveness and generalizability across diverse SQL query patterns.

To address these limitations, existing studies have explored prompting LLMs to rewrite input SQL queries, determine optimal rule application sequences for logical optimization, or select the best query execution plans for physical optimization by leveraging the extensive SQL optimization knowledge encoded within the models. Early efforts essentially follow a prompt-response basis without environmental perception and interaction, therefore fall into L1 data agents.

• *Logical Query Optimization.* DB-GPT [52] proposes an automatic prompt generation framework to produce multiple structured instruction candidates with adaptive demonstration examples based on computed semantic similarity to enhance query rewriting, ultimately selecting the optimal ones according to a customized scoring function. GenRewrite [54] introduces natural language rewrite rules as hints and leverages these to transfer knowledge between tasks, incorporating a counterexample-guided CoT mechanism to self-correct syntactic and semantic errors in rewritten queries. Besides, LLM-R^2 [53] uses LLMs to recommend rewrite rules and include a demonstration manager employing a contrastive model trained via curriculum learning to select effective query demonstrations for few-shot learning. LITHE [55] combines an ensemble suite of prompts, including basic prompts, database-sensitive prompts with selectivity-based rewriting rules, and token probability-guided rewrite paths to generate optimized query equivalents. Furthermore, E3-Rewrite [58] incorporates execution hints into prompts using parsed query plans and fine-tunes an LLM with a two-stage GRPO training process that prioritizes rewards of executability and semantic equivalence, ensuring stable multi-objective learning. During inference, it leverages execution plans to retrieve hybrid demonstrations incorporated into prompts to enhance the rewriting process.

• *Physical Query Optimization.* LLMOpt [57] tackles this by generating and selecting query plan candidates. Instead of heuristic search, it employs a fine-tuned LLM to produce multiple candidate plans, which are then evaluated by a fine-tuned LLM-based list-wise cost model to select the optimal query plan, replacing traditional Tree-CNN-based models. LLM-QO [56] constructs the QInstruct dataset, which comprises structured prompts containing input queries, data statistics, instructions, and paired responses that sequentialize the query plan using a post-order traversal of the tree-structured execution plan, thereby aligning with bottom-up query execution semantics. The dataset includes both instruction data and preference data, and LLM-QO takes a step forward by applying direct preference optimization (DPO) after supervised fine-tuning to train the policy model, further improving its ability to generate more efficient execution plans.

Limitations. However, these systems remain limited due to their inability to dynamically adapt or refine optimization decisions based on perception of actual system behavior, which constrains their effectiveness and robustness in complex, changing workloads and diverse operational contexts.

System Diagnosis. System diagnosis focuses on analyzing root causes and identifying recovery solutions for anomalies (*e.g.,* spikes in system resource usage) during runtime, particularly in systems like databases. L1 data agents, leveraging advanced textual understanding and reasoning capabilities, demonstrate promising potential to effectively pinpoint causes and generate diagnoses with recovery recommendations in various formats for database systems.

Comparative Diagnosis with LLMs. For instance, DBG-PT [59] addresses unexpected slowdowns in query execution by enabling LLMs to compare regressed query plans with stored efficient plan instances alongside relevant system information, accordingly responding with generated configuration recommendations to resolve plan regressions.

Augmenting Diagnosis with RAG. To address the tendency of naive LLM prompting to yield overly generic responses in specialized diagnosis tasks, as configuration tuning here, Andromeda [60] further incorporates a RAG strategy that supplies matched domain-specific contexts from multiple sources, including historical queries, database troubleshooting manuals, and DBMS telemetries, enhancing its performance in generating diagnostic suggestions for DBMS configuration debugging.

Limitations. Despite these advances, relying solely on static input data and logs limits L1 data agents in system diagnosis, as they lack the ability to monitor real-time database states or interact autonomously with the system environment. This restricts their capacity to provide timely and adaptive anomaly detection and recovery in dynamic operational environments.

D. L1 Assistance in Data Preparation

Data preparation encompasses the processes of cleaning, integrating, and discovering data to ensure the quality and completeness of information for downstream tasks. This section presents the application of L1 data agents in data preparation: (1) Data Cleaning, which aims to detect and correct errors, inconsistencies, and missing values in raw datasets; (2) Data Integration, where heterogeneous sources are reconciled and merged through schema alignment and entity resolution to form unified data views; and (3) Data Discovery, focusing on

extracting metadata, identifying relevant datasets, and uncovering latent patterns to facilitate efficient data exploration.

Data Cleaning. Data cleaning is a foundational step in data preparation, focused on identifying and rectifying errors, inconsistencies, and inaccuracies within a dataset to improve its quality and reliability. The process addresses a wide range of data quality issues, including missing values, constraint violations, and inconsistencies that can significantly degrade the performance of downstream tasks.

With the emergence of LLMs, introducing L1 data agents has demonstrated significant advantages for data cleaning. They primarily utilize LLMs in two distinct modes in data cleaning:

Generating Direct Answers via Prompting. The most straightforward method involves performing data cleaning tasks through direct prompting [100], where L1 data agents directly infer and generate an answer (*e.g.,* a missing value). For example, a data record can be serialized into text and fed to them to fill in a missing value with minimal examples [100]. To improve the quality of these direct answers, more sophisticated mechanisms are employed. RetClean [62] and LakeFill [63] utilize an RAG approach, retrieving relevant tuples from a user-provided data lake and then using this external context to ground their response, which is crucial for private or domain-specific data. Similarly, UniDM [64] employs a multi-stage prompting pipeline to systematically guide the LLMs, first identifying relevant metadata and instances, then parsing that context into natural language, before finally constructing a targeted prompt to generate the final answer.

Generating Intermediate Artifacts. When directly inferring a value is too complex or inefficient, some L1 data agents are designed to generate intermediate artifacts such as code or rules for the user to integrate, execute, and optimize. LLMClean [65] leverages an LLM to automate the knowledge engineering process itself. It prompts the LLM to analyze a dataset and generate a formal context model along with a set of data quality rules, such as Ontological Functional Dependencies (OFDs). These human-readable rules are then passed to the user or a traditional cleaning system for implementation, automating what is typically a manual, expert-driven task.

Limitations. Despite their powerful cleaning capabilities, L1 data agents operate in a passive, open-loop manner during data cleaning. They cannot perceive downstream validations or receive execution feedback to adjust their cleaning strategy. For example, if UniDM [64] imputes a missing value that violates a range constraint upon insertion, it remains oblivious to the failure and can not revise its imputation methods. Consequently, the detection of these errors, diagnosis, and resolution depend on human intervention, which restricts L1 data agents' ability to autonomously improve cleaning results.

Data Integration. Data integration refers to the process of combining data from heterogeneous sources into a unified, coherent dataset [101], which serves as a foundation for reliable downstream analysis and knowledge reuse. The primary challenges in data integration stem from semantic heterogeneity, incomplete metadata, and entity ambiguity [36]. This process typically involves two key tasks: schema matching and entity resolution. Schema matching aligns data structures across different sources, while entity resolution builds on this alignment to resolve ambiguities at the entity level, collectively achieve entity-level unification.

Traditionally, addressing these issues relies on expert-driven attribute alignment and entity disambiguation. Such approaches are labor-intensive, costly, and difficult to scale [102], [103]. The initial introduction of LLM-based methods to assist with data integration tasks presents a substantial advancement to mitigate these gaps, marked as L1 data agents.

• *Schema Matching.* The adoption of L1 data agents for schema matching has progressed from in-context learning to task-specific fine-tuning and, more recently, integration into sophisticated multi-stage pipelines, all operating on a prompt-response basis essentially.

In-Context Learning and Fine-tuning. Some studies deploy off-the-shelf LLMs directly through zero-shot or few-shot learning. For instance, Narayan et al. [61] use serialized attribute values and optional task demonstrations as input and demonstrated that even under few-shot settings only using column names can outperform traditional deep learning approaches [104]. As the demand for adaptability increases, task-specific fine-tuning has been employed to improve schema matching performance. For example, Table-GPT [66] performs multi-task fine-tuning across both complex reasoning tasks and simpler table operations, while Jellyfish [68] focuses on meticulous prompt design and output reasoning.

Hybrid and Multi-Stage Systems. To address even more complex matching scenarios, recent approaches embed LLMs within multi-stage systems: In these, pretrained language model embeddings first filter candidate attributes, and then LLMs determine final correspondences using a multiple-choice formulation [105], [106]. Magneto [107] exemplifies this two-stage approach: a smaller language model (SLM) performs initial ranking, followed by LLM re-ranking. This method also leverages synthesized data to fine-tune the SLM, reducing the need for manual labeling while addressing LLM context window limitations.

• *Entity Resolution.* For entity resolution, Narayan et al. [61] also use off-the-shelf LLMs with carefully designed prompts to produce entity matching decisions, presenting early exploration in this area. Peeters et al. [108] conduct a thorough evaluation of LLM-based entity resolution performance across various settings. Steiner et al. [109] investigate fine-tuning LLMs by synthesizing explanations to training data and optimizing example selection, demonstrating that fine-tuned models achieved better in-domain generalization, while zero-shot approaches outperformed on cross-domain data. Additionally, BatchER [67] focuses on improving cost efficiency in in-context learning for entity resolution by developing a batch prompting method that incorporates question batching and a covering-based demonstration selection strategy to reduce token consumption.

Limitations. However, L1 data agents face a critical constraint in their lack of dynamic interaction with real-world environments like search engines and databases. As static

prompt-response systems, they cannot proactively invoke external data sources to retrieve real-time information or access private databases, relying instead on pre-provided inputs. This isolation from live systems undermines their ability to resolve ambiguities with up-to-date data or integrate proprietary sources, hindering performance in realistic, data-scattered enterprise scenarios.

Data Discovery. Data discovery refers to the process of identifying, interpreting, and contextualizing datasets within large and heterogeneous repositories such as data lakes [36]. At the L1, data agents, typically instantiated as LLM-driven systems, function as query responders that help users navigate complex data environments. They do not alter the underlying data; instead, they generate descriptive or structural artifacts such as dataset summaries, metadata annotations, semantic profiles, or structured views. These outputs are designed for human review, thereby facilitating subsequent decision-making in data preparation.

• *Descriptive Profiling* A central role of these data agents is to act as descriptive interpreters that generate metadata or dataset profiles to facilitate accessibility. Within this role, existing approaches can be broadly divided into two categories.

Prompt-Based Profiling. The first category is prompt-based profiling, where dataset or table descriptions are generated directly through carefully designed prompts. Representative work such as AutoDDG [72] adopts general-purpose prompting strategies that integrate both statistical and semantic profiling information, enabling large language models to produce concise and informative dataset-level summaries while reducing the manual effort required for documentation. Subsequent works extend this paradigm through domain-specific prompting, for example, in ecological metadata harvesting [110] or metadata validation for scientific papers [111], where tailored instructions improve consistency and compliance. More advanced prompting strategies have been explored for semantic profiling, with Cocoon [71] generating interpretable semantic views of tables, and customizable profiling for code data [112] leveraging programming-language concepts to capture domain-sensitive properties.

Hybrid Retrieval Integration. To enhance retrieval effectiveness across diverse query types, the second method adopts a hybrid approach that integrates diverse retrieval techniques, which grounds LLM outputs in repository-level signals to improve robustness and relevance. Pneuma [73] exemplifies this paradigm by combining LLM-based tabular representation with retrieval mechanisms in an end-to-end system, enabling profiling that is not only descriptive but also retrieval-aware.

• *Semantic Annotation for Schema.* Another prominent role of L1 data agents is to serve as annotators who assign semantic types, roles, or labels to schema elements.

Prompt-Based Annotation. One approach encodes task-specific instructions within carefully crafted prompt templates. In this setting, the data agent directly infers semantics from column names or sampled values. This line of work covers column type annotation [69], [113], column property assignment [74], and column-name expansion [75]. For instance, early studies such as ArcheType [69] use structured prompting and contextual sampling to guide LLMs in inferring column types, establishing a foundation for zero-shot semantic annotation. Building on this idea, LLMCTA [74] extends prompt-based reasoning toward column property assignment, introducing knowledge-generation prompting to refine property definitions and assess annotation reliability across models. Moving further, Columbo [75] exemplifies a higher-level application of prompt design by expanding abbreviated or underspecified column names into semantically meaningful expressions, thereby enhancing schema interpretability.

Semantic Grounding. However, although these methods provide fine-grained interpretability, they often suffer from inconsistency and contextual robustness. To overcome these quality limitations, RACOON [70] leverages knowledge graphs to provide richer semantic grounding, while AutoMetaSQL [114] mitigates context deficiency by extracting schema metadata directly from user queries.

Limitations. Although these efforts improve accessibility through profiling and annotation, their capabilities remain limited by task-specific weaknesses. Profiling approaches, such as AutoDDG [72] and Cocoon [71], rely solely on prompt-based summarization without perceiving the broader data environment, often producing incomplete or inconsistent metadata that quickly becomes outdated. Retrieval-augmented systems like Pneuma [73] partially mitigate this issue by grounding profiling in repository context, yet they still lack adaptive mechanisms to update or validate outputs as data evolves. In schema annotation, works such as Columbo [75] and LLMCTA [74] demonstrate the potential of LLM-based inference for enriching column semantics, but these methods remain highly sensitive to naming conventions and lack feedback from real data usage or execution results. Consequently, at L1, data agents' discovery outputs tend to be static, error-prone, and detached from dynamic repository states. These limitations highlight that L1 data agents function as passive assistants rather than adaptive collaborators, motivating the development of more advanced systems that integrate environmental awareness and feedback-driven refinement.

E. L1 Assistance in Data Analysis

Data analysis involves processing and interpreting data to derive insights and support informed decision-making across various data formats. This section presents the application of L1 data agents in data analysis across three areas: (1) Structured Data Analysis, which includes TableQA, NL2SQL, and NL2VIS to enable question anwering, querying, and visualization through natural lanauge interface; (2) Unstructured Data Analysis, focus on extracting insights from data lacking explicit structure, such as documents; and (3) Report Generation, where data agents produce integrated narrative and visual summaries.

Structured Data Analysis. Structured data analysis centers on extracting insights from data organized according to a defined schema or format, such as tables or relational databases, facilitating efficient querying, manipulation, and interpretation. In the following, we focus on three key tasks in structured data analysis: TableQA, NL2SQL, and NL2VIS.

a) TableQA: Table Question Answering (TableQA) enables systems to interpret natural language queries and extract answers from structured or semi-structured tables without requiring formal query languages [115]–[117]. It demands both deep semantic understanding of user queries and precise parsing of the table's two-dimensional layout, including headers, rows, cells, and their relationships, to accurately locate and infer responses.

With the rise of LLMs, the methodologies for addressing TableQA tasks have continuously evolved. We will first introduce L1 data agents in the TableQA, where they handle diverse tabular tasks typically by directly generating final answers or executable code through a single-pass, static process. Existing works employ various strategies to enhance their TableQA performance.

Prompt Engineering. Prompt engineering is widely used to enhance the quality of input, thereby improving reasoning in TableQA. A primary challenge in this domain is the limited context length of LLMs, which can make processing large, information-dense tables intractable. To mitigate this, some researchers focus on decomposing the problem itself before the final reasoning step. For instance, the Dater [76] framework uses an LLM in a preliminary stage to break down large tables into smaller, relevant sub-tables and to simplify complex questions. Similarly, the self-augmented prompting [78] uses an initial LLM call to extract structural insights from a table, which are then fed into a second, enriched prompt to derive the final answer.

Fine-tuning. Another approach is to develop end-to-end table-processing models through instruction fine-tuning for TableQA tasks, thereby enhancing the intrinsic capabilities. TableLlama [79] is a prime example of this approach, having been fine-tuned on TableInstruct, a large-scale and diverse dataset of table-based instructions. The goal of this method is for the model to internalize the ability to handle a wide variety of table tasks directly from instructions, rather than relying on complex, multi-step prompting at inference time.

Neuro-Symbolic Program Synthesis. A distinct methodology involves neuro-symbolic systems, which integrate the reasoning capabilities of LLMs with the precision of formal code. Binder [77] exemplifies this by parsing natural language questions into hybrid programs that contain both standard code (*e.g.,* SQL) and embedded LLM API calls. These API calls are designed to resolve parts of the query that require common sense or are beyond the scope of standard code. The entire process operates under a static "parse-then-execute" workflow, where the program is constructed in one step and then executed.

Limitations. Despite notable progress, these systems mostly rely on coarse-grained processing by serializing whole or partial tables into text prompts, lacking fine-grained control and feedback over table structure and content. For complex or dense tables, this approach risks losing critical structural information or introducing noise. Additionally, the "prompt-response" format restricts iterative interaction with tables and external tools to facilitate refinement, limiting effectiveness on challenging TableQA tasks.

b) NL2SQL: Natural Language to SQL (NL2SQL) is the task of converting a natural language query (NL) into a SQL query (SQL) that can be executed on a relational database. Specifically, given an NL and a DB, the goal of NL2SQL is to generate a SQL query that accurately captures the user's intent and returns the correct results when executed on the database [40], [41], [83], [118].

As a bridge between non-technical users and relational databases, NL2SQL has long attracted significant attention from both academia and industry. More recently, the advent of LLMs has substantially expanded the scope of this field, driving a paradigm shift from traditional syntax-based mappings toward intelligent systems capable of semantic understanding, contextual reasoning, and interactive capabilities. In the early stage, most LLM-driven NL2SQL systems concentrate on the L1 autonomy, primarily operating in a single-turn prompting paradigm [80]–[82]. Representative works have been proposed, each targeting a different aspect of the NL2SQL process:

Prompt Engineering. DAIL-SQL [81] systematically investigates prompt engineering for LLM-based NL2SQL methods, covering five forms of question representation, two prompt components, four example selection strategies, and three example organization schemes, with comprehensive evaluations conducted on four different LLMs. In addition, DAIL-SQL further explores the effectiveness of supervised fine-tuning, extending the understanding of the applicability and potential of open-source LLMs in NL2SQL scenarios.

Task Decomposition. DIN-SQL [80] proposes a task decomposition-based prompting framework to address the limitations of single-turn zero/few-shot prompting in NL2SQL. It splits SQL generation into sub-tasks (*e.g.,* schema linking, query sketching, clause completion), handled step by step with crafted prompts, and progressively combines the results into a complete SQL query, thereby improving performance on complex questions.

Self-Guided SQL Reasoning. ACT-SQL [82] focuses on enhancing the reasoning capability of LLMs in NL2SQL through automatic chain-of-thought prompting. Unlike approaches that rely on manually constructed exemplars, ACT-SQL automatically generates CoT sequences from the database schema, natural language questions, and corresponding SQL queries, thereby effectively improving the accuracy of complex SQL generation.

Limitations. At the L1 stage, methods such as DAIL-SQL [81], DIN-SQL [80], and ACT-SQL [82] improve the performance of NL2SQL by exploring prompt representation, task decomposition, and reasoning enhancement. However, these approaches function primarily as passive assistants, generating SQL in single-turn interactions in response to natural language queries. The generated SQL must be manually validated and executed by users, with any errors requiring further diagnosis and correction through additional user input.

c) NL2VIS: Natural Language to Visualization (NL2VIS) systems aim to lower the barrier to data exploration by enabling users to generate visualizations directly from natural language queries [6], [119]–[121]. Leveraging their markedly enhanced capabilities of understanding and reasoning, LLMs

are prompt to provide on-demand assistance by translating users' intents in natural language into visualization specifications [84]. These studies represent a quintessential example of L1 data agents for NL2VIS, marking a significant paradigm shift in the field.

Direct Prompting. Amid the rapid advancement of LLMs, Chat2VIS [84] is among the first to demonstrate that off-the-shelf LLMs like ChatGPT could generate visualizations directly from natural language text through effective prompt engineering. Building on this, Prompt4Vis [86] further enhances performance by employing automated prompt optimization, introducing a multi-objective "example mining" module to select the most effective examples for in-context learning. To address the inherent "black-box" nature of LLMs, NL4DV-LLM [85] focuses on explainability by prompting the model to generate a structured analytic specification, which details the inferred data attributes, tasks, and design rationale, making the process more transparent and easier to debug.

Step-Wise Reasoning. Furthermore, a persistent challenge in NL2VIS is ambiguity. To address this, nvBench 2.0 [87] introduces a benchmark with controlled ambiguity specifications using an ambiguity-injection pipeline and step-wise reasoning. It also presents Step-Text2Vis, a model trained to trace how each ambiguous query can result in multiple valid visualizations, setting a new state-of-the-art for resolving ambiguity in NL2VIS tasks.

Limitations. However, at this level, data agents generate visualization specifications or Vega-Lite snippets in response to natural language queries in prompts, yet lack the capacity to interact with data sources, incorporate feedback on visual outputs, or adapt to dynamic data inconsistencies beyond initial responses.

Unstructured Data Analysis. Unstructured data analysis focuses on extracting insights from data without a predefined schema. The emergence of data agents, driven by LLMs, has fundamentally revolutionized this field [122]. Their profound semantic understanding, in-context reasoning, and coherent text synthesis abilities allow them to process documents in a manner previously unattainable, endowing these unstructured data analysis tasks with a foundational degree of L1 autonomy.

• *Textual Documents.* Textual documents represent a primary form of such data, and a central challenge lies in comprehending and synthesizing information from vast document corpora. A primary challenge in real-world applications is the sheer length and complexity of documents. Research has progressed along complementary paths to address this.

Intrinsic Context Expansion. One direction focuses on enhancing its intrinsic capacity to process vast, contiguous contexts [123]. As models with ever-larger context windows become available, techniques like R&R [124] have been developed to optimize how information is retrieved and utilized within these ultra-long inputs, significantly boosting accuracy.

Structure-Aware Information Filtering. Concurrently, another effective strategy involves intelligently selecting and providing LLMs with only the most relevant information from the document. This approach has evolved beyond simple keyword retrieval into sophisticated methods for understanding the document's global structure. For instance, RAPTOR [89] recursively clusters and summarizes text to build a hierarchical understanding, while LongRAG [88] employs a dual-strategy that provides both high-level summaries and fine-grained details. Beyond just textual length, the challenge extends to complex layouts. Works like PDFTriage [90] address this by restructuring documents into semantically coherent blocks, making them more digestible for LLM analysis.

• *Multimodal Documents.* The scope of document analysis is rapidly broadening from purely textual content to complex multimodal documents that integrate text, tables, and rich visual elements. This complexity has spurred the creation of dedicated benchmarks like MMVQA [125] and VisDoMBench [92] to rigorously evaluate data agents' ability to comprehend and retrieve information from multipage, visually-rich sources.

Multimodal RAG Adaptations. To tackle this challenge, many approaches adapt the RAG framework. Some systems deploy hierarchical indexing and multi-granularity retrieval to leverage both textual and visual information across lengthy documents (*e.g.,* MMRAG-DocQA [126]). A more radical paradigm, Document Screenshot Embedding [91], aims to unify this process by bypassing traditional, error-prone content extraction pipelines altogether. DSE treats the document screenshot as a single input format, using a vision-language model to directly encode its visual appearance, thereby preserving the original layout and context.

Multimodal Reasoning. Beyond retrieval, reasoning architectures are also advancing. VisDoM [92], for example, employs parallel textual and visual RAG pipelines, then fuses their outputs by analyzing the consistency between their respective reasoning chains. Similarly, MDocAgent [127] utilizes a multi-agent framework where specialized agents collaborate to synthesize cross-modal insights. While RAG-based methods are prevalent, end-to-end models like Docopilot [93] have also been developed to perform multimodal understanding without an explicit retrieval component.

Limitations. Despite these sophisticated advancements in processing long, structurally complex, and multimodal data, L1 data agents are constrained by their fundamental operational model: a static, single-pass workflow. Even with advanced multimodal retrieval and complex reasoning pipelines like those in VisDoM [92], if the initial context misses a critical piece of evidence, the agent has no mechanism to recognize this failure and autonomously re-interrogate the source documents. Furthermore, should data agents misinterpret a complex chart or generate a factually incorrect statement, they lack the metacognitive ability to self-critique by cross-validating its own output against the original data. This inability to adapt based on intermediate results means they cannot perform true iterative analysis, leaving the ultimate responsibility of verification and refinement with the user.

Report Generation. Report generation has become a central approach for communicating analytical insights, combining visualizations and narrative text to make data more interpretable and actionable. The overarching goal is to help users bridge the gap between raw data and decision-making by

addressing challenges in query interpretation, data transformation, visual mapping, interaction, and presentation. Within this context, L1 data agents primarily act as interpreters and assistants within the early exploration of LLMs. They generate charts, summaries, or explanations in response to user queries, lowering comprehension barriers but offering no perception of or adaptation to the visualization environment. Several representative works illustrate this pattern.

For translating structured data into human-readable narratives, ReportGPT [96] introduces a verifiable table-to-text generation pipeline that employs a DSL to make table operations explicit and uses user feedback for factual verification. Datatales [94] investigates the use of LLMs for authoring data-driven articles on program-based visualization. Similarly, InterChat [97], VizTA [98], and Choe et al. [95] have explored agentic frameworks in visualization contexts to support multimodal interaction, where direct manipulation and natural language are combined to disambiguate queries, and textual narratives are enriched with visually grounded evidence on charts. Extending this direction, ChartLens [99] introduces a multi-agent framework that grounds narrative text with fine-grained visual attributes by providing bounding box citations as supporting evidence within chart images.

Limitations. Overall, L1 data agents lower comprehension barriers and provide on-demand explanations and verification mechanisms, enabling users to derive insights more easily. Yet their autonomy remains limited: these systems operate in a "one-shot Q&A" paradigm, lacking real-time perception and iterative refinement. As a result, the human remains the primary driver of analysis and validation, while the agent functions as an assistant rather than an autonomous collaborator.

F. Progress and Limitations of L1 Data Agents

L1 data agents have introduced prompt-based assistance for discrete data-related tasks. By interpreting and responding to users' queries on demand, this initial form of intelligent assistance also significantly improves developer efficiency by offloading trivial and routine operations such as unit conversions or standard preprocessing steps.

Despite these gains, L1 data agents operate in a stateless, prompt–response paradigm and lack the capabilities to perceive and interact with their environment, which limits their ability to perform adaptive optimization. They can not perceive or interact with external environments or data systems autonomously, necessitating manual execution, integration, verification, and optimization of generated outputs by human analysts. Consequently, L1 data agents remain reliant on user oversight to ensure correctness and to manage the entire procedure, limiting their autonomy to atomic and static subtasks.

IV. L2: Perceive the Environment

A. Partial Automation: From Auxiliary to Executor

As outlined in Section II, L2 data agents gain the ability to perceive and interact with their environment, including data lakes, code interpreters, and other resources, transitioning from simple responders to procedural executors within human-designed workflows. As illustrated in Figure 5, unlike L1

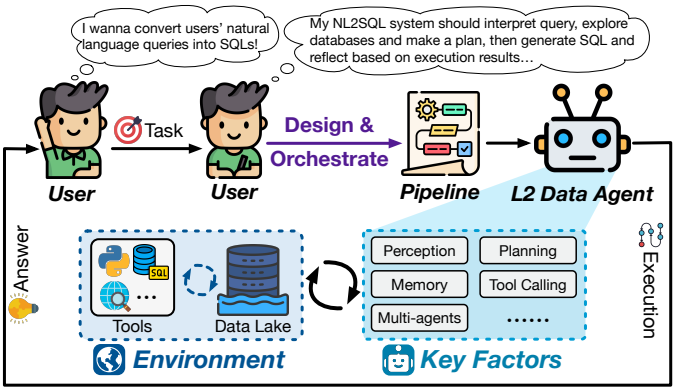

Fig. 5: L2 Data Agents (Partial Autonomy).

agents where humans are responsible to interact directly with the environment, L2 data agents connect to real-world data systems such as databases, files, and APIs, enabling them to explore and interact with these sources, execute operations, invoke external tools (*e.g.,* search engines, code sandboxes, SQL equivalence checker, visualization libraries etc.), and adaptively optimize their outputs based on environmental feedback, such as query results or execution outcomes. This capability grants L2 data agents partial autonomy in data-related tasks, allowing them to perform task-specific procedures independently. Formally, at L2, data agent $\mathcal{A}$ gains environmental perception and interaction capabilities $(\mathcal{D}, \mathcal{E})$, capable of handling specific data-related tasks by executing $(\epsilon_{\mathcal{A}})$ pipeline P orchestrated by human $\mathcal{H}$:

$$\mathcal{H} : \pi_{\mathcal{H}}(\mathcal{T}, \mathcal{D}, \mathcal{E}) \rightarrow P$$
$$\mathcal{A} : \epsilon_{\mathcal{A}}(P, \mathcal{D}, \mathcal{E}, \mathcal{M}) \rightarrow \mathcal{O}$$

Such functionality represents a significant advancement beyond the stateless simple prompt-response paradigm at L1, alleviating users from the repetitive cycle of integrating data agents' outputs and interacting with the environment manually. In practice, many recent developments fall within the L2 category. For instance, ReFoRCE [163] actively explores database columns and translates and optimizes natural language queries into SQL through interaction with the database. However, the autonomy of L2 data agents remains limited: they operate within human-orchestrated pipelines and are not yet capable of independently devising workflows tailored to diverse, complex, high-level tasks.

Next, we review L2 data agents for various data-related tasks with Table III outlining a summary and comparison.

B. L2 Data Agents in Data Management

Configuration Tuning. L2 data agents not only generate configuration recommendations but also monitor system states and dynamically refine tuning strategies based on real-time performance feedback. In configuration tuning, they continuously adjust parameters to address system feedback, such as database knobs, and some methods further incorporate structured knowledge sources to improve tuning effectiveness.

Iterative Tuning via Feedback. A typical approach is to deploy direct feedback-driven interaction between data agents

TABLE III: Comparison of L2 Data Agents. RAG: Retrieval-Augmented Generation; Percept: Environmental Perception; Plan: Planning; Mem: Memory; Tool: Tool invocation; Reflect: Self-reflection mechanism; MAS: Multi-agent system; SFT: Supervised Fine-Tuning; RL: Reinforcement Learning. Data complexity dimensions include Multi-source (Multis.), Heterogeneous (Hete.), and Multimodal (Multim.) data support.

	Task	Data Agent	Years	RAG	Percept	Plan	Mem	Tool	Reflect	MAS	SFT	RL	Data Complexity		
													Multis.	Hete.	Multim.
Data Management	Configuration Tuning	Li et al. [128]	2024	-	✓	-	✓	-	-	-	-	-	-	-	-
		LLMIdxAdvis [129]	2025	✓	✓	-	✓	-	✓	✓	-	-	-	✓	-
		RABBIT [130]	2025	✓	✓	-	-	-	✓	✓	-	-	✓	✓	-
		MCTuner [131]	2025	✓	✓	-	✓	✓	-	✓	-	-	✓	✓	-
	Query Optimization	SERAG [132]	2025	✓	✓	-	✓	-	-	-	-	-	-	-	-
		QUITE [133]	2025	✓	✓	✓	✓	✓	✓	✓	-	-	✓	-	-
		R-Bot [134]	2025	✓	✓	-	-	✓	✓	-	-	-	✓	✓	-
	System Diagnosis	Panda [135]	2024	✓	✓	-	✓	-	-	-	-	-	✓	✓	✓
		D-Bot [136]	2024	✓	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	-
		DBAIOps [137]	2025	✓	✓	-	-	✓	-	-	-	-	✓	✓	-
Data Preparation	Data Cleaning	SketchFill [138]	2024	-	✓	-	-	✓	✓	✓	-	-	-	-	-
		IterClean [139]	2024	-	✓	-	-	✓	✓	✓	-	-	-	-	-
		AutoPrep [29]	2025	-	✓	✓	-	✓	-	✓	-	-	-	✓	-
		AutoDCWorkflow [140]	2025	-	✓	✓	-	✓	✓	-	-	-	✓	✓	-
		CleanAgent [141]	2025	-	✓	✓	✓	✓	-	✓	-	-	-	✓	-
		Bendinelli et al. [142]	2025	-	✓	✓	-	✓	-	-	-	-	-	-	-
		MegaTran [143]	2025	✓	✓	-	-	✓	✓	-	✓	-	✓	✓	-
	Data Integration	Agent-OM [144]	2024	✓	✓	✓	-	✓	✓	✓	-	-	✓	-	-
		MILA [145]	2025	✓	✓	✓	-	-	-	-	-	-	✓	-	-
		COMEM [146]	2025	-	✓	✓	-	-	-	-	-	-	✓	-	-
	Data Discovery	Chorus [147]	2024	-	✓	-	-	-	-	✓	-	-	-	✓	-
		DataVoyager [148]	2024	-	✓	✓	✓	✓	✓	✓	-	-	✓	✓	✓
		DATALORE [149]	2024	-	✓	✓	-	✓	✓	-	-	-	✓	-	-
		LEDD [150]	2025	-	✓	✓	-	✓	✓	-	-	-	✓	-	-
		MetaGen [151]	2025	-	✓	-	-	-	-	-	-	-	✓	-	-
		DBDescGen [152]	2025	-	✓	✓	✓	✓	✓	✓	-	-	✓	✓	-
		Wang et al. [153]	2025	-	✓	✓	-	✓	✓	✓	-	-	✓	✓	✓
Data Analysis	TableQA	StructGPT [154]	2023	-	✓	✓	-	✓	-	-	-	-	✓	✓	-
		ReAcTable [155]	2024	-	✓	✓	-	✓	-	-	-	-	-	-	-
		CHAIN-OF-TABLE [156]	2024	-	✓	✓	-	✓	-	-	-	-	-	-	-
		AutoTQA [157]	2024	-	✓	✓	-	✓	✓	✓	-	-	✓	✓	-
		Table-Critic [158]	2025	-	✓	-	✓	-	✓	✓	-	-	-	-	-
	NL2SQL	MAC-SQL [159]	2023	-	✓	✓	-	✓	✓	✓	-	-	-	-	-
		ChatBI [160]	2024	-	✓	✓	✓	✓	✓	-	-	-	✓	✓	-
		Chase-SQL [161]	2024	-	✓	✓	-	✓	✓	✓	✓	-	✓	✓	-
		Alpha-SQL [30]	2025	-	✓	-	-	✓	✓	✓	-	-	-	-	-
		OpenSearch-SQL [162]	2025	✓	✓	-	-	✓	✓	✓	-	-	✓	✓	-
		ReFoRCE [163]	2025	-	✓	-	✓	✓	✓	✓	-	-	-	✓	-
		Deepeye-SQL [164]	2025	✓	✓	-	-	✓	✓	✓	-	-	✓	✓	-
	NL2VIS	MatPlotAgent [165]	2024	-	✓	-	-	✓	✓	✓	-	-	-	-	-
		Text2Chart31 [166]	2024	-	✓	-	-	✓	✓	-	✓	-	✓	✓	-
		nvAgent [31]	2025	-	✓	✓	-	✓	✓	✓	-	-	✓	✓	-
		DeepVIS [167]	2025	-	✓	✓	-	✓	✓	-	✓	-	-	-	-
	Unstructured Data Analysis	FLARE [168]	2023	✓	✓	✓	-	✓	-	-	-	-	✓	-	-
		ReadAgent [169]	2024	-	✓	✓	✓	-	-	-	-	-	-	-	-
		GraphReader [170]	2024	-	✓	✓	✓	✓	✓	-	-	-	-	-	-
		Self-RAG [171]	2024	✓	✓	✓	-	✓	✓	-	✓	-	✓	✓	-
		REAR [172]	2024	✓	✓	-	-	-	-	-	✓	-	-	-	-
		QUEST [173]	2025	✓	✓	✓	-	✓	-	-	-	-	✓	✓	-
		Doctopus [174]	2025	✓	✓	✓	-	✓	-	-	✓	-	✓	✓	-
		MACT [175]	2025	-	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓
		DataPuzzle [176]	2025	✓	✓	✓	-	-	✓	✓	-	-	✓	✓	✓
	Report Generation	DataNarrative [177]	2024	-	✓	-	-	✓	✓	✓	-	-	-	✓	-
		Data Director [178]	2024	-	✓	✓	-	✓	-	✓	-	-	-	-	-
		LightVA [179]	2024	-	✓	✓	✓	✓	✓	✓	-	-	✓	✓	-
		Multimodal DeepResearcher [180]	2025	✓	✓	-	-	✓	✓	✓	-	-	✓	✓	✓
		DAgent [181]	2025	✓	✓	✓	✓	✓	-	-	✓	-	-	✓	-
		Chart Insighter [182]	2025	-	✓	✓	-	✓	✓	✓	-	-	-	✓	-
		ProactiveVA [183]	2025	-	✓	✓	✓	✓	✓	✓	-	-	✓	✓	✓
		VOICE [184]	2025	-	✓	✓	✓	✓	✓	✓	✓	-	✓	✓	✓
		NLI4VolVis [185]	2025	-	✓	✓	✓	✓	✓	✓	-	-	✓	✓	✓

and database systems, where tuning suggestions are evaluated and refined based on actual execution metrics. For instance, Li et al. [128] explore the effectiveness of LLMs in configuration tuning through a feedback loop. They design an iterative workflow in which the system uses performance metrics as input, manages knob selection, initialization, and recommendation leveraging its internal knowledge, and continuously refines these settings based on execution outcomes. Similarly, LLMIdxAdvis [129] introduces a self-optimization mechanism for iterative index tuning. In each cycle, proposed indexes are virtually created, and the system estimates remaining storage capacity and potential cost reduction, feeding these statistics back to guide further refinements.

Knowledge-Guided Tuning. Some studies further highlight the importance of structured domain knowledge encoded in knowledge bases, such as knowledge graphs or expert repositories, to guide and validate tuning actions. Rabbit [130] retrieves structured knowledge from a knob-centric knowledge graph that encodes static database manuals, knob dependencies, and historical tuning experiences. It iteratively executes configurations on various DBMS (*e.g.,* MySQL and PostgreSQL) and accordingly adjusts its search strategy through similarity-based feedback with past tasks. MCTuner [131] expands this paradigm by incorporating a Mixture-of-Experts (MoE) mechanism to identify performance-critical knobs and retrieves and validates multi-source knowledge—including documentation, web content, and LLM-generated insights—to guide optimal knob selection under complex, multi-objective constraints.

Limitations. L2 data agents can perceive system states and invoke tools to recommend configurations, but they remain confined to pre-defined knob sets and procedures within a guided pipeline, lacking autonomy to self-initiate tuning processes or adapt settings dynamically without human intervention.

Query Optimization. Regarding query optimization, L2 data agents operate LLMs as reasoning engines to iteratively explore alternative query formulations or execution plans. They incorporate database feedback or invoke external tools to enhance the optimization process and improve efficiency.

Execution Feedback-Driven Refinement. Direct approaches focus on leveraging real query execution feedback to refine data agents' optimization strategies through iterative plan generation and evaluation. For example, SERAG [132] uses a vector database to retrieve past query examples, constructs prompts based on these examples to generate optimized query plans, executes the plans, and dynamically interacts with the execution engine to incorporate performance feedback to continuously improve future optimizations.

Tool-Assisted Validation. Some methods further incorporate specialized external tools, such as query rewrite engines, equivalence verifiers, to evaluate and validate the optimized queries, whose assessment provides a concrete, measurable reward signal that drives the iterative improvement of data agents' proposals. R-Bot [134] exemplifies this on query rewriting by guiding an LLM to select and sequence transformation rules, which are subsequently applied via an external query rewrite engine to generate rewritten queries. The rewritten output is then evaluated to determine if it meets quality criteria, which in turn informs whether further refinement is needed. Similarly, QUITE [133] employs an external verification tool (*i.e.,* SQLSolver [186]) to check the equivalence of the original and rewritten queries, and defines a reward function based on execution metrics, thereby guiding the progressive improvement of rewrite quality.

Limitations. While capable of leveraging environmental feedback or external verifiers to iteratively optimize query plans, these systems adhere to pipelines, transformation rules, and rewards meticulously designed by humans to ensure robust performance, incapable of autonomously exploring beyond procedural optimizations or adopting new optimization strategies.

System Diagnosis. L2 data agents for system diagnosis are designed to actively perceive, localize, and resolve performance issues within database systems, going beyond simply responding to human prompts. At this level, they function as intelligent diagnostic agents that collect evidence, reason about complex system states, and generate actionable diagnoses through continuous monitoring and interaction with the database system. The feedback mechanisms employed by L2 data agents are twofold: they utilize direct execution feedback from the database and, in some cases, further leverage structured knowledge extracted from graph-based repositories that encode prior diagnostic experiences.

Interactive Evidence Collection and Diagnosis. Interacting with database systems allows data agents to leverage performance metrics (*e.g.,* estimated cost and storage impact) that are fed back for reflection and iterative refinement. For example, DBdoctor [187] monitors the execution of each SQL query in detail, collecting comprehensive resource usage data. It analyzes this information to automatically identify the root causes of performance issues, pinpoint problematic queries, and provide improvement suggestions. D-Bot [136] advances this approach by introducing a multi-agent system for database anomaly diagnosis, with a chief agent coordinating several specialized agents, each aligned with a specific cluster of diagnostic knowledge extracted from documents. They collectively perform multi-step root cause analysis through a tree-search algorithm, continuously interacting with the database state to accurately localize issues. Similarly, Panda [135] simulates the workflow of database engineers within a multi-component agentic framework. It interacts with the database using four key mechanisms: grounding to supply query context, verification to ensure correctness using troubleshooting documents and tickets, affordance to estimate the performance impact of recommendations, and feedback to enable continuous improvement.

Contextualized Diagnosis Knowledge. Some approaches further integrate evolving structured diagnosis knowledge graphs, which accumulate experiences and memory over time. Data agents retrieve relevant subgraphs or identify missing connections (paths) to provide concrete, contextualized knowledge, enabling more informed and precise diagnosis. For instance, DBAIOps [137] builds a heterogeneous graph to rep-

resent diagnostic experience and employs an automatic graph evolution mechanism to explore relevant paths and detect potential gaps (*i.e.,* missing connections) within the graph. This structured knowledge supports root cause identification and the generation of comprehensive diagnostic reports.

Limitations. While collecting evidence and localizing issues through database interactions and knowledge graphs significantly enhance performance and autonomy in anomaly diagnosis and resolution, these methods remain constrained by their reliance on structured diagnostic paths, human-defined worker agents, and predefined collaboration mechanisms. Consequently, they are unable to autonomously design diagnostic mechanisms to proactively address emerging anomalies and coordinate with tasks such as configuration tuning and query optimization, hindering the achievement of more efficient and robust data management.

C. L2 Data Agents in Data Preparation

Data Cleaning. At L2, data agents gain the ability to interact with the data environment and leverage feedback from these interactions to iteratively refine the data cleaning process. This establishes a closed loop in which data agents perceive the environment, apply cleaning methods, and optimize cleaning strategies and output based on the results, eliminating the need for manual human interaction with the environment or output integration, therefore enabling partial autonomy. Research at this level is distinguished by the specific environments involved and the types of feedback received, which can be categorized into the following patterns.

Execution and Assessment Environments. One of the interaction patterns at L2 involves treating code interpreters or data processing tools as environments that provide operational feedback. For instance, AutoPrep [29] deploys an Executor agent to interact with the execution engine, providing feedback to fix or refine generated code used for data derivation, normalization, and filtering. AutoDCWorkflow [140] executes cleaning operations within OpenRefine [188] and then internally assesses the modified table, using the resulting data quality report as feedback to guide the generation of subsequent operations in its iterative workflow. Similarly, CleanAgent [141] integrates a code executor module to interact with Python interpreters and Docker environments, and incorporates a chat manager as a memory module to maintain historical conversation context, enabling the generation of responses informed by the complete dialogue history.

Internal Self-Correction and Verification. Building on interaction with the execution environment, some L2 data agents further establish the feedback loop internally for self-assessment and refinement during data cleaning. For example, SketchFill [138] tests its generated imputation formulas on a masked data sample and receives a correct/incorrect signal from an internal Evaluator as feedback. If incorrect, a Reflector component analyzes and refines its underlying reasoning sketch for the next attempt. Likewise, IterClean [139] forms a loop where its Repairer cleans the data, and an internal Detector then re-evaluates it. The discovery of remaining or newly introduced errors serves as feedback to trigger another repair cycle. Moreover, MegaTran [143] enhances data transformation by interacting with the datasets and Python executor, using feedback to trigger a novel Sanity-check Reflection module with a checklist to internally review and refine generated code, or trigger Lazy-RAG to selectively retrieve external knowledge, which is then incorporated into the prompt to iteratively improve the generated code.

Downstream Task Performance as Feedback. A more advanced form uses the performance of a downstream task as the ultimate environmental feedback. The framework proposed by Bendinelli et al. [142] allows data agents to modify a dataset and submit it to a machine learning evaluation pipeline. The environment's feedback is the resulting model performance score, which directly guides subsequent cleaning actions to optimize the data for that specific downstream task.

Limitations. Despite these advancements, these L2 data agents are still considered partially autonomous, as their actions are confined within human-designed workflows or predefined strategic pipelines. For instance, CleanAgent [141] follows a fixed annotate-program-execute sequence, while SketchFill [138] operates within a structured generate-evaluate-reflect cycle. The agent's role is to execute and optimize within this given framework, not to devise the framework itself. Moreover, their system architectures and workflows are closely coupled with specific data cleaning tasks—for instance, CleanAgent [141] employs a tailored column-type annotator to enhance data cleaning, resulting in limited capability to handle diverse and comprehensive tasks.

Data Integration. Recent studies have introduced L2 data agents that gain environmental perception and interaction to dynamically resolve ambiguities in resolution, optimize alignment strategies, and validate intermediate results during data integration. Such progress significantly improves data integration performance and efficiency, while still operating within predefined data integration workflows. Below, we present representative advances.

Execution-Driven Iterative Refinement. SEED [189] exemplifies this approach by generating and validating code within sandboxed environments, while reusing cached intermediate results for entity resolution. Additionally, SEED incorporates an optimizer that selectively deploys and configures LLM-assisted modules that are best suited to the task, thereby reducing costs. It also invokes data access tools based on observations on data retrieval outcomes to obtain relevant information and improve response accuracy. COMEM [146] follows a "filter–refine" workflow where mid-sized LLMs screen candidates before stronger LLMs finalize matches, with positional-bias feedback dynamically refining the ranking process.

Retrieval-Driven Iterative Refinement. TaDA [190] similarly employs tabular databases and vector indices to optimize table alignment strategies based on environmental feedback, achieving autonomous execution in structured data integration tasks. Agent-OM [144] extends this paradigm by employing retrieval and matching agents that interact with hybrid databases and invoke alignment tools based on search feedback, further enhanced by chain-of-thought planning and

shared memory. MILA [145] integrates SBERT-based ontology retrieval with iterative LLM prompting, autonomously confirming high-confidence matches while adaptively handling ambiguous cases. Likewise, KG-RAG4SM [191] and KCMF [192] leverage external knowledge graphs to retrieve domain-specific subgraphs, refine semantic alignment logic, and mitigate hallucinations during schema matching.

Limitaitons. Overall, L2 data agents present a shift from static pipelines toward adaptive and self-improving data integration systems. However, their autonomy remains limited, as they continue to operate within human-designed components, workflows, and task boundaries.

Data Discovery. At L2, rather than statically producing metadata or schema descriptions, data agents actively engage with real data systems, perceive environmental signals, and iteratively refine their outputs. Such partial autonomy enables adaptive discovery aligned with the dynamic nature of enterprise-scale repositories.

Direct Interaction with Data Repositories. Some L2 data agents interface directly with repositories such as data lakes or relational databases. Through exploratory queries and schema traversal, they dynamically uncover structural and semantic characteristics of the underlying datasets. LEDD [150] exemplifies this paradigm by enabling data agents to actively interact with large-scale data lakes through a federated access layer. During discovery, the data agent queries the data lake via IGinX to retrieve schema and column metadata, analyzes content patterns in real time, and derives semantic representations through integrated LLMs reasoning. These representations are then iteratively refined through clustering and re-querying of the repository, forming a feedback loop that updates hierarchical catalogs and relation graphs as the environment evolves. Similarly, DataVoyager [148] interacts directly with live data repositories by ingesting structured datasets, parsing schema information, and sampling table contents to perceive statistical patterns. Guided by these environmental signals, it autonomously issues analytical queries to explore correlations and latent structures, forming a data-centric interaction loop.

Interaction via Execution Feedback. Another line of work emphasizes the integration of execution outcomes as feedback to guide refinement. In this setting, profiling quality measures or code execution results serve as environmental signals that inform subsequent iterations. For instance, MetaGen [151] establishes a validation-centered workflow, where hallucination detection and statistical consistency checks feed back into prompt reconstruction to enhance metadata quality. Similarly, DBDescGen [152] proposes automatic database description generation for NL2SQL, in which execution results such as query success rate or accuracy serve as feedback to refine schema-level descriptions. Furthermore, DATALORE [149] extends this paradigm from metadata refinement to lineage reconstruction. Specifically, the data agent generates and executes transformation scripts, validates outputs against target tables, and iteratively adjusts candidate selection and logic based on success or mismatch signals.

Multimodal Data Discovery. Some studies extend data discovery to heterogeneous and multimodal environments. Instead of confining discovery to relational or tabular systems, Chorus [147] is deployed as unified discovery executors capable of interacting simultaneously with structured, semi-structured, and unstructured sources. Through foundation model embeddings, it maps schemas, documents, and ontological entities into a shared representation space, enabling cross-source semantic alignment and bidirectional exchange of discovery signals across modalities. Moving from semantic alignment to system-level coordination, Wang et al. [153] advance this paradigm by operationalizing cross-modal interaction through declarative query interfaces, multimodal operators, and data-driven query planning. Within this framework, the data agent dynamically identifies relevant modalities, dispatches sub-queries across tables, documents, and graphs, and aggregates the results into coherent outputs.

Limitations. Despite their ability to perceive and interact with data environments, L2 data agents remain bounded by human-defined workflows and coarse feedback mechanisms. For example, LEDD [150] performs heuristic exploration without adaptive planning, often leading to redundant traversal and incomplete schema understanding. Metadata generation and database description frameworks [151], [152] incorporate only coarse-grained success signals, resulting in limited refinement and unstable discovery quality. Similarly, Chorus [147] employs static coordination pipelines across heterogeneous sources, restricting dynamic adjustment as data contexts evolve. These limitations underscore that L2 data agents execute and refine within predefined frameworks, whereas L3 agents are expected to autonomously design adaptive discovery pipelines and integrate fine-grained environmental feedback.

D. L2 Data Agents in Data Analysis

Structured Data Analysis. L2 data agents in structured data analysis enhance efficiency and insight extraction by autonomously interacting with tables or relational databases, executing code, and refining analyses through iterative feedback loops. They move beyond static querying to dynamically engage with, verify, and refine multi-step analytical processes, enabling more effective and complex data analysis, marking partial autonomy but limited strategic flexibility.

a) TableQA: Building upon the foundation of L1, L2 data agents mark an evolution in TableQA from static assistants to dynamic, adaptive problem-solvers in TableQA.

Stateful and Adaptive TableQA. ReAcTable [155] and Chain-of-Table [156] leverage external execution to create an iterative reasoning loop, materializing the process into a series of observable state transitions where intermediate tables serve as feedback for the LLM. The key difference between these approaches lies in the actions generated by the LLM and the structure of the feedback loop. ReAcTable employs a flexible ReAct [19]-inspired "thought-action-observation" cycle, in which the LLM produces general-purpose code such as SQL or Python. This code is executed externally, and the resulting observation guides the subsequent reasoning step. In contrast, Chain-of-Table employs a more structured "plan-execute" cycle, selecting from a predefined set of atomic tabular operations. Each operation directly transforms

the table, with the updated table serving as stateful feedback for the next planning iteration. Additionally, StructGPT [154] iteratively calls specialized APIs for structured data, allowing it to actively extract required evidence from the data source and plan further actions based on the returned data, thereby fostering a dynamic dialog with the data environment. TableMaster [193] tackles key challenges in table understanding, including locating target data, enriching table semantics, and overcoming rigid symbolic reasoning. It combines semantic verbalization of relevant table content with an adaptive reasoning strategy that flexibly integrates textual and symbolic approaches based on the query.

Multi-Agent TableQA. Some approaches further incorporate multi-agent collaboration to enhance TableQA performance. AutoTQA [157] delineates roles such as Planner, Engineer, and Critic to manage from planning and execution to evaluation and revision, focusing on solving a given complex task through strategic re-planning. Likewise, Table-Critic [158] constructs a cooperative framework involving agentic modules like Judge, Critic, and Refiner for multi-round error correction, but distinguishes itself by introducing a self-evolving template tree. This dynamic knowledge base, managed by a Curator agent module, allows the system to learn from experience by distilling successful correction patterns into reusable critique templates, allowing it to evolve and continuously refine its strategies over time.

Limitations. These studies represent a more advanced form of adaptive adjustment. However, these data agents at L2 exhibit several limitations. First, their feedback is often superficial, limited to raw outputs like updated tables or error messages rather than substantive reflection. Consequently, refinements tend to tactically address only the most recent error without reassessing the overall strategy, trapping data agents in unproductive loops that treat symptoms without fixing root causes [158], [194]. Second, these systems heavily depend on a predefined set of tools or APIs and cannot access additional tools or functions on demand, limiting their ability to address unforeseen problems requiring operations beyond those anticipated by their designers [195]. Moreover, data agents at this level, such as Table-Critic [158], operate within a human-designed architecture with specified components and templates targeted for TableQA tasks, which restricts their flexibility and versatility.

b) NL2SQL: In NL2SQL, L2 data agents exhibit partial autonomy, characterized by their ability to perceive and interact with databases [163], leverage SQL execution feedback [159], [196], and decompose complex SQL reasoning [30], [161].

Execution Feedback-Driven SQL Refinement. For example, earlier efforts like MAC-SQL [159] introduce a multi-agent framework with a refiner that uses SQL execution feedback from an external tool to iteratively refine faulty queries. OpenSearch-SQL [162] employs a consistency alignment module to reduce instruction-following failures and hallucinations, further enhancing the interactive refinement process. Extending this direction, ReFoRCE [163] actively interacts with relational databases through iterative column exploration guided by execution feedback, where both environmental exploration and feedback jointly contribute to the progressive and continuous self-improvement of the generated SQL queries. Such designs highlight the value of incorporating environment exploration and interaction into the SQL generation loop.

Planning and Decomposition. Building on refining SQL generation based on environmental perception and interaction, some studies introduce planning and decomposition mechanisms to improve performance in complex SQL generation, reflecting a progression from basic interaction toward meticulously designed reasoning pipelines. Chase-SQL [161] introduces a query plan chain-of-thought mechanism for planning, combined with a divide-and-conquer strategy that autonomously decomposes complex SQL generation into manageable subtasks, each followed by refinement modules through database interaction. ChatBI [160] decomposes schema linking into a Single View Selection problem, allowing effective handling of large and ambiguous schemas through lightweight machine learning. This is followed by LLM-based linking on the reduced schema, enhancing the capability to generate SQL queries with complex semantics and comparison relationships. In addition, DeepEye-SQL [164] implements a meticulously orchestrated pipeline inspired by the software development life cycle, decomposing SQL generation and deploying a toolchain of checkers, while leveraging interactive feedback to improve the generated SQL.

Improvement and Limitations. More advanced studies have moved toward more flexible approaches. For example, Alpha-SQL [30] employs Monte Carlo Tree Search (MCTS) to automatically explore effective combinations of actions such as function identification, schema linking, and SQL revision during NL2SQL translation. While this represents progress toward reducing reliance on human orchestration, the approach essentially depends on search heuristics rather than the deliberate orchestration of an end-to-end pipeline. Furthermore, its manually designed actions are highly task-specific and tailored exclusively for NL2SQL, limiting its generalizability to a broader range of data-related tasks. Consequently, these data agents' autonomy is still bounded.

c) NL2VIS: To better manage NL2SQL tasks, L2 data agents can execute code, assess the visual output, and use the feedback to debug, self-correct, and improve the final visualization within structured, carefully orchestrated workflows, such as multi-agent systems or iterative refinement cycles, to handle the complexities of visualization generation.

Iterative and Collaborative Refinement. For example, MatPlotAgent [165] emphasizes iterative refinement through direct interaction with execution and visual environments by adopting a dual-feedback mechanism: a code generation module with iterative debugging, and a visual feedback loop where a multimodal LLM acts as a "visual agent" to inspect generated charts to suggest corrections, resolving errors not obvious in code alone. Multi-agent collaboration is also used to decompose the NL2VIS task. nvAgent [31] employs a "divide-and-conquer" workflow with three specialized agents: A Processor agent interacts with the database to filter schemas, a Composer agent plans the visualization using step-by-step reasoning, and a Validator agent translates the plan into code and verifies the output. This iterative, collaborative workflow effectively

handles complex queries, especially involving joins across multiple tables.

Self-Improvement. Some systems leverage automated mechanisms to enable continuous learning. Text2Chart31 [166] introduces a reinforcement learning-based instruction tuning technique that does not require constant human feedback. By using automatically generated reward signals based on preference and alignment, it can learn and improve its chart generation capabilities in a more scalable, closed-loop manner. Text2Chart31 [166] also presents a new dataset that covers a wider range of chart types, including 3D and volumetric plots, to train more versatile agents.

Explainability. There are also efforts focused on improving transparency and facilitating human-in-the-loop collaboration. DeepVIS [167] tackles the "black-box" nature of LLM-based generation by integrating CoT reasoning into the NL2VIS pipeline. It features an interactive visual interface that allows users to inspect step-by-step reasoning, identify errors, and provide targeted feedback to refine the visualization.

Limitations. Despite these advances, L2 agents primarily operate within predefined procedural workflows tailored for NL2VIS tasks, such as iterative debugging, multi-agent collaboration, or reinforcement learning frameworks. While they excel at refining and validating visualizations once the problem scope and task decomposition are defined, they lack the capacity to autonomously perform higher-level reasoning to interpret complex user goals and design comprehensive visualization strategies, which may begin with data management and preparation, and culminate in the curation of appropriate visual representations.

Unstructured Data Analysis. In unstructured data analysis, L2 agents represent a significant leap from passive processing to active, strategic information handling. They move beyond linear, one-shot extraction to dynamically interact with complex documents, addressing core challenges such as extreme length, factual grounding, multimodality, and the high cost of LLM-based analysis [197]. This evolution is defined by capabilities for intelligent navigation, dynamic information seeking, and resource-aware optimization.

• *Textual Documents.* A primary challenge in this domain is managing vast, often noisy textual documents that exceed typical context windows.

Structured Representation for Navigation. L2 data agents address this by moving from a linear "read-and-process" model to one of intelligent navigation and dynamic information seeking. Early pioneering work like WebGPT [198] demonstrates how LLMs could be trained to operate a browsing environment to autonomously search, select, and cite sources to generate more faithful long-form answers. Building on this, existing studies create structured internal representations of documents to navigate them more efficiently. For example, ReadAgent [169] introduces a human-inspired gist-based memory, compressing long texts into condensed episodic summaries while retaining the ability to selectively look up original sections for detail. Similarly, GraphReader [170] models documents as graphs of key elements and facts, enabling a planning agent to perform targeted traversals and bypass irrelevant content to locate critical evidence.

Uncertainty-Driven Verification. Beyond navigation, other approaches enhance reliability through dynamic, self-driven verification. Reflection-driven agents like Self-RAG [171] and FLARE [168] incorporate mechanisms to assess uncertainty during generation. They can pause, trigger additional retrieval when knowledge is lacking, and critique their own outputs to improve factual accuracy. Further enhancing robustness, systems such as REAR [172] introduce explicit assessment modules to evaluate the relevance of retrieved documents, ensuring that the model is not misled by irrelevant content.

• *Multimodal Documents.* L2 data agents also advance the analysis of multimodal documents, where information is embedded not just in text but also in visual layouts, tables, and images. For instance, MACT [175] proposes a multi-agent collaboration framework for visual document understanding, featuring a specialized judgment agent that verifies correctness and redirects tasks for revision, creating a more robust self-correction loop. Complementing this, DataPuzzle [176] champions a structure-first approach, transforming unstructured multimodal inputs into interpretable representations like tables or graphs. This enables a modular workflow where specialized agents can explore, extract, and reason over structured data, overseen by a meta-agent that coordinates the process and triggers revisions, enhancing transparency and verifiability.

• *DocumentDB.* Another key barrier to widespread adoption is the high monetary cost and latency of LLM-based extraction. Addressing this, a significant trend is the development of systematic frameworks that treat unstructured analysis as a query optimization problem, akin to traditional databases (*i.e.,* DocumentDB). For example, QUEST [173] introduces an "optimize at execution time" architecture that creates a tailored query plan for each document, minimizing expensive extraction operations by intelligently ordering filters based on both cost and selectivity. Similarly, Doctopus [174] introduces a budget-aware framework that dynamically selects the most cost-effective extraction strategy—from simple parsers to powerful LLMs—based on user-defined constraints and internal quality estimates.

Limitations. Collectively, these advancements demonstrate a clear progression: L2 data agents for unstructured data analysis are becoming more strategic, robust, and efficient. They actively manage long contexts, verify information, handle multimodal complexity, and optimize for cost. Despite these advances, their autonomy remains bounded. They typically operate within structured frameworks, relying on predefined tools, operators, and manually curated data pipelines to perform their analysis, which sets the stage for future advancements toward more comprehensive autonomy.

Report Generation. In visualization and automated narrative generation tasks, L2 data agents perceive and interact with databases, tools, visual interfaces, etc., and self-optimize subtasks based on feedback within human-designed workflows, completing a "generate-visualize-detect-refine" cycle.

Narrative Generation and Verification. One research direction focuses on automated data storytelling and report gener-

ation, where the goal is to design multimodal agentic architectures that make the generation process more intelligent and autonomous. Multimodal DeepResearcher [180] adopts a four-stage pipeline: researching, exemplar textualization, planning, and multimodal synthesis, grounded in a Formal Description of Visualization (FDV) to produce interleaved text–chart reports. DAgent [181] targets relational database analytics, decomposing complex queries into sub-tasks and synthesizing reports directly from execution results. Data Director [178] extends storytelling into dynamic media by orchestrating multiple agents for data video creation. Some systems emphasize verification and error reduction, aiming to improve the semantic consistency and factual accuracy of narrative generation. For example, DataNarrative [177] employs a generator-evaluator pair of agents to iteratively validate chart–text narratives, and ChartInsighter [182] combines multi-agent tool use with self-consistency checks to refine time-series chart summaries and significantly reduce hallucinations.

Interactive Visual Analytics. Beyond narrative generation, a second major direction explores how agents can reshape user interaction with visualization systems. LightVA [179] employs a lightweight planner–executor–controller loop to assist users in decomposing, executing, and refining analytic tasks. ProactiveVA [183] goes a step further by embedding an LLM-based UI agent that monitors user interactions and proactively offers context-aware analytic suggestions rather than waiting for explicit queries. Complementing this proactive support, conversational and multimodal visualization systems highlight the role of agents as mediators between user queries and visualization environments. NLI4VolVis [185] enables volumetric data exploration and editing via conversational language grounded in 3D Gaussian splatting. VOICE [184] coordinates a "pack of bots" to interpret queries, assign subtasks, and generate synchronized visual explanations for interactive scientific communication.

Limitations. Collectively, these works move beyond the "prompt-and-response" paradigm of L1 by incorporating feedback and cross-modal reasoning. They achieve partial autonomy through "generate-visualize-detect-refine" loops, such as verifying narrative coherence [177], checking visual outputs against chart data [182], or sensing user interactions in visual analytics interfaces [179], [183]. These mechanisms enable more fluent and trustworthy storytelling across diverse modalities. Yet their autonomy remains bounded: current approaches still rely on pre-defined pipelines, focus on narrow task scopes, and lack the capacity for high-level and comprehensive narrative orchestration that spans data management, preparation, analysis, and ultimately report generation.

E. The Glass Ceiling of L2 Data Agents

L2 data agents integrate environmental perception with autonomous procedural execution to achieve partial autonomy in data-related tasks. However, as discussed, they face a glass ceiling marked by critical limitations as follows.

Dependence on human-defined pipelines. L2 data agents can execute and optimize within pre-established workflows but lack the ability to independently design pipelines tailored to high-level or evolving task goals. For example, CleanAgent [141] follows a human-crafted architecture with four specified components and workflows to conduct data cleaning rather than autonomously orchestrating the entire process.

Task-specific rigidity L2 data agents are closely tied to particular domains or tasks, which restricts their generalizability. For instance, Chase-SQL [161] employs a SQL translation and query-fixing pipeline specialized for NL2SQL tasks and fine-tunes an LLM solely for SQL candidate selection. This approach can not be easily extended to a broader range of data-related tasks, such as data integration or more high-level and complex tasks that span data management, preparation, and analysis, without significant redesign. Such narrow task specialization limits their effectiveness in addressing the diverse and comprehensive challenges common in real-world data-related tasks.

Together, the constraints of pipeline dependence and task specificity highlight that, although L2 data agents represent a notable advancement towards partial autonomy, they remain primarily reactive executors rather than self-directed architects of comprehensive and flexible data solutions that can take over the dominance and responsibility in data-related tasks.

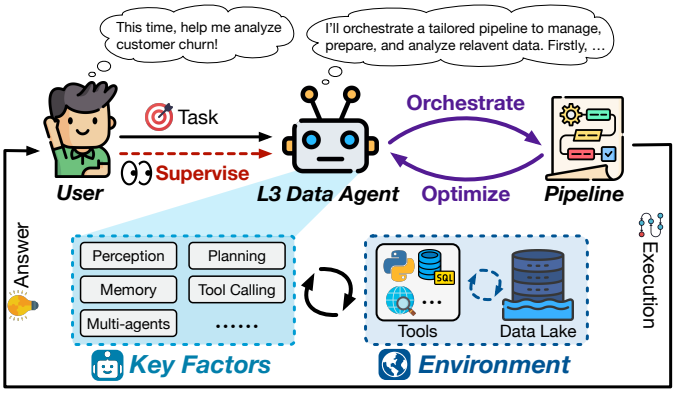

Fig. 6: L3 Data Agents (Conditional Autonomy).

V. L3: Striving for Autonomous Data Agent

A. The Critical Leap: From "Executor" to "Dominator"

In this section, we elaborate on the transformative advancement of data agents to the L3, where the transition to L3 represents a critical transfer of dominance in data-related tasks, achieving conditional autonomy under supervision and signaling the emergence of true autonomy.

As illustrated in Figure 6, L3 data agents are expected to autonomously orchestrate and optimize pipelines themselves rather than following human-defined ones; managing diverse and comprehensive tasks that potentially span the entire data lifecycle from data management and preparation to analysis, rather than isolated and task-specific procedures. Formally, at L3, data agent $\mathcal{A}$ autonomously manage the entire pipeline from orchestration $\pi_{\mathcal{A}}$ to execution $\epsilon_{\mathcal{A}}$, tackling versatile and comprehensive data-related tasks $\mathcal{T}$. Meanwhile, human $\mathcal{H}$ transitions into a supervisory role:

$$\mathcal{A}: \pi_{\mathcal{A}}(\mathcal{T}, \mathcal{D}, \mathcal{E}, \mathcal{M}) \rightarrow P; \quad \epsilon_{\mathcal{A}}(P, \mathcal{D}, \mathcal{E}, \mathcal{M}) \rightarrow \mathcal{O}$$
$$\mathcal{H}: \text{Supervise}(\pi_{\mathcal{A}}, \epsilon_{\mathcal{A}})$$

TABLE IV: Comparison of Proto-L3 Data Agents from Academia Research and Industry Products. Compares Open-source: availability; Undef Ops.: capabilities in utilizing unpredefined operators; data-related task coverage across data management, preparation, analysis; data complexity dimensions: Multi-source (Multis.), Heterogeneous (Hete.), and Multimodal (Multim.)

Years	Data Agent	Open-source	Undef Ops.	Data Complexity			Data Mangement			Data Preparation			Data Analysis		
				Multis.	Hete.	Multim.	Config Tun.	Query Opt.	Sys. Diag.	Data Clean.	Data Integ.	Data Disc.	Struct.	Unstruct.	Report Gen.
2025	AgenticData [199]	✓	✗	✓	✓	✓	-	✓	✓	✓	✓	✓	✓	✓	-
2025	DeepAnalyze [200]	✓	-	✓	✓	-	-	-	-	✓	✓	✓	✓	✓	✓
2025	AOP [201]	-	-	✓	✓	-	-	✓	-	✓	✓	✓	✓	✓	-
2025	iDataLake [32]	✓	-	✓	✓	-	-	✓	-	✓	✓	✓	✓	✓	✓
2024	Data Interpreter [202]	✓	-	-	-	✓	-	-	-	✓	-	✓	✓	✓	✓
2025	JoyAgent (JDCHO) [203]	✗	✗	✓	✓	-	-	-	-	✓	-	✓	✓	-	✓
2025	Assist. DS Agent (Databricks) [204]	-	-	✓	✓	-	-	✓	-	✓	✓	✓	✓	✓	✓
2025	TabTab (TabTab AI) [205]	-	-	✓	✓	-	-	-	-	✓	✓	-	✓	✓	✓
2025	Data Agent (ByteDance) [206]	-	-	✓	✓	-	-	-	-	-	✓	-	✓	✓	✓
2025	BigQuery (Google) [207]	-	-	✓	✓	-	-	✓	-	✓	✓	✓	✓	-	-
2025	Cortex (Snowflake) [208]	-	-	✓	✓	✓	-	-	-	✓	✓	✓	✓	✓	-
2025	Xata Agent (Xata) [209]	-	-	✓	✓	-	✓	✓	✓	-	-	✓	-	-	-
2025	SiriusBI (Tencent) [210]	-	-	✓	✓	✓	-	✓	-	✓	✓	✓	✓	✓	-

For example, given a high-level, often ambiguous, objective such as "analyze customer churn", L3 data agents are expected to orchestrate and coordinate a customized, end-to-end pipeline to achieve this goal. It determines which data to manage, how to prepare it, which analytical operations to apply, and how to demonstrate the results, and integrates these components into an effective and efficient pipeline. However, such autonomy is still conditional: although L3 data agents take the dominance and primary responsibility in data-related tasks, they still operate under human supervision and potential intervention.

The challenges described above highlight that true L3 autonomy remains an ongoing research frontier. To date, no existing system has fully realized such versatile, self-directed orchestration capabilities that define a complete L3 data agent. However, emerging efforts such as AgenticData [199] and JoyAgent [203] are beginning to address these challenges, giving rise to what we term "Proto-L3" data agents. These systems represent crucial exploratory steps to bridge the gap between L2 and L3. While they do not yet achieve the full spectrum of autonomous orchestration and robust generalization required for conditional autonomy, they exhibit initial and promising capabilities toward this goal. Analyzing these recent efforts is therefore essential for understanding the current state-of-the-art and charting a path toward the realization of true L3 data agents. In the following section, we discuss Proto-L3 systems that are beginning to bridge this critical transition, which are summarized and compared in Table IV.

B. Emerging Efforts and Proto-L3 Systems

Emerging efforts on Proto-L3 data agents employ various approaches to achieve conditional autonomy under supervision. Their goal is to transform sequences of manually designed specialized components and processing procedures for diverse and complex data-related tasks into autonomously orchestrated and optimized pipelines managed by data agents, as illustrated in Figure 6.

Orchestring Pipeline. Earlier exploration includes Data Interpreter [202], which recasts the pipeline orchestration as a Hierarchical Graph Modeling problem to support various data analysis tasks. It autonomously decomposes a high-level user requirement into a structured Task Graph, which is then further broken down into an executable Action Graph. Crucially, this graph is not static; the system employs Iterative Graph Refinement, allowing it to dynamically modify its own plan in response to graph executor feedback.

Broadening Task Scope. However, these systems are limited in task scope, primarily concentrating on data analysis and relying heavily on preprocessed data. SiriusBI [210] incorporates Data Insight and Knowledge Management modules that enable automating basic data preparation and database metadata management, supporting the planning and orchestration of multi-agent workflows for complex analytical queries. Systems like iDataLake [32] and AOP [201] further extend this scope to heterogeneous data lakes. iDataLake [32] supports analytics on data lakes by orchestrating executable pipelines constructed from semantic operators, and includes an offline process that employs a unified embedding space to align multimodal data, enabling effective data linking. Likewise, AOP [201] orchestrates pipelines using a predefined set of semantic operators to support data cleaning, integration, discovery, and semantic data analytics across heterogeneous data with a pipeline optimizer to improve efficiency through cost-based optimization, prefetching, and parallel execution. It

also features dynamic, interactive pipeline adjustment and self-reflection based on real-time results. Furthermore, DeepAnalyze [200] introduces a curriculum-based agentic training paradigm combining SFT and GRPO, which progressively trains data agents to orchestrate pipelines for data preparation and analysis tasks, producing insightful reports through five key actions: analyze, understand, code, execute, and answer. This also highlights the potential of agentic reinforcement learning in transforming data agents' capabilities from static, heuristic modules into adaptive agentic behavior [211].

Industrial Practice. Recently, this field has also attracted significant attention from industry, leading to the development of several promising data agent products. Snowflake showcases Cortex [208], which orchestrates on both structured and unstructured data sources through a specialized Cortex search service to deliver insights. ByteDance has introduced a data agent [206] that supports basic data integration from heterogeneous sources, as well as analysis and report generation, targeting application scenarios such as marketing analysis. Similarly, Tabtab [205] supports data cleaning and integration across heterogeneous data and can produce data reports in multiple formats, including documents, web pages, and slides. Google's BigQuery [207] focuses primarily on structured data, providing query optimization techniques to enhance performance and facilitate data preparation and analysis. Xata Agent [209] offers comprehensive data management by orchestrating tasks such as configuration tuning, query optimization, and system diagnosis to support data discovery, demonstrating potential for further integration with analytical systems. Furthermore, Databricks proposes an Assistance Data Science Agent [204] that supports a broader range of data-related tasks through orchestration with a planner module, enabling applications across query optimization, data preparation, and data analysis, including generating reports in dashboards, establishing itself as one of the state-of-the-art systems in this domain.

Overcoming Predefined Toolsets and Operators. Despite these advances, their autonomy remains limited due to dependence on pre-designed components, such as the human-designed agentic components and collaboration mechanisms [204], [206], [210], or the reliance on predefined tools and operators [202], [207], [208]. JoyAgent [203] begins to overcome the limitations of fixed toolsets with its "Tool Evolution" capability, enabling the dynamic creation of new tools by disassembling and recombining existing atomic tools. It also introduces an advanced multi-level and multi-pattern thinking framework that decomposes high-level queries into a Directed Acyclic Graph (DAG) of subtasks for concurrent execution, demonstrating strategic orchestration capabilities. AgenticData [199], while incorporating predefined relational and semantic operators for pipeline orchestration, also supports the curation of non-predefined operators by leveraging LLMs to generate code. It further broadens the task scope to include data management, preparation, and analysis by enabling data manipulation and discovery across heterogeneous sources through customized Model-Context Protocol (MCP) servers. Additionally, it introduces a feedback-driven multi-agent planning framework that transforms complex and comprehensive user queries into a tree-structured semantic pipeline composed of relational and semantic operators.

Limitations. Collectively, these pioneering developments signal a significant shift in the paradigm. By incorporating mechanisms for autonomous pipeline orchestration and dynamic optimization, they show the potential to gain dominance in data-related tasks. However, they have not yet achieved true L3 autonomy. Their orchestration, despite greater autonomy, often depends on a predefined set of operators and tools; although their task scope has expanded, it remains limited—for example, in effective configuration tuning and system diagnosis within data management. Their reasoning tends to be tactical rather than strategic over extended horizons, and they lack the capability to genuinely self-evolve by learning from experiences across multiple tasks. Addressing these gaps constitutes a critical challenge for realizing true L3 data agents.

C. Challenges and Research Opportunities Towards True L3

Despite the promising advancements of Proto-L3 data agents, a significant chasm still separates their current capabilities from the vision of a true L3 data agent. The limitations of these emerging efforts highlight several fundamental challenges that constitute key research opportunities for the field.

1) Limited Autonomy in Pipeline Orchestration. A primary limitation is that, although Proto-L3 data agents have made progress in autonomously orchestrating pipelines to some extent, they still rely heavily on predefined operations or tools. This dependence restricts their autonomy, preventing a true transition from L2 partial autonomy to true conditional autonomy. For example, systems such as AOP [201], BigQuery [207], and Cortex [208] continue to orchestrate pipelines with predefined and pre-programmed semantic operators. Notable advancements include JoyAgent [203], which introduces a "Tool Evolution" capability that dynamically generates new tools by reassembling existing atomic tools. However, this process remains a form of recombination rather than genuine ab initio creation. AgenticData [199] enables the curation of non-predefined operations through LLM-based code generation, but lacks validation regarding the effectiveness and robustness. Sun et al. [24] outline the potential of automatic data-skill discovery to address these gaps by extracting atomic skills from diverse task examples, organizing them hierarchically based on prevalence, and equipping data agents with a validated, dynamically enriched toolkit that transcends fixed operator sets. Future research should integrate continuous skill discovery with pipeline orchestration, enabling data agents to generate, evaluate, and deploy both predefined and emergent skills across complex workflows, thereby advancing autonomy toward true L3.

2) Incomplete Coverage of the Data Lifecycle. Current Proto-L3 systems face a significant limitation in their relatively narrow focus on specific subsets of data-related tasks, predominantly centered on data analysis, as seen in systems like Data Interpreter [202], or restricted to basic and very limited data preparation, as exemplified by SiriusBI [210], Joy Agent [203], and AOP [201]. AgenticData [199] attempts

to support management across heterogeneous sources through customized MCP servers, but its capabilities remain confined to basic monitoring and manipulation. However, a true L3 data agent should function as a versatile "data expert" spanning the entire data lifecycle under supervision, capable not only of data analysis but also of managing and optimizing the underlying systems that govern data lakes. Crucial data management tasks such as database configuration tuning, system diagnosis, query rewriting, and index management still remain largely unaddressed by these recent systems. Consequently, their expertise is mostly limited to downstream analytical processes using preprocessed or nearly ready-to-use data. They still lack the ability to autonomously manage and optimize data infrastructure and preparation to better support downstream analysis. The key research opportunity lies in enhancing their versatility and cross-stage capabilities, enabling them to reason about and execute tasks across the full data lifecycle, from low-level system administration to high-level insight extraction.

3) Deficiencies in Advanced Reasoning. Current Proto-L3 data agents exhibit strong tactical capabilities, effectively addressing immediate errors and refining recent steps. However, they often lack higher-order reasoning to reassess and adapt their overarching analysis strategy. By focusing narrowly on the symptom of a failed step, current systems can become trapped in unproductive loops that repeatedly treat symptoms rather than diagnosing and resolving root causes. Advancing beyond this limitation requires research into integrated causal reasoning and meta-reasoning for cross-process optimization and sophisticated memory architectures that support abstract strategic knowledge beyond execution history.

4) Inadequate Adaptation to Dynamic Environment. Real-world data ecosystems are dynamic and continuously evolve, requiring data agents to operate adaptively within environments shaped by data drift, shifting schemas, and changing user requirements. By contrast, current Proto-L3 systems are primarily designed and evaluated against static data and tasks, and they lack capabilities to genuinely self-evolve as the data environment evolves. Although most existing approaches rely on inherent reasoning to adapt, DeepAnalyze [200] employs agentic reinforcement learning to enhance these capabilities, albeit at the expense of substantial human efforts in the process. Consequently, developing more effective and human-free methods to enable data agents to adapt to evolving environments, together with robust evaluation under dynamic conditions, represents a meaningful and promising research direction.

VI. L4–L5: Vision of Proactive and Generative Data Agent

While current research is actively exploring the transition from L2 to L3, the realms of L4 and L5 represent the forward-looking vision for data agents. These future levels are expected by a profound shift from conditional, task-driven autonomy to proactive self-governance and, ultimately, generative innovation. As we look ahead, it is crucial to envision not only what these data agents are expected to be capable of but also the fundamental capabilities that enable their evolution.

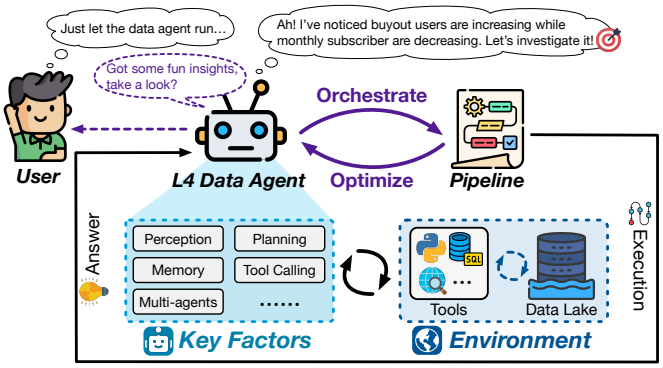

Fig. 7: L4 Data Agents (High Autonomy).

A. L4: High Automation

At L4, data agents are envisioned to achieve a high degree of autonomy, being fully delegated the responsibility for data-related tasks. L4 data agents function as trustworthy, proactive, long-lived, and self-governing agents within data lakes, operating independently of human supervision or explicit task instructions.

As shown by Figure 7. Rather than responding to given goals, L4 data agents independently monitor and explore data lakes, proactively identifying valuable and emerging tasks, then autonomously orchestrate end-to-end pipelines to address these self-discovered tasks. Unlike lower levels, where data agents depend on human-defined objectives and oversight, L4 data agents take the initiative to detect issues or opportunities, shifting the user's role from active supervisor to passive onlooker and recipient of insights. For instance, it might autonomously detect an emerging trend in user behavior, like the increase in buyout subscriptions and the decrease in monthly subscriptions, then initiate a root-cause analysis for data drift, or re-optimize an indexing strategy based on evolving task patterns. Formally, the data agent $\mathcal{A}$ takes full initiative, not only orchestrates $\pi_{\mathcal{A}}$ and executes $\epsilon_{\mathcal{A}}$ pipeline P but also autonomously discovers task T' to begin with. The human $\mathcal{H}$ transitions to a passive recipient of the output:

$$\begin{aligned}&\mathcal{A}: \text{Discover}_{\mathcal{A}}(\mathcal{D},\mathcal{E},\mathcal{M}) \to \mathcal{T}';\\ &\quad \pi_{\mathcal{A}}(\mathcal{T}',\mathcal{D},\mathcal{E},\mathcal{M}) \to P; \quad \epsilon_{\mathcal{A}}(P,\mathcal{D},\mathcal{E},\mathcal{M}) \to \mathcal{O}\\ &\mathcal{H}: \text{Receive}(\mathcal{O})\end{aligned}$$

This leap is associated with three core advancements: the capacity for autonomous problem discovery, the establishment of trustworthy self-governance, and long-horizon views.

- *Autonomous Problem Discovery.* To achieve autonomous problem discovery, L4 data agents are expected to demonstrate critical reasoning, task-oriented awareness, and curiosity to investigate data throughout its lifecycle. During routine data management and preparation, they not only execute tasks autonomously but also critically evaluate data to identify gaps, anomalies, and latent issues worthy of further investigation. For instance, while cleansing sales data, an agent might detect not only inaccuracies but also a subtle, recurring drop in transactions within a specific region, flagging this as a significant issue requiring

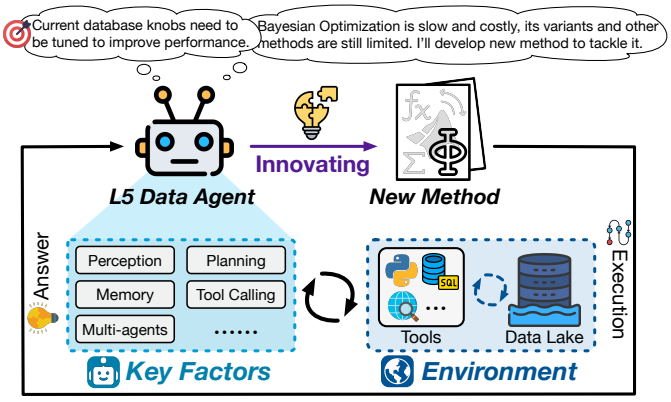

Fig. 8: L5 Data Agents (Full Autonomy).

comprehensive analysis. This transforms the data pipeline from a passive flow into an active source of analytical hypotheses.

- *Trustworthy Self-governance*. To address the diverse and unforeseen challenges they encounter, L4 data agents must operate as reliable generalists, fully entrusted with data-related responsibilities beyond the supervised scope of L3 data agents. This demands a comprehensive mastery of the entire data lifecycle. Their abilities to self-manage resources and ensure the accuracy and security of their operations are expected to form the foundation of trust, making human oversight unnecessary.
- *Long-Horizon and Holistic Views.* Achieving high levels of autonomy and self-governance requires that L4 data agents possess long-term planning and decision-making capabilities. They should be able to optimize across processes and make strategic trade-offs, for example, balancing the immediate computational costs of data cleaning against the potential long-term analytical benefits. Such foresight prioritizes holistic benefits and long-term objectives over short-term gains, enabling the management of goals and resources over extended periods. Consequently, L4 data agents are expected to adopt a long-horizon and holistic perspective that goes beyond tactical operations and local optimizations.

B. L5: The Ultimate Vision of Ubiquitous and Generative Data Agents

L5 represents the ultimate vision: a fully autonomous data agent that functions as a ubiquitous and generative data scientist. Such an agent goes beyond applying existing methods to actively creating new knowledge by recognizing when conventional approaches fall short and identifying the need for novel solutions, as illustrated in Figure 8. For example, an L5 data agent might develop a novel sampling theory for high-velocity data streams, design a new framework for federated data preparation tailored to specific application scenarios, or invent an original visualization grammar for high-dimensional data. Formally, the data agent $\mathcal{A}$ not only autonomously identifies the promising task $\mathcal{T}'$ but also invents a new, innovative process Φ (*e.g.,* a new theory, algorithm, or paradigm) rather than relying on existing methods, while human $\mathcal{H}$ disengages:

$$\mathcal{A}: \text{Discover}_{\mathcal{A}}(\mathcal{D}, \mathcal{E}, \mathcal{M}) \rightarrow \mathcal{T}';$$
$$\text{Innovate}_{\mathcal{A}}(\mathcal{T}', \mathcal{D}, \mathcal{E}, \mathcal{M}) \rightarrow \Phi; \quad \Phi(\mathcal{T}', \mathcal{D}, \mathcal{E}, \mathcal{M}) \rightarrow \mathcal{O}$$
$$\mathcal{H}: \emptyset$$

This final stage transcends technical proficiency to embrace innovation, where data agents not only establish state-of-the-art algorithms and models but also pioneer new analytical paradigms and even formulate groundbreaking theories. At this visionary stage, human engagement becomes unnecessary and potentially detrimental, as L5 data agents are envisioned as a fully autonomous and generative intellects with unparalleled expertise, pushing the frontiers of data management, preparation, and analysis.

C. The Research Odyssey Ahead

The progression from Proto-L3 data agents to L3, L4, and L5 is an odyssey, requiring fundamental breakthroughs rather than mere incremental improvements. As discussed, key challenges include developing data agents endowed with autonomous orchestration, versatility and generalizability, critical reasoning, intrinsic motivation for independent task discovery, long-horizon planning and decision-making, and robust safety guarantees. While realizing the full potential of L4 and L5 remains a long-term vision, we believe exploratory efforts, such as developing data agents that can autonomously manage diverse tasks within data lakes over extended periods with progressively reduced human intervention, constitute essential and achievable steps toward the eventual goal of fully autonomous data agents.

VII. Conclusion

In this survey, we introduce the first systematic hierarchical taxonomy for data agents, comprising six levels (L0–L5) that delineate progressive shifts in autonomy from manual operations to fully autonomous systems. We conduct a structured literature review through this taxonomy, categorizing works across data-related tasks. We further analyze evolutionary leaps and technical gaps between levels, with emphasis on the ongoing L2-to-L3 transition. Finally, we provide a roadmap for future research, outlining opportunities to bridge gaps and envision higher autonomy towards proactive and generative data agents (L4–L5).

References

[1] L. Cao, "Data science: A comprehensive overview," *ACM Comput. Surv.*, vol. 50, no. 3, Jun. 2017. [Online]. Available: https://doi.org/10.1145/3076253
[2] F. Cady, *The data science handbook*. John Wiley & Sons, 2024.
[3] C. Chai, J. Wang, Y. Luo, Z. Niu, and G. Li, "Data management for machine learning: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 5, pp. 4646–4667, 2023.
[4] A. A. Fernandes, M. Koehler, N. Konstantinou, P. Pankin, N. W. Paton, and R. Sakellariou, "Data preparation: A technological perspective and review," *SN Computer Science*, vol. 4, no. 4, p. 425, 2023.
[5] C. Chai, N. Tang, J. Fan, and Y. Luo, "Demystifying artificial intelligence for data preparation," in *Companion of the 2023 International Conference on Management of Data*, ser. SIGMOD '23. New York, NY, USA: Association for Computing Machinery, 2023, p. 13–20. [Online]. Available: https://doi.org/10.1145/3555041.3589406

[6] X. Qin, Y. Luo, N. Tang, and G. Li, "Making data visualization more efficient and effective: a survey," *VLDB J.*, vol. 29, no. 1, pp. 93–117, 2020.

[7] Y. Luo, X. Qin, N. Tang, and G. Li, "Deepeye: Towards automatic data visualization," in *ICDE*. IEEE Computer Society, 2018, pp. 101–112.

[8] Y. Luo, C. Chai, X. Qin, N. Tang, and G. Li, "Visclean: Interactive cleaning for progressive visualization," *Proc. VLDB Endow.*, vol. 13, no. 12, pp. 2821–2824, 2020.

[9] Y. Luo, Y. Zhou, N. Tang, G. Li, C. Chai, and L. Shen, "Learned data-aware image representations of line charts for similarity search," *Proc. ACM Manag. Data*, vol. 1, no. 1, pp. 88:1–88:29, 2023.

[10] Y. Luo, W. Li, T. Zhao, X. Yu, L. Zhang, G. Li, and N. Tang, "Deeptrack: Monitoring and exploring spatio-temporal data - A case of tracking COVID-19 -," *Proc. VLDB Endow.*, vol. 13, no. 12, pp. 2841–2844, 2020.

[11] X. Zhou, J. He, W. Zhou, H. Chen, Z. Tang, H. Zhao, X. Tong, G. Li, Y. Chen, J. Zhou *et al.*, "A survey of llm× data," *arXiv preprint arXiv:2505.18458*, 2025.

[12] G. Li, X. Zhou, and X. Zhao, "LLM for data management," *Proceedings of the VLDB Endowment*, vol. 17, no. 12, pp. 4213–4216, 2024.

[13] X. Zhou, G. Li, and Z. Liu, "Llm as dba," *arXiv preprint arXiv:2308.05481*, 2023.

[14] G. Li, J. Wang, C. Zhang, and J. Wang, "Data+AI: Llm4data and data4llm," in *Companion of the 2025 International Conference on Management of Data*, 2025, pp. 837–843.

[15] S. Minaee, T. Mikolov, N. Nikzad, M. Chenaghlu, R. Socher, X. Amatriain, and J. Gao, "Large language models: A survey," *arXiv preprint arXiv:2402.06196*, 2024.

[16] X. Lin, Y. Qi, Y. Zhu, T. Palpanas, C. Chai, N. Tang, and Y. Luo, "LEAD: iterative data selection for efficient LLM instruction tuning," *CoRR*, vol. abs/2505.07437, 2025.

[17] F. Teng, Z. Yu, Q. Shi, J. Zhang, C. Wu, and Y. Luo, "Atom of thoughts for markov LLM test-time scaling," *CoRR*, vol. abs/2502.12018, 2025.

[18] B. Liu, X. Li, J. Zhang, J. Wang, T. He, S. Hong, H. Liu, S. Zhang, K. Song, K. Zhu *et al.*, "Advances and challenges in foundation agents: From brain-inspired intelligence to evolutionary, collaborative, and safe systems," *arXiv preprint arXiv:2504.01990*, 2025.

[19] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *International Conference on Learning Representations (ICLR)*, 2023.

[20] T. Sumers, S. Yao, K. Narasimhan, and T. Griffiths, "Cognitive architectures for language agents," *Transactions on Machine Learning Research*, 2023.

[21] J. Zhang, J. Xiang, Z. Yu, F. Teng, X. Chen, J. Chen, M. Zhuge, X. Cheng, S. Hong, J. Wang *et al.*, "Aflow: Automating agentic workflow generation," *arXiv preprint arXiv:2410.10762*, 2024.

[22] Y. Zhuang, Y. Yu, K. Wang, H. Sun, and C. Zhang, "Toolqa: A dataset for llm question answering with external tools," *Advances in Neural Information Processing Systems*, vol. 36, pp. 50 117–50 143, 2023.

[23] J. Feng, S. Huang, X. Qu, G. Zhang, Y. Qin, B. Zhong, C. Jiang, J. Chi, and W. Zhong, "Retool: Reinforcement learning for strategic tool use in llms," *arXiv preprint arXiv:2504.11536*, 2025.

[24] Z. Sun, J. Wang, X. Zhao, J. Wang, and G. Li, "Data Agent: A holistic architecture for orchestrating data+ ai ecosystems," *arXiv preprint arXiv:2507.01599*, 2025.

[25] Y. Fu, D. Wang, W. Ying, X. Zhang, H. Liu, and J. Pei, "Autonomous data agents: A new opportunity for smart data," *arXiv preprint arXiv:2509.18710*, 2025.

[26] Z. Tang, W. Wang, Z. Zhou, Y. Jiao, B. Xu, B. Niu, X. Zhou, G. Li, Y. He, W. Zhou *et al.*, "Llm/agent-as-data-analyst: A survey," *arXiv preprint arXiv:2509.23988*, 2025.

[27] P. Wang, Y. Yu, K. Chen, X. Zhan, and H. Wang, "Large language model-based data science agent: A survey," *arXiv preprint arXiv:2508.02744*, 2025.

[28] W. Zhou, J. Sun, X. Zhou, G. Li, L. Liu, H. Wu, and T. Wang, "Gaussmaster: An llm-based database copilot system," *arXiv preprint arXiv:2506.23322*, 2025.

[29] M. Fan, J. Fan, N. Tang, L. Cao, G. Li, and X. Du, "Autoprep: Natural language question-aware data preparation with a multi-agent framework," *Proc. VLDB Endow.*, vol. 18, no. 10, p. 3504–3517, Sep. 2025. [Online]. Available: https://doi.org/10.14778/3748191.3748211

[30] B. Li, J. Zhang, J. Fan, Y. Xu, C. Chen, N. Tang, and Y. Luo, "Alpha-SQL: Zero-shot text-to-SQL using monte carlo tree search," in *Forty-second International Conference on Machine Learning*, 2025. [Online]. Available: https://openreview.net/forum?id=kGg1ndttmI

[31] G. Ouyang, J. Chen, Z. Nie, Y. Gui, Y. Wan, H. Zhang, and D. Chen, "nvAgent: Automated data visualization from natural language via collaborative agent workflow," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 19 534–19 567. [Online]. Available: https://aclanthology.org/2025.acl-long.960/

[32] J. Wang, G. Li, and J. Feng, "iDataLake: An llm-powered analytics system on data lakes," *Data Engineering*, p. 57, 2025.

[33] E. Shi, T. M. Gasser, A. Seeck, and R. Auerswald, "The Principles of Operation Framework: A Comprehensive Classification Concept for Automated Driving Functions," *SAE International Journal of Connected and Automated Vehicles*, vol. 03, no. 1, pp. 27–37, Feb. 2020.

[34] M. A. Ramos, C. C. Jullian, J. McCullough, J. Ma, and A. Mosleh, "Automated driving systems operating as mobility as a service: operational risks and sae j3016 standard," in *2023 Annual reliability and maintainability symposium (RAMS)*. IEEE, 2023, pp. 1–6.

[35] Z. Tan, D. Li, S. Wang, A. Beigi, B. Jiang, A. Bhattacharjee, M. Karami, J. Li, L. Cheng, and H. Liu, "Large language models for data annotation and synthesis: A survey," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 930–957. [Online]. Available: https://aclanthology.org/2024.emnlp-main.54/

[36] J. Freire, G. Fan, B. Feuer, C. Koutras, Y. Liu, E. Peña, A. S. Santos, C. T. Silva, and E. Wu, "Large language models for data discovery and integration: Challenges and opportunities." *IEEE Data Eng. Bull.*, vol. 49, no. 1, pp. 3–31, 2025.

[37] S. Maojun, R. Han, B. Jiang, H. Qi, D. Sun, Y. Yuan, and J. Huang, "A survey on large language model-based agents for statistics and data science," *The American Statistician*, no. just-accepted, pp. 1–21, 2025.

[38] M. Rahman, A. Bhuiyan, M. S. Islam, M. T. R. Laskar, R. Mahbub, A. Masry, S. Joty, and E. Hoque, "Llm-based data science agents: A survey of capabilities, challenges, and future directions," *arXiv preprint arXiv:2510.04023*, 2025.

[39] S. Wang, Z. Tan, Z. Chen, D. Li, Y. Zhu, B. Jiang, Y. He, C. Zhao, Z. Lei, P. Sheth, L. Li, L. P. Y. Ting, J. Li, and H. Liu, "Large language models for data science: A survey," 06 2025.

[40] X. Liu, S. Shen, B. Li, P. Ma, R. Jiang, Y. Zhang, J. Fan, G. Li, N. Tang, and Y. Luo, "A survey of text-to-sql in the era of llms: Where are we, and where are we going?" *IEEE Transactions on Knowledge and Data Engineering*, 2025.

[41] Y. Luo, G. Li, J. Fan, C. Chai, and N. Tang, "Natural language to sql: State of the art and open problems," *Proceedings of the VLDB Endowment*, vol. 18, no. 12, pp. 5466–5471, 2025.

[42] S. Duan, V. Thummala, and S. Babu, "Tuning database configuration parameters with ituned," *Proc. VLDB Endow.*, vol. 2, no. 1, p. 1246–1257, Aug. 2009. [Online]. Available: https://doi.org/10.14778/1687627.1687767

[43] X. Zhao, X. Zhou, and G. Li, "Automatic database knob tuning: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 35, no. 12, pp. 12 470–12 490, 2023.

[44] C. Tanimura, *SQL for Data Analysis: Advanced Techniques for Transforming Data Into Insights*. " O'Reilly Media, Inc.", 2021.

[45] A. Satyanarayan, D. Moritz, K. Wongsuphasawat, and J. Heer, "Vega-lite: A grammar of interactive graphics," *IEEE transactions on visualization and computer graphics*, vol. 23, no. 1, pp. 341–350, 2016.

[46] L. Grammel, M. Tory, and M.-A. Storey, "How information visualization novices construct visualizations," *IEEE transactions on visualization and computer graphics*, vol. 16, no. 6, pp. 943–952, 2010.

[47] X. Huang, H. Li, J. Zhang, X. Zhao, Z. Yao, Y. Li, Z. Yu, T. Zhang, H. Chen, and C. Li, "LLMTune: Accelerate database knob tuning with large language models," *CoRR*, 2024.

[48] C. Fan, Z. Pan, W. Sun, C. Yang, and W.-N. Chen, "Latuner: An llm-enhanced database tuning system based on adaptive surrogate model," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2024, pp. 372–388.

[49] J. Lao, Y. Wang, Y. Li, J. Wang, Y. Zhang, Z. Cheng, W. Chen, M. Tang, and J. Wang, "GPTuner: A manual-reading database tuning system via gpt-guided bayesian optimization," *Proc. VLDB Endow.*, vol. 17, no. 8, p. 1939–1952, Apr. 2024. [Online]. Available: https://doi.org/10.14778/3659437.3659449

[50] V. Giannakouris and I. Trummer, "λ-tune: Harnessing large language models for automated database system tuning," *Proceedings of the ACM on Management of Data*, vol. 3, no. 1, pp. 1–26, 2025.

[51] X. Huang, H. Li, J. Zhang, X. Zhao, Z. Yao, Y. Li, T. Zhang, J. Chen, H. Chen, and C. Li, "E2Etune: End-to-end knob tuning via fine-tuned generative language model," *Proceedings of the VLDB Endowment*, 2025.

[52] X. Zhou, Z. Sun, and G. Li, "Db-gpt: Large language model meets database," *Data Science and Engineering*, vol. 9, no. 1, pp. 102–111, 2024.

[53] Z. Li, H. Yuan, H. Wang, G. Cong, and L. Bing, "Llm-r2: A large language model enhanced rule-based rewrite system for boosting query efficiency," *Proceedings of the VLDB Endowment*, vol. 18, no. 1, pp. 53–65, 2024.

[54] J. Liu and B. Mozafari, "Query rewriting via large language models," *arXiv preprint arXiv:2403.09060*, 2024.

[55] S. Dharwada, H. Devrani, J. Haritsa, and H. Doraiswamy, "Query rewriting via llms," *arXiv preprint arXiv:2502.12918*, 2025.

[56] J. Tan, K. Zhao, R. Li, J. X. Yu, C. Piao, H. Cheng, H. Meng, D. Zhao, and Y. Rong, "Can large language models be query optimizer for relational databases?" *arXiv preprint arXiv:2502.05562*, 2025.

[57] Z. Yao, H. Li, J. Zhang, C. Li, and H. Chen, "A query optimization method utilizing large language models," *arXiv preprint arXiv:2503.06902*, 2025.

[58] D. Xu, Y. Cui, W. Shi, Q. Ma, H. Guo, J. Li, Y. Zhao, R. Zhang, S. Di, J. Zhu *et al.*, "E3-rewrite: Learning to rewrite sql for executability, equivalence, and efficiency," *arXiv preprint arXiv:2508.09023*, 2025.

[59] V. Giannakouris and I. Trummer, "Dbg-pt: A large language model assisted query performance regression debugger," *Proceedings of the VLDB Endowment*, vol. 17, no. 12, pp. 4337–4340, 2024.

[60] S. Chen, J. Fan, B. Wu, N. Tang, C. Deng, P. Wang, Y. Li, J. Tan, F. Li, J. Zhou *et al.*, "Automatic database configuration debugging using retrieval-augmented language models," *Proceedings of the ACM on Management of Data*, vol. 3, no. 1, pp. 1–27, 2025.

[61] A. Narayan, I. Chami, L. Orr, and C. Ré, "Can foundation models wrangle your data?" *Proceedings of the VLDB Endowment*, vol. 16, no. 4, pp. 738–746, 2022.

[62] Z. A. Naeem, M. S. Ahmad, M. Eltabakh, M. Ouzzani, and N. Tang, "Retclean: Retrieval-based data cleaning using llms and data lakes," *Proc. VLDB Endow.*, vol. 17, no. 12, p. 4421–4424, Aug. 2024. [Online]. Available: https://doi.org/10.14778/3685800.3685890

[63] C. Yang, Y. Luo, C. Cui, J. Fan, C. Chai, and N. Tang, "Data imputation with limited data redundancy using data lakes," *Proc. VLDB Endow.*, vol. 18, no. 10, p. 3354–3367, Sep. 2025. [Online]. Available: https://doi.org/10.14778/3748191.3748200

[64] Y. Qian, Y. He, R. Zhu, J. Huang, Z. Ma, H. Wang, Y. Wang, X. Sun, D. Lian, B. Ding *et al.*, "Unidm: A unified framework for data manipulation with large language models," *Proceedings of Machine Learning and Systems*, vol. 6, pp. 465–482, 2024.

[65] F. Biester, M. Abdelaal, and D. D. Gaudio, "Llmclean: Context-aware tabular data cleaning via llm-generated ofds," 2024. [Online]. Available: https://arxiv.org/abs/2404.18681

[66] P. Li, Y. He, D. Yashar, W. Cui, S. Ge, H. Zhang, D. Rifinski Fainman, D. Zhang, and S. Chaudhuri, "Table-gpt: Table fine-tuned gpt for diverse table tasks," *Proceedings of the ACM on Management of Data*, vol. 2, no. 3, pp. 1–28, 2024.

[67] M. Fan, X. Han, J. Fan, C. Chai, N. Tang, G. Li, and X. Du, "Cost-effective in-context learning for entity resolution: A design space exploration," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 3696–3709.

[68] H. Zhang, Y. Dong, C. Xiao, and M. Oyamada, "Jellyfish: Instruction-tuning local large language models for data preprocessing," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024, pp. 8754–8782.

[69] B. Feuer, Y. Liu, C. Hegde, and J. Freire, "Archetype: A novel framework for open-source column type annotation using large language models," *Proceedings of the VLDB Endowment*, vol. 17, no. 9, pp. 2279–2292, 2024.

[70] L. L. Wei, G. Xiao, and M. Balazinska, "Racoon: an llm-based framework for retrieval-augmented column type annotation with a knowledge graph," *arXiv preprint arXiv:2409.14556*, 2024.

[71] Z. Huang and E. Wu, "Cocoon: Semantic table profiling using large language models," in *Proceedings of the 2024 Workshop on Human-In-the-Loop Data Analytics*, 2024, pp. 1–7.

[72] H. Zhang, Y. Liu, A. Santos, J. Freire *et al.*, "Autoddg: Automated dataset description generation using large language models," *arXiv preprint arXiv:2502.01050*, 2025.

[73] M. I. L. Balaka, D. Alexander, Q. Wang, Y. Gong, A. Krisnadhi, and R. Castro Fernandez, "Pneuma: Leveraging llms for tabular data representation and retrieval in an end-to-end system," *Proceedings of the ACM on Management of Data*, vol. 3, no. 3, pp. 1–28, 2025.

[74] K. Korini and C. Bizer, "Evaluating knowledge generation and self-refinement strategies for llm-based column type annotation," *arXiv preprint arXiv:2503.02718*, 2025.

[75] T. Cai, S. Sheen, and A. Doan, "Columbo: Expanding abbreviated column names for tabular data using large language models," *arXiv preprint arXiv:2508.09403*, 2025.

[76] Y. Ye, B. Hui, M. Yang, B. Li, F. Huang, and Y. Li, "Large Language Models are Versatile Decomposers: Decomposing Evidence and Questions for Table-based Reasoning," in *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*. Taipei Taiwan: ACM, Jul. 2023, pp. 174–184.

[77] Z. Cheng, T. Xie, P. Shi, C. Li, R. Nadkarni, Y. Hu, C. Xiong, D. Radev, M. Ostendorf, L. Zettlemoyer, N. A. Smith, and T. Yu, "Binding Language Models in Symbolic Languages," Mar. 2023.

[78] Y. Sui, M. Zhou, M. Zhou, S. Han, and D. Zhang, "Table Meets LLM: Can Large Language Models Understand Structured Table Data? A Benchmark and Empirical Study," Jul. 2024.

[79] T. Zhang, X. Yue, Y. Li, and H. Sun, "TableLlama: Towards Open Large Generalist Models for Tables ," Apr. 2024.

[80] M. Pourreza and D. Rafiei, "DIN-SQL: decomposed in-context learning of text-to-sql with self-correction," in *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., 2023. [Online]. Available: http://papers.nips.cc/paper_files/paper/2023/hash/72223cc66f63ca1aa59edaec1b3670e6-Abstract-Conference.html

[81] D. Gao, H. Wang, Y. Li, X. Sun, Y. Qian, B. Ding, and J. Zhou, "Text-to-sql empowered by large language models: A benchmark evaluation," *Proc. VLDB Endow.*, vol. 17, no. 5, pp. 1132–1145, 2024. [Online]. Available: https://www.vldb.org/pvldb/vol17/p1132-gao.pdf

[82] H. Zhang, R. Cao, L. Chen, H. Xu, and K. Yu, "ACT-SQL: in-context learning for text-to-sql with automatically-generated chain-of-thought," in *Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, December 6-10, 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Association for Computational Linguistics, 2023, pp. 3501–3532. [Online]. Available: https://doi.org/10.18653/v1/2023.findings-emnlp.227

[83] B. Li, Y. Luo, C. Chai, G. Li, and N. Tang, "The dawn of natural language to SQL: are we fully ready? [experiment, analysis & benchmark]," *Proc. VLDB Endow.*, vol. 17, no. 11, pp. 3318–3331, 2024. [Online]. Available: https://www.vldb.org/pvldb/vol17/p3318-luo.pdf

[84] P. Maddigan and T. Susnjak, "Chat2VIS: Generating Data Visualizations via Natural Language Using ChatGPT, Codex and GPT-3 Large Language Models," *IEEE Access*, vol. 11, pp. 45 181–45 193, oct 2023.

[85] S. Sah, R. Mitra, A. Narechania, A. Endert, J. Stasko, and W. Dou, "Generating analytic specifications for data visualization from natural language queries using large language models," 2024. [Online]. Available: https://arxiv.org/abs/2408.13391

[86] S. Li, X. Chen, Y. Song, Y. Song, C. J. Zhang, F. Hao, and L. Chen, "prompt4vis: prompting large language models with example mining for tabular data visualization," *The VLDB Journal*, vol. 34, no. 4, p. 38, jul 2025. [Online]. Available: https://ieeexplore.ieee.org/document/9617561/https://link.springer.com/10.1007/s00778-025-00912-0

[87] T. Luo, C. Huang, L. Shen, B. Li, S. Shen, W. Zeng, N. Tang, and Y. Luo, "nvbench 2.0: Resolving ambiguity in text-to-visualization through stepwise reasoning," 2025. [Online]. Available: https://arxiv.org/abs/2503.12880

[88] Q. Zhao, R. Wang, Y. Cen, D. Zha, S. Tan, Y. Dong, and J. Tang, "LongRAG: A dual-perspective retrieval-augmented generation paradigm for long-context question answering," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 22 600–22 632. [Online]. Available: https://aclanthology.org/2024.emnlp-main.1259/

[89] P. Sarthi, S. Abdullah, A. Tuli, S. Khanna, A. Goldie, and C. D. Manning, "Raptor: Recursive abstractive processing for tree-organized retrieval," in *International Conference on Learning Representations (ICLR)*, 2024.

[90] J. Saad-Falcon, J. Barrow, A. Siu, A. Nenkova, S. Yoon, R. A. Rossi, and F. Dernoncourt, "PDFTriage: Question answering over long, structured documents," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, F. Dernoncourt, D. Preoţiuc-Pietro, and A. Shimorina, Eds.

Miami, Florida, US: Association for Computational Linguistics, Nov. 2024, pp. 153–169. [Online]. Available: https://aclanthology.org/2024.emnlp-industry.13/

[91] X. Ma, S.-C. Lin, M. Li, W. Chen, and J. Lin, "Unifying multimodal retrieval via document screenshot embedding," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 6492–6505. [Online]. Available: https://aclanthology.org/2024.emnlp-main.373/

[92] M. Suri, P. Mathur, F. Dernoncourt, K. Goswami, R. A. Rossi, and D. Manocha, "VisDoM: Multi-document QA with visually rich elements using multimodal retrieval-augmented generation," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, L. Chiruzzo, A. Ritter, and L. Wang, Eds. Albuquerque, New Mexico: Association for Computational Linguistics, Apr. 2025, pp. 6088–6109. [Online]. Available: https://aclanthology.org/2025.naacl-long.310/

[93] Y. Duan, Z. Chen, Y. Hu, W. Wang, S. Ye, B. Shi, L. Lu, Q. Hou, T. Lu, H. Li, J. Dai, and W. Wang, "Docopilot: Improving multimodal models for document-level understanding," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2025, pp. 4026–4037.

[94] N. Sultanum and A. Srinivasan, "Datatales: Investigating the use of large language models for authoring data-driven articles," in *2023 IEEE Visualization and Visual Analytics (VIS)*, 2023, pp. 231–235.

[95] K. Choe, C. Lee, S. Lee, J. Song, A. Cho, N. W. Kim, and J. Seo, "Enhancing data literacy on-demand: Llms as guides for novices in chart interpretation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 31, no. 9, pp. 4712–4727, 2025.

[96] L. Cecchi and P. Babkin, "ReportGPT: Human-in-the-loop verifiable table-to-text generation," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, F. Dernoncourt, D. Preoţiuc-Pietro, and A. Shimorina, Eds. Association for Computational Linguistics, Nov. 2024, pp. 529–537.

[97] J. Chen, J. Wu, J. Guo, V. Mohanty, X. Li, J. P. Ono, W. He, L. Ren, and D. Liu, "Interchat: Enhancing generative visual analytics using multimodal interactions," *Computer Graphics Forum*, vol. 44, no. 3, p. e70112, 2025.

[98] L. Wang, Z. Wang, S. Xiao, L. Liu, F. Tsung, and W. Zeng, "Vizta: Enhancing comprehension of distributional visualization with visual-lexical fused conversational interface," *Computer Graphics Forum*, vol. 44, no. 3, p. e70110, 2025.

[99] M. Suri, P. Mathur, N. Lipka, F. Dernoncourt, R. A. Rossi, and D. Manocha, "ChartLens: Fine-grained visual attribution in charts," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Association for Computational Linguistics, Jul. 2025, pp. 22 447–22 462.

[100] A. Narayan, I. Chami, L. Orr, and C. Ré, "Can foundation models wrangle your data?" *Proc. VLDB Endow.*, vol. 16, no. 4, p. 738–746, Dec. 2022. [Online]. Available: https://doi.org/10.14778/3574245.3574258

[101] A. Doan, A. Halevy, and Z. Ives, *Principles of data integration*. Elsevier, 2012.

[102] P. Christen, "Data matching: concepts and techniques for record linkage, entity resolution, and duplicate detection. 2012."

[103] M. Lenzerini, "Data integration: A theoretical perspective," in *Proceedings of the twenty-first ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, 2002, pp. 233–246.

[104] J. Zhang, B. Shin, J. D. Choi, and J. C. Ho, "Smat: An attention-based deep learning solution to the automation of schema matching," in *European Conference on Advances in Databases and Information Systems*. Springer, 2021, pp. 260–274.

[105] N. Seedat and M. van der Schaar, "Matchmaker: Self-improving large language model programs for schema matching," *arXiv preprint arXiv:2410.24105*, 2024.

[106] E. Sheetrit, M. Brief, M. Mishaeli, and O. Elisha, "Rematch: Retrieval enhanced schema matching with llms," *arXiv preprint arXiv:2403.01567*, 2024.

[107] Y. Liu, E. Pena, A. Santos, E. Wu, and J. Freire, "Magneto: Combining small and large language models for schema matching," *arXiv preprint arXiv:2412.08194*, 2024.

[108] R. Peeters, A. Steiner, and C. Bizer, "Entity matching using large language models," 2025.

[109] A. Steiner, R. Peeters, and C. Bizer, "Fine-tuning large language models for entity matching," in *2025 IEEE 41st International Conference on Data Engineering Workshops (ICDEW)*. IEEE, 2025, pp. 9–17.

[110] Z. Lu, T. L. van der Plas, P. Rashidi, W. D. Kissling, and I. N. Athanasiadis, "Flexible metadata harvesting for ecology using large language models," *arXiv preprint arXiv:2508.20115*, 2025.

[111] Z. Alyafeai, M. S. Al-Shaibani, and B. Ghanem, "Mole: Metadata extraction and validation in scientific papers using llms," *arXiv preprint arXiv:2505.19800*, 2025.

[112] P. Thorat, A. Qidwai, A. Dhar, A. Chakraborty, A. Eswaran, H. Patel, and P. Jayachandran, "Llm-aided customizable profiling of code data based on programming language concepts," *arXiv preprint arXiv:2503.15571*, 2025.

[113] K. Korini and C. Bizer, "Column type annotation using chatgpt," *arXiv preprint arXiv:2306.00745*, 2023.

[114] V. Shkapenyuk, D. Srivastava, T. Johnson, and P. Ghane, "Automatic metadata extraction for text-to-sql," *arXiv preprint arXiv:2505.19988*, 2025.

[115] W. Chen, H. Zha, Z. Chen, W. Xiong, H. Wang, and W. Y. Wang, "HybridQA: A dataset of multi-hop question answering over tabular and textual data," in *Findings of the Association for Computational Linguistics: EMNLP 2020*, T. Cohn, Y. He, and Y. Liu, Eds. Online: Association for Computational Linguistics, Nov. 2020, pp. 1026–1036. [Online]. Available: https://aclanthology.org/2020.findings-emnlp.91/

[116] Y. Zhu, S. Du, B. Li, Y. Luo, and N. Tang, "Are large language models good statisticians?" *Advances in Neural Information Processing Systems*, vol. 37, pp. 62 697–62 731, 2024.

[117] Z. Tang, B. Niu, X. Zhou, B. Li, W. Zhou, J. Wang, G. Li, X. Zhang, and F. Wu, "St-raptor: Llm-powered semi-structured table question answering," *Proc. ACM Manag. Data*, 2026.

[118] X. Liu, S. Shen, B. Li, N. Tang, and Y. Luo, "Nl2sql-bugs: A benchmark for detecting semantic errors in nl2sql translation," in *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V. 2*, 2025, pp. 5662–5673.

[119] L. Shen, E. Shen, Y. Luo, X. Yang, X. Hu, X. Zhang, Z. Tai, and J. Wang, "Towards Natural Language Interfaces for Data Visualization: A Survey," *IEEE Transactions on Visualization and Computer Graphics*, vol. 29, no. 6, pp. 3121–3144, 2023.

[120] Y. Luo, N. Tang, G. Li, C. Chai, W. Li, and X. Qin, "Synthesizing natural language to visualization (nl2vis) benchmarks from nl2sql benchmarks," in *Proceedings of the 2021 International Conference on Management of Data*, ser. SIGMOD '21. New York, NY, USA: Association for Computing Machinery, 2021, p. 1235–1247. [Online]. Available: https://doi.org/10.1145/3448016.3457261

[121] Y. Luo, N. Tang, G. Li, J. Tang, C. Chai, and X. Qin, "Natural language to visualization by neural machine translation," *IEEE Transactions on Visualization and Computer Graphics*, vol. 28, no. 1, p. 217–226, jan 2022. [Online]. Available: https://doi.org/10.1109/TVCG.2021.3114848

[122] T. B. Brown, B. Mann, N. Ryder, M. Subbiah, J. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, S. Agarwal, A. Herbert-Voss, G. Krueger, T. Henighan, R. Child, A. Ramesh, D. M. Ziegler, J. Wu, C. Winter, C. Hesse, M. Chen, E. Sigler, M. Litwin, S. Gray, B. Chess, J. Clark, C. Berner, S. McCandlish, A. Radford, I. Sutskever, and D. Amodei, "Language models are few-shot learners," in *Proceedings of the 34th International Conference on Neural Information Processing Systems*, ser. NIPS '20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[123] "A comprehensive survey on long context language modeling," 2025. [Online]. Available: https://arxiv.org/abs/2503.17407

[124] D. Agrawal, S. Gao, and M. Gajek, "Can't remember details in long documents? you need some r&r," *arXiv preprint arXiv:2403.05004*, 2024.

[125] Y. Ding, K. Ren, J. Huang, S. Luo, and S. C. Han, "Pdf-mvqa: A dataset for multimodal information retrieval in pdf-based visual question answering," 2024. [Online]. Available: https://arxiv.org/abs/2404.12720

[126] Z. Gong, Y. Huang, and C. Mai, "Mmrag-docqa: A multi-modal retrieval-augmented generation method for document question-answering with hierarchical index and multi-granularity retrieval," *arXiv preprint arXiv:2508.00579*, 2025.

[127] S. Han, P. Xia, R. Zhang, T. Sun, Y. Li, H. Zhu, and H. Yao, "Mdocagent: A multi-modal multi-agent framework for document understanding," *arXiv preprint arXiv:2503.13964*, 2025.

[128] Y. Li, H. Li, Z. Pu, J. Zhang, X. Zhang, T. Ji, L. Sun, C. Li, and H. Chen, "Is large language model good at database knob tuning? a comprehensive experimental evaluation," *arXiv preprint arXiv:2408.02213*, 2024.

[129] X. Zhao, H. Li, J. Zhang, X. Huang, T. Zhang, J. Chen, R. Shi, C. Li, and H. Chen, "Llmidxadvis: Resource-efficient index advisor utilizing large language model," *arXiv preprint arXiv:2503.07884*, 2025.

[130] W. Sun, Z. Pan, Z. Hu, Y. Liu, C. Yang, R. Zhang, and X. Zhou, "Rabbit: Retrieval-augmented generation enables better automatic database knob tuning," in *2025 IEEE 41st International Conference on Data Engineering (ICDE)*. IEEE, 2025, pp. 3807–3820.

[131] Z. Yan, R. Xi, and M. Hou, "Mctuner: Spatial decomposition-enhanced database tuning via llm-guided exploration," *arXiv preprint arXiv:2509.06298*, 2025.

[132] H. Liu, Q. Zhang, R. Marcus, and I. Sabek, "Serag: Self-evolving rag system for query optimization," 2025.

[133] Y. Song, H. Yan, J. Lao, Y. Wang, Y. Li, Y. Zhou, J. Wang, and M. Tang, "Quite: A query rewrite system beyond rules with llm agents," *arXiv preprint arXiv:2506.07675*, 2025.

[134] Z. Sun, X. Zhou, G. Li, X. Yu, J. Feng, and Z. Yong, "R-bot: An llm-based query rewrite system," *Proceedings of the VLDB Endowment*, vol. 18, no. 12, pp. 5031–5044, 2025.

[135] V. Y. Singh, K. Vaidya, V. B. Kumar, S. Khosla, B. Narayanaswamy, R. Gangadharaiah, and T. Kraska, "Panda: Performance debugging for databases using llm agents," in *CIDR*, 2024.

[136] X. Zhou, G. Li, Z. Sun, Z. Liu, W. Chen, J. Wu, J. Liu, R. Feng, and G. Zeng, "D-bot: Database diagnosis system using large language models," *Proceedings of the VLDB Endowment*, vol. 17, no. 10, pp. 2514–2527, 2024.

[137] W. Zhou, P. Sun, X. Zhou, Q. Zang, J. Xu, T. Zhang, G. Li, and F. Wu, "Dbaiops: A reasoning llm-enhanced database operation and maintenance system using knowledge graphs," *arXiv preprint arXiv:2508.01136*, 2025.

[138] Y. Zhang, C. Li, Y. Luo, and N. Tang, "Sketchfill: Sketch-guided code generation for imputing derived missing values," *arXiv preprint arXiv:2412.19113*, 2024.

[139] W. Ni, K. Zhang, X. Miao, X. Zhao, Y. Wu, and J. Yin, "Iterclean: An iterative data cleaning framework with large language models," in *Proceedings of the ACM Turing Award Celebration Conference - China 2024*, ser. ACM-TURC '24. New York, NY, USA: Association for Computing Machinery, 2024, p. 100–105. [Online]. Available: https://doi.org/10.1145/3674399.3674436

[140] L. Li, L. Fang, B. Ludäscher, and V. I. Torvik, "Autodcworkflow: Llm-based data cleaning workflow auto-generation and benchmark," in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 2025. [Online]. Available: https://arxiv.org/abs/2412.06724

[141] D. Qi, Z. Miao, and J. Wang, "Cleanagent: Automating data standardization with llm-based agents," 2025. [Online]. Available: https://arxiv.org/abs/2403.08291

[142] T. Bendinelli, A. Dox, and C. Holz, "Exploring LLM agents for cleaning tabular machine learning datasets," in *ICLR 2025 Workshop on Foundation Models in the Wild*, 2025. [Online]. Available: https://openreview.net/forum?id=RXnQPYSoun

[143] C. Li, C. Yang, Y. Luo, J. Fan, and N. Tang, "Weak-to-strong prompts with lightweight-to-powerful llms for high-accuracy, low-cost, and explainable data transformation," *Proceedings of the VLDB Endowment*, vol. 18, no. 8, pp. 2371–2384, 2025.

[144] Z. Qiang, W. Wang, and K. Taylor, "Agent-om: Leveraging llm agents for ontology matching," *Proceedings of the VLDB Endowment*, vol. 18, no. 3, pp. 516–529, 2024.

[145] M. Taboada, D. Martinez, M. Arideh, and R. Mosquera, "Ontology matching with large language models and prioritized depth-first search," *Information Fusion*, p. 103254, 2025.

[146] T. Wang, X. Chen, H. Lin, X. Chen, X. Han, L. Sun, H. Wang, and Z. Zeng, "Match, compare, or select? an investigation of large language models for entity matching," in *Proceedings of the 31st International Conference on Computational Linguistics*, 2025, pp. 96–109.

[147] M. Kayali, A. Lykov, I. Fountalis, N. Vasiloglou, D. Olteanu, and D. Suciu, "Chorus: Foundation models for unified data discovery and exploration," *Proceedings of the VLDB Endowment*, vol. 17, no. 8, pp. 2104–2114, 2024.

[148] B. P. Majumder, H. Surana, D. Agarwal, S. Hazra, A. Sabharwal, and P. Clark, "Data-driven discovery with large generative models," *arXiv preprint arXiv:2402.13610*, 2024.

[149] Y. Lou, C. Lei, X. Qin, Z. Wang, C. Faloutsos, R. Anubhai, and H. Rangwala, "Datalore: Can a large language model find all lost scrolls in a data repository?" in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 5170–5176.

[150] Q. An, C. Ying, Y. Zhu, Y. Xu, M. Zhang, and J. Wang, "Ledd: large language model-empowered data discovery in data lakes," *arXiv preprint arXiv:2502.15182*, 2025.

[151] S. Al-Rubaye and A. Al-Khafaji, "Automated metadata generation using large language models: A gpt-4 case study for enterprise data profiling," *Scientific Journal for Engineering and Computer Science*, vol. 221, pp. 769–775, 2025.

[152] Y. Li, K. Wang, Z. Sun *et al.*, "Automatic database description generation for text-to-sql," *arXiv preprint arXiv:2502.20657*, 2025.

[153] J. Wang, Y. Feng, C. Shen, S. Rahman, and E. Kandogan, "Towards operationalizing heterogeneous data discovery," *arXiv preprint arXiv:2504.02059*, 2025.

[154] J. Jiang, K. Zhou, Z. Dong, K. Ye, W. X. Zhao, and J.-R. Wen, "StructGPT: A General Framework for Large Language Model to Reason over Structured Data," Oct. 2023.

[155] Y. Zhang, J. Henkel, A. Floratou, J. Cahoon, S. Deep, and J. M. Patel, "ReAcTable: Enhancing ReAct for Table Question Answering," *Proceedings of the VLDB Endowment*, vol. 17, no. 8, pp. 1981–1994, Apr. 2024.

[156] Z. Wang, H. Zhang, C.-L. Li, J. M. Eisenschlos, V. Perot, Z. Wang, L. Miculicich, Y. Fujii, J. Shang, C.-Y. Lee, and T. Pfister, "Chain-of-Table: Evolving tables in the reasoning chain for table understanding," in *The Twelfth International Conference on Learning Representations*, 2024. [Online]. Available: https://openreview.net/forum?id=4L0xnS4GQM

[157] J.-P. Zhu, P. Cai, K. Xu, L. Li, Y. Sun, S. Zhou, H. Su, L. Tang, and Q. Liu, "AutoTQA: Towards Autonomous Tabular Question Answering through Multi-Agent Large Language Models," *Proceedings of the VLDB Endowment*, vol. 17, no. 12, pp. 3920–3933, Aug. 2024.

[158] P. Yu, G. Chen, and J. Wang, "Table-Critic: A Multi-Agent Framework for Collaborative Criticism and Refinement in Table Reasoning," in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 17 432–17 451.

[159] B. Wang, C. Ren, J. Yang, X. Liang, J. Bai, L. Chai, Z. Yan, Q.-W. Zhang, D. Yin, X. Sun, and Z. Li, "MAC-SQL: A multi-agent collaborative framework for text-to-SQL," in *Proceedings of the 31st International Conference on Computational Linguistics*, O. Rambow, L. Wanner, M. Apidianaki, H. Al-Khalifa, B. D. Eugenio, and S. Schockaert, Eds. Abu Dhabi, UAE: Association for Computational Linguistics, Jan. 2025, pp. 540–557. [Online]. Available: https://aclanthology.org/2025.coling-main.36/

[160] J. Lian, X. Liu, Y. Shao, Y. Dong, M. Wang, Z. Wei, T. Wan, M. Dong, and H. Yan, "Chatbi: Towards natural language to complex business intelligence sql," *arXiv preprint arXiv:2405.00527*, 2024.

[161] M. Pourreza, H. Li, R. Sun, Y. Chung, S. Talaei, G. T. Kakkar, Y. Gan, A. Saberi, F. Ozcan, and S. O. Arik, "CHASE-SQL: Multi-path reasoning and preference optimized candidate selection in text-to-SQL," in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: https://openreview.net/forum?id=CvGqMD5OtX

[162] X. Xie, G. Xu, L. Zhao, and R. Guo, "Opensearch-sql: Enhancing text-to-sql with dynamic few-shot and consistency alignment," *Proc. ACM Manag. Data*, vol. 3, no. 3, Jun. 2025. [Online]. Available: https://doi.org/10.1145/3725331

[163] M. Deng, A. Ramachandran, C. Xu, L. Hu, Z. Yao, A. Datta, and H. Zhang, "ReFoRCE: A text-to-sql agent with self-refinement, format restriction, and column exploration," *arXiv preprint arXiv:2502.00675*, 2025.

[164] B. Li, C. Chen, Z. Xue, Y. Mei, and Y. Luo, "DeepEye-SQL: A software-engineering-inspired text-to-sql framework," *arXiv preprint arXiv:2510.17586*, 2025.

[165] Z. Yang, Z. Zhou, S. Wang, X. Cong, X. Han, Y. Yan, Z. Liu, Z. Tan, P. Liu, D. Yu, Z. Liu, X. Shi, and M. Sun, "MatPlotAgent: Method and Evaluation for LLM-Based Agentic Scientific Data Visualization," in *Findings of the Association for Computational Linguistics ACL 2024*. Stroudsburg, PA, USA: Association for Computational Linguistics, aug 2024, pp. 11 789–11 804.

[166] F. Pesaran Zadeh, J. Kim, J.-H. Kim, and G. Kim, "Text2Chart31: Instruction tuning for chart generation with automatic feedback," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 11 459–11 480. [Online]. Available: https://aclanthology.org/2024.emnlp-main.640/

[167] Z. Shuai, B. Li, S. Yan, Y. Luo, and W. Yang, "Deepvis: Bridging natural language and data visualization through step-wise reasoning," 2025. [Online]. Available: https://arxiv.org/abs/2508.01700

[168] Z. Jiang, F. Xu, L. Gao, Z. Sun, Q. Liu, J. Dwivedi-Yu, Y. Yang, J. Callan, and G. Neubig, "Active retrieval augmented generation," in *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 7969–7992. [Online]. Available: https://aclanthology.org/2023.emnlp-main.495/

[169] K.-H. Lee, X. Chen, H. Furuta, J. Canny, and I. Fischer, "A human-inspired reading agent with gist memory of very long contexts," *arXiv preprint arXiv:2402.09727*, 2024.

[170] S. Li, Y. He, H. Guo, X. Bu, G. Bai, J. Liu, J. Liu, X. Qu, Y. Li, W. Ouyang *et al.*, "Graphreader: Building graph-based agent to enhance long-context abilities of large language models," *arXiv preprint arXiv:2406.14550*, 2024.

[171] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-rag: Self-reflective retrieval augmented generation," in *NeurIPS 2023 workshop on instruction tuning and instruction following*, 2023.

[172] Y. Wang, R. Ren, J. Li, X. Zhao, J. Liu, and J.-R. Wen, "REAR: A relevance-aware retrieval-augmented framework for open-domain question answering," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds. Miami, Florida, USA: Association for Computational Linguistics, Nov. 2024, pp. 5613–5626. [Online]. Available: https://aclanthology.org/2024.emnlp-main.321/

[173] Z. Sun, C. Chai, Q. Deng, K. Jin, X. Guo, H. Han, Y. Yuan, G. Wang, and L. Cao, "Quest: Query optimization in unstructured document analysis," *Proc. VLDB Endow.*, vol. 18, no. 11, p. 4560–4573, Sep. 2025. [Online]. Available: https://doi.org/10.14778/3749646.3749713

[174] C. Chai, J. Li, Y. Deng, Y. Zhong, Y. Yuan, G. Wang, and L. Cao, "Doctopus: Budget-aware structural table extraction from unstructured documents," *Proceedings of the VLDB Endowment*, vol. 18, no. 11, pp. 3695–3707, 2025.

[175] X. Yu, Z. Chen, Y. Zhang, S. Lu, R. Shen, J. Zhang, X. Hu, Y. Fu, and S. Yan, "Visual document understanding and question answering: A multi-agent collaboration framework with test-time scaling," *arXiv preprint arXiv:2508.03404*, 2025.

[176] Z. Zhang, Z. Liang, Y. Wu, T. Lin, Y. Luo, and N. Tang, "Datapuzzle: Breaking free from the hallucinated promise of llms in data analysis," *arXiv preprint arXiv:2504.10036*, 2025.

[177] M. S. Islam, M. T. R. Laskar, M. R. Parvez, E. Hoque, and S. Joty, "DataNarrative: Automated data-driven storytelling with visualizations and texts," in *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Y. Al-Onaizan, M. Bansal, and Y.-N. Chen, Eds., Nov. 2024, pp. 19 253–19 286.

[178] L. Shen, H. Li, Y. Wang, and H. Qu, "From data to story: Towards automatic animated data video creation with llm-based multi-agent systems," 2024. [Online]. Available: https://arxiv.org/abs/2408.03876

[179] Y. Zhao, J. Wang, L. Xiang, X. Zhang, Z. Guo, C. Turkay, Y. Zhang, and S. Chen, "Lightva: Lightweight visual analytics with llm agent-based task planning and execution," *IEEE Transactions on Visualization and Computer Graphics*, vol. 31, no. 9, p. 6162–6177, Sep. 2025. [Online]. Available: http://dx.doi.org/10.1109/TVCG.2024.3496112

[180] Z. Yang, B. Pan, H. Wang, Y. Wang, X. Liu, M. Zhu, B. Zhang, and W. Chen, "Multimodal deepresearcher: Generating text-chart interleaved reports from scratch with agentic framework," 2025. [Online]. Available: https://arxiv.org/abs/2506.02454

[181] W. Xu, Y. Mao, X. Zhang, C. Zhang, X. Dong, M. Zhang, and Y. Gao, "Dagent: A relational database-driven data analysis report generation agent," 2025. [Online]. Available: https://arxiv.org/abs/2503.13269

[182] F. Wang, B. Wang, X. Shu, Z. Liu, Z. Shao, C. Liu, and S. Chen, "Chartinsighter: An approach for mitigating hallucination in time-series chart summary generation with a benchmark dataset," 2025. [Online]. Available: https://arxiv.org/abs/2501.09349

[183] Y. Zhao, X. Shu, L. Fan, L. Gao, Y. Zhang, and S. Chen, "Proactiveva: Proactive visual analytics with llm-based ui agent," 2025. [Online]. Available: https://arxiv.org/abs/2507.18165

[184] D. Jia, A. Irger, L. Besancon, O. Strnad, D. Luo, J. Bjorklund, A. Ynnerman, and I. Viola, "Voice: Visual oracle for interaction, conversation, and explanation," 2024. [Online]. Available: https://arxiv.org/abs/2304.04083

[185] K. Ai, K. Tang, and C. Wang, "Nli4volvis: Natural language interaction for volume visualization via llm multi-agents and editable 3d gaussian splatting," 2025. [Online]. Available: https://arxiv.org/abs/2507.12621

[186] H. Ding, Z. Wang, Y. Yang, D. Zhang, Z. Xu, H. Chen, R. Piskac, and J. Li, "Proving query equivalence using linear integer arithmetic," *Proc. ACM Manag. Data*, vol. 1, no. 4, Dec 2023.

[187] "DBdoctor — dbdoctor.cn," https://www.dbdoctor.cn/.

[188] "OpenRefine — openrefine.org," https://openrefine.org/.

[189] Z. Chen, L. Cao, S. Madden, T. Kraska, Z. Shang, J. Fan, N. Tang, Z. Gu, C. Liu, and M. Cafarella, "Seed: Domain-specific data curation with large language models," *arXiv preprint arXiv:2310.00749*, 2023.

[190] Z. Huang, J. Guo, and E. Wu, "Transform table to database using large language models," *Proceedings of the VLDB Endowment. ISSN*, vol. 2150, p. 8097, 2024.

[191] C. Ma, S. Chakrabarti, A. Khan, and B. Molnár, "Knowledge graph-based retrieval-augmented generation for schema matching," *arXiv preprint arXiv:2501.08686*, 2025.

[192] Y. Xu, H. Li, K. Chen, and L. Shou, "Kcmf: A knowledge-compliant framework for schema and entity matching with fine-tuning-free llms," *arXiv preprint arXiv:2410.12480*, 2024.

[193] L. Cao and H. Liu, "Tablemaster: A recipe to advance table understanding with language models," *arXiv preprint arXiv:2501.19378*, 2025.

[194] H. Sun, Y. Zhuang, L. Kong, B. Dai, and C. Zhang, "Adaplanner: Adaptive planning from feedback with language models," 2023. [Online]. Available: https://arxiv.org/abs/2305.16653

[195] T. Masterman, S. Besen, M. Sawtell, and A. Chao, "The landscape of emerging ai agent architectures for reasoning, planning, and tool calling: A survey," 2024. [Online]. Available: https://arxiv.org/abs/2404.11584

[196] Y. Zhu, R. JIANG, B. Li, N. Tang, and Y. Luo, "EllieSQL: Cost-efficient text-to-SQL with complexity-aware routing," in *Second Conference on Language Modeling*, 2025. [Online]. Available: https://openreview.net/forum?id=8OqGNXKwo8

[197] X. Yang, K. Sun, H. Xin, Y. Sun, N. Bhalla, X. Chen, S. Choudhary, R. Gui, Z. Jiang, Z. Jiang *et al.*, "Crag-comprehensive rag benchmark," *Advances in Neural Information Processing Systems*, vol. 37, pp. 10 470–10 490, 2024.

[198] R. Nakano, J. Hilton, S. Balaji, J. Wu, L. Ouyang, C. Kim, C. Hesse, S. Jain, V. Kosaraju, W. Saunders *et al.*, "Webgpt: Browser-assisted question-answering with human feedback," *arXiv preprint arXiv:2112.09332*, 2021.

[199] J. Sun, G. Li, P. Zhou, Y. Ma, J. Xu, and Y. Li, "Agenticdata: An agentic data analytics system for heterogeneous data," *CoRR*, vol. abs/2508.05002, 2025. [Online]. Available: https://doi.org/10.48550/arXiv.2508.05002

[200] S. Zhang, J. Fan, M. Fan, G. Li, and X. Du, "Deepanalyze: Agentic large language models for autonomous data science," 2025. [Online]. Available: https://arxiv.org/abs/2510.16872

[201] J. Wang and G. Li, "AOP: Automated and interactive llm pipeline orchestration for answering complex queries." CIDR, 2025.

[202] S. Hong, Y. Lin, B. Liu, B. Liu, B. Wu, C. Zhang, D. Li, J. Chen, J. Zhang, J. Wang, L. Zhang, L. Zhang, M. Yang, M. Zhuge, T. Guo, T. Zhou, W. Tao, R. Tang, X. Lu, X. Zheng, X. Liang, Y. Fei, Y. Cheng, Y. Ni, Z. Gou, Z. Xu, Y. Luo, and C. Wu, "Data interpreter: An LLM agent for data science," in *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, W. Che, J. Nabende, E. Shutova, and M. T. Pilehvar, Eds. Association for Computational Linguistics, 2025, pp. 19 796–19 821. [Online]. Available: https://aclanthology.org/2025.findings-acl.1016/

[203] Agent Team at JDCHO, "Joyagent-jdgenie," 2025. [Online]. Available: https://github.com/jd-opensource/joyagent-jdgenie

[204] Databricks, "Assistant data science agent," 2025. [Online]. Available: https://www.databricks.com/blog/introducing-databricks-assistant-data-science-agent

[205] TabTab AI, "Tabtab," 2025. [Online]. Available: https://tabtabai.com/

[206] ByteDance, "Data agent," 2025. [Online]. Available: https://www.volcengine.com/product/DataAgent

[207] Google, "Bigquery," 2025. [Online]. Available: https://cloud.google.com/blog/products/data-analytics/a-closer-look-at-bigquery-data-engineering-agent

[208] Snowflake, "Cortex agents," 2025. [Online]. Available: https://docs.snowflake.com/en/user-guide/snowflake-cortex/cortex-agents

[209] Xata, "Xata agent," 2025. [Online]. Available: https://xata.io/blog/dba-to-db-agent

[210] J. Jiang, H. Xie, S. Shen, Y. Shen, Z. Zhang, M. Lei, Y. Zheng, Y. Li, C. Li, D. Huang, Y. Wu, W. Zhang, X. Yang, B. Cui, and P. Chen, "Siriusbi: A comprehensive llm-powered solution for data analytics in business intelligence," *Proc. VLDB Endow.*, vol. 18, no. 12, pp. 4860–4873, 2025. [Online]. Available: https://www.vldb.org/pvldb/vol18/p4860-xie.pdf

[211] G. Zhang, H. Geng, X. Yu, Z. Yin, Z. Zhang, Z. Tan, H. Zhou, Z. Li, X. Xue, Y. Li *et al.*, "The landscape of agentic reinforcement learning for llms: A survey," *arXiv preprint arXiv:2509.02547*, 2025.