Towards Autonomous Data Science with LLM Agents: A Survey and Outlook

Anonymous ACL submission

Abstract

Large Language Models (LLMs) are reshaping data science automation by enabling agents to interpret analytical goals, generate executable code, interact with heterogeneous data sources, and coordinate multi-step workflows. However, reliable autonomous data science remains difficult: agents must understand imperfect and multimodal data, make sound analytical decisions, and execute long-horizon workflows under real system constraints. This survey reviews LLM-based data science agents through a perception–reasoning–operation framework. We organize existing work into three complementary perspectives: data-world perception, which enables agents to understand and process heterogeneous data; agentic data reasoning, which supports methodological decision-making and workflow optimization; and system-level operation, which provides the runtime, infrastructure, and governance needed for reliable execution. We further analyze representative systems, compare their design principles and limitations, and discuss open challenges toward robust autonomous data science agents.

1 Introduction

Data science workflows are inherently complex and multi-stage, requiring a tightly integrated pipeline that transforms raw and heterogeneous data into actionable insights. In practice, this process often involves multiple stages (e.g., data cleaning, feature engineering, statistical modeling, and visualization) (Liang et al., 2025; Trirat et al., 2024). Traditionally, these stages are implemented through fragmented toolchains and manually written scripts, which makes the workflow difficult to scale and maintain. Although AutoML systems (He et al., 2021) have reduced human effort in certain components, such as model selection and hyperparameter tuning, their automation scope remains limited and typically relies on predefined pipeline structures.

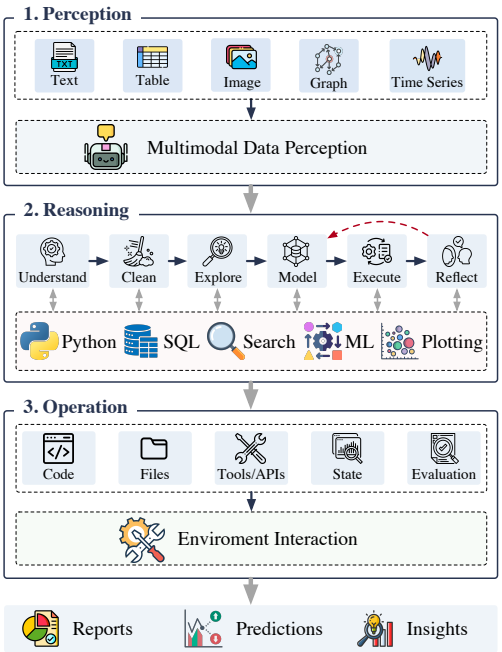

Figure 1: Perception-Reasoning-Operation stack of data science agents.

LLMs have introduced a new paradigm for data science automation (Tang et al., 2025; Yuksekgonul et al., 2024): agents can interpret analytical goals, generate code, and coordinate multi-step workflows through natural-language interaction, program synthesis, and reasoning (Cao et al., 2024). This capability points toward end-to-end autonomous data science, but reliable deployment remains difficult. Agents must perceive heterogeneous data (e.g., tables, graphs, time series, text, and images), make methodological decisions across long-horizon reasoning, modeling, coding, and execution, and operate within complex environments that demand reliable execution, scalable infrastructure, and robust external-system interaction.

Recent surveys on LLM-based and data-science agents (Tang et al., 2025; Zhu et al., 2025) often focus on individual aspects, such as data types, agent capabilities, autonomy levels, or lifecycle stages (Rahman et al., 2025; Sun et al., 2025b;

Chen et al., 2025a). However, data science agents are not defined by a single capability. Their core lies in the coordination among three layers: perceiving data, reasoning over analytical choices, and operating within executable systems, as shown in Figure 1. As summarized in Table 1, existing surveys pay less attention to this cross-layer interaction, especially how perception, reasoning, and operation jointly support heterogeneous, long-horizon workflows. To fill this gap, we organize LLM-based data science agents along three complementary perspectives. *Data-world perception* studies how agents understand and process heterogeneous data. *Agentic data reasoning* covers knowledge grounding, policy learning, and workflow optimization for analytical decision-making. *System-level operation* examines the runtimes, infrastructure, and governance mechanisms needed for reliable execution. This taxonomy highlights not only individual techniques, but also how they work together to support end-to-end autonomous data science.

The major contributions of this paper are threefold: (1) We propose a perception–reasoning–operation framework for LLM-based autonomous data science agents, reframing prior work through coupled agent–data–system interactions, as shown in Figure 2. This unified framework clarifies how agents perceive heterogeneous data, make analytical decisions, and execute long-horizon workflows. (2) We organize representative methods into data-world perception, agentic data reasoning, and system-level operation, as summarized in Figure 3, and compare their support for heterogeneous data handling, multi-step reasoning, workflow orchestration, and reproducible execution. (3) We identify key gaps in agent–data interaction, long-horizon decision-making, and robust execution, and outline future directions toward reliable autonomous data science agents.

2 Data-World Perception

In this section, we introduce data-world perception techniques, providing the perceptual foundation for autonomous data science agents. Before modeling or reasoning, agents actively explore heterogeneous raw data, querying, summarizing, and observing it, to form an internal understanding of its structure, content, and key features. We focus on two central challenges in this process: heterogeneous data perception and imperfect data processing.

2.1 Heterogeneous Data Perception

Heterogeneous data perception equips agents to turn tables, text, images, and other data forms into a unified semantic view for downstream analysis (Fu et al., 2025). Its role is to preserve modality-specific structure while linking descriptions, visualizations, and numeric or categorical records for holistic dataset reasoning. Existing methods fall into three categories: *modality-aware encoding*, which captures native structures and semantic relations; *executable data probing*, which refines understanding through code-based queries and feedback; and *cross-modal data grounding*, which aligns evidence across modalities so that one modality can inform reasoning over another.

2.1.1 Modality-Aware Encoding

A central challenge in heterogeneous data perception is the diversity of data modalities, as tables, text, and images exhibit distinct structural and geometric properties. Early approaches (Wang et al., 2021; Lee et al., 2023) that linearized all inputs into plain text often caused severe information loss, particularly for spatially structured data. Modality-aware encoding addresses this by preserving modality-native structures through specialized encoders prior to cross-modal reasoning. Recent methods explicitly model structural and spatial properties, representing data as structural units (e.g., cells, patches, or layout blocks) with associated row-column indices or 2D coordinates. Structure-aware transformers, such as TUTA (Wang et al., 2021) and Pix2Struct (Lee et al., 2023), incorporate spatial embeddings and structure-aware attention to reason over semantic content and structural relationships, including hierarchical dependencies and cross-region alignments. Masking strategies (Huang et al., 2022) enhance robustness by filtering unreliable or sparse regions during training. While effective for high-fidelity, structure-aware representations, these methods remain passive and lack mechanisms for active interaction or verification, limiting their performance in complex reasoning tasks.

2.1.2 Executable Data Probing

Executable data probing introduces an interactive paradigm in which agents explore data by generating and executing code, overcoming the limitations of passive, purely representational perception (Yao et al., 2022; Gao et al., 2023). Agents iteratively produce executable scripts (e.g., SQL queries or

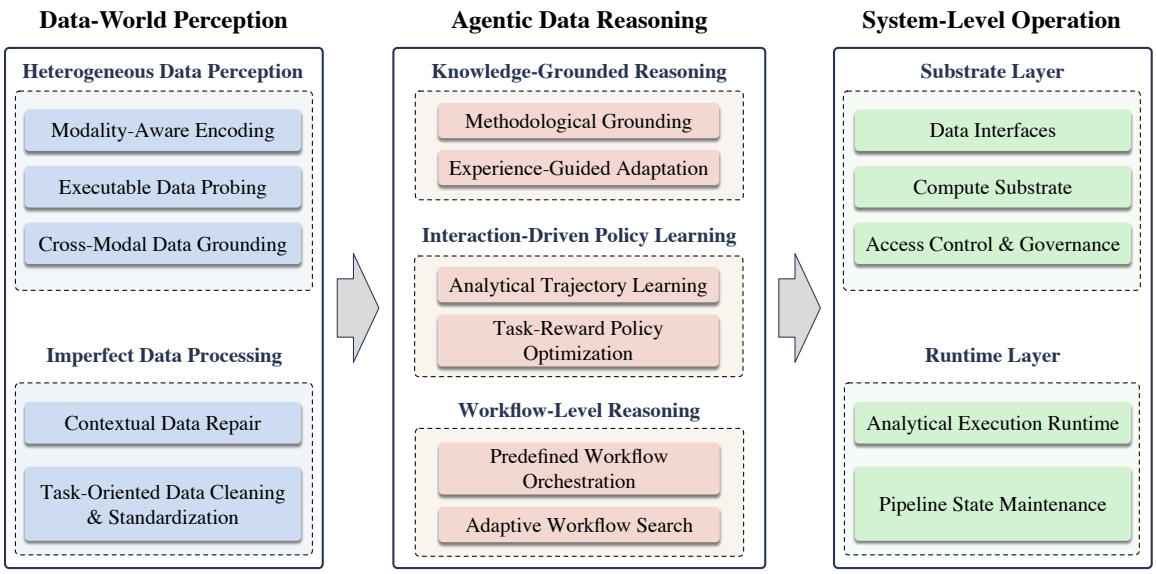

Figure 2: A prospective framework for autonomous data science with LLM agents.

Python), run them in the environment, and observe outputs, forming a closed-loop execution-feedback process (Yao et al., 2022) that updates the agent's internal state based on results or error messages. Complex tasks are decomposed into sequences of executable steps, enabling incremental refinement rather than one-shot inference. Failed executions, such as syntax errors or empty results, provide explicit supervision signals that allow the agent to correct reasoning and discard invalid intermediate hypotheses, reinforcing only execution-consistent trajectories. Systems such as Spider2-V (Cao et al., 2024) implement this paradigm in real-world data science workflows, autonomously generating, executing, and refining code to complete end-to-end tasks. While this approach improves grounding and robustness, it primarily operates within individual systems and lacks seamless integration across heterogeneous data sources.

2.1.3 Cross-Modal Data Grounding

Real-world data analysis requires reasoning across modalities, where isolated encoding or single-system execution is often insufficient. Cross-modal data grounding combines modality-aware representations with executable interactions in a unified framework. DataCrossAgent (Qi et al., 2026), for example, decomposes global tasks into modality-specific sub-tasks handled by specialized agents, exchanges intermediate results for cross-modal reasoning, and traces inconsistencies to their sources for iterative refinement of perception and reasoning. Despite these advances, heterogeneous data perception remains largely passive. Most systems assume schemas, metadata, or input tables are already available and reason only over exposed information. TRUST-SQL (Jian et al., 2026) exemplifies early active perception, exploring databases in unknown-schema settings by retrieving relevant structural evidence and grounding SQL generation on verified schema subsets. Active perception will likely become increasingly important as real-world data science involves multi-source, heterogeneous datasets, yet current systems still lack comprehensive cross-source understanding and reasoning.

2.2 Imperfect Data Processing

Imperfect data processing converts raw, noisy, incomplete, and inconsistent data into analysis-ready forms. It addresses common real-world issues such as missing values, duplicates, inconsistent formats, and heterogeneous conventions across records or columns. This module supports downstream objectives through three categories: *contextual data repair*, which infers and fixes corrupted or missing values using surrounding context; *task-oriented data cleaning*, which plans cleaning operations according to the analytical goal; and *semantic data standardization*, which normalizes heterogeneous representations into consistent, usable formats.

2.2.1 Contextual Data Repair

A fundamental challenge in imperfect data processing is that many datasets are incomplete or corrupted at the value level. Missing values, erroneous entries, duplicates, and anomalous records can propagate errors and bias into downstream models if left unresolved. Traditional heuristic strategies (e.g., mean or mode imputation) are computationally efficient, but they often ignore contex-

tual dependencies among attributes, failing to preserve the semantic integrity of the original data. In contrast, recent agent-based approaches treat contextual data repair as a context-sensitive inference problem. For example, LDI (Omidvartehrani and Rafiei, 2025) selects a localized subset of relevant attributes and tuples for each missing value, enabling more precise and interpretable imputations. Similarly, LLMDR (Keshav et al., 2026) combines representative sample selection with multi-LLM reasoning to recover missing values in mixed-type datasets, particularly under high-missingness or low-resource conditions. Compared with fixed statistical rules, these approaches make repair more context-aware and better aligned with the structural and semantic properties of the underlying table.

2.2.2 Task-Oriented Data Cleaning

Task-oriented data cleaning is the core of imperfect data processing, aiming to transform raw, inconsistent, or noisy data into forms suitable for downstream analysis. In practice, cleaning rarely involves applying isolated fixes; decisions about what to clean, in what order, and to what extent depend on the intended analytical task. LLM-based agents address this by interpreting data, reasoning about inconsistencies, and generating executable cleaning operations tailored to the task. AutoDCWorkflow (Li et al., 2024a) exemplifies goal-driven task-oriented cleaning. The agent identifies relevant columns based on the raw table and the analysis goal, evaluates their quality, and plans appropriate actions (e.g., repairing missing values or standardizing formats), translating them into executable steps. The workflow is iterative: the agent monitors outcomes and refines operations until the data meet the required quality. More advanced systems extend this paradigm. MLZero (Fang et al., 2025) integrates repair, transformation, and cleaning into a multi-agent loop, iteratively refining operations based on execution feedback. ForeAgent (Zheng et al., 2026) adopts a Predict-then-Verify approach, proactively detecting potential issues, applying corrections, and validating results. By reasoning about data, planning and executing repair and transformation steps, and validating outcomes, these LLM agents ensure that prepared datasets are consistent, reliable, and ready for downstream analysis.

2.2.3 Semantic Data Standardization

After task-oriented cleaning identifies and resolves major data-quality issues, semantic data standardization further improves data usability by normalizing heterogeneous representations into consistent formats. In real-world datasets, values such as dates, addresses, phone numbers, categorical labels, or measurement units often vary across sources. These variations may not be erroneous, but they hinder querying, aggregation, feature construction, and downstream modeling. LLM-based agents address this challenge by interpreting the semantic meaning of values and transforming them into task-consistent representations. For example, CleanAgent (Qi et al., 2024) uses natural-language specifications to guide automated, context-aware standardization, reducing manual coding while improving reliability and scalability. Jellyfish (Zhang et al., 2023) further demonstrates how LLMs can support broader data preparation, including error detection, schema alignment, and column-level normalization. By combining semantic reasoning with pattern-based heuristics, such agents can align heterogeneous representations, correct format-level anomalies, and produce standardized data that better supports downstream analytical workflows.

3 Agentic Data Reasoning

This section reviews agentic data reasoning techniques that form the decision-making core of autonomous data science agents. These techniques operate at complementary levels: *knowledge-grounded reasoning* enriches the agent's context with task-relevant external knowledge; *interaction-driven policy learning* improves decision behavior through feedback from execution, evaluation, or user interaction; and *workflow-level reasoning* structures and optimizes action sequences through orchestration or search.

3.1 Knowledge-Grounded Reasoning

Knowledge-grounded reasoning improves agent decision-making by incorporating task-relevant external knowledge into the reasoning process. In autonomous data science, such knowledge is especially important for solving complex analytical problems, where agents must formulate computational objectives, select appropriate modeling strategies, and adapt workflows under task-specific constraints. Existing methods mainly follow two directions: *methodological grounding*, which supplies explicit modeling principles or structured method knowledge to guide problem formulation and modeling, and *experience-guided adaptation*, which reuses historical task experience to support initialization, planning, and refinement.

3.1.1 Methodological Grounding

A central challenge in autonomous data science is model formulation: translating an open-ended real-world goal into a computational objective and executable analytical plan. This requires principled choices about modeling assumptions, workflow design, and optimization strategy, which are hard to make reliably from implicit model knowledge alone. Methodological grounding addresses this gap by introducing explicit technical knowledge, such as modeling principles, structured method libraries, or relational knowledge, to constrain planning and guide modeling decisions. MM-Agent (Liu et al., 2025a) instantiates this idea with the Hierarchical Mathematical Modeling Library (HMML), which organizes methods by domains, subdomains, and method nodes; agents retrieve relevant methods from HMML to guide abstraction and modeling for open-ended mathematical tasks. KAPSO (Nadafian et al., 2026) extends this paradigm to program synthesis, where externalized knowledge supports execution-aware code generation and optimization. Together, these studies show that explicit methodological structures can ground agent reasoning in established modeling knowledge and improve decision reliability in complex data science workflows.

3.1.2 Experience-Guided Adaptation

While methodological grounding relies on explicit modeling knowledge, experience-guided adaptation improves agent decision-making by reusing prior task experience. Rather than solving each problem from scratch, the agent retrieves related cases, workflows, or distilled heuristics and adapts them to the current task. This is valuable in data science, where tasks often share transferable modeling choices, preprocessing routines, evaluation strategies, and optimization patterns, allowing agents to initialize plans more effectively and reduce redundant exploration. DS-Agent (Guo et al., 2024) follows this case-based paradigm by constructing a case bank from high-quality human reports and successful executions. For a new task, it retrieves similar cases, adapts them into an experiment plan, executes the plan in a sandbox, and refines it using feedback such as error logs and model metrics. This retrieve–adapt–execute–retain loop supports both task grounding and continual improvement. MLCopilot (Zhang et al., 2024b) reuses higher-level experience distilled from prior machine learning tasks, organizing heterogeneous artifacts into experience pools and retrieving abstract knowledge, such as modeling insights and optimization heuristics, to guide planning for new tasks.

3.2 Interaction-Driven Policy Learning

Unlike knowledge-grounded reasoning, which relies on external methodological or historical knowledge, interaction-driven policy learning improves agent policies through environmental feedback. This direction is crucial for autonomous data science, where agents must learn modeling, planning, tool-use, and execution behaviors from long-horizon interactions with data, code, and evaluation signals rather than static supervision alone. Recent methods therefore optimize policies with rollout trajectories and task-level rewards, following two lines: *analytical trajectory learning*, which structures exploratory problem-solving interactions, and *task-reward policy optimization*, which designs rewards to evaluate trajectories and refine behavior.

3.2.1 Analytical Trajectory Learning

Analytical trajectory learning improves agent policies by structuring how agents interact with data environments over multi-step rollouts. In autonomous data science, the environment may include databases, raw files, execution sandboxes, intermediate outputs, error logs, and evaluation feedback. A central challenge is that task-relevant information is often only partially observable at the beginning: the agent must decide what to inspect, which hypotheses to test, when to execute code, and how to revise its plan based on delayed or sparse feedback. Thus, trajectory learning focuses not only on producing the final answer, but also on learning effective interaction processes for exploration, verification, execution, and refinement. TRUST-SQL (Jian et al., 2026) illustrates this idea in unknown-schema text-to-SQL tasks, where large databases may contain hundreds of tables with noisy or incomplete metadata. Since the relevant schema cannot be assumed to be fully available, TRUST-SQL formulates SQL generation as a four-stage trajectory: *Explore*, *Propose*, *Generate*, and *Confirm*. Its Dual-Track GRPO framework separates rewards for schema exploration from rewards for final SQL execution, explicitly addressing the credit-assignment gap between information acquisition and answer generation. Yang et al. (Yang et al., 2025b) extend this trajectory-learning perspective to machine learning engineering agents, where rollouts involve long and variable-duration

code-execution processes. They address two key difficulties: asynchronous updates may favor faster but weaker actions, and final test performance provides overly sparse feedback. To mitigate these issues, they introduce duration-aware gradient updates and environment instrumentation, which extracts partial-credit rewards from intermediate execution progress. This enables agents to learn more effective exploration, debugging, and refinement behaviors during complex analytical workflows.

3.2.2 Task-Reward Policy Optimization

Task-reward policy optimization studies how to evaluate agent actions and trajectories after environmental interaction. In autonomous data science, agents make sequential decisions over planning, data inspection, code generation, execution, and refinement, while supervision often appears only as delayed outcomes or final metrics. Reward design is therefore critical for turning sparse feedback into learnable signals. Recent work defines rewards at different granularities: DeepAnalyze (Zhang et al., 2025b) uses a curriculum to train atomic skills such as planning, data understanding, and code generation before reinforcement learning in realistic environments, reducing reward sparsity for end-to-end policies; ML-Agent (Liu et al., 2025b) improves action-level decisions with step-wise online reinforcement learning and exploration-enriched fine-tuning; SQL-R1 (Ma et al., 2025) designs multi-dimensional rewards for text-to-SQL over format, execution, correctness, and length; and Table-R1 (Yang et al., 2025c) jointly optimizes region evidence and answer correctness to ground table reasoning in relevant structure.

3.3 Workflow-Level Reasoning

Workflow-level reasoning concerns how autonomous data science agents organize execution at inference time. Unlike interaction-driven policy learning, which improves the agent's internal decision policy through feedback, workflow-level reasoning focuses on how analytical steps, tools, execution contexts, and specialized roles are structured to complete complex tasks. Existing work can be broadly organized into two paradigms: *predefined workflow orchestration*, where agents follow a manually designed workflow structure, and *adaptive workflow search*, where the workflow itself is optimized during inference.

3.3.1 Predefined Workflow Orchestration

Predefined workflow orchestration coordinates data science tasks with a fixed or manually specified execution structure, either through a single agent in a shared context or multiple specialized agents across stages. Single-agent designs preserve continuity by tracking intermediate states, execution history, and evolving task context. Operand Quant (Sahney et al., 2025), for example, organizes machine learning engineering as an IDE-centric loop over observation, command generation, execution, and integration, avoiding cross-agent handoffs by operating directly on files, notebooks, and kernels. Multi-agent designs decompose workflows across specialized modules for heterogeneous expertise, parallelism, or stage-wise coordination. The Multi-Agent Blackboard System (Salemi et al., 2025) uses a shared blackboard for decentralized information discovery; SPIO (Seo et al., 2025) assigns agents to preprocessing, feature engineering, modeling, and hyperparameter tuning before selecting or ensembling strategies; and AutoM3L (Luo et al., 2024) coordinates modality recognition, feature engineering, encoder selection, and fusion for multimodal learning.

3.3.2 Adaptive Workflow Search

Adaptive workflow search studies how agents can automatically discover effective workflow structures instead of relying on manually designed pipelines. In this paradigm, the workflow itself becomes an inference-time optimization target: the agent searches over stage compositions, action sequences, or workflow topologies to identify more effective reasoning and execution processes. This is important in autonomous data science because performance often depends not only on choosing appropriate tools or models, but also on arranging intermediate actions in the right order and structure. AFlow (Zhang et al., 2024a) formulates workflow construction as a search problem over code-represented agentic workflows. It uses Monte Carlo Tree Search to explore workflow topology, prompt design, and control flow, showing that optimization can target the organization of agent behavior rather than only model parameters or prompts. SELA (Chi et al., 2024) applies a similar search perspective to automated machine learning, exploring stage-wise processes such as exploratory data analysis, preprocessing, feature engineering, and model training. By combining planning with iterative refinement, these methods enable agents to

optimize workflow configurations and modeling strategies dynamically during inference.

4 System-Level Operation

This section reviews system-level operation techniques that provide the infrastructure for autonomous data science agents to function reliably in real-world environments. We organize the discussion into two layers: the *substrate layer*, which supports data access, sandboxed execution, and governance, and the *runtime layer*, which maintains long-running analytical processes.

4.1 Substrate Layer

The substrate layer defines the basic system infrastructure through which autonomous data science agents access data, execute actions, and operate under practical constraints. It consists of three core components: *data interfaces*, which connect agents to structured or heterogeneous data sources; *compute substrate*, which provides executable environments for code, tools, and feedback; and *access control and analytical governance*, which regulate permissions, isolation, and compliant behavior.

4.1.1 Data Interfaces

Unlike data perception, which concerns how agents represent and understand data, data interfaces define how such understanding is translated into valid operations over data systems. Autonomous data science agents must query, transform, and refine data across heterogeneous sources under system constraints, making interfaces a key part of the substrate layer. Current work mainly focuses on structured data. ReFoRCE (Deng et al., 2025) supports large-schema interaction through table compression, column exploration, and self-refinement. MARS-SQL (Yang et al., 2025a) enables iterative query construction over live databases through a multi-agent execution framework, while AutoLink (Wang et al., 2026) treats scalable schema exploration as part of the interface design. In enterprise settings, MAIA (Kapoor et al., 2025) integrates data transformation into text-to-SQL pipelines, connecting user intent with executable operations over business data. Despite these advances, most interfaces remain centered on tables and database schemas, leaving multi-source and multimodal data interfaces underexplored.

4.1.2 Compute Substrate

While data interfaces determine how agents access data, the compute substrate determines where agent actions are executed and verified. Its role is to provide controlled, feedback-rich environments in which agents can run code, invoke tools, observe system responses, and validate intermediate results. DSGym (Nie et al., 2026) standardizes datasets, metrics, and execution within a containerized framework for evaluating and training data science agents, thereby grounding reasoning in real data-dependent feedback. MLE-Dojo (Qiang et al., 2025) provides an interactive environment for machine learning engineering, enabling agents to experiment, debug, and refine solutions in Kaggle-style workflows. In enterprise-oriented settings, RP-ReAct (Molinari and Ciravegna, 2025) shows how planner–executor loops can be grounded in controlled runtimes, keeping reasoning and action tightly coupled through iterative feedback.

4.1.3 Access Control and Analytical Governance

Beyond data access and execution, autonomous data science agents must operate within organizational constraints that specify what they can access, modify, and execute. Access control and analytical governance therefore provide the permission, isolation, and policy mechanisms needed when agents interact with sensitive data, production systems, or shared analytical assets. Systems such as RP-ReAct (Molinari and Ciravegna, 2025) and *app.build* (Kniazev et al., 2025) point to this requirement by grounding agent behavior in controlled runtimes and deployment constraints. More broadly, governance mechanisms ensure that agentic analytics remain observable, auditable, and aligned with safety and compliance requirements.

4.2 Runtime Layer

The runtime layer sustains long-running data science workflows once data access and execution environments are in place. It has two main components: *analytical execution runtime*, which coordinates, schedules, and monitors multi-stage processes, and *pipeline state maintenance*, which preserves continuity, resource stability, and failure recovery across extended runs.

4.2.1 Analytical Execution Runtime

Analytical execution runtime concerns how autonomous data science systems coordinate multi-stage workflows. Rather than simply launching steps, the runtime must schedule analytical phases, monitor progress, coordinate agent roles, and incorporate feedback as workflows evolve. AutoK-

aggle (Li et al., 2024b) decomposes data science competitions into background understanding, exploratory analysis, data cleaning, feature engineering, and model building, while coordinating specialized agents across phases. ADP-MA (Khurana, 2026) separates planning, orchestration, and monitoring to dynamically construct and revise data-processing pipelines. AISAC (Bhattacharya and Som, 2025) introduces driver/helper roles to make multi-agent scientific workflows more transparent and controllable, while AgenticData (Sun et al., 2025a) manages heterogeneous analytics workflows with attention to quality and cost. These systems show that runtime design is essential for controllable, feedback-driven execution across complex data science pipelines.

4.2.2 Pipeline State Maintenance

Pipeline state maintenance focuses on preserving the continuity and reliability of long-running analytical workflows. Since data science agents repeatedly interact with code, tools, databases, and intermediate artifacts, the runtime must maintain execution state, manage resources, isolate environments, recover from failures, and support diagnosis. ALE (Wang et al., 2025) addresses this issue through large-scale sandbox allocation, resource reclamation, and fault isolation for agentic learning. *app.build* (Kniazev et al., 2025) provides production-oriented runtime scaffolding for reliable execution under deployment constraints, especially when agents interact with external systems. Vinay (Vinay, 2025) further emphasizes runtime reliability by studying monitoring and diagnosis mechanisms that expose, localize, and interpret failures. These works indicate that stable agent deployment requires not only executable environments, but also persistent state management and failure-aware runtime support.

5 Open issues and Future Direction

Despite recent progress, autonomous data science agents remain far from reliable end-to-end automation. Following our taxonomy, we highlight key research directions across data-world perception, agentic data reasoning, and system-level operation.

Data-World Perception Challenges. Current agents still struggle to understand heterogeneous and multi-source datasets in realistic environments. Most existing systems assume that structured schemas or curated datasets are already available, which rarely holds in practice. Future work should focus on developing *active data perception* mechanisms that enable agents to explore unknown data environments, discover schema structures, and dynamically construct task-relevant representations. Another key challenge lies in scalable data curation. Real-world datasets often contain noisy, incomplete, and inconsistent information that requires complex repair and transformation operations. Designing agents that can automatically perform context-aware data repair and cleaning remains an open research problem.

Agentic Data Reasoning Challenges. Although LLMs provide strong reasoning capabilities, their decision policies remain fragile in long-horizon analytical workflows. A key direction is to improve policy learning with richer feedback signals and better reward design; reinforcement learning from execution feedback may better align agent behavior with practical analytical objectives. Another direction is workflow-level reasoning: rather than following fixed pipelines, agents should dynamically discover and optimize analytical workflows. Search-based workflow discovery and hierarchical planning may therefore be central to more flexible and adaptive data science agents.

System-Level Operation Challenges. Reliable deployment requires system infrastructure that reflects real-world data ecosystems. Current agents often run in simplified environments, so future work should build scalable execution platforms for large datasets, heterogeneous sources, and long-running analytical workflows. Safety is equally critical: incorrect code generation or data manipulation can cause serious failures. Robust sandboxes, verification mechanisms, and monitoring frameworks are therefore essential for deploying autonomous data science agents in practice.

6 Conclusions

This survey reviewed LLM-based agents for autonomous data science and organized the field around three perspectives: data-world perception for understanding and curating heterogeneous data, agentic data reasoning for analytical decision-making and workflow optimization, and system-level operation for reliable execution and scalable infrastructure. Across these perspectives, we analyzed representative systems, summarized their design principles, and identified key limitations in robustness, scalability, and real-world deployment.

Limitations

First, although this survey covers representative LLM-based data-science agents, the field is evolving rapidly, and some recent systems or unpublished industrial deployments may not be fully covered. Our goal is therefore to provide a conceptual organization rather than an exhaustive catalog. Second, our perception–reasoning–operation taxonomy emphasizes cross-layer coordination, but some systems naturally span multiple categories. For example, executable data probing involves both perceptual grounding and runtime execution, while workflow orchestration involves both reasoning-level planning and system-level scheduling. We assign each work according to its main technical contribution, but these boundaries should be viewed as analytical rather than strict.

References

Chandrachur Bhattacharya and Sibendu Som. 2025. Aisac: An integrated multi-agent system for transparent, retrieval-grounded scientific assistance. *arXiv preprint arXiv:2511.14043*.

Ruisheng Cao, Fangyu Lei, Haoyuan Wu, Jixuan Chen, Yeqiao Fu, Hongcheng Gao, Xinzhuang Xiong, Hanchong Zhang, Yuchen Mao, Wenjing Hu, et al. 2024. Spider2-v: How far are multimodal agents from automating data science and engineering workflows? *Advances in Neural Information Processing Systems*, 37:107703–107744.

Jun Shern Chan, Neil Chowdhury, Oliver Jaffe, James Aung, Dane Sherburn, Evan Mays, Giulio Starace, Kevin Liu, Leon Maksin, Tejal Patwardhan, et al. 2025. Mle-bench: Evaluating machine learning agents on machine learning engineering. In *International Conference on Learning Representations*, volume 2025, pages 50466–50494.

Ke Chen, Peiran Wang, Yaoning Yu, Xianyang Zhan, and Haohan Wang. 2025a. Large language model-based data science agent: A survey. *arXiv preprint arXiv:2508.02744*.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, et al. 2025b. Scienceagentbench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations*, volume 2025, pages 96934–96990.

Yizhou Chi, Yizhang Lin, Sirui Hong, Duyi Pan, Yaying Fei, Guanghao Mei, Bangbang Liu, Tianqi Pang, Jacky Kwok, Ceyao Zhang, et al. 2024. Sela: Tree-search enhanced llm agents for automated machine learning. *arXiv preprint arXiv:2410.17238*.

Minghang Deng, Ashwin Ramachandran, Canwen Xu, Lanxiang Hu, Zhewei Yao, Anupam Datta, and Hao Zhang. 2025. Reforce: A text-to-sql agent with self-refinement, format restriction, and column exploration. In *ICLR 2025 Workshop: VerifAI: AI Verification in the Wild*.

Alex Egg, Martin Iglesias Goyanes, Friso Kingma, Andreu Mora, Leandro von Werra, and Thomas Wolf. 2025. Dabstep: Data agent benchmark for multi-step reasoning. *arXiv preprint arXiv:2506.23719*.

Haoyang Fang, Boran Han, Nick Erickson, Xiyuan Zhang, Su Zhou, Anirudh Dagar, Jiani Zhang, Ali Caner Turkmen, Cuixiong Hu, Huzefa Rangwala, et al. 2025. Mlzero: A multi-agent system for end-to-end machine learning automation. *arXiv preprint arXiv:2505.13941*.

Yanjie Fu, Dongjie Wang, Wangyang Ying, Xinyuan Wang, Xiangliang Zhang, Huan Liu, and Jian Pei. 2025. Autonomous data agents: A new opportunity for smart data. *arXiv preprint arXiv:2509.18710*.

Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023. Pal: Program-aided language models. In *International conference on machine learning*, pages 10764–10799. PMLR.

Aniketh Garikaparthi, Manasi Patwardhan, and Arman Cohan. 2026. Researchgym: Evaluating language model agents on real-world ai research. *arXiv preprint arXiv:2602.15112*.

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

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

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

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. 2023. Mlagentbench: Evaluating language agents on machine learning experimentation. *arXiv preprint arXiv:2310.03302*.

Yiming Huang, Jianwen Luo, Yan Yu, Yitong Zhang, Fangyu Lei, Yifan Wei, Shizhu He, Lifu Huang, Xiao Liu, Jun Zhao, et al. 2024. Da-code: Agent data science code generation benchmark for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13487–13521.

Yupan Huang, Tengchao Lv, Lei Cui, Yutong Lu, and Furu Wei. 2022. Layoutlmv3: Pre-training for document ai with unified text and image masking. In *Proceedings of the 30th ACM international conference on multimedia*, pages 4083–4091.

Ai Jian, Xiaoyun Zhang, Wanrou Du, Jingqing Ruan, Jiangbo Pei, Weipeng Zhang, Ke Zeng, and Xunliang Cai. 2026. Trust-sql: Tool-integrated multi-turn reinforcement learning for text-to-sql over unknown schemas. *arXiv preprint arXiv:2603.16448*.

Liqiang Jing, Zhehui Huang, Xiaoyang Wang, Wenlin Yao, Wenhao Yu, Kaixin Ma, Hongming Zhang, Xinya Du, and Dong Yu. 2024. Dsbench: How far are data science agents from becoming data science experts? *arXiv preprint arXiv:2409.07703*.

Sara Mani Kapoor, Soumya Banerjee, et al. 2025. Chatting with your data: Llm-enabled data transformation for enterprise text to sql. In *NeurIPS 2025 Workshop: Generative AI in Finance*.

Durga Keshav, GVD Praneeth, Chetan Kumar Patruni, Vivek Yelleti, and U Sai Ram. 2026. Llmdr: Large language model driven framework for missing data recovery in mixed data under low resource regime. *arXiv preprint arXiv:2601.22916*.

Udayan Khurana. 2026. Autonomous data processing using meta-agents. *arXiv preprint arXiv:2602.00307*.

Evgenii Kniazev, Arseny Kravchenko, Igor Rekun, James Broadhead, Nikita Shamgunov, Pranav Sah, Pratik Nichite, and Ivan Yamshchikov. 2025. app. build: A production framework for scaling agentic prompt-to-app generation with environment scaffolding. *arXiv preprint arXiv:2509.03310*.

Eugenie Lai, Gerardo Vitagliano, Ziyu Zhang, Om Chabra, Sivaprasad Sudhir, Anna Zeng, Anton A Zabreyko, Chenning Li, Ferdi Kossmann, Jialin Ding, et al. 2025. Kramabench: A benchmark for ai systems on data-to-insight pipelines over data lakes. *arXiv preprint arXiv:2506.06541*.

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

Kenton Lee, Mandar Joshi, Iulia Raluca Turc, Hexiang Hu, Fangyu Liu, Julian Martin Eisenschlos, Urvashi Khandelwal, Peter Shaw, Ming-Wei Chang, and Kristina Toutanova. 2023. Pix2struct: Screenshot parsing as pretraining for visual language understanding. In *International Conference on Machine Learning*, pages 18893–18912. PMLR.

Lan Li, Liri Fang, and Vetle I Torvik. 2024a. Autodcworkflow: Llm-based data cleaning workflow auto-generation and benchmark. *arXiv preprint arXiv:2412.06724*.

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

Zujie Liang, Feng Wei, Wujiang Xu, Lin Chen, Yuxi Qian, and Xinhui Wu. 2025. I-mcts: Enhancing agentic automl via introspective monte carlo tree search. *arXiv preprint arXiv:2502.14693*.

Fan Liu, Zherui Yang, Cancheng Liu, Tianrui Song, Xiaofeng Gao, and Hao Liu. 2025a. Mm-agent: Llm as agents for real-world mathematical modeling problem. *arXiv preprint arXiv:2505.14148*.

Zexi Liu, Jingyi Chai, Xinyu Zhu, Shuo Tang, Rui Ye, Bo Zhang, Lei Bai, and Siheng Chen. 2025b. Ml-agent: Reinforcing llm agents for autonomous machine learning engineering. *arXiv preprint arXiv:2505.23723*.

An Luo, Jin Du, Xun Xian, Robert Specht, Fangqiao Tian, Ganghua Wang, Xuan Bi, Charles Fleming, Ashish Kundu, Jayanth Srinivasa, et al. 2026. Agentds technical report: Benchmarking the future of human-ai collaboration in domain-specific data science. *arXiv preprint arXiv:2603.19005*.

Daqin Luo, Chengjian Feng, Yuxuan Nong, and Yiqing Shen. 2024. Autom3l: An automated multimodal machine learning framework with large language models. In *Proceedings of the 32nd ACM International Conference on Multimedia*, pages 8586–8594.

Peixian Ma, Xialie Zhuang, Chengjin Xu, Xuhui Jiang, Ran Chen, and Jian Guo. 2025. Sql-r1: Training natural language to sql reasoning model by reinforcement learning. *arXiv preprint arXiv:2504.08600*.

Gianni Molinari and Fabio Ciravegna. 2025. Reason-plan-react: A reasoner-planner supervising a react executor for complex enterprise tasks. *arXiv preprint arXiv:2512.03560*.

Alireza Nadafian, Alireza Mohammadshahi, and Majid Yazdani. 2026. Kapso: A knowledge-grounded framework for autonomous program synthesis and optimization. *arXiv preprint arXiv:2601.21526*.

Fan Nie, Junlin Wang, Harper Hua, Federico Bianchi, Yongchan Kwon, Zhenting Qi, Owen Queen, Shang Zhu, and James Zou. 2026. Dsgym: A holistic framework for evaluating and training data science agents. *arXiv preprint arXiv:2601.16344*.

Soroush Omidvartehrani and Davood Rafiei. 2025. Ldi: Localized data imputation for text-rich tables. *arXiv preprint arXiv:2506.16616*.

Danrui Qi, Zhengjie Miao, and Jiannan Wang. 2024. Cleanagent: Automating data standardization with llm-based agents. *arXiv preprint arXiv:2403.08291*.

Ruyi Qi, Zhou Liu, and Wentao Zhang. 2026. Datacross: A unified benchmark and agent framework for cross-modal heterogeneous data analysis. *arXiv preprint arXiv:2601.21403*.

Rushi Qiang, Yuchen Zhuang, Yinghao Li, Rongzhi Zhang, Changhao Li, Ian Shu-Hei Wong, Sherry Yang, Percy Liang, Chao Zhang, Bo Dai, et al. 2025. Mle-dojo: Interactive environments for empowering llm agents in machitang2025llm engineering. *arXiv preprint arXiv:2505.07782*.

Bo Qiao, Liqun Li, Xu Zhang, Shilin He, Yu Kang, Chaoyun Zhang, Fangkai Yang, Hang Dong, Jue Zhang, Lu Wang, et al. 2023. Taskweaver: A code-first agent framework. *arXiv preprint arXiv:2311.17541*.

Mizanur Rahman, Amran Bhuiyan, Mohammed Saidul Islam, Md Tahmid Rahman Laskar, Ridwan Mahbub, Ahmed Masry, Shafiq Joty, and Enamul Hoque. 2025. Llm-based data science agents: A survey of capabilities, challenges, and future directions. *arXiv preprint arXiv:2510.04023*.

Arjun Sahney, Ram Gorthi, Cezary Łastowski, and Javier Vega. 2025. Operand quant: A single-agent architecture for autonomous machine learning engineering. *arXiv preprint arXiv:2510.11694*.

Alireza Salemi, Mihir Parmar, Palash Goyal, Yiwen Song, Jinsung Yoon, Hamed Zamani, Tomas Pfister, and Hamid Palangi. 2025. Llm-based multi-agent blackboard system for information discovery in data science. *arXiv preprint arXiv:2510.01285*.

Wonduk Seo, Juhyeon Lee, Yanjun Shao, Qingshan Zhou, Seunghyun Lee, and Yi Bu. 2025. Spio: Ensemble and selective strategies via llm-based multi-agent planning in automated data science. *arXiv preprint arXiv:2503.23314*.

Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, et al. 2025. Paperbench: Evaluating ai's ability to replicate ai research. *arXiv preprint arXiv:2504.01848*.

Ji Sun, Guoliang Li, Peiyao Zhou, Yihui Ma, Jingzhe Xu, and Yuan Li. 2025a. Agenticdata: An agentic data analytics system for heterogeneous data. *arXiv preprint arXiv:2508.05002*.

Maojun Sun, Ruijian Han, Binyan Jiang, Houduo Qi, Defeng Sun, Yancheng Yuan, and Jian Huang. 2025b. A survey on large language model-based agents for statistics and data science. *The American Statistician*, pages 1–14.

Zirui Tang, Weizheng Wang, Zihang Zhou, Yang Jiao, Bangrui Xu, Boyu Niu, Dayou Zhou, Xuanhe Zhou, Guoliang Li, Yeye He, et al. 2025. Llm/agent-as-data-analyst: A survey. *arXiv preprint arXiv:2509.23988*.

Patara Trirat, Wonyong Jeong, and Sung Ju Hwang. 2024. Automl-agent: A multi-agent llm framework for full-pipeline automl. *arXiv preprint arXiv:2410.02958*.

Vaishali Vinay. 2025. Failure modes in llm systems: A system-level taxonomy for reliable ai applications. *arXiv preprint arXiv:2511.19933*.

Weixun Wang, XiaoXiao Xu, Wanhe An, Fangwen Dai, Wei Gao, Yancheng He, Ju Huang, Qiang Ji, Hanqi Jin, Xiaoyang Li, et al. 2025. Let it flow: Agentic crafting on rock and roll, building the rome model within an open agentic learning ecosystem. *arXiv preprint arXiv:2512.24873*.

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

Ziyang Wang, Yuanlei Zheng, Zhenbiao Cao, Xiaojin Zhang, Zhongyu Wei, Pei Fu, Zhenbo Luo, Wei Chen, and Xiang Bai. 2026. Autolink: Autonomous schema exploration and expansion for scalable schema linking in text-to-sql at scale. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 33809–33817.

Haolin Yang, Jipeng Zhang, Zhitao He, and Yi R Fung. 2025a. Mars-sql: A multi-agent reinforcement learning framework for text-to-sql. *arXiv preprint arXiv:2511.01008*.

Sherry Yang, Joy He-Yueya, and Percy Liang. 2025b. Reinforcement learning for machine learning engineering agents. *arXiv preprint arXiv:2509.01684*.

Zheyuan Yang, Lyuhao Chen, Arman Cohan, and Yilun Zhao. 2025c. Table-r1: Inference-time scaling for table reasoning. *arXiv preprint arXiv:2505.23621*.

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

Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. 2024. Textgrad: Automatic" differentiation" via text. *arXiv preprint arXiv:2406.07496*.

Dan Zhang, Sining Zhoubian, Min Cai, Fengzu Li, Lekang Yang, Wei Wang, Tianjiao Dong, Ziniu Hu, Jie Tang, and Yisong Yue. 2025a. Datascibench: An llm agent benchmark for data science. *arXiv preprint arXiv:2502.13897*.

Haochen Zhang, Yuyang Dong, Chuan Xiao, and Masafumi Oyamada. 2023. Jellyfish: A large language model for data preprocessing. *arXiv preprint arXiv:2312.01678*.

Jiayi Zhang, Jinyu Xiang, Zhaoyang Yu, Fengwei Teng, Xionghui Chen, Jiaqi Chen, Mingchen Zhuge, Xin Cheng, Sirui Hong, Jinlin Wang, et al. 2024a. Aflow: Automating agentic workflow generation. *arXiv preprint arXiv:2410.10762*.

Lei Zhang, Yuge Zhang, Kan Ren, Dongsheng Li, and Yuqing Yang. 2024b. Mlcopilot: Unleashing the power of large language models in solving machine learning tasks. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2931–2959.

Shaolei Zhang, Ju Fan, Meihao Fan, Guoliang Li, and Xiaoyong Du. 2025b. Deepanalyze: Agentic large language models for autonomous data science. *arXiv preprint arXiv:2510.16872*.

Yuge Zhang, Qiyang Jiang, XingyuHan XingyuHan, Nan Chen, Yuqing Yang, and Kan Ren. 2024c. Benchmarking data science agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5677–5700.

Jingsheng Zheng, Jintian Zhang, Yujie Luo, Yuren Mao, Yunjun Gao, Lun Du, Huajun Chen, and Ningyu Zhang. 2026. Can we predict before executing machine learning agents? *arXiv preprint arXiv:2601.05930*.

Yizhang Zhu, Liangwei Wang, Chenyu Yang, Xiaotian Lin, Boyan Li, Wei Zhou, Xinyu Liu, Zhangyang Peng, Tianqi Luo, Yu Li, et al. 2025. A survey of data agents: Emerging paradigm or overstated hype? *arXiv preprint arXiv:2510.23587*.

A Comparison with Existing Surveys

Table 1 contrasts our survey with recent work on LLM-based data analysis, data agents, and data science agents. Prior surveys typically center on a single axis, data modalities, analysis tasks, agent capabilities, autonomy levels, or lifecycle stages. While useful for mapping specific subareas, they offer limited coverage of how data understanding, agentic reasoning, and executable systems interact to support heterogeneous, long-horizon data science workflows.

In contrast, our survey takes a perception–reasoning–operation view that connects the data, agent, and system layers. The *data* layer covers how agents perceive, clean, and represent heterogeneous data; the *agent* layer focuses on reasoning, planning, learning, and workflow optimization; and the *system* layer examines execution environments, data interfaces, runtime support, and governance. By foregrounding how these layers interact, our taxonomy moves beyond isolated capability lists and explains how they jointly enable end-to-end autonomous data science.

B Preliminaries

This section introduces the fundamental components of LLM-based agents for automated data science, including task representation and agent interaction with computational environments.

Data Science Tasks. A data science task refers to an analytical problem in which a user seeks to extract insights or build predictive models from data. These tasks typically arise in real-world scenarios where datasets must be explored, processed, and analyzed to answer specific questions. Broadly, they can be divided into two categories: (1) *Data Analysis*, which focuses on understanding and summarizing datasets through techniques such as data cleaning, preprocessing, exploratory analysis, statistical summarization, and visualization; and (2) *Data Modeling*, which involves constructing predictive or explanatory models, including classification, regression, forecasting, and other machine learning approaches. Datasets may include multiple modalities, such as tabular, graph, time-series, text, or image data. The goal of a data science agent is to transform the task specification into actionable outputs, such as statistical summaries, predictive models, visualizations, or interpretable insights (Hu et al., 2024; Huang et al., 2024).

Agent Interaction Framework. Given a data science task, an agent iteratively reasons over the problem, interacts with computational tools, and produces a solution. LLMs serve as the reasoning core, guiding the agent through a loop of *thought*, *action*, and *observation*. At each step, the agent plans the next operation (thought), generates an executable action (e.g., Python code, SQL query, or visualization command), and receives feedback from the environment (observation), such as computed statistics, query results, or error messages. This iterative process continues until the agent produces a final answer or reaches a predefined maximum number of steps. The interaction history, including prior reasoning and executed actions, serves as context for subsequent steps, enabling long-horizon, multi-step analytical workflows (Yao et al., 2022).

C Taxonomy Overview

Figure 3 summarizes the organization used in this survey. The taxonomy follows the perception–reasoning–operation stack: data-world perception focuses on how agents understand and process heterogeneous data; agentic data reasoning captures how agents make analytical and workflow decisions; and system-level operation describes the execution environments, interfaces, and runtime support needed for reliable deployment. This overview is intended as a compact map of the surveyed literature and as a guide to the three main technical sections of the paper.

D Benchmark and Evaluation

Table 2 summarizes representative benchmarks for LLM agents in data science. We focus on public artifacts, task realism, and whether each benchmark evaluates an executable workflow rather than only a final answer. We observe two main findings. First, no single benchmark yet covers the full data workflow: DS-1000 (Lai et al., 2023) remains useful for API-heavy coding, MLE-bench (Chan et al., 2025) is strong for Kaggle-style ML engineering, and DABstep (Egg et al., 2025) and KramaBench (Lai et al., 2025) better approximate real analytics work, but none has become a universal standard. Second, the benchmark substrate itself is becoming a research object: frameworks such as DSGym (Nie et al., 2026) and ResearchGym (Garikaparthi et al., 2026) package tasks with tools, compute limits, traces, and auditing mechanisms, shifting evaluation from isolated question answering toward reproducible workflow execution.

Table 1: Comparison of related surveys from different organizational perspectives. ✓indicates explicit coverage, and •indicates partial coverage.

Survey	Data	Agent	System	DS Lifecycle
LLM/Agent-as-Data-Analyst: A Survey (Tang et al., 2025)	✓	–	–	•
A Survey of Data Agents: Emerging Paradigm or Overstated Hype? (Zhu et al., 2025)	✓	–	–	✓
LLM-Based Data Science Agents: A Survey of Capabilities, Challenges, and Future Directions (Rahman et al., 2025)	–	✓	–	✓
A Survey on Large Language Model-based Agents for Statistics and Data Science (Sun et al., 2025b)	–	✓	–	–
Large Language Model-based Data Science Agent: A Survey (Chen et al., 2025a)	–	✓	–	✓
Towards Autonomous Data Science with LLM Agents: A Survey and Outlook (Ours)	✓	✓	✓	✓

Table 2: Comparison of representative benchmarks for LLM/data agents in data science.

Benchmark	Year	Primary Focus	Task/Data Scale	Core Evaluation Style
DS-1000 (Lai et al., 2023)	2023	Data-science code generation	1,000 problems spanning seven Python libraries	Executable tests and surface-form constraints; reliability against false positives
InfiAgent-DABench (Hu et al., 2024)	2024	Agentic data analysis on CSV files	Approximately 311 public validation questions over 55 CSV files; broader benchmark framework around DAEval	Closed-form correctness with sandboxed execution; accuracy metric
DSEval (Zhang et al., 2024c)	2024	Whole data-science lifecycle	Four benchmark families: Exercise, Stack Overflow, LeetCode, and Kaggle; 825 total problems across sets	Pass rate, error propagation, intactness, and presentation-error variants
TaskWeaver (Qiao et al., 2023)	2024	Stateful data-analytics agent framework	23 internal evaluation cases plus transformed DS-1000, DABench, and DSEval tasks	Normalized aggregate score across benchmark suites
MLAgentBench (Huang et al., 2023)	2024	End-to-end ML experimentation	13 tasks across image, text, graph, tabular, time series, and recent research datasets	Success rate based on final artifact quality in controlled environments
DSBench (Jing et al., 2024)	2024	Realistic data analysis and data modeling	466 data-analysis tasks and 74 Kaggle-style modeling tasks	Accuracy for analysis; RPG and task success for modeling; human baselines included
DA-Code (Huang et al., 2024)	2024	Agent data-science code generation	500 tasks across data wrangling, machine learning, and exploratory data analysis	Executable environment; total score, completion rate, average steps, and executable-code rate
MLE-bench (Chan et al., 2025)	2025	ML engineering via Kaggle competitions	75 competitions; Lite split of 22 competitions	Kaggle-style scoring mapped to leaderboard percentiles and Any Medal percentage
DataSciBench (Zhang et al., 2025a)	2025	Broad data-science prompt benchmark with fine-grained metrics	23 evaluated models; 519 TFC test cases derived from curated prompts	Success rate, completion rate, VLM judging, 25 aggregate-function metrics, and total score
DABstep (Egg et al., 2025)	2025	Realistic multi-step data analysis with heterogeneous documents	450+ enterprise data-analysis tasks from Adyen	Factoid-style automatic correctness; hard/easy split; cost reporting; public leaderboard
KramaBench (Lai et al., 2025)	2025	Data-to-insight pipelines over data lakes	104 curated pipelines, 1,700 data files, 24 sources, and 6 domains	End-to-end and subtask evaluation; exact/approximate string and numeric scoring; runtime per task
ScienceAgentBench (Chen et al., 2025b)	2025	Data-driven scientific discovery workflows	102 tasks from 44 peer-reviewed papers in four disciplines	Generated Python programs scored by execution, results, and cost; multiple frameworks per model
PaperBench (Starace et al., 2025)	2025	Replicating frontier ML research papers	20 ICML 2024 spotlight/oral papers	Hierarchical research-task rubrics for replication
AgentDS (Luo et al., 2026)	2026	Domain-specific data science and human–AI collaboration	17 challenges across 6 industries; 29 teams and 80 participants	Competition ranking; AI-only versus human–AI comparison
DSGym (Nie et al., 2026)	2026	Standardized evaluation and training framework	Standardized/refined benchmark suite plus DSBio and DSPredict; 2,000-example training case study	Unified execution environments; quality and shortcut-solvability filtering
ResearchGym (Garikaparthi et al., 2026)	2026	Real-world AI research agents	5 research environments and 39 subtasks	Improvement over strong baselines plus subtask completion in fixed containers

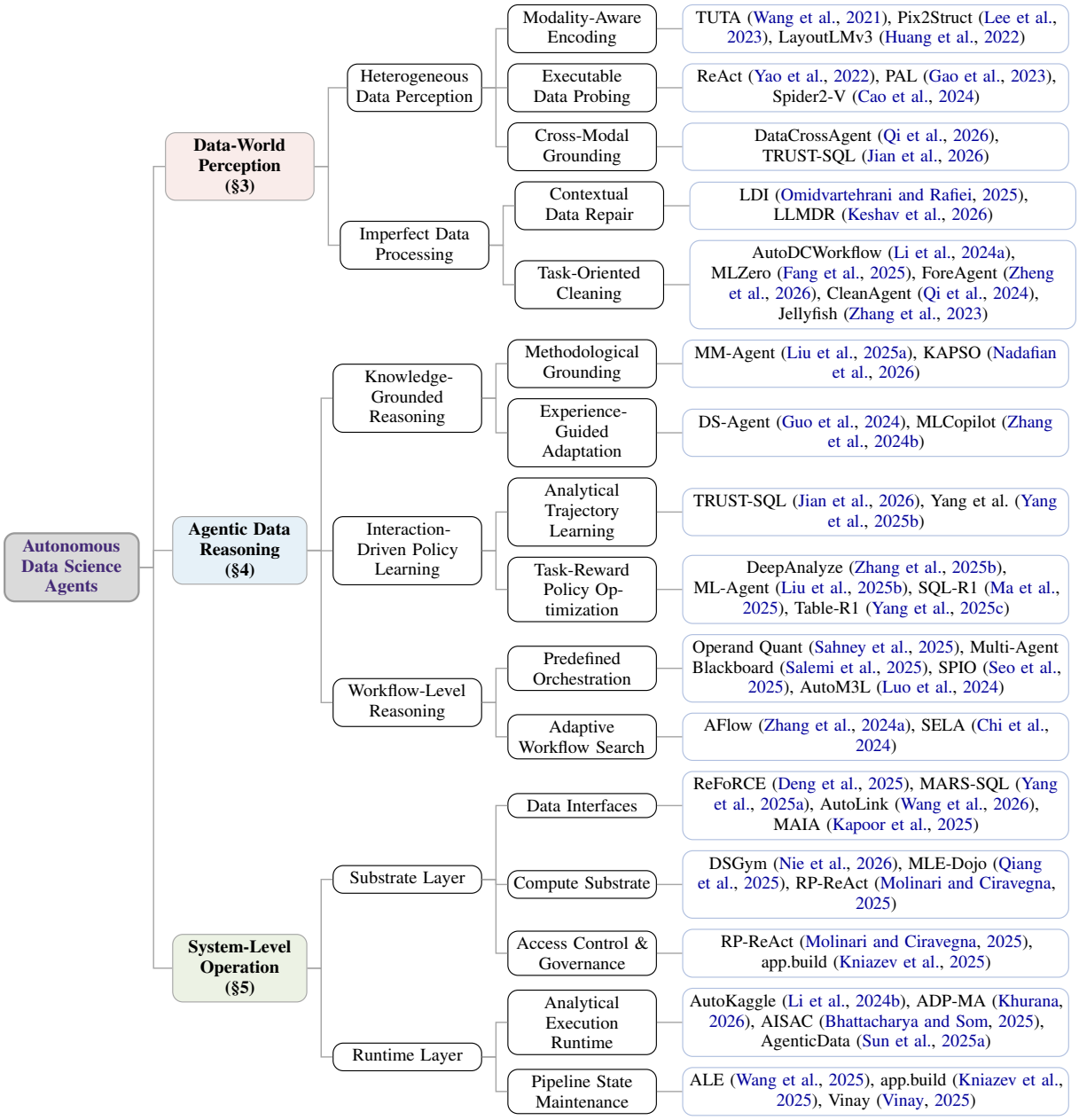

Figure 3: Taxonomy of LLM-based autonomous data science agents. We organize the field into three complementary perspectives: *Data-World Perception* (§ 3), which equips agents to understand and prepare heterogeneous data; *Agentic Data Reasoning* (§ 4), which underpins decision-making through knowledge grounding, policy learning, and workflow-level reasoning; and *System-Level Operation* (§ 5), which provides the substrate and runtime infrastructure for reliable end-to-end execution.