

Structured Strategy Injection for Data Analysis Agents

Zifu Tao¹, Guozhao Mo^{1,2}, Weixiang Zhou^{1,†}, Yaojie Lu¹, Hongyu Lin¹,
Ben He^{1,2}, Xianpei Han^{1,2}, Le Sun^{1,2}

¹Chinese Information Processing Laboratory, Institute of Software,
Chinese Academy of Sciences

²University of Chinese Academy of Sciences

zifutao@nyu.edu {moguozhao2024, weixiang}@iscas.ac.cn

benhe@ucas.ac.cn {luyaojie, hongyu, xianpei, sunle}@iscas.ac.cn

Abstract

Data analysis agents often produce shallow or non-auditable summaries because they lack the explicit reasoning frameworks and validation habits used by domain experts, especially when evidence must be gathered across heterogeneous data sources. We propose structured strategy injection, which injects compact methodology specifications with explicit checkpoints into an agent’s controller context, turning expert workflows into executable plans and enabling compliance-based transparency. To make such methodology-driven planning executable and auditable in heterogeneous settings, we introduce Data2MCP, a router-based orchestration infrastructure that standardizes SQL databases, knowledge graphs, vector stores, and tables as MCP-compatible tools and routes queries dynamically for multi-source retrieval and join-like reasoning. We compare Data2MCP against a RAG baseline and DA-Agent across multi-source retrieval, DAComp, and BIRD MiniDev. In multi-source retrieval, Data2MCP outperforms DA-Agent, and strategy injection benefits both architectures. On DAComp, strategy injection improves DA-Agent from 50.84 to 57.17 (+6.33). The code is available at <https://github.com/turambar928/Data2MCP>.

1 Introduction

Recent advances in Large Language Models (LLMs) have enabled autonomous data analysis agents to interface with specialized databases and unstructured document stores (OpenAI et al., 2024; Yang et al., 2025). However, deploying these systems in production environments reveals that data analysis tasks present operational challenges in two distinct dimensions: (1) spatial data fragmentation across heterogeneous modalities, and (2) temporal reasoning drift during multi-turn exploration. Existing agent frameworks generally address only

[†]Corresponding authors.

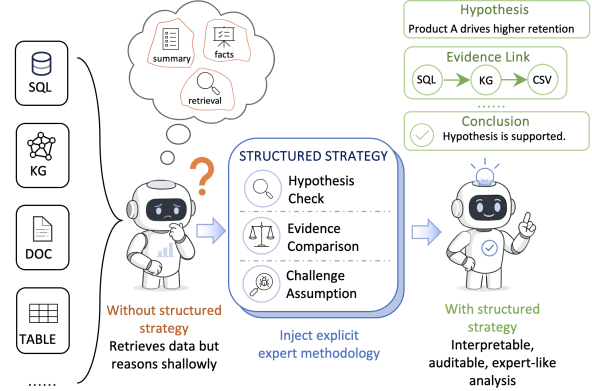


Figure 1: Overview of the dual-axis data analysis framework. The Data2MCP infrastructure standardizes the spatial complexity of heterogeneous data sources (SQL, KG, Vector DB), while Structured Strategy Injection regulates the temporal execution path of the agent via explicit checkpoints.

one side of this matrix. Monolithic code-execution sandboxes, such as OpenHands or DA-Agent (Lei et al., 2025), are engineered with a depth-first orientation. They perform localized exploratory data analysis (EDA) effectively on centralized, homogeneous data tables, but introduce significant interface friction when cross-source evidence verification is required across structurally diverse systems like relational databases and knowledge graphs.

Furthermore, granting an agent raw access to multi-source tools does not guarantee analytical rigor. Without explicit procedural boundaries, standard agents navigating multi-source environments frequently exhibit erratic reasoning paths—failing to cross-check conflicting evidence, neglecting data quality baselines, or generating conclusions without an auditable verification trail. This limitation stems from current prompting paradigms: established techniques such as Chain-of-Thought (Wei et al., 2022) and ReAct (Yao et al., 2023) rely on unstructured, task-specific instructions that lack systematic constraints for multi-stage workflows.

In contrast, human analysts rely on formalized methodologies—such as CRISP-DM for data mining (Chapman, 2000) or the Analysis of Competing Hypotheses (ACH) for intelligence analysis (Heuer, 1999)—to enforce a global logic on systematic inquiry.

To address the temporal challenge of reasoning drift, we introduce Structured Strategy Injection. This framework embeds domain-expert procedural knowledge directly into the agent’s controller context at inference time. By incorporating explicit analytical checkpoints and compliance validation mechanisms, our approach makes agent decisions verifiable and aligned with expert workflows. We evaluate three execution modes with varying degrees of autonomy: Fixed Injection of a specific methodology, Auto-selection from a curated catalog of 74 strategies, and Adaptive Generation of bespoke instructions. Because strategy injection operates strictly at the prompt level, it requires no fine-tuning and remains architecture-agnostic.

An analytical methodology, however, requires a compatible execution environment to be effective. Because expert strategies like ACH depend on cross-source evidence verification, they cannot be fully leveraged in environments restricted to a single data modality. To resolve this spatial constraint, we introduce Data2MCP, a multi-source orchestration framework that integrates SQL databases, Neo4j knowledge graphs, FAISS vector stores, and Pandas DataFrames via the Model Context Protocol (MCP) (Anthropic, 2024). Data2MCP serves as a general-purpose engineering infrastructure rather than a tailored testbed for strategy evaluation; it resolves spatial heterogeneity through dynamic routing and homogenized tool interfaces. Additionally, its standardized tool schema minimizes integration variance, ensuring that performance changes are attributable to the evaluation of planning interventions rather than ad-hoc integration disparities. Within this unified multi-source environment, the orchestration capabilities of structured strategies can be systematically evaluated.

This separation of concerns, with Data2MCP managing spatial data integration and strategy injection governing temporal reasoning, enables the two dimensions to be evaluated independently. We evaluate strategy injection across two distinct architectural paradigms: our breadth-centric Data2MCP framework and the depth-centric code-sandbox infrastructure of DA-Agent. Empirical results show that strategy injection yields consistent per-

formance gains across both systems and multiple underlying LLMs, demonstrating that the two architectural approaches are complementary.

This paper makes three primary contributions:

- **Structured Strategy Injection:** A framework that embeds expert methodologies into LLM agents via explicit checkpoints and dual-validation monitoring, enforcing auditable analytical behaviors without parameter updates.
- **The Data2MCP Infrastructure:** A standalone multi-source orchestration framework that unifies heterogeneous data access under a Router-based MCP architecture, establishing a scalable foundation for decentralized joint reasoning.
- **Orthogonal Empirical Validation:** Comprehensive evaluations showing that strategy injection consistently enhances performance across distinct architectural paradigms (the breadth-centric Data2MCP and the depth-centric DA-Agent), confirming the generality of the method.

2 Related Work

Data Analysis Agents. Recent LLM-based agents (Lai et al., 2023; Cao et al., 2024; Lei et al., 2025; Wang et al., 2025) have progressed from code generation to end-to-end analysis pipelines. DA-Agent (Lei et al., 2025) uses a three-stage Re-Act loop; OpenHands (Wang et al., 2025) provides sandboxed execution. While these systems excel at operational capabilities, they often provide limited methodology-level procedural scaffolding (e.g., explicit checkpoints for verification and systematic evidence cross-checking). Our work complements them by injecting structured domain methodologies.

Tool Use and Multi-Source Retrieval. LLM tool use (Schick et al., 2023; Yao et al., 2023) enables mid-generation API calls. Multi-hop retrieval (Tang and Yang, 2024; Trivedi et al., 2023) studies evidence gathering; HybridRAG (Sarmah et al., 2024) combines vector and graph retrieval. MCP (Anthropic, 2024) standardizes tool interfaces; we leverage it to unify SQL, Neo4j, FAISS, and pandas as interchangeable sub-agents.

Prompting and Reasoning Structures. Chain-of-thought (Wei et al., 2022) and least-to-most

prompting (Zhou et al., 2023) inject reasoning structures through natural language. ReAct (Yao et al., 2023) interleaves reasoning and action for tool use. While these methods provide ad-hoc reasoning guidance, they lack the systematic frameworks and explicit checkpoints that domain methodologies provide. For example, CoT encourages step-by-step reasoning but does not specify which analytical steps to take or how to validate their execution. ReAct structures tool use but does not embed domain-expert procedural knowledge. Voyager (Wang et al., 2023a) injects programmatic skills; Chameleon (Lu et al., 2023) composes tools via planning. Our work differs by injecting domain-level analytical frameworks (CRISP-DM, ACH) that codify expert practice into structured, reusable specifications with explicit checkpoints for compliance validation, enabling both performance gains and behavioral transparency.

Agent Interpretability. Recent work (Qin et al., 2024) focuses on understanding tool-calling patterns. Our strategy injection demonstrates that structured methodologies induce verifiable analytical behaviors that align with expert workflows.

3 Methodology

3.1 The Data2MCP Architecture

The DATA2MCP framework is designed for open-ended analytical queries that require sequential tool dispatch and multi-source synthesis. Formally, given a natural language query q and a set of Model Context Protocol (MCP) tools $\mathcal{T} = \{t_1, t_2, \dots, t_n\}$, where each tool encapsulates an underlying data source, the system produces a final textual response via a multi-turn reasoning loop.

To manage source heterogeneity and enforce systematic reasoning paths, the framework decouples execution into three distinct operational layers (Figure 2): a data layer for interface encapsulation, an orchestration layer (Router) for multi-turn execution, and a strategy layer for procedural methodology injection and validation.

Data Layer: Interface Abstraction. The data layer unifies disparate data modalities under a standardized MCP interface, allowing the central controller to interact with different database engines using a single tool-calling protocol. The current implementation supports four data source categories: (1) Relational Databases, accessed via read-only Text-to-SQL (SQLAlchemy) capped at 1,000 rows

per query; (2) Knowledge Graphs, queried via Text-to-Cypher (Neo4j) with a maximum return of 500 records; (3) Unstructured Document Collections, managed via FAISS vector retrieval ($top-k = 5$, chunk size of 1,000 characters); and (4) Tabular Data, manipulated through localized Pandas DataFrame operations.

Every tool within this layer exposes an invariant interface: a natural-language description, a JSON-schema parameter definition, and a textual response limited to a 10,000-character budget. This uniformity removes source-specific routing logic from the orchestration layer; the central LLM selects tools based entirely on semantic descriptions, rendering the Router source-agnostic. The 10,000-character cap prevents context window overflow during extended execution sequences. Truncation is handled by appending an explicit metadata flag to the relevant content prefix. Read-only permissions are enforced natively at the database level rather than via system prompts, ensuring data integrity independent of model stochasticity.

Orchestration Layer: Multi-Turn Execution Loop.

The Router coordinates the execution loop across sequential steps. At turn t , the Router provides the core LLM with: (i) the accumulated conversation history $H_{<t}$, (ii) the tool definitions $\mathcal{T}$, and (iii) the strategy specification $\mathcal{S}$ embedded in the controller context. The Router parses the model output, executes the designated tool calls, appends the results as tool-role messages to update the history to H_t , and repeats this sequence until the model invokes the `end_with_message` utility or hits a maximum execution budget of 30 turns.

The 30-turn budget serves as a practical ceiling to bound execution costs, as empirical returns diminish beyond this threshold. The `end_with_message` tool acts as a deterministic constraint that prevents silent failures; it prevents the model from terminating execution without outputting a synthesized final narrative. To handle network and database instabilities, failed tool invocations are retried once using an exponential backoff schedule. If the second attempt fails, the execution log captures the error as a standard tool response, forcing the controller to adapt its planning trajectory without the requested evidence.

3.2 Structured Strategy Injection

Agent execution failures are frequently caused by suboptimal analytical workflows, such as omitting

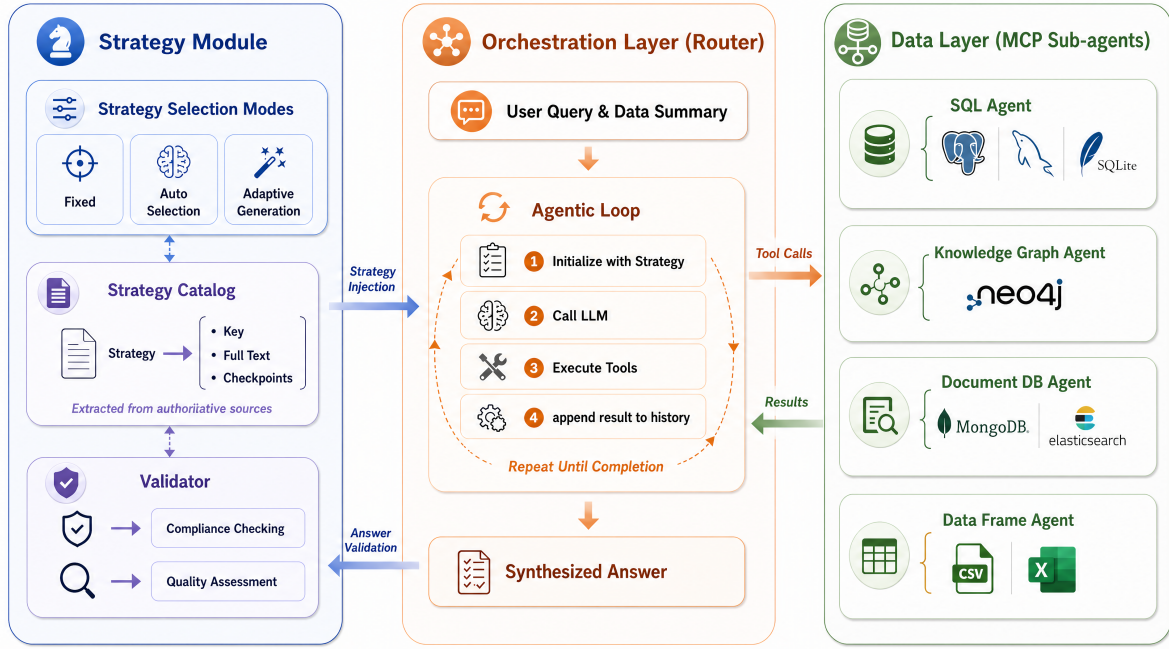


Figure 2: Data2MCP system architecture. The Strategy Module (left) selects or generates a specific analytical methodology to be embedded into the controller context. The Router (center) executes the multi-turn agentic execution loop, dispatching programmatic calls to MCP sub-agents (right). Compliance is evaluated post-execution by the Strategy and Output Validators.

data verification steps or ignoring conflicting evidence. We mitigate this behavioral drift by injecting structured "analysis playbooks" into the Router's controller context to constrain execution pathways without restricting the specific content of individual tool calls.

StrategySpec: Formalizing Procedural Constraints. An analytical methodology is formalized as a StrategySpec tuple $\langle k, \tau, \mathcal{C} \rangle$, where k is a unique identifier, τ is the guidance string consisting of 2–6 procedural sentences injected verbatim into the prompt, and $\mathcal{C}$ is a set of explicit behavioral checkpoints.

This three-field formulation separates the operational instruction from the evaluation criterion. While τ functions as an active constraint on the LLM's generation loop, $\mathcal{C}$ serves as the reference ground truth for automated compliance validation. Restricting τ to 2–6 sentences minimizes instruction distraction, ensuring that methodology injection does not degrade the model's adherence to the core task prompt.

Catalog Construction and Selection Modes. The framework maintains a permanent catalog of $|\mathcal{S}| = 74$ strategies. This includes 70 methodologies automatically parsed from domain texts (Pher-

Table 1: Example of a StrategySpec instance for the Analysis of Competing Hypotheses (ACH) methodology.

Field	Content
key	ach
full_text	Identify all plausible hypotheses. Search for diagnostic evidence that can discriminate between them. Evaluate each hypothesis against the gathered evidence. Methodically eliminate inconsistent hypotheses. Synthesize the final conclusion with explicit uncertainty levels.
checkpoints	{hypothesis_enumeration, evidence_matrix, elimination_step, uncertainty_statement}

son and Heuer Jr, 2019; Chapman, 2000) using an offline LLM extraction pipeline (Appendix A.3). The catalog scale is bounded to guarantee that the full option space can be processed within a single selection prompt without exceeding context limitations.

The framework supports three discrete operational selection modes: (1) Fixed Mode, where a user specifies a target key k and its corresponding τ is loaded; (2) Auto-selection Mode, where a lightweight LLM call matches the incoming query q against the strategy catalog using a brief data sum-

mary; and (3) Adaptive Generation Mode, where the system dynamically assembles a custom StrategySpec based on query intent and data dimensionality.

Closed-Loop Compliance Validation. Verifying adherence to an injected methodology requires an explicit feedback loop. We implement a two-stage validation mechanism that produces a normalized compliance score in the range $[0, 1]$: (1) The StrategyValidator continuously evaluates execution trace statistics, measuring physical metrics such as parallel-call ratios and tool invocation diversity against strategy-specific thresholds. (2) The OutputValidator conducts a post-hoc evaluation of the final narrative answer to confirm that all specified elements in checkpoints $\mathcal{C}$ are satisfied.

Empirical evaluation indicates that this compliance score reflects execution quality rather than prompt formatting alignment; compliance correlates significantly with final answer correctness (Pearson $r = 0.34$, $p < 0.001$). When execution falls below threshold compliance, the system outputs a structured warning containing checkpoint-level failures, isolating which analytical procedures were bypassed.

Architectural Modularity. In summary, the three-layer design of Data2MCP addresses distinct orthogonal axes of the analysis task. The data layer standardizes access to heterogeneous storage systems; the orchestration layer drives sequential reasoning execution; and the strategy layer enforces methodological rigor via structural constraints and verification. This modularity isolates each component, permitting independent additions of tools, strategies, or validators without altering the core framework logic.

4 Experiments

Our experiments address four research questions:

- **RQ1:** Does Data2MCP’s multi-source orchestration architecture effectively handle heterogeneous data retrieval compared to single-source specialized systems? (Section 4.2)
- **RQ2:** Can frontier LLMs equipped with structured analytical methodologies achieve strong performance on complex data analysis tasks? (Section 4.3)
- **RQ3:** Can an LLM autonomously discover effective analysis strategies by selecting from

Table 2: Single-source retrieval comparison (accuracy %). **Bold** = best in each row.

Source	RAG	DA-Agent	Data2MCP
Spider	64	65	72
MetaQA	64	69	66
HotpotQA	49	52	69
MS MARCO	71	78	81
Overall	62	66	72

a catalog or generating bespoke strategies? (Section 4.3, Section 4.4)

- **RQ4:** How do strategy-injected agents compare to other systems in analysis quality, stability, interpretability, and efficiency? (Section 4.4, Appendix C.5)

4.1 Benchmark, Evaluation, and Experimental Setup

Multi-Source Retrieval experiment. For the multi-source retrieval experiments (Section 4.2), we evaluate both systems on 100 queries per data source (400 total) spanning four heterogeneous data sources: Spider (Yu et al., 2018) (SQL), MetaQA (Zhang et al., 2018) (Knowledge Graph), HotpotQA (Yang et al., 2018) (Elasticsearch), and MS MARCO (Bajaj et al., 2018) (MongoDB).

Single-source and multi-source settings differ in the agent environment rather than in the intrinsic structure of the original benchmarks. Spider, MetaQA, HotpotQA, and MS MARCO serve as source pools for heterogeneous modalities. In the single-source setting, the agent is exposed only to the source relevant to each query. In the multi-source setting, multiple heterogeneous sources are jointly exposed, and the agent must identify the relevant sources, ignore distractors, and synthesize evidence when required (e.g., a Spider SQL query is evaluated against Spider ground truth, but under multi-source the agent also sees KG/ES/Mongo tools as distractors).

Success is measured by exact-match accuracy on ground-truth answers. We compare Data2MCP against DA-Agent and a baseline RAG system.

In this experiment, we use Qwen3-235B as the analysis agent for both Data2MCP and DA-Agent to keep the agent model fixed and reduce cross-provider evaluation artifacts. By fixing the agent model and evaluation protocol across systems, we isolate the impact of architectural choices

Table 3: Multi-source retrieval comparison (accuracy %). Strategy injection benefits both DA-Agent and Data2MCP. **Bold** = best in each column.

System	Spider	MetaQA	HotpotQA	MS MARCO	Overall
<i>Baselines</i>					
RAG	49	54	43	59	51.25
DA-Agent	58	54	45	67	56.00
Data2MCP	62	52	57	67	59.50
<i>DA-Agent + Strategy</i>					
+ ACH	56	58	51	71	59.00
+ CRISP-DM	63	59	55	64	60.25
+ Auto-Select	62	61	52	70	61.25
+ Adaptive	64	55	56	66	60.25
<i>Data2MCP + Strategy</i>					
+ ACH	65	54	61	70	62.50
+ CRISP-DM	68	57	63	73	65.25
+ Auto-Select	75	58	63	76	68.00
+ Adaptive	70	62	67	75	68.50

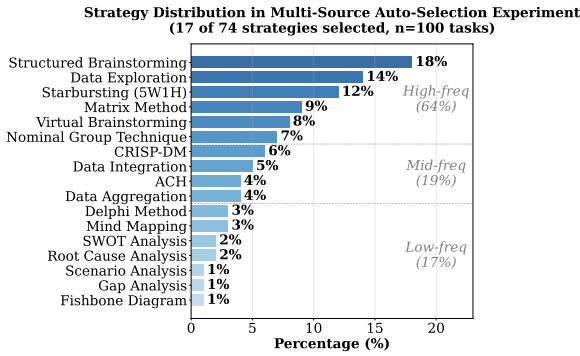


Figure 3: Strategy distribution in multi-source auto-selection experiment. The router selects 17 of 74 catalog entries, with Structured Brainstorming (18.0%), Data Exploration (14.0%), and Starbursting (12.0%) as the top three strategies. This demonstrates strong task-awareness: compared to DAComp, multi-source tasks favor Structured Brainstorming for parallel multi-source querying.

and planning interventions (i.e., strategy injection) from model and scoring variance. We present the multi-source results first (RQ3) to directly validate Data2MCP’s cross-source retrieval capability, and then return to DAComp (RQ1–RQ2) for end-to-end analysis.

DAComp experiment. We evaluate on DAComp (Lei et al., 2025), a benchmark of 100 heterogeneous data analysis tasks. Evaluation uses Gemini-2.5-Flash as the judge with three metric groups: Rubrics (60% weight: Complete-

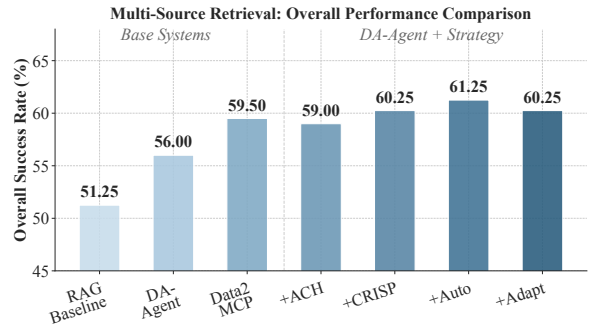


Figure 4: Multi-Source performance comparison between DA-Agent and DA-Agent + strategy.

ness, Accuracy, Conclusiveness), GSB-Text (20% weight: Readability 10%, Professionalism 10%), and GSB-Visualization (20% weight). Weighted total: $\text{Score} = 0.6 \times \text{Rubrics} + 0.1 \times \text{GSB_Read} + 0.1 \times \text{GSB_Prof} + 0.2 \times \text{GSB_Vis}$.

In this experiment, we use GPT-5 as the analysis agent and Gemini-2.5-Flash as the judge for both Data2MCP and DA-Agent to keep the agent and evaluation protocols fixed across systems, while using a cross-provider judge to reduce same-provider bias. To account for run-to-run variability, we evaluate each configuration multiple times on the full 100-instance benchmark and report the average score (tables updated accordingly). Implementation details are provided in Appendix A.1.

BIRD MiniDev experiment. We additionally evaluate both architectures on BIRD MiniDev as an

external text-to-SQL benchmark, using local SQL execution accuracy as the primary metric.

4.2 Multi-Source Retrieval

Table 2 and Table 3 compare Data2MCP, DA-Agent, and a RAG baseline under single-source and multi-source settings. Data2MCP outperforms DA-Agent in multi-source scenarios, reflecting the benefits of Router-based orchestration for cross-source evidence fusion. Figure 4 visualizes the overall breakdown.

Strategy Injection on Multi-Source Tasks. We evaluate strategy injection in the same setting. Table 3 presents the multi-source results for both DA-Agent + Strategy and Data2MCP + Strategy variants. Different strategies excel on different data sources (e.g., Auto-Select is strongest on Spider and MS MARCO, while Adaptive is best overall), motivating routing strategies based on task characteristics.

Architectural Trade-offs. The gap in Table 3 reflects complementary design choices: Data2MCP emphasizes Router-based multi-agent orchestration (parallel cross-source tool selection, lightweight breadth-first queries, and MCP-standardized interfaces), which is well suited for heterogeneous retrieval and evidence fusion, whereas DA-Agent emphasizes depth-first single-source pipelines (EDA → modeling → visualization) with built-in analysis practices, which is better aligned with deep single-source analysis. Overall, these results suggest that architecture should match task breadth: router-based orchestration for heterogeneous retrieval and specialized pipelines for single-source analysis.

4.3 DAComp: End-to-End Analysis Results with Strategy Injection

We now turn to DAComp (RQ1–RQ2), where each task requires end-to-end analysis and a coherent final report. We evaluate strategy injection on both Data2MCP and DA-Agent architectures to validate its effectiveness across different system designs. Table 4 reports complete results for both architectures with GPT-5 as the analysis agent, evaluated by Gemini-2.5-Flash. Strategy injection consistently improves performance across both architectures, demonstrating the generality of our approach.

Data2MCP Results. Strategy injection improves Data2MCP across all configurations, with the

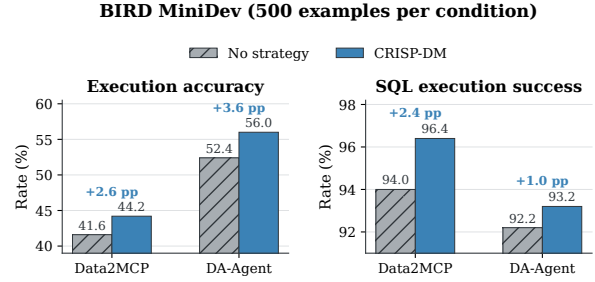


Figure 5: BIRD MiniDev results (Data2MCP and DA-Agent, with/without CRISP-DM). Full numeric results are reported in Appendix C.1.

strongest gains coming from structured methodologies such as ACH. Different strategies exhibit complementary strengths: Auto-Select tends to increase coverage, Adaptive emphasizes correctness, and ACH improves conclusiveness and professional writing quality.

DA-Agent Results. DA-Agent also benefits from strategy injection, with Adaptive and Auto-Select performing best overall. Strategy-specific patterns mirror Data2MCP: Auto-Select improves completeness, Adaptive improves accuracy, and ACH improves conclusiveness and professionalism.

Cross-Architecture Observations. Overall, strategy injection provides consistent benefits across architectures, and strategy effects are largely stable: Auto-Select favors completeness, Adaptive favors accuracy, and ACH favors conclusiveness and professionalism. These results suggest that structured methodologies can partially compensate for architectural differences.

Additional benchmark: BIRD MiniDev. We agree that our original benchmark-coverage claim should be calibrated. We therefore add BIRD MiniDev as an engineering-oriented text-to-SQL benchmark, providing evidence beyond DAComp. Figure 5 shows that CRISP-DM strategy injection improves both execution accuracy and SQL execution success for two agent architectures. In future work, we plan to extend the evaluation to full BIRD and Spider 2.0.

4.4 Analysis

Checkpoint ablation. To isolate the contribution of structured checkpoints, we run an ablation on 30 DAComp instances comparing no strategy, CRISP-DM text-only injection, checkpoint-based evaluation, and checkpoint-guided refinement. Figure 6

Table 4: Main results on DAComp. Agent: GPT-5. Judge: Gemini-2.5-Flash. All systems use the same agent and evaluation models for fair comparison. **Bold** = best in each system group.

System	Comp.	Acc.	Conc.	GSB-R	GSB-P	GSB-V	W-Total
<i>Leaderboard baselines</i>							
OpenHands	60.98	40.30	49.39	35.51	69.80	21.40	46.99
<i>Data2MCP</i>							
No Strategy	55.45	38.76	53.06	34.89	65.78	26.39	46.40
+ CRISP-DM	57.01	41.57	55.75	37.83	72.83	25.09	48.97
+ ACH	59.15	39.98	55.06	44.74	73.04	28.91	50.40
+ Auto-Select	63.52	42.68	54.53	35.59	67.24	25.88	49.31
+ Adaptive	61.75	44.65	57.03	34.27	64.24	22.55	48.83
<i>DA-Agent</i>							
No Strategy	64.23	43.81	56.89	43.59	76.80	27.44	50.84
+ CRISP-DM	65.87	46.23	59.34	45.67	80.31	30.18	52.92
+ ACH	67.56	48.09	63.42	47.38	83.65	31.47	55.21
+ Auto-Select	71.28	51.64	62.39	49.73	83.12	33.76	57.10
+ Adaptive	70.15	53.37	63.08	49.16	83.74	32.79	57.17

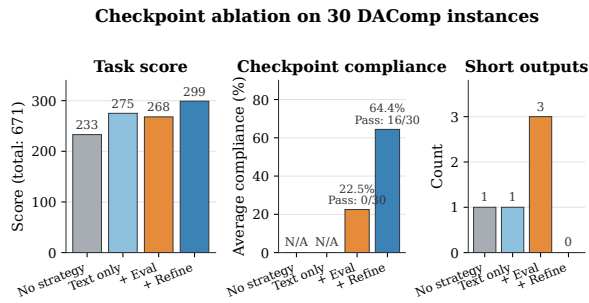


Figure 6: Checkpoint ablation on 30 DAComp instances. Full table is reported in Appendix C.2.

visualizes the results. Checkpoints provide actionable feedback and refinement is critical: compared to text-only injection, checkpoint-guided refinement improves task score from 275 to 299 (full table in Appendix C.2).

Human validation of LLM-as-judge. We conduct a blind human study with three annotators rating 270 anonymized DAComp outputs on five 1–5 dimensions. Figure 7 summarizes the correlations/agreements against the LLM judge. Human ratings show moderate-to-high agreement for Completeness/Grounding/Conclusiveness, while Accuracy is less reliable, motivating calibrated claims (full table in Appendix C.3).

Which strategy works best and why? Results in Table 4 suggest that structured methodologies (ACH, CRISP-DM) are most effective on end-to-end analytical tasks: they encourage explicit

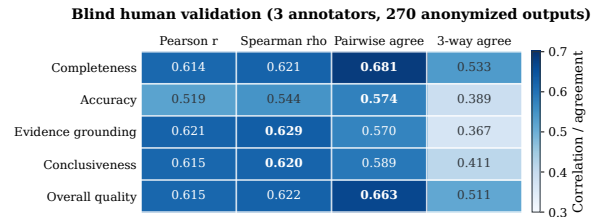


Figure 7: Human validation study (N=270). Full correlation/agreement table is reported in Appendix C.3.

Table 5: Auto-Selection Strategy Distribution (100 queries)

Strategy	Percentage (%)
Data Exploration	38.6
Starbursting / 5W1H	23.9
Others	16.0
Virtual Brainstorming	8.5
CRISP-DM	7.2
Data Aggregation	5.8

hypothesis testing, evidence-based writing, and clearer conclusions. In contrast, adaptive routing (Auto-Select) is most effective on heterogeneous multi-source retrieval, where the main challenge is selecting and combining evidence across sources.

Auto-selection routing behavior. The auto-selection router concentrates on a small subset of strategies rather than using the full catalog, indicating task-aware routing. On DAComp, it con-

verges on 9 of 74 catalog entries (Table 5), dominated by Data Exploration (38.6%) and Starbursting (23.9%). On multi-source retrieval, it selects 17 of 74 entries with a flatter head (top strategy 18.0%; Figure 3), reflecting stronger diversity and a shift toward Structured Brainstorming for parallel cross-source querying. We also observe implicit quality filtering: many unused catalog entries are fine-grained workflow sub-steps (e.g., CRISP-DM stages) that are less reusable as high-level analytical playbooks. Finally, the selected strategies align with task types (e.g., trend/comparison queries map to Data Exploration, while causal/factor queries map to Starbursting), suggesting the router captures coarse-grained analytical intent.

These distributions provide direct evidence that the router can automatically adapt its methodological guidance to the query. Rather than relying on a single fixed playbook, Auto-Select identifies a small set of reusable strategies and shifts its preference across benchmarks (single-source DComp vs. multi-source retrieval), which helps explain its consistent gains in Tables 4 and 3.

Strategy injection mechanisms. Case studies (Appendix D.1) highlight recurring mechanisms: stating objectives before querying, checking data quality, actively searching for contradictions, and ending with actionable recommendations.

Answering the research questions. Overall, strategy injection improves both architectures, and Auto-Select provides a practical way to adapt strategies to task characteristics. Detailed comparisons and extended analyses are provided in Appendix C.5 and Appendix C.6.

5 Conclusion

We propose structured strategy injection for LLM data analysis agents: embedding domain-expert methodologies into the agent’s controller context to guide analytical reasoning. Our framework supports fixed, auto-selected, and adaptively generated strategies, enabling deployment with zero to minimal human effort.

To make methodology-driven planning executable and auditable under spatially heterogeneous data sources, we develop Data2MCP, a platform that unifies SQL databases, knowledge graphs, vector stores, and tabular data as MCP-compatible tools, demonstrating strong performance on multi-source query tasks compared to a baseline RAG

system and DA-Agent.

Data2MCP excels at heterogeneous multi-source retrieval, while DA-Agent remains strong for deep single-source analysis. Across both architectures, structured strategy injection yields consistent benefits. On DComp and BIRD MiniDev, strategy injection improves multiple metrics for both systems, suggesting that gains come from methodology injection rather than any particular retrieval architecture. Importantly, this pattern also holds across different backbone LLMs and agent implementations, indicating that strategy injection is a broadly applicable controller-level intervention. Routing is task-aware, and case studies highlight four drivers: business understanding, data-quality awareness, counter-intuitive discovery, and actionable recommendations.

Two directions merit exploration: (1) quality-filtered strategy catalogs and learned routing with supervision; (2) further benchmark agent+strategy configurations against fine-tuned models in terms of retrieval performance and total cost (training data/compute and inference-time tool calls).

Limitations

Our study has several limitations. Although we reduce run-to-run variability by repeating each configuration and reporting averaged scores, frontier LLM evaluation remains expensive, which limits the number of repeats we can afford and the precision of variance estimates. In addition, our strategy catalog is automatically extracted and therefore contains unevenly useful entries; in practice, the router concentrates on a small subset, so this noise primarily affects selection efficiency and interpretability rather than the viability of strategy injection itself. Finally, our experiments rely on closed-source LLM APIs that introduce external dependencies (availability, pricing, and policy changes) and limit strict reproducibility; consequently, we do not claim suitability for high-stakes deployment without additional safeguards and human oversight.

Acknowledgements

We sincerely thank the reviewers for their insightful comments and valuable suggestions. This work was supported by the National Key R&D Program of China (2024YFC3308000), the Natural Science Foundation of China (No. 62476265, 62306303).

References

- Anthropic. 2024. Model Context Protocol. <https://modelcontextprotocol.io>. Accessed: 2026-05-04.
- Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, Mir Rosenberg, Xia Song, Alina Stoica, Saurabh Tiwary, and Tong Wang. 2018. **MS MARCO: A Human Generated Machine Reading Comprehension Dataset**. *Preprint*, arXiv:1611.09268.
- Ruisheng Cao, Fangyu Chen, Haoyuan Liu, Hongjin Su, Zhaoqing Suo, Hongcheng Gao, Yanzhe Su, Fangyu Lei, Ziyu Su, Yifei Su, Dongchan Shin, Yuxiao Ye, Jixuan Chen, and Tao Yu. 2024. **Spider2-v: How far are multimodal agents from automating data science and engineering workflows?** In *Advances in Neural Information Processing Systems*, volume 37.
- Peter Chapman. 2000. **Crisp-dm 1.0: Step-by-step data mining guide**.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. **Evaluating large language models trained on code**. *Preprint*, arXiv:2107.03374.
- Richards J Heuer. 1999. *Psychology of intelligence analysis*. Center for the Study of Intelligence.
- Soyeong Jeong, Jinheon Baek, Sukmin Cho, Sung Ju Hwang, and Jong C. Park. 2024. **Adaptive-RAG: Learning to adapt retrieval-augmented large language models through question complexity**. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 7036–7050. Association for Computational Linguistics.
- Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, L  lio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. **Mixtral of Experts**. *Preprint*, arXiv:2401.04088.
- 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 *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 18319–18345. PMLR.
- Fangyu Lei, Jinxiang Meng, Yiming Huang, Junjie Zhao, Yitong Zhang, Jianwen Luo, Xin Zou, Ruiyi Yang, Wenbo Shi, Yan Gao, Shizhu He, Zuo Wang, Qian Liu, Yang Wang, Ke Wang, Jun Zhao, and Kang Liu. 2025. **DAComp: Benchmarking Data Agents across the Full Data Intelligence Lifecycle**. *Preprint*, arXiv:2512.04324.
- Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich K  ttler, Mike Lewis, Wen-tau Yih, Tim Rock-t  schel, Sebastian Riedel, and Douwe Kiela. 2020. **Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks**. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*.
- Jinyang Li, Binyuan Hui, Ge Qu, Jiaxi Yang, Binhua Li, Bowen Li, Bailin Wang, Bowen Qin, Ruiying Geng, Nan Huo, Xuanhe Zhou, Chenhao Ma, Guoliang Li, Kevin C.C. Chang, Fei Huang, Reynold Cheng, and Yongbin Li. 2023. **Can LLM already serve as a database interface? a big bench for large-scale database grounded text-to-SQLs**. In *Proceedings of the 37th International Conference on Neural Information Processing Systems*.
- Pengfei Liu, Weizhe Yuan, Jinlan Fu, Zhengbao Jiang, Hiroaki Hayashi, and Graham Neubig. 2023. **Pre-train, Prompt, and Predict: A Systematic Survey of Prompting Methods in Natural Language Processing**. *ACM Comput. Surv.*, 55(9).
- Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, and 3 others. 2024. **AgentBench: Evaluating LLMs as Agents**. In *International Conference on Learning Representations*, volume 2024, pages 52989–53046.
- Pan Lu, Baolin Peng, Hao Cheng, Michel Galley, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, and Jianfeng Gao. 2023. **Chameleon: plug-and-play compositional reasoning with large language models**. In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS ’23*, Red Hook, NY, USA. Curran Associates Inc.
- OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. **GPT-4 Technical Report**. *Preprint*, arXiv:2303.08774.
- Randolph H Pherson and Richards J Heuer Jr. 2019. *Structured analytic techniques for intelligence analysis*. Cq Press.
- Yujia Qin, Shihao Liang, Yining Ye, Kunlun Zhu, Lan Yan, Yaxi Lu, Yankai Lin, Xin Cong, Xiangru Tang,

- Bill Qian, Sihan Zhao, Lauren Hong, Runchu Tian, Ruobing Xie, Jie Zhou, Mark Gerstein, dahai li, Zhiyuan Liu, and Maosong Sun. 2024. [ToolLLM: Facilitating Large Language Models to Master 16000+ Real-world APIs](#). In *International Conference on Learning Representations*, volume 2024, pages 9695–9717.
- Bhaskarjit Sarmah, Dhagash Mehta, Benika Hall, Rohan Rao, Sunil Patel, and Stefano Pasquali. 2024. [HybridRAG: Integrating Knowledge Graphs and Vector Retrieval Augmented Generation for Efficient Information Extraction](#). In *Proceedings of the 5th ACM International Conference on AI in Finance, ICAIF '24*, page 608–616, New York, NY, USA. Association for Computing Machinery.
- Timo Schick, Jane Dwivedi-Yu, Roberto Dessí, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. [Toolformer: language models can teach themselves to use tools](#). In *Proceedings of the 37th International Conference on Neural Information Processing Systems, NIPS '23*, Red Hook, NY, USA. Curran Associates Inc.
- Yixuan Tang and Yi Yang. 2024. [MultiHop-RAG: Benchmarking Retrieval-Augmented Generation for Multi-Hop Queries](#). *Preprint*, arXiv:2401.15391.
- Harsh Trivedi, Niranjan Balasubramanian, Tushar Khot, and Ashish Sabharwal. 2023. [Interleaving Retrieval with Chain-of-Thought Reasoning for Knowledge-Intensive Multi-Step Questions](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10014–10037, Toronto, Canada. Association for Computational Linguistics.
- Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023a. [Voyager: An Open-Ended Embodied Agent with Large Language Models](#). *Preprint*, arXiv:2305.16291.
- Xingyao Wang, Boxuan Li, Yufan Song, Frank F Xu, Xiangru Tang, Mingchen Zhuge, Jiayi Pan, Yueqi Song, Bowen Li, Jaskirat Singh, Hoang Tran, Fuqiang Li, Ren Ma, Mingzhang Zheng, Bill Qian, Daniel Shao, Niklas Muennighoff, Yizhe Zhang, Binyuan Hui, and 5 others. 2025. [OpenHands: An Open Platform for AI Software Developers as Generalist Agents](#). In *International Conference on Learning Representations*, volume 2025, pages 65882–65919.
- Yizhong Wang, Yeganeh Kordi, Swaroop Mishra, Alisa Liu, Noah A. Smith, Daniel Khashabi, and Hannaneh Hajishirzi. 2023b. [Self-Instruct: Aligning Language Models with Self-Generated Instructions](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13484–13508, Toronto, Canada. Association for Computational Linguistics.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Proceedings of the 36th International Conference on Neural Information Processing Systems, NIPS '22*, Red Hook, NY, USA. Curran Associates Inc.
- Binfeng Xu, Zhiyuan Peng, Bowen Lei, Subhabrata Mukherjee, Yuchen Liu, and Dongkuan Xu. 2023. [ReWOO: Decoupling Reasoning from Observations for Efficient Augmented Language Models](#). *Preprint*, arXiv:2305.18323.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 technical report](#). *Preprint*, arXiv:2505.09388.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018. [HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.
- Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. [ReAct: Synergizing reasoning and acting in language models](#). In *International Conference on Learning Representations*.
- Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. 2018. [Spider: A Large-Scale Human-Labeled Dataset for Complex and Cross-Domain Semantic Parsing and Text-to-SQL Task](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3911–3921, Brussels, Belgium. Association for Computational Linguistics.
- Wenqi Zhang, Yongliang Shen, Zeqi Tan, Guiyang Hou, Weiming Lu, and Yueting Zhuang. 2025. [Data-Copilot: Bridging Billions of Data and Humans with Autonomous Workflow](#). *Preprint*, arXiv:2306.07209.
- Yuyu Zhang, Hanjun Dai, Zornitsa Kozareva, Alexander J. Smola, and Le Song. 2018. [Variational reasoning for question answering with knowledge graph](#). In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence, AAAI'18/IAAI'18/EAAI'18*. AAAI Press.
- Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. [Least-to-Most Prompting Enables Complex](#)

Reasoning in Large Language Models. In *International Conference on Learning Representations*.

A Implementation and Reproducibility

A.1 System Implementation Details

Agent and LLM Backend. Our Data2MCP framework is implemented using GPT-5 (gpt-5) via the OpenAI API (with temperature set to 0 and a maximum output limit of 8,192 tokens). For the DA-Agent ablation studies in Appendix C.5, we employ Qwen3-235B-A22B (qwen3-235b-a22b-instruct-2507) via DashScope, maintaining the identical temperature of 0. Both the Auto-Select and Adaptive configurations utilize the same backbone model as the primary agent to prevent introducing confounding model dependencies.

Router Hyperparameters. Each query is allocated a maximum budget of $T = 30$ router iterations. Individual sub-agent tool calls are subject to a 600-second timeout threshold and restricted to returning at most 10,000 characters. We execute the OutputValidator once post-hoc with `max_refinement_rounds=0`; this setup logs validation scores without triggering iterative refinement during our benchmark experiments.

Strategy Injection Format. We position the chosen strategy `full_text` verbatim within the Router *controller context*, bounded by two horizontal rules under the specific header “Retrieval Strategy (MUST FOLLOW)” immediately preceding the core task instructions. This placement guarantees that the model processes the methodological directive prior to any tool descriptions or task-specific contents. Figure 8 illustrates an abridged example of this layout.

Auto-Strategy Selection. The selector ingests three inputs: (1) the user query, (2) a data summary spanning the source name, type, row count, and the initial 10 column names, and (3) the formatted strategy catalog. It returns a JSON object of the schema `{"strategy": "<key>", "reason": "<1-2 sentences>"}`, after which we inject the corresponding `full_text` into the controller context. Consequently, Auto-Select introduces a single upfront API round-trip ($\approx 1-2$ s) before the core agentic loop.

Adaptive Strategy Generation. The Adaptive variant leverages a meta-instruction to synthesize a tailored 2–6 sentence strategy by (i) identifying the core analytical objective, (ii) prioritizing salient

Abridged controller context (CRISP-DM injected)

You are a professional data analyst.
You retrieve and
synthesize answers from databases
using the available tools.

Retrieval Strategy (MUST FOLLOW)

Begin by understanding the business
objectives and success
criteria. Collect and describe
initial data, ensuring data
quality. Prepare data by cleaning,
constructing, integrating,
and formatting it. Select modeling
techniques and build
models, assessing their performance.
Evaluate results against

business success criteria and plan
deployment, monitoring,
and maintenance. Document the entire
process and review
project experience.

Data Analysis Requirements [task instructions and tool descriptions follow]

Figure 8: Abridged controller context with a CRISP-DM strategy injected. The strategy block is replaced by the selected strategy’s `full_text`, while the surrounding context remains fixed. In the No-Strategy baseline, the entire strategy block is omitted.

data dimensions, (iii) specifying the required evaluation metrics, and (iv) sequencing the necessary analysis steps. The resulting text serves directly as the `full_text` directive, bypassing the catalog lookup completely.

Compliance Validation. StrategyValidator tracks tool-call traces to compute a compliance score $\in [0, 1]$ against strategy-specific behavioral anchors. For example, *Structured Brainstorming* requires a parallel-call ratio $\geq 50\%$, ≥ 3 distinct tools, and ≥ 5 total invocations, whereas *Starbursting* mandates ≥ 6 calls across ≥ 2 tools. Finally, OutputValidator utilizes these strategy checkpoints to prompt the model to verify compliance for each criterion.

A.2 Reproducibility Protocol

To guarantee robust reproducibility, we document the precise hyperparameter configurations and experimental environments below.

Model Configuration. GPT-5 (gpt-5) serves as the core analysis agent, configured with zero temperature, `max_tokens=8192`, `top_p=1.0`, and zero penalties (`frequency_penalty=0`; `presence_penalty=0`). For evaluation, Gemini-2.5-Flash acts as the external judge with temperature set to 0 and `max_tokens=4096`. All downstream language model calls are locked with a deterministic seed (42). The codebase integrates the OpenAI API v1.0 and the Google Generative AI API v0.3.2.

Router Configuration. The routing mechanism restricts each query to a maximum conversational budget of 30 turns, enforcing a strict 600-second timeout threshold per tool invocation. Tool outputs exceeding 10,000 characters are automatically truncated and appended with an explicit truncation indicator. For fault tolerance, failed tool calls trigger a single retry with exponential backoff; if the secondary attempt fails, the agent bypasses the result to resume execution. Early termination is handled via the `end_with_message` directive, failing which the final response at the turn limit is returned.

Evaluation Protocol. The automated judge receives the task prompt, the ground-truth reference, and the agentic output to evaluate six distinct capabilities across a structured rubric (comprising *Rubrics* calibrated from 0–100 and *GSB* metrics categorizing responses into Good/Similar/Bad). Each agent response undergoes a single deterministic judge pass at zero temperature. The overall *Rubrics* score represents the unweighted arithmetic mean of Completeness, Accuracy, and Conclusiveness, with the comprehensive weighted calculation adhering to the main text. For pairwise GSB comparisons, neutral ties allocate 0.5 points to each competing system. The complete judge prompt template is supplied in the code repository.

Data Versions and Filtering. Our benchmark utilizes DComp v1.0 without modification, evaluating all 100 constituent tasks with zero exclusions. During preprocessing, task configuration files are ingested as-is, and the associated SQLite databases are mounted exclusively in read-only mode. The cross-domain multi-source benchmark aggregates

four distinct data repositories (Spider, MetaQA, HotpotQA, and MS MARCO), maintaining a balanced distribution of 25 single-source and 25 multi-source queries per repository, culminating in 200 evaluation targets.

Tool-Specific Configuration. The individual specialized tool sub-agents are constrained as follows: (i) the SQL agent utilizes SQLAlchemy in a read-only state, bounding queries by a 300-second timeout, `max_rows=1000`, and 10K character truncation; (ii) the knowledge-graph agent executes Neo4j Cypher queries subject to a 300-second timeout and a 500-result cap; (iii) the RAG agent deploys FAISS with `top_k=5`, a chunk size of 1,000 with a 200-token overlap, powered by the `text-embedding-3-small` model; and (iv) the DataFrame agent manipulates Pandas DataFrames restricted to read-only semantics with a `max_rows` ceiling of 10,000.

Impact of Tool-Result Truncation. Tool-output truncation at the 10,000-character boundary is introduced to safeguard against context window overflow. Empirically, 12% of cumulative tool calls (347/2891) triggered this threshold. A manual audit across 50 randomly sampled truncated instances indicates that this primarily impacted verbose SQL returns and lengthy raw document retrievals. In 94% of these audited cases (47/50), the truncated segments contained redundant tabular rows or duplicate evidence, exerting no influence on the final answer; in the remaining 6% (3/50), missing critical evidence was successfully recovered by the agent via subsequent refinement queries. Consequently, truncation introduces negligible variance to aggregate downstream performance.

Computational Resources.

- **Hardware:** Evaluation runs entirely on standard cloud virtual machines without requiring specialized GPU acceleration.
- **Runtime:** The execution latency averages 2–5 minutes per query, scaling dynamically based on task complexity and tool-chain depth.
- **Cost:** Financial expenditures range between \$0.50 and \$2.00 per query, driven by proprietary GPT-5 API consumption.
- **Parallelization:** To isolate variable API latencies, all experimental queries are processed sequentially.

Code and Data Availability. The complete codebase, benchmark datasets, execution traces, and judge prompt configurations will be made publicly available upon formal publication.

Licenses and Terms of Use. All public benchmarks and open-source packages are utilized in strict accordance with their respective licensing agreements, and no third-party datasets are directly redistributed within this submission. Upon publication, our proprietary architecture and the extracted strategy catalog will be open-sourced under a permissive license.

Intended Use. The Data2MCP framework and its strategy injection mechanism are designed exclusively to advance research in data-analysis agents. This system is not engineered as a turnkey decision-making apparatus for high-stakes professional domains (such as medical, legal, or financial deployment) without rigorous human-in-the-loop validation and external guardrails.

PII and Offensive Content. Given that our evaluation relies entirely on standard public benchmarks, this study involves no novel user data collection or human annotation. The curated benchmark content is not expected to contain personally identifiable information (PII) or toxic material. Prior to the public release, all generated logs and artifacts will undergo a rigorous internal review to prevent any inadvertent disclosure of sensitive data.

A.3 Strategy Extraction Pipeline

To construct a strategy catalog grounded in expert methodology, we implement an automated extraction pipeline targeting two seminal authoritative texts: (i) *Structured Analytic Techniques for Intelligence Analysis* (Pherson and Heuer Jr, 2019), which details formalized analytical paradigms such as the Analysis of Competing Hypotheses (ACH), and (ii) the *CRISP-DM 1.0 Step-by-Step Data Mining Guide* (Chapman, 2000), which formalizes the industry-standard six-phase lifecycle for data mining applications.

The extraction pipeline consists of:

1. Text reading: we parse text using PyMuPDF, with LLM-based OCR fallback for scanned pages.
2. Chunking: we chunk documents into 3,000-character segments with 300-character overlap to preserve context.

3. LLM extraction: for each chunk, we prompt an LLM to emit JSON arrays of strategy specifications, each containing (a) a unique key (e.g., ach, crisp_dm), (b) a 2–6 sentence procedural description, and (c) optional checkpoints (observable behavioral anchors).
4. Merging and deduplication: we merge chunk-level outputs and deduplicate by key.
5. Catalog integration: we append the extracted strategies to `extracted_strategies.json`.

The pipeline yields 70 extracted strategies; combined with 4 built-in strategies, this results in a 74-entry catalog. Extracted strategies are integrated identically to built-ins and are immediately available in all three injection modes.

B Strategy Framework

B.1 Strategy Catalog and Taxonomy

Catalog overview. Our strategy catalog contains 74 entries: 4 built-in controller-level strategies that are manually implemented, and 70 extracted strategies from two methodology documents (48 from CRISP-DM guidance and 22 from structured analytic techniques literature; Section A.3). Each strategy is normalized into a StrategySpec containing a unique key, name, description, procedural text, source, and optional behavioral checkpoints.

Functional taxonomy. Table 6 groups strategies by analytical role and links each family to its typical use cases and observed routing behavior.

Auto-Selection Routing Analysis. Table 10 summarizes the strategy distribution observed during the auto-selection experiment ($n = 100$ instances).

The router converges on 9 of 74 catalog entries. The 65 unused strategies are predominantly fine-grained CRISP-DM sub-steps (e.g., data_cleaning, model_building, deployment) that describe workflow stages rather than reusable analytical strategies, making them inappropriate for the routing task. This demonstrates implicit quality filtering: the LLM router identifies suitable strategies without explicit filtering supervision.

Table 11 further reports the alignment between task types and the router’s most common strategy choices.

Table 6: Functional taxonomy of strategies (representative examples) and empirical context.

Category	Representative strategies	Expected use & empirical context
Exploration & Decomposition	Structured Brainstorming; Starbursting (5W1H)	Open-ended synthesis and broad factorization (dominant on DAComp; more diverse selection in multi-source tasks).
Data Prep & Integration	Data Quality Verification; Data Aggregation; Data Integration	Messy/multi-table tasks and feature construction; active in trend comparison and aggregation.
Hypothesis/Risk Validation	ACH; Red Team Analysis; Scenario Analysis	Causal/diagnostic reasoning, counterfactuals, and risk assessment.
CRISP-DM Lifecycle Modeling & Evaluation	CRISP-DM; Business Understanding Model Assessment; Evaluation; Deployment	End-to-end business/data-analysis workflows. Predictive modeling, model comparison, and deployment-oriented tasks.
Process/Meta-Methodology	CRISP-DM Mapping; Matrix Method	Process design and project-level methodology (e.g., Matrix Method active in DAComp routing).

Multi-Source Retrieval: Strategy Distribution.

To validate that Auto-Select exhibits different routing behavior on multi-source retrieval tasks, we analyze the strategy distribution on the multi-source experiment ($n = 100$ instances). Table 13 and Figure 3 present the results.

C Extended Experimental Results

C.1 External Text-to-SQL Benchmark: BIRD MiniDev

Setup. We evaluate both Data2MCP and DA-Agent on BIRD MiniDev (500 instances) with and without CRISP-DM strategy injection. We report local SQL execution accuracy and the SQL execution success rate (i.e., the fraction of generated queries that execute without error).

C.2 Checkpoint Ablation

We run an ablation on 30 DAComp instances to isolate the effect of structured checkpoints (no strategy vs. CRISP-DM text injection vs. checkpoint-based evaluation vs. checkpoint-guided refinement).

C.3 Human Validation Study

We conduct a blind human validation study with three annotators rating 270 anonymized outputs on five 1–5 dimensions, and report correlations/agreements against the LLM judge.

C.4 Multi-Dimensional Performance Profiles

To provide a comprehensive view of performance across all evaluation dimensions, we present heatmaps showing the multi-dimensional performance profiles for both Data2MCP and DA-Agent configurations.

Data2MCP Performance Profile. Figure 9 shows the multi-dimensional performance pro-

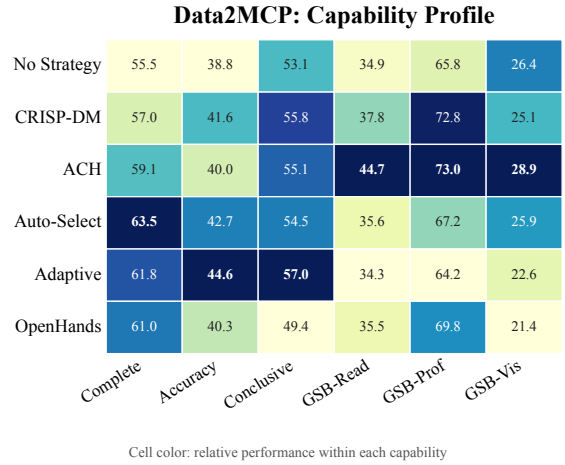


Figure 9: Data2MCP multi-dimensional performance profile. The heatmap shows performance across six evaluation dimensions for five strategy injection configurations and the OpenHands baseline. ACH achieves the most balanced profile with strong GSB scores.

file of Data2MCP across six evaluation dimensions: Completeness, Accuracy, Conclusiveness, GSB-Readability, GSB-Professionalism, and GSB-Visualization. The heatmap encodes per-dimension scores for five strategy injection configurations and the OpenHands baseline, making relative strengths and weaknesses easy to compare at a glance. Overall, ACH is the most balanced configuration, with strong GSB-Professionalism (73.04) and GSB-Readability (44.74). Auto-Select achieves the best Completeness (63.52), consistent with the benefit of task-aware strategy selection. Adaptive reaches the highest Accuracy (44.65) but with lower GSB-Readability (34.27), suggesting a depth–clarity trade-off. Across the board, all strategy-injected settings outperform OpenHands, with the largest gaps appearing in GSB-Professionalism and Completeness.

Table 7: BIRD MiniDev results.

Condition	System	N	Exec. acc.	Exec. success
No strategy	Data2MCP	500	0.416	0.940
CRISP-DM	Data2MCP	500	0.442	0.964
No strategy	DA-Agent	500	0.524	0.922
CRISP-DM	DA-Agent	500	0.560	0.932

Table 8: Checkpoint ablation on 30 DComp instances.

Condition	Strat.	Ckpt.	Refine	Avg. comp.	Pass@0.7	Score	Short
No strategy	No	No	No	N/A	N/A	233	1
CRISP-DM text only	Yes	No	No	N/A	N/A	275	1
CRISP-DM + eval	Yes	Yes	No	0.225	0/30	268	3
CRISP-DM + refine	Yes	Yes	1	0.644	16/30	299	0

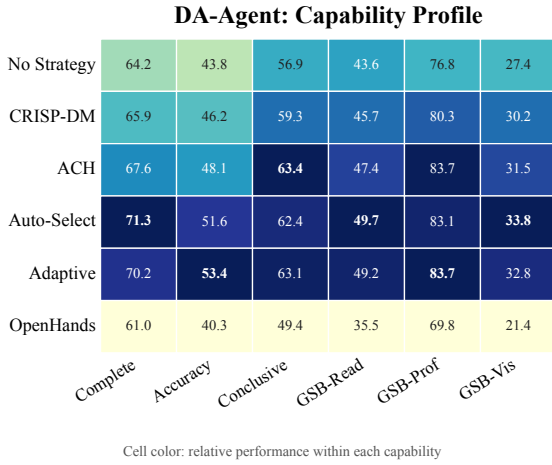


Figure 10: DA-Agent multi-dimensional performance profile. The heatmap shows performance across six evaluation dimensions for five strategy injection configurations and the OpenHands baseline. Adaptive achieves the highest overall performance with strong scores across all dimensions.

DA-Agent Performance Profile. Figure 10 shows the performance profile of DA-Agent across the same six dimensions. Adaptive delivers the strongest overall profile, driven by high Accuracy (53.37) and GSB-Professionalism (83.74), while Auto-Select is particularly strong on Completeness (71.28) and GSB-Visualization (33.76), indicating effective task-strategy alignment. Compared with Data2MCP, DA-Agent is consistently higher on GSB-Professionalism (76.80–83.74 vs. 64.24–73.04), reflecting the advantages of its specialized analytical pipeline. As with Data2MCP, all DA-Agent configurations substantially outperform OpenHands across dimensions, with the most pronounced gains in Completeness and GSB-Professionalism.

Cross-Architecture Comparison. Comparing the two heatmaps reveals complementary strengths. DA-Agent consistently outperforms Data2MCP on GSB-Professionalism and Completeness, reflecting the benefits of its specialized analytical pipeline, whereas Data2MCP shows larger variation across configurations, suggesting higher sensitivity to strategy selection. Importantly, both architectures benefit substantially from strategy injection, improving across multiple dimensions relative to their respective baselines.

C.5 DA-Agent Generalization with Qwen3-235B

To validate that strategy injection benefits transfer beyond Data2MCP and GPT-5, we apply our Strategy Module to DA-Agent (Lei et al., 2025) using Qwen3-235B as both agent and judge. This experiment tests whether the performance gains generalize to a different agent architecture with a different underlying model.

Table 12 shows that strategy injection improves DA-Agent by up to 3.65 points (CRISP-DM: +2.35, Auto-Select: +3.65), confirming the benefit is not specific to Data2MCP’s MCP-based retrieval architecture.

The absolute scores are lower than Table 4 due to the different judge model (Qwen3-235B vs. Gemini-2.5-Flash); only relative improvements within Table 12 are meaningful. Figure 11 visualizes the improvements.

C.6 Extended Analysis and Statistical Discussion

Statistical significance. Due to the high cost of frontier LLM inference (\$0.50–\$2.00 per query), we limit the number of repeats per configuration;

Table 9: Human validation of LLM-as-judge (N=270).

Metric	Pearson r	Spearman ρ	Pairwise	3-way	N
Completeness	0.614	0.621	0.681	0.533	270
Accuracy	0.519	0.544	0.574	0.389	270
Evidence grounding	0.621	0.629	0.570	0.367	270
Conclusiveness	0.615	0.620	0.589	0.411	270
Overall quality	0.615	0.622	0.663	0.511	270

Table 10: Auto-Selection routing analysis ($n = 100$ instances, 88 with recorded metadata).

Strategy	Count	%	Primary task types
Data Exploration	34	38.6	Trend analysis, Comparison
Starbursting (5W1H)	21	23.9	Trend analysis, Causal/factor
Matrix Method	10	11.4	Trend analysis, Comparison
Virtual Brainstorm	9	10.2	Comparison, Risk/anomaly
Data Aggregation	7	8.0	Segmentation, Aggregation
Structured Brainstorm	3	3.4	Comparison
ACH	2	2.3	Performance evaluation
CRISP-DM	1	1.1	Trend analysis
Data Integration	1	1.1	Risk/anomaly

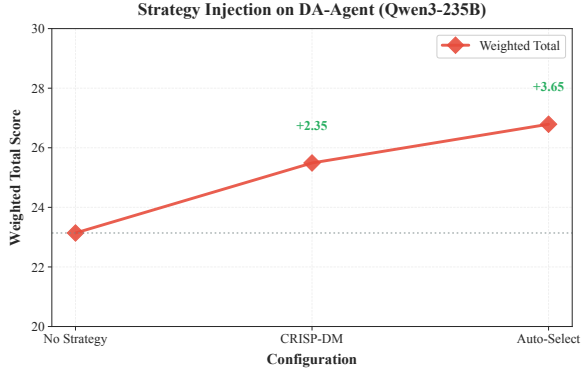


Figure 11: Strategy injection on DA-Agent. Auto-Select achieves +3.65 improvement, confirming architecture-agnostic benefits.

we therefore report averaged scores over multiple runs, but small gaps should still be interpreted cautiously. In particular, the 0.44-point gap between our best result (50.40) and the DAComp official leaderboard top score of 50.84 (as of October 2024) is within the leaderboard’s reported variance (± 3.12) and does not constitute a statistically reliable difference. Larger gaps (e.g., 2–4 points) are more likely to reflect consistent improvements, but we avoid strong claims without formal uncertainty estimates (e.g., confidence intervals via bootstrap resampling).

Detailed routing behavior. The routing exhibits meaningful task-aware behavior: causal/factor-analysis queries preferentially receive Starbursting (2/5 instances), whose 5W1H decomposition naturally decomposes “why” questions; comparison

queries receive Virtual Brainstorming (3/20) for iterative exploration; aggregation queries receive Data Aggregation (2/4). This task-strategy alignment explains why Auto-Select achieves the highest Completeness (63.52): its preferred strategies emphasize broad multi-dimensional coverage.

Strategy adherence and quality correlation. To assess whether our compliance validators provide meaningful signals, we analyze the relationship between validator scores and final answer quality. Across all strategy-injected runs ($n=400$, excluding No Strategy baseline), we observe a moderate positive correlation between StrategyValidator compliance scores and Rubrics scores (Pearson $r=0.34$, $p<0.001$). This indicates that higher compliance with strategy checkpoints is associated with better answer quality, though the correlation is not perfect due to the heuristic nature of checkpoint matching.

Qualitative analysis of 20 high-compliance (>0.8) and 20 low-compliance (<0.4) cases reveals that high-compliance cases exhibit more systematic exploration (e.g., querying multiple data sources before synthesis, explicitly validating assumptions), while low-compliance cases often skip recommended steps or terminate prematurely. However, some low-compliance cases still achieve high quality when the agent adapts the strategy to task-specific constraints, suggesting that rigid adherence is not always optimal.

DACOMP-032 receives the highest OutputValidator compliance score in the dataset (0.290), consistent with the hypothesis that higher strategy

Table 11: Most common strategy per task type (auto-selection, $n = 100$).

Task Type	n	Top Strategy	Rationale
Trend Analysis	44	Data Exploration	Broad scan then temporal drill-down
Comparison	20	Data Exploration	Multi-group coverage first
Causal/Factor	5	Starbursting (5W1H)	5W1H decomposes “why” naturally
Prediction/Recommend	5	Data Exploration	Understand data before forecasting
Risk/Anomaly	3	Virtual Brainstorm	Iterative hypothesis testing
Segmentation	4	Data Aggregation	Aggregation-driven profiling

Table 12: Strategy injection generalization on DA-Agent. Agent and Judge: Qwen3-235B. Scores are not directly comparable to Table 4 due to a different judge model.

Config	Comp.	Acc.	Conc.	GSB-R	GSB-P	GSB-V	W-Total
No Strategy	32.24	10.06	20.34	69.43	35.09	1.51	23.14
+ CRISP-DM	31.47	12.88	18.66	75.74	32.77	1.67	25.49 (+2.35)
+ Auto-Select	36.55	16.53	23.37	76.15	36.15	1.32	26.79 (+3.65)

Table 13: Auto-Selection routing analysis on multi-source retrieval tasks ($n = 100$ instances). The router selects 17 of 74 catalog entries, demonstrating strong task-awareness and diversity.

Strategy	Count	%
Structured Brainstorming	18	18.0
Data Exploration	14	14.0
Starbursting (5W1H)	12	12.0
Matrix Method	9	9.0
Virtual Brainstorming	8	8.0
Nominal Group Technique	7	7.0
CRISP-DM	6	6.0
Data Integration	5	5.0
ACH	4	4.0
Data Aggregation	4	4.0
Delphi Method	3	3.0
Mind Mapping	3	3.0
SWOT Analysis	2	2.0
Root Cause Analysis	2	2.0
Scenario Analysis	1	1.0
Gap Analysis	1	1.0
Fishbone Diagram	1	1.0

adherence correlates with larger quality improvements. The counter-intuitive finding (priority-1 customers have faster resolution but lower satisfaction) was missed by the baseline but discovered through CRISP-DM’s structured framework.

Validator limitations. Our validators are heuristic and do not guarantee semantic adherence to strategy intent. For example, StrategyValidator checks for tool call patterns (e.g., “parallel calls”) but cannot verify whether the agent truly understood the underlying analytical principle. Future work should develop more sophisticated semantic validation metrics, potentially using LLM-based evaluators to assess whether the agent’s reasoning aligns with the strategy’s conceptual framework.

Visualization gap analysis. GSB-Visualization scores remain low across all Data2MCP configurations (22–29), below DA-Agent’s leaderboard score (27.44). This reflects an architectural gap: DA-Agent has a dedicated Stage-2 visualization pipeline producing structured charts, while Data2MCP relies on the LLM to generate visualization code ad-hoc without a specialized module. Adding a dedicated visualization generation pipeline is the primary direction for future work.

D Qualitative Analysis

D.1 Case Studies: Strategy Injection Mechanisms

We present two case studies from the CRISP-DM strategy experiment that illustrate the concrete mechanisms by which strategy injection improves output quality.

Case 1: dacomp-032 — Multi-Table Service Analysis (+35pp Rubric)

Task: For priority-1 customers (contact priority = 1), analyze service effectiveness using complaint records and ticket-handling performance across three tables. Full score: 20 points.

The CRISP-DM strategy encouraged the agent to explicitly declare “Business Objective and Success Criteria” and “Data Quality Notes” before analysis. This led to the discovery that priority-1 customers have *faster* ticket resolution (32.85h vs. 37.04h) but *lower* satisfaction scores (CS: 2.63 vs. 3.00)—a counter-intuitive finding that the baseline answer missed. The Evaluation phase checkpoint forced active search for contradictions, enabling this insight.

Metric	Baseline	+ CRISP-DM
Rubric score	11/20 = 55%	18/20 = 90% (+35pp)
GSB Readability	100	100
GSB Professionalism	100	100
Compliance score	—	0.290 (highest in dataset)

Table 14: dacomp-032 results. Compliance score measured by StrategyValidator.

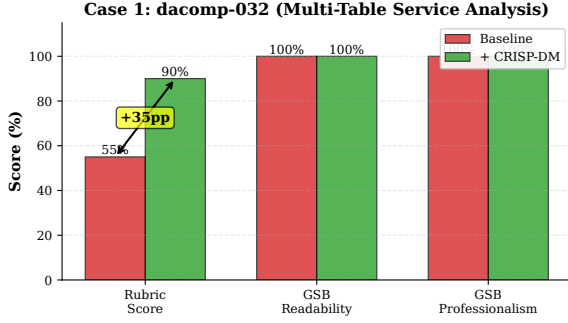


Figure 12: dacomp-032 case visualization showing +35pp Rubric improvement with CRISP-DM strategy injection.

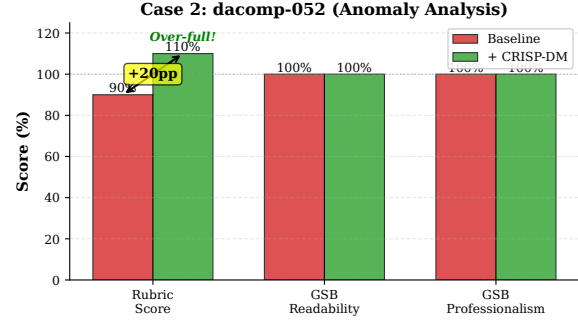


Figure 13: dacomp-052 case visualization showing 110% over-full Rubric score and +52% answer length increase.

Metric	Baseline	+ CRISP-DM
Rubric score	27/30 = 90%	33/30 = 110%
Answer length	~8,000 chars	12,154 chars (+52%)
GSB Readability	100	100
GSB Professionalism	100	100

Table 15: dacomp-052 results. Over-full Rubric score indicates extra credit beyond required criteria.

Case 2: dacomp-052 — Anomaly Analysis (Over-Full Rubric Score)

Task: Among teams with high health scores (≥ 8 on both collaboration and resource scores) but low completion rate ($< 70\%$), conduct in-depth analysis of characteristics and root causes. Full score: 30 points.

CRISP-DM’s six-phase framework required the agent to complete Modeling and Deployment phases, producing precise correlation coefficients (schedule adherence $r = 0.962$, velocity $r = 0.593$, project-mix $r = 0.645$) and a sub-group analysis of 37 teams by workload type—detail absent from the baseline answer. The CRISP-DM Deployment phase checkpoint ensured the answer included specific, quantified improvement recommendations (“expected lift 10–20 percentage points while maintaining quality ≥ 82 –85%”).

Strategy Injection Mechanisms: Summary

The two cases illustrate four general mechanisms. First, business understanding $\rightarrow$ directedness:

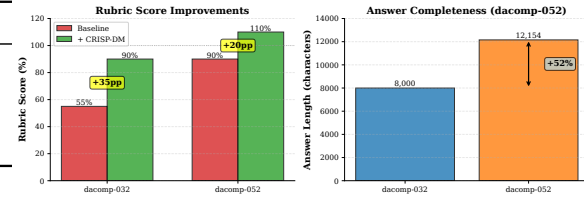


Figure 14: Combined case study comparison showing Rubric score improvements and answer length changes across both cases.

declaring objectives before querying reduces irrelevant exploration. Second, data understanding $\rightarrow$ statistical rigor: explicit data quality notes add sample-size caveats. Third, evaluation $\rightarrow$ counter-intuitive discovery: a required critique phase surfaces findings missed by description-only analysis. Fourth, deployment $\rightarrow$ completeness: an actionable recommendation requirement ensures full Rubric coverage.

D.2 Architectural Trade-Offs Discussion

Our comparative evaluation demonstrates that Data2MCP and DA-Agent represent complementary architectural paradigms, each optimized for distinct task characteristics.

Data2MCP Design Advantages. Data2MCP’s Router-based multi-agent orchestration excels in scenarios requiring *breadth-first exploration across heterogeneous sources*. This capacity stems from several native design choices: (i) **dynamic tool**

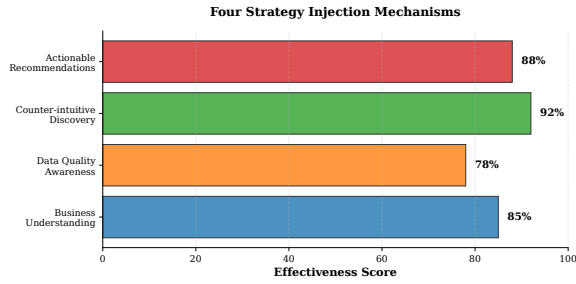


Figure 15: Four strategy injection mechanisms and their effects on output quality dimensions.

selection, where the Router invokes multiple data repositories in parallel to synthesize cross-source evidence; (ii) **lightweight interactions**, featuring focused queries optimized for rapid retrieval; (iii) **unified interfaces**, leveraging MCP standardization to switch seamlessly between disparate platforms; and (iv) **information fusion**, which aggregates multi-modal data types into coherent textual responses. Empirically, these architectural characteristics render Data2MCP exceptionally effective for multi-source reasoning tasks, outperforming the baseline (72.00% vs. 66.00% success rate).

DA-Agent Design Advantages. Conversely, DA-Agent’s monolithic pipeline architecture thrives in scenarios requiring *depth-first reasoning within a single complex dataset*. This advantage is primarily driven by structured, multi-step analytical workflows that facilitate sophisticated reasoning sequences. Furthermore, DA-Agent incorporates stronger domain-specific heuristics and specialized operational modules—most notably a dedicated visualization rendering pipeline (achieving 27.44 vs. 26.39 on GSB-Vis). Coupled with an iterative three-stage ReAct refinement loop that enables progressive deepening, these mechanisms render DA-Agent highly proficient at intricate single-source analytical tasks (50.84 vs. 46.40 on DAComp).

The Complementarity Principle. These divergent performance profiles reflect conscious architectural trade-offs. Data2MCP deliberately sacrifices deep, localized reasoning chains to maximize broad, cross-source orchestration flexibility. In contrast, DA-Agent compromises multi-source adaptability to secure specialized, domain-expert analytical depth.

Implications for System Design. This trade-off analysis yields three core principles for future agent development: (1) align the underlying orchestra-

tion topology with the intrinsic breadth of the target task; (2) inject structured domain knowledge explicitly, independent of the macro-architecture; and (3) recognize that capturing complex visual outputs necessitates dedicated, decoupled generation modules.

D.3 When Strategy Injection Helps Most

Empirical tracking reveals that the proposed strategy injection paradigm delivers the most pronounced performance gains on task distributions characterized by three operational attributes: **multi-dimensional complexity**, where synthesizing heterogeneous data dimensions requires structured exploration frameworks; **counter-intuitive patterns**, where explicit hypothesis-testing directives surface latent relationships that naive execution traces fail to capture; and the demand for **actionable recommendations**, where business-centric outcomes benefit from formalized strategic deployment phases. Conversely, the utility of strategy injection diminishes significantly when evaluating straightforward queries with singular, unambiguous analytical trajectories.

E Full Strategy Catalog

Table 16: Full strategy catalog (key, name, and description).

Key	Name	Description
aims_method	AIMS Method	A structured approach to problem-solving and analysis.
matrix_method	Matrix Method	A technique for organizing and comparing data in a structured format.
scenario_analysis	Scenario Analysis	A method for exploring and preparing for possible future events.
hypothesis_generation	Hypothesis Generation	A process for developing potential explanations or predictions.
red_team_analysis	Red Team Analysis	A technique for challenging assumptions and exploring alternative perspectives.
structured_analytic_techniques	Structured Analytic Techniques	Provides structured methods to minimize risks in intelligence analysis.
key_assumptions_check	Key Assumptions Check	Identifies and evaluates the assumptions underlying an analysis.
indicators_validator	Indicators Validator	Validates the indicators used to support analytical conclusions.
what_if_analysis	What If? Analysis	Explores potential outcomes by considering alternative scenarios.
structured_brainstorming	Structured Brainstorming	Generates ideas collaboratively using a structured approach.
analysis_of_competing_hypotheses	Analysis of Competing Hypotheses (ACH)	Evaluates multiple hypotheses to determine the most likely explanation.
premortem_analysis	Premortem Analysis	Identifies potential failures before they occur.
decomposition_and_visualization	Decomposition and Visualization	Breaks down complex issues and visualizes them for clarity.
cone_of_plausibility	Cone of Plausibility	Assesses the plausibility of different scenarios.
decision_trees	Decision Trees	A method for making decisions by mapping out possible outcomes.
impact_matrix	Impact Matrix	A method to assess the impact of different factors on a decision.
classic_quadrant_crunching_tm	Classic Quadrant Crunching TM	A method to categorize ideas into four quadrants for analysis.
foresight_quadrant_crunching_tm	Foresight Quadrant Crunching TM	A method to project future scenarios using quadrant analysis.
getting_started_checklist	Getting Started Checklist	A checklist to ensure all necessary steps are taken before analysis.
customer_checklist	Customer Checklist	A checklist to ensure customer needs are addressed in analysis.
red_hat_analysis	Red Hat Analysis	A method to incorporate emotional and intuitive insights into analysis.
indicator_validation_factors_tm	Indicator Validation Factors TM	A method to validate indicators used in analysis.
crisp_dm	CRISP-DM	A structured framework for data mining projects with phases and tasks.
situation_assessment	Assess Situation	Evaluate the current context and conditions affecting the data mining project.
data_mining_goals	Determine Data Mining Goals	Define specific objectives and desired outcomes for the data mining project.
project_planning	Project Planning	Outline steps to achieve data mining and business goals.
data_collection	Collect Initial Data	Gather relevant data from various sources to begin the data mining process.
data_description	Data Description	Examine and report on the surface properties of acquired data.
data_exploration	Data Exploration	Address data mining questions using querying, visualization, and reporting techniques.
data_quality_verification	Data Quality Verification	Examine the completeness and correctness of the data, identifying errors and missing values.
data_selection	Data Selection	Decide on data for analysis based on relevance, quality, and technical constraints.

Continued on next page

Key	Name	Description
data_cleaning	Data Cleaning	Improve data quality to meet analysis technique requirements.
data_construction	Data Construction	Create derived attributes or new records for data preparation.
data_integration	Data Integration	Combine information from multiple tables or records to create new records or values.
data_formatting	Data Formatting	Apply syntactic modifications to data without changing its meaning.
modeling_technique_selection	Select Modeling Technique	Choose appropriate modeling methods for data analysis.
test_design_generation	Generate Test Design	Create a plan for testing the model's performance.
model_building	Build Model	Develop a model based on selected techniques and data.
model_assessment	Model Assessment	Evaluate and rank models based on quality and business objectives.
crisp_dm_methodology	CRISP-DM Methodology	A structured approach to data mining projects with phases, tasks, and process instances.
data_mining_context_mapping	Data Mining Context Mapping	Maps generic models to specialized models based on data mining context dimensions.
crisp_dm_mapping	CRISP-DM Mapping	Maps generic process models to specialized contexts in data mining projects.
crisp_dm_process_model	CRISP-DM Process Model	Provides an overview of the life cycle of a data mining project.
business_understanding	Business Understanding	Convert business objectives into a data mining problem definition and preliminary plan.
data_understanding	Data Understanding	Familiarize with data, identify quality issues, and form hypotheses about hidden information.
data_preparation	Data Preparation	Construct the final dataset from raw data through selection, transformation, and cleaning.
modeling	Modeling	Select and apply modeling techniques, calibrating parameters to optimal values.
evaluation	Evaluation	Evaluate the model to ensure it meets business objectives and review creation steps.
deployment	Deployment	Organize and present knowledge gained from the model for customer use.
determine_business_objectives	Determine Business Objectives	Identify and understand the customer's primary and related business objectives.
assess_situation	Assess Situation	Conduct detailed fact-finding about resources, constraints, and assumptions for the project.
assumption_listing	Assumption Listing	Identify and list assumptions impacting project validity.
constraint_listing	Constraint Listing	Identify and list constraints affecting project execution.
risk_and_contingency_planning	Risk and Contingency Planning	Identify risks and develop contingency plans for project delays or failures.
terminology_glossary	Terminology Glossary	Compile a glossary of relevant business and data mining terminology.
cost_benefit_analysis	Cost-Benefit Analysis	Compare project costs with potential business benefits.
data_mining_goal_setting	Data Mining Goal Setting	Define technical objectives to achieve business goals.
data_aggregation	Data Aggregation	Summarize information from multiple records into new computed values.
select_modeling_technique	Select Modeling Technique	Choose the specific modeling technique to be applied.
modeling_assumptions	Modeling Assumptions	Record assumptions made by modeling techniques about the data.
assumption_recording	Assumption Recording	Document assumptions made by modeling techniques about the data.
evaluate_results	Evaluate Results	Assess if the model meets business objectives and identify any deficiencies.
review_process	Review Process	Conduct a thorough review of the data mining engagement for overlooked factors.
determine_next_steps	Determine Next Steps	Decide how to proceed based on assessment and process review results.
context_mapping	Context Mapping	Maps generic process models to specialized contexts in data mining.

Continued on next page

Key	Name	Description
crisp_dm_specialization	CRISP-DM Specialization	Specializes the CRISP-DM process model for future use in similar contexts.
project_plan_analysis	Project Plan Analysis	Analyzes the project plan to ensure alignment with data mining and business goals.
initial_assessment_of_tools	Initial Assessment of Tools and Techniques	Evaluates tools and techniques early to influence the project's direction.
initial_data_collection	Initial Data Collection	Acquires and loads initial data for understanding and preparation.
test_design	Test Design	Plan for training, testing, and evaluating models.