Discussion of LAMBDA: A Large Model Based Data Agent

Xihong Lin
Harvard University

1 Introduction

(Sun et al., 2025a) introduce LAMBDA, a natural-language-driven, two-agent architecture for data analysis. In this framework, a programmer agent generates executable code from user instructions, while an inspector agent evaluates execution errors and proposes possible corrections. This design represents a significant advancement in an agent-based data analysis system, offering a practically useful tool for both practitioners and educators.

LAMBDA includes several additional strengths. It provides a mechanism that incorporates domain-specific knowledge into analytical workflows, allowing the system to adapt to specialized applications and improving the interpretability of results. Its human-in-the-loop design further enables users to interactively adjust individual steps of the analysis, positioning LAMBDA as a collaborative assistant rather than a black-box automated pipeline.

The authors demonstrate LAMBDA's versatility and performance across diverse datasets, including genomic, image, and text data, and illustrate how it empowers non-technical domain scientists to perform various data analyses with ease. This interface is especially valuable in educational settings, where learners benefit from engaging directly with iterative refinement of models, code implementations, and analytical results.

This discussion focuses on enhancing the practical utility of LAMBDA through improving analytic and scientific reasoning capabilities and by developing two collaborative agents: an analysis-planning agent and a result agent.

2 The Analysis-Planning Agent with Reasoning

The two-agent setup in LAMBDA (programmer + inspector) translates user instructions into code and iteratively corrects execution errors. This workflow assumes users already know which analyses to run, what statistical methods to use, and what domain-specific community standards to apply. This can be a challenging expectation for non-expert researchers. Extending LAMBDA with an analysis-planning agent could address this limitation. Such an agent interacts with users to construct an analysis plan that answers the scientific questions of interest, adheres to field norms, and incorporates appropriate analytic reasoning.

Many disciplines rely on well-established analytic pipelines and conventions from quality control, analysis, validation, to result interpretation. For example, Genome-Wide

Association Studies (GWAS) commonly require genotype QC (Hardy-Weinberg equilibrium tests, missing-call filters), population-structure correction using ancestry principal components, association testing with mixed models to handle relatedness, and multiple-testing correction, e.g., Bonferroni. These steps represent community consensus developed over years of methodological and empirical research and are critical for reproducible science.

An analysis-planning agent would collaborate with users to:

1. Identify the scientific domain of the data and refine the user's research goals and questions.

2. Map the user's research questions to appropriate analytic procedures by applying statistical and domain-specific reasoning rules and tailoring them to the data at hand.

3. Propose a comprehensive analysis pipeline that adheres to community standards and lays the groundwork for a comprehensive analysis report.

Such a pipeline would include QC procedures, data cleaning and preprocessing, exploratory analysis, modeling choices, result validation strategies, plans for generating intermediate results to solicit user feedback, and components of the final analysis report that addresses the user's scientific questions while following field conventions.

In doing so, rather than relying on users to specify individual tasks, the analysis-planning agent could partner with the users to generate a domain-aware workflow that addresses the scientific questions of interest, explains each analytic step, flags common pitfalls, and highlights matters requiring user input. Embedding domain knowledge into developing data preprocessing and modeling steps would allow the agent to work with the user to reason about the appropriateness of the analytic choices and make specific recommendations so the analysis plan mirrors best practices to address the scientific questions of interest. The developed analysis plan can then be pushed into the programmer and inspector agents for execution. In doing so, this agent would transform LAMBDA into a domain-aware analytic assistant that ensures analyses both answer the user's scientific questions and align with community standards, with human oversight and customization.

3 The Result Agent with Reasoning

Trustworthy findings and contextualized result interpretation are essential for an intelligent data analysis agent to be practically useful. LAMBDA's knowledge-integration mechanisms provide a good foundation for developing a result agent that collaborates with the users to map analytic outputs onto field-specific interpretive frameworks and reporting norms, while interpreting results with deeper scientific insight and suggesting additional analyses.

Domain knowledge and community standards guide not only how data are analyzed but also how results should be validated, interpreted, and reported. For example, GWAS findings are typically validated in independent cohorts and followed by functional follow-up of implicated variants. A result agent should therefore be able to map results to domain-field-specific established interpretive frameworks, compare findings with the literature to assess novelty and significance, evaluate result validity, provide interpretations that incorporate the existing domain-specific knowledge, and caution against overinterpretation. Accomplishing this requires scientific reasoning that integrates curated literature, ontologies, domain-specific reporting templates, and field guidelines.

Human-in-the-loop interaction is a strength in LAMBDA, although its current implementation focuses mainly on execution code error correction rather than deeper scientific dialogue. To increase real-world utility, human-agent dialogue should be adapted in multiple agents, including both the analysis-planning and result agents by engaging users in deeper scientific conversation. Specifically, the analysis-planning agent collaborates with the user on designing conversational mechanisms that develop a comprehensive analysis plan that meets the user's scientific goals. The result agent enhances result intelligence by collaborating with the user through a dialogue to examine the results, make suggestions to improve analysis, and provide result interpretation. For example, incorporating interactive visual interfaces in the result agent can allow users to examine whether the analysis plan and the results make sense by casting them in the literature, and provide recommendations for additional analysis. This feedback loop would allow the system to adapt dynamically and ensure that the analytical results remain aligned with the user's scientific goals, while providing scientific insights that guide further analysis and support meaningful, trustworthy result interpretation.

Achieving this alignment requires improving analytic reasoning from procedural checks to conceptual and consequential evaluation. Although the programmer and inspector agents execute the code following the user instructions and fix execution errors, they do not evaluate whether the analytic procedure and results are scientifically meaningful and casting them within the literature, and do not provide scientific insights of the results. To improve analytical and scientific reasoning, future development could explore multi-layer reasoning architectures, e.g., assisted by a reasoning agent, that evaluate the conceptual validity of an analysis plan and the strengths and weaknesses of results, combining statistical and data-science reasoning with domain-expert reasoning to ensure scientific justification and findings are sound and interpretable and make suggestions of additional analyses based on the current results. A tree-of-thought-style component could further deepen reasoning by evaluating alternative analytic routes and interpretations before committing to a specific approach.

4 Summary

LAMBDA represents a significant step toward a natural-language-driven system that translates user instructions into executable analyses, leveraging domain knowledge and human-in-the-loop refinement. To enhance its real-world utility, we suggest augmenting LAMBDA with improved statistical and data-science reasoning and domain-expert reasoning capabilities, along with two collaborative agents: an analysis-planning agent that partners with users to design analytic pipelines aligned with their scientific goals and domain-specific standards, and a results agent that assists users in interpreting results, contextualizing them within field standards and existing literature, and identifying potential additional analyses to enhance scientific discovery. Together, these enhancements could ensure that analyses and results remain scientifically rigorous, interpretable, and more closely aligned with users' objectives, as well as providing greater practical utility and scientific contributions.

References

Sun, M., Han, R., Jiang, B., Qi, H., Sun, D., Yuan, Y., and Huang, J. (2025a). Lambda: A large model based data agent. *Journal of the American Statistical Association*, pages 1–13.

Sun, M., Han, R., Jiang, B., Qi, H., Sun, D., Yuan, Y., and Huang, J. (2025b). A survey on large language model-based agents for statistics and data science. *The American Statistician*, pages 1–14.