

# Response to Discussion 4: Self-Evolving Agents and Data-Aware Retrieval

Team LAMBDA  
The Hong Kong Polytechnic University  
Hong Kong SAR, China

We are grateful to Professor Fan Zhou and Professor Bang Liu for their insightful and forward-looking discussion of our work on LAMBDA [Sun et al., 2025a]. Their expertise in large language models and agentic systems provides valuable perspectives on how LAMBDA can be extended to become a more adaptive, intelligent, and collaborative data analysis system. They outline several promising directions that align closely with our ongoing research agenda. We address each of their key suggestions below.

Our survey on large language model-based agents for statistics and data science [Sun et al., 2025b] provides the foundational context for understanding the current landscape and future directions of AI-powered data analysis systems.

## 1 Toward Self-Evolving Agent Design

Professor Zhou and Professor Liu correctly observe that the interactive Programmer-Inspector design in LAMBDA focuses on external feedback to improve performance, but each task is approached anew, and any improvements or mistakes made during the iterative process do not persist beyond that single session. They suggest complementing external-feedback loops with mechanisms for the agent to learn from its own experiences over time.

We agree that this represents an exciting direction for future research. A self-evolving agent that accumulates knowledge as it solves more tasks could gradually enhance its capabilities. The authors cite prior work such as Voyager, which developed an LLM-driven agent in a Minecraft environment that continuously learns new skills by storing successful solutions in a growing library and retrieving them for future tasks.

Translating this idea to a data-analysis agent, we might equip LAMBDA with a long-term memory or repository of past problem-solving episodes. Over time, the agent could build up a library of useful code snippets, data transformations, and analytical techniques that it discovered or received as feedback. Faced with a new task, the agent could retrieve relevant past cases or code from its memory to guide the current solution.

In addition to skill reuse, the agent could maintain an episodic memory of mistakes and fixes, enabling it to avoid repeating errors. Techniques like Reflexion have demonstrated the power of an agent generating self-reflective feedback and storing it in memory to improve

later decisions. By internally recording what went wrong and why, an LLM agent can adjust its reasoning on subsequent attempts without external intervention.

Professor Zhou and Professor Liu also point out that LAMBDA currently treats each task in isolation, failing to carry over knowledge from previous successes or failures. To address this, designing Self-Evolving Agents is a priority for future work. This would incorporate a long-term memory module that allows the agent to curate a personal library of “successful code skills” and “debugged error patterns,” enabling compound growth in capability and preventing the repetition of past mistakes across different sessions.

Incorporating these ideas, a future LAMBDA-like agent could become increasingly autonomous and proficient the more it is used. It would start to exhibit continual learning, refining its internal knowledge base of coding patterns, analytical methods, and domain insights with each user interaction. Such a self-evolving design moves closer to an AI data scientist that, like a human, grows in expertise through experience.

## 2 Domain Knowledge Integration and RAG

Professor Zhou and Professor Liu provide an excellent analysis of the current limitations of LAMBDA’s Knowledge Integration Mechanism. They correctly note that the retrieval process relies solely on natural language instructions provided by the user and lacks any direct conditioning on the dataset itself or intermediate analytical outputs. As a result, the system may fail to surface relevant analytical strategies that are well-suited to the statistical characteristics of the data at hand.

This limitation is particularly important in empirical data analysis, where datasets with similar statistical structures often benefit from similar modeling pipelines. For instance, time series datasets typically share characteristics such as autocorrelation, seasonality, and missing data patterns regardless of the application domain. However, a purely language-driven retrieval mechanism may not recognize the shared structure if these similarities are not explicitly described in the prompt.

Moreover, many useful patterns in data science practice are not tied to specific task names or domains but emerge from data-driven considerations such as skewness, multimodality, sparsity, or high cardinality. A retrieval system unaware of these properties risks retrieving generic or suboptimal routines.

We have developed a Data-Aware Retrieval framework that synthesizes user intent with statistical characteristics of the dataset. Rather than treating the dataset as a passive input, we make it an active component in the retrieval process. Specifically, retrieval is conditioned on a Data Profile—a vector that encodes key meta-features of the dataset, such as dimensionality, skewness, sparsity, and correlation structure.

In this framework, retrieval operates jointly on both the data context and the semantic context. The final retrieval score is computed as a weighted sum:  $\text{Score} = \alpha S_{\text{data}} + \beta S_{\text{semantic}}$ , where  $S_{\text{data}}$  measures the similarity between the target dataset’s profile and the metadata associated with candidate code snippets, and  $S_{\text{semantic}}$  measures the alignment with user instructions. By fine-tuning the weights and the underlying embeddings, the system retrieves knowledge that is not only semantically appropriate but also data-consistent.

Regarding the suggestion to build a dynamic knowledge base, we agree that a large,

continuously updated repository of statistical methods and data science techniques would significantly enhance LAMBDA’s capabilities. Such a repository could contain foundational methods like regression and classification algorithms as well as domain-specific techniques such as time-series forecasting, anomaly detection, and deep learning-based approaches. It could also store real-world challenges and datasets from competitions along with corresponding solutions and best practices.

However, building and maintaining such a knowledge repository presents several challenges. Ensuring the quality, diversity, and accuracy of the methods included is crucial, as is keeping the repository updated. Additionally, efficient and context-aware retrieval remains an open challenge, particularly when integrating the repository with the RAG framework.

### **3 Enhanced Statistical Reasoning via Reinforcement Learning**

Professor Zhou and Professor Liu suggest framing automated data analysis as a reinforcement learning environment where the agent learns to reason through data analysis rather than merely operating over it. We find this suggestion compelling and have begun exploring this direction.

Recent breakthroughs in language model reasoning, such as DeepSeek-R1, demonstrate that reinforcement learning can substantially enhance Chain-of-Thought capabilities. By optimizing outcome-based reward signals, these models are able to develop generalizable reasoning strategies that extend beyond the limitations of traditional supervised fine-tuning.

LAMBDA’s existing modular architecture provides an excellent foundation for building such a reinforcement learning-based environment. In this setup, agent actions could include invoking specific data transformations, revising prompts or code, or triggering self-correction routines. The environment states would evolve to reflect the changing analytic context, including updates to data transformations or the analytic model itself. Rewards could be defined based on various outcomes, such as predictive performance, efficiency metrics, or intermediate objectives like interpretability and statistical coherence.

This RL formulation offers several unique advantages. Outcome-supervised learning allows rewards to be defined based on downstream task success, encouraging the agent to explore effective reasoning paths. Scalability through dataset diversity exposes the agent to a wide range of distributional patterns, helping it generalize reasoning strategies across domains. The rich state and action space fosters the emergence of structured reasoning and decision-making strategies essential for solving complex real-world problems.

Framing data analysis as an RL environment thus presents not only a systems-oriented opportunity but also a principled setting for studying how reasoning abilities can emerge, specialize, and transfer within agent-driven language models.

### **4 Expanding Agentic Roles and Workflows**

Professor Zhou and Professor Liu note that LAMBDA’s streamlined two-agent architecture (Programmer and Inspector) emphasizes simplicity and clarity but may need to be extended

for more complex and open-ended analytical projects. They suggest increasing the granularity of specialization among AI agents, with separate agents for data preprocessing, model building, visualization, and reporting.

While we acknowledge this potential, we approach this expansion with caution. Although agent frameworks like AutoGen and MetaGPT demonstrate the power of specialized multi-agent systems, recent work shows that agentic capability does not improve simply by increasing the amount of supervision data. We believe this hypothesis similarly applies to the number of agents; mastering agency requires understanding its essence, not scaling the role of agents.

Data science differs fundamentally from general software engineering or scientific discovery, where modularity often allows agents to operate independently. Data analysis is inherently state-dependent and sequential: each step relies on the exact variable definitions, transformations, and memory state established previously. Introducing too many specialized agents can exacerbate the risk of “Context Confusion,” where the overhead of serializing state and communicating context between agents outweighs the benefits of specialization.

Therefore, rather than an expansive village of agents, we propose a streamlined Atomic Team with skilled atomic agents sufficient to handle the majority of data science workflows while maintaining tight context coherence:

- **The Planner:** Responsible for high-level roadmap and reasoning.
- **The Data Scientist:** A unified execution role responsible for end-to-end coding.
- **The Inspector:** The critical quality control layer that reviews code and outputs.
- **The Reporter:** Dedicated to final synthesis, organizing files, and translating technical results.

This multi-agent framework is streamlined, avoids unnecessary agent roles, and facilitates clear and well-structured management of context.

However, we recognize that for more complex analytical projects, adaptive workflow design may be beneficial. Exciting recent research has started to tackle automated optimization of agentic workflows. For instance, AFlow uses search algorithms to automatically discover effective sequences of LLM agent actions, while Automated Design of Agentic Systems (ADAS) proposes a meta-learning approach where a meta-agent can experiment with different multi-agent organizational patterns.

In the context of data analysis, an adaptive agentic system might decide to spawn an extra debugging agent if a problem is particularly tricky or re-order the workflow based on the data characteristics.

## 5 Deepening Human-in-the-Loop Collaboration

Professor Zhou and Professor Liu suggest that there is further potential to amplify human insight and oversight in agentic AI systems, moving toward a deeply collaborative human-AI partnership. They propose allowing more nuanced and conversational interactions between

human experts and AI agents, including natural-language feedback and AI-initiated clarifying questions.

We agree that current implementations of human feedback in LAMBDA (via direct code edits or re-prompts) represent only a first step. Future systems could allow users to give natural-language feedback such as “The approach you used to imputing missing data is too simplistic; consider using regression imputation or consult domain knowledge on likely values.” The AI agents would then interpret this instruction and adjust their plan accordingly.

Another possibility is enabling the AI to ask the human clarifying questions during the process, such as “Do you trust the outliers in this dataset, or should I remove them?” Or presenting interim results for guidance: “Here is a preliminary model; would you like to prioritize interpretability or predictive accuracy?” By incorporating richer two-way communication, the human can intervene at a higher level of abstraction, injecting expertise or preferences throughout the analysis.

The ultimate vision is an “amplified analyst” paradigm where human and AI form a tightly integrated team. The goal of agentic systems like LAMBDA should not be to replace the human analyst but to enhance them, allowing the human to focus on creative and judgment-intensive aspects while the AI handles laborious computations and explorations.

By expanding the ways in which humans can guide and partner with the AI, future systems will further amplify human value in the analytical process, ensuring that AI serves as a force multiplier for human intelligence rather than a mere replacement.

## 6 Conclusion

We thank Professor Fan Zhou and Professor Bang Liu for their comprehensive and forward-looking discussion. Their suggestions regarding self-evolving agents, data-aware retrieval, reinforcement learning for statistical reasoning, expanded agent roles, and deepened human-AI collaboration represent important directions for our ongoing research. We are committed to developing LAMBDA into a system that not only automates data analysis tasks but learns from experience, collaborates fluidly with human experts, and reasons adaptively in complex, uncertain environments.

## References

- Maojun Sun, Ruijian Han, Binyan Jiang, Houduo Qi, Defeng Sun, Yancheng Yuan, and Jian Huang. Lambda: A large model based data agent. *Journal of the American Statistical Association*, pages 1–13, 2025a.
- Maojun Sun, Ruijian Han, Binyan Jiang, Houduo Qi, Defeng Sun, Yancheng Yuan, and Jian Huang. A survey on large language model-based agents for statistics and data science. *The American Statistician*, pages 1–14, 2025b.