StarlightSearch Launches Reflect: Utility-Ranked Memory System for Self-Improving AI Agents

New approach closes the feedback loop between agent observability and performance, enabling continuous improvement without prompt engineering

SAN FRANCISCO, CA, UNITED STATES, April 9, 2026 /EINPresswire.com/ -- StarlightSearch, a startup building infrastructure for self-improving AI agents, today announced the launch of Reflect, a utility-ranked memory layer that ranks retrieved guidance by actual outcomes rather than semantic similarity alone.

> “Agents in production need utility, that is, outcome-informed learnable score, not a similarity score.”
>
> *Sonam Pankaj*

The announcement addresses a persistent gap in production AI systems: while most organizations now have robust observability stacks capturing agent traces and evaluation frameworks measuring pass/fail rates, these systems rarely connect. Agents start each task from a blank slate, unable to learn from previous failures.

"Every AI team we talk to has the same frustration," said Sonam Pankaj, founder of StarlightSearch. "They can see exactly where their agents fail. They have dashboards full of traces. But turning those failures into better behavior requires manual intervention. We built Reflect to automate that learning loop."

How Reflect Works: The Utility Difference

Traditional memory systems for large language models rely on semantic similarity: they retrieve content that sounds relevant to the current query. Reflect adds a second dimension — utility, a score that tracks whether following a particular piece of retrieved advice actually led to success.

The system uses a weighted scoring formula where the score balances semantic relevance against proven effectiveness. A memory that has been retrieved multiple times and consistently contributed to successful outcomes will rank higher than one that merely sounds similar to the current task.

Think of it like a credit score," Pankaj explained. "It doesn't just record that you had a loan. It

tracks whether you paid it back. Similarly, utility tracks whether a memory actually helped the agent succeed."

From Facts to Reasoning

Unlike conventional memory layers that store static facts — user preferences, document chunks, conversation history — Reflect stores reasoning about outcomes. When a trace is reviewed (marked as pass or fail), Reflect generates a reflection: a compressed lesson about what went wrong and what should be done differently next time.

For example, a customer support agent handling refund requests might initially retrieve advice to "issue an immediate refund" for duplicate charge complaints. If that leads to a double-refund because settlement status wasn't checked, the utility of that memory drops. Simultaneously, Reflect stores a new reflection: "For duplicate charge complaints, check settlement status before initiating any refund." On subsequent similar tickets, the new reflection ranks higher while the old one is deprioritized — all without human prompt engineering.

Three-Layer Integration

Reflect sits between three existing components in production agent architectures:

- Observability: Traces capture every tool call, LLM completion, and exception
- Evaluation: Reviews mark outcomes as pass, fail, or provide detailed feedback
- Action: The agent retrieves memories before executing

The company's approach treats traces not as passive audit logs but as training signal. When a review marks a trace as failed, the system extracts a reflection and stores it as task-linked memory. When a similar task arrives, that reflection surfaces with an updated utility score.

"What makes this production-ready is that it's outcome-addressable," Pankaj said. "You're not retrieving by keyword. You're retrieving by semantic similarity weighted by whether those memories have historically helped or hurt. The eval outcome becomes a first-class signal in retrieval ranking."

Market Context

The launch comes as enterprises increasingly deploy AI agents for customer support, code review, and task automation — use cases where consistent, reliable behavior matters more than one-off heroic performance.

Existing memory frameworks have focused primarily on user continuity: personalization, preferences, and conversation history. Academic work including Reflexion demonstrated that agents can learn from verbal self-reflection, achieving 91 percent pass rates on coding

benchmarks. However, these approaches do not incorporate learned utility signals that rank which experiences to surface.

Reflect's differentiation lies in its quantitative ranking layer. While semantic memory retrieves what sounds relevant, Reflect retrieves what has earned trust through repeated reviewed runs.

Availability and Integration

Reflect is available today via Python SDK and REST API. The SDK provides a context manager that handles memory retrieval before agent execution and automatic trace submission with review attachment afterward.

The company offers hosted cloud infrastructure with enterprise features including project isolation, API key management, and audit logging. Self-hosted deployments are available for organizations with specific compliance requirements.

About StarlightSearch

StarlightSearch builds infrastructure for outcome-informed AI agents. The company was

Sonam Pankaj
StarlightSearch INC
email us here
Visit us on social media:
LinkedIn