State Machine Structured Agents for Physical Science Reasoning

Jenelle Millison[1], Jennifer Sleeman[1], Javesh Sood[2], Jay Brett[1], Alexander Chen[1], Caroline Tang[1], Adeline Hillier[1], Chace Ashcraft [1]

[1]Johns Hopkins Applied Physics Laboratory
[2]The University of Maryland
jenelle.millison@jhuapl.edu, jennifer.sleeman@jhuapl.edu

Abstract

Large language models (LLMs) have shown promise as scientific assistants capable of reasoning, tool invocation, and autonomous analysis. However, their use in the physical sciences remains limited by the need for strong guarantees of explainability, traceable reasoning over long computational workflows, and the high consequences of subtle errors. Many existing LLM-based agent systems rely on unstructured conversational context or loosely coupled tool calls, which are insufficient for scientific domains governed by strict physical constraints and nonlinear dynamics. In this study, we explore the use of an AI assistant for an Earth system science use case. Unique to this work, we introduce a state machine agentic architecture that enables LLMs to perform structured, physically grounded scientific analysis through an explicit and enforceable notion of agent state. We define the agent state as a structured tuple, maintained across agent iterations using explicit update rules and strongly typed interfaces that enforce physical feasibility during tool invocation. We evaluate this methodology for a case study of ocean overturning circulations and a potential slowing or collapse of this system. The AI assistant interacts with a reduced-order ocean model and in-house developed deep learning models. The AI assistant is able to decompose complex questions into actionable steps that are answered using the set of tools developed. Compared to traditional grid search approaches, the agent autonomously designs parameter explorations, invokes high-dimensional models within valid physical bounds, and produces interpretable scientific summaries while reducing simulation count. The results demonstrate that the structured agent state is a key enabler of reliable LLM-based scientific assistants and is broadly applicable to constrained tool-centric scientific workflows.

Introduction

Large language models (LLMs) have had a profound impact on many AI tasks and their increased capabilities have suggested that Artificial General Intelligence (AGI) could be a near term possibility (Zhao et al. 2023), where LLMs will reach, if not surpass, human intelligence. Research related to multi-agent systems has seen a recent resurgence (Li et al. 2024), where LLMs act as agents that work together to complete challenging computational tasks. LLM-agents have become so powerful that their use in the physical and biological sciences has also increased (Schmidgall et al. 2025; Ren et al. 2025; Pantiukhin et al. 2025), although adoption is still slow in these disciplines (Kim et al. 2025). This slow adoption is driven not only by stringent demands for rigor and explainability, but also by limited trust in the black-box nature of deep learning (ŞAHiN, Arslan, and Özdemir 2025) and by challenges related to human–machine alignment (Ilievski et al. 2025).

Specifically, within the physical and biological sciences, a critical gap remains between the surface-level correctness of LLM-generated outputs and their epistemological quality, particularly in contexts of the Earth system (Bulian et al. 2024). LLMs frequently produce fluent, coherent, and executable outputs that appear scientifically plausible but may be incomplete, misleading, or physically invalid (Oliveira et al. 2025). In domains governed by nonlinear dynamics and strict physical constraints, such failures are especially problematic, as subtle errors can propagate silently through long computational workflows, undermining scientific conclusions without triggering obvious warnings.

In this study, a state machine is used to introduce structure and traceability for agent interactions. It becomes the underpinning for a multi-agent-based approach to scientific question answering for an Earth systems use case, making use of a climate model and a set of deep learning models as tools. The agent's answers are compared to those of a human oceanographer, showing a significant reduction in the number of simulations needed to answer scientific questions pertaining to the global ocean circulation system while maintaining consistency with the physics of the system. The stateful agent interactions are also compared with those of a non-stateful agent, showing that the stateful agent outperforms the non-stateful agent in efficiency using the tools available to answer the set of questions.

Background

Earth system science investigates processes governing the atmosphere, oceans, cryosphere, and land surface. Forecasting short, medium, and long-range trajectories helps scientists understand how the Earth is changing on both near-term and decadal timescales. These forecasts typically rely on high performance computing, yet even such resources can be computationally limiting. When exploring abrupt Earth

system changes, events with potentially large societal impacts, traditional modeling approaches often break down.

Recently, AI-based methods have been explored to forecast abrupt state changes in the Earth system (Bury et al. 2021; Lenton et al. 2024; Zhuge, Li, and Chen 2025) and to identify initial conditions that lead to regime shifts (Sleeman et al. 2023b). One such study (Sleeman et al. 2023a) investigated a deep generative approach to characterize the initial conditions that lead to a collapse of the Atlantic Meridional Overturning Circulation (AMOC). The AMOC is a large-scale circulation system in which warm saline surface waters flow northward into the North Atlantic, cool, and sink into the deep ocean as their density increases (McCarthy et al. 2017). Recent work has highlighted the potential for AMOC collapse (Dijkstra and van Westen 2025) due to the severe climatic and societal consequences associated with such a disruption. Reduced-order models are commonly used to study the mechanisms and drivers of AMOC collapse (Gnanadesikan, Kelson, and Sten 2018).

In this work, a reduced-order model (Gnanadesikan et al. 2024) of the Meridional Overturning Circulation (MOC) dynamics was used to study the AMOC and what leads to collapse and/or recovery. The reduced-order model represents the oceans in terms of six boxes, including a deep ocean box, North Pacific and low latitude Pacific surface boxes, North Atlantic and low latitude Atlantic surface boxes, and an inter-basin exchange between the Atlantic and Pacific. The overturning in both the Atlantic and the Pacific can exhibit deep, intermediate, or shallow circulation patterns. The Atlantic overturning is deep in today's current climate and would be in a collapsed state when experiencing shallow circulation, as modeled in this six box model.

Even in this reduced model, with arguably fewer differential equations than a global climate model, understanding the drivers of a collapsed AMOC is challenging. Deep learning models were built to better understand the initial conditions that could lead to collapse and to forecast when these events may occur by anticipating sudden statistical changes before an abrupt shift occurs (Sleeman et al. 2023a). However, for oceanographers to meaningfully interrogate these models, an accessible natural language interface is needed, as direct interaction with deep learning architectures presents a barrier to scientific exploration. Recent advances in large language models (LLMs), particularly their ability to interpret scientific queries and orchestrate tool use, make their integration as scientific assistants a promising direction. In this work, a multi-agent AI Assistant architecture is explored that addresses the scientific challenge of addressing questions rooted in first principles Earth system dynamics.

Related Work

Before the wide adoption of LLMs for scientific assistance, early work by Ashcraft et al. (Ashcraft et al. 2023) built a neuro-symbolic method to answer scientific questions related to climate science. Their natural language processing (NLP) method coupled with climate model simulations demonstrated substantial potential to accelerate scientific inquiry in Earth system science. This early work employed a bidirectional text-to-program translation architecture that mapped natural language queries to executable program calls interfacing directly with deep learning models, enabling scientists to query and interrogate AI systems without requiring expertise in machine learning. As LLMs became more prolific, their adoption in the sciences has increased and more recent advances in LLMs have further expanded these capabilities (Ren et al. 2025). However, gaps still persist, even with recent efforts that have sought to adapt LLMs more closely to scientific domains. Some recent efforts include domain-specific fine-tuning (Thulke et al. 2024) and integration of external domain-aware tools that allow LLMs to invoke simulations, data processing pipelines, and analysis routines (Widlansky and Komar 2025; Bran et al. 2024). Building on this foundation, more sophisticated agentic systems have emerged that incorporate long-horizon autonomous planning and iterative experiment execution, enabling end-to-end scientific workflows that span hypothesis generation, experimentation, analysis, and manuscript drafting (Yamada et al. 2025; Mitchener et al. 2025). More recent advances have introduced the notion of state through various implementations, such as a world state (Weidener et al. 2026) or a memory agent (Jin et al. 2025). However, these systems have been primarily developed and evaluated in materials and life sciences and have not been thoroughly assessed in other physical sciences, such as on Earth system science tasks. Moreover, even when applied to scientific discovery, existing agentic frameworks typically operate over relatively low to moderate dimensional tools and models and have not been demonstrated in settings requiring repeated, physically constrained exploration of high-dimensional simulation or deep learning models.

Recent work in Earth sciences has demonstrated the promise of LLM-driven agentic frameworks that integrate planning modules, domain-specific datasets, and computational tools (Guo et al. 2025). While such systems are capable of automating complex analysis pipelines, their authors note an important limitation is that LLM-generated code is often syntactically correct and executable, yet can produce scientifically invalid or erroneous results without triggering runtime errors, thereby compromising entire workflows. To mitigate this risk, EarthLink (Guo et al. 2025) incorporates iterative result-checking loops; however, these checks primarily target generic execution failures such as NaN values, empty outputs, or malformed files, rather than domain-specific violations of physical validity. Consequently, there is no formal mechanism to prevent invalid parameter regimes, inappropriate modeling assumptions, or extrapolation beyond calibrated ranges. As a result, extending such systems beyond cases where established diagnostic recipes, prior literature, or previously encoded workflows exist remains a challenge. Furthermore, such systems can be challenging for domain scientists to use directly, as the progression of the analysis is implicitly distributed across generated plans, executable code, intermediate data products, and natural-language reasoning. Consequently, while EarthLink enables post-hoc inspection of results, it remains difficult to trace how specific assumptions change over long computational workflows or to formally verify whether individual agent decisions were physically admissible at the

time they were made.

A key challenge underlying these limitations is the absence of an explicit, enforceable notion of agent state in most existing LLM-based scientific assistants. In many agentic systems, progress through a scientific workflow is encoded implicitly in conversational context, generated code, or accumulated artifacts, making it difficult to reason about what assumptions are currently active, which constraints are being enforced, and which actions are admissible at a given point in the analysis. State machines provide a natural abstraction for addressing this challenge. They offer a formal mechanism for representing the state of the system, governing allowable transitions, and enforcing invariants across long-running processes. Specifically, finite state machines make system evolution explicit, auditable, and constrained by design (Gladyshev and Patel 2004). By adopting a state-machine-oriented architecture, we elevate the agent state from an implicit byproduct of interaction to an object that governs tool invocation, parameter exploration, and reasoning flow. This framing directly targets the traceability and physical-admissibility gaps identified above, enabling scientific workflows in which each agent action is both context-aware and formally constrained by the current state of the system.

LLM-based Scientific Assistants Grounded by a State Machine

Towards this vision, this paper presents a state machine agentic architecture and demonstrates its capabilities in Earth system science, specifically analyzing the MOC. Although the case study centers on an analysis of the ocean overturning circulation, the architecture itself is domain-agnostic and designed to support structured, tool-driven scientific reasoning across disciplines. By enforcing stateful behavior and strongly typed, physically grounded tool interfaces, the framework enables an agent to operate complex computational models, produce physically consistent simulations, and deliver interpretable results, thereby connecting the flexibility of LLMs with the disciplined workflows required in Earth systems scientific research.

This work addresses four fundamental questions central to advancing LLM-based scientific assistants:

1. Can LLM agents act as scientific assistants that accurately interpret and answer domain-grounded scientific questions?
2. Can LLM agents reliably invoke complex, multi-parameter scientific tools for simulation, discovery, and forecasting while ensuring physical feasibility?
3. Can LLM agents present scientific results in interpretable and domain-relevant forms?
4. Can LLM agents reduce the time required for scientific workflows while preserving interpretability, accuracy, and reproducibility?

Methodology

We propose a state-machine-oriented agentic architecture designed to support structured, physically grounded scientific reasoning in Earth system workflows. We organize the system around an explicit and enforceable notion of agent state, mediated through a stateful control loop, and strongly typed interfaces that govern all tool interactions.

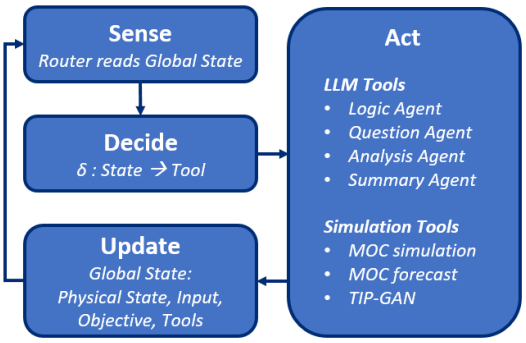


Figure 1: High-level system diagram of the state-machine-oriented assistant. The figure illustrates the execution loop, including transitions, tool invocation, and state updates.

At a high level, the architecture consists of three tightly coupled components:

1. **State Controller** that governs agent transitions and tool selection,
2. **Typed Global State** that encodes scientific context, assumptions, and intermediate results,
3. **Multi-Agent Composition** in which specialized agents operate over a shared state representation.

The execution loop is summarized in Figure 1. This loop iterates until the specified scientific objective is satisfied. In contrast to conversational agent architectures that rely on implicit context accumulation, all progression in our system is mediated through explicit state transitions, enabling traceability, auditability, and physical admissibility at each step.

State Machine Architecture

We formalize the agent controller as a state machine:

$$M = (S, \Sigma, \delta, s_0, F, \mathcal{T}), \tag{1}$$

where:

- S denotes the set of admissible scientific states (e.g., circulation regimes or experimental phases),
- Σ represents incoming signals, including user queries, tool outputs, and execution errors,
- δ is the transition function,
- s_0 is the initial, unperturbed system state,
- F denotes the active scientific objective,
- $\mathcal{T}$ is the set of available scientific tools.

At each iteration, the router agent operates on a structured context:

$$C = (S, \Sigma, F, \mathcal{T}), \tag{2}$$

and LLM inference on that structured context acts as the transition function

$$\delta : C \to S', \tag{3}$$

producing a new state that governs subsequent actions.

State Data Structure

```
class SystemState
    goal ← optional string
    Atlantic_state ← latest integer or null
    Pacific_state ← latest integer or null
    current_step ← latest value
    user_input ← latest value
    llm_output ← latest value
    error ← latest value
    tool_input ← dictionary (default empty)
```

Figure 2: Global typed state used by the state-machine-oriented agent. The state maintains intermediate scientific assumptions, tool invocation results, and control variables enabling traceable transitions between agent steps during multi-step reasoning and tool chaining.

Typed State Representation and Enforcement

To instantiate the state machine in executable form, we represent the global agent state using a strongly typed data model implemented via Pydantic (Colvin et al. 2023). The state encodes both scientific variables and the agent's reasoning context in a persistent structured format outlined in Figure 2. This typed state serves several critical functions:

- Encodes the state S and active goal F
- Captures incoming signals Σ such as tool outputs and errors
- Persists across all agent and tool invocations, ensuring continuity and traceability

By making the agent state explicit and enforceable, the system avoids reliance on unstructured conversational memory and enables formal verification of admissibility for each transition.

Multi-Agent Composition over Shared State

The architecture employs a set of specialized agents that operate over the shared typed state:

- **Router Agent**: Implements the transition function δ, selecting tools or subagents based on the current state.
- **Logic Agent**: Performs reasoning audit.
- **Question Agent**: Verifies proper information is present in high-level scientific queries to generate actionable subtasks.
- **Analysis Agent**: Generates executable code for data analysis, model result interrogation, and visualization.
- **Summary Agent**: Produces interpretable scientific narratives and summaries.

All agents read from and write to the same global state, ensuring coherent reasoning across planning, execution, and interpretation stages.

High-Dimensional Scientific Tools and Typed Interfaces

To evaluate the architecture in a demanding Earth systems setting, the system integrates three high-dimensional scientific tools:

- **Six Box MOC Simulation Model**: An extension of the model introduced by (Gnanadesikan et al. 2024), parameterized by 41 physical variables.
- **MOC Forecasting Model**: A sequence-to-sequence time series forecasting model based on the Temporal Fusion Transformer architecture (Lim et al. 2020), operating on multivariate time series with input windows of 200 time steps and output windows of 100 time steps.
- **TIP-GAN**: A generative model adapted from (Sleeman et al. 2023c), with 30 structured physical inputs and 14 tunable hyperparameters.

Each tool is accessed through a typed interface that constrains LLM-generated inputs to physically meaningful ranges. This design mitigates a common failure mode in LLM-assisted workflows, where generated code may be syntactically valid and executable yet still produce scientifically invalid results without triggering runtime errors. By enforcing validation at the interface boundary, the system prevents invalid parameter regimes, inappropriate model assumptions, and extrapolation beyond calibrated domains at invocation rather than relying on post-hoc error detection.

Dynamic State Evolution and Traceability

Every tool or agent invocation results in:

1. Execution of the selected operation
2. Structured update of the global typed state
3. Selection of the next state transition by the router agent

This yields a dynamic execution trace:

$$S \to \delta(C) \to S' \to \delta(C') \to \cdots, \tag{4}$$

where each transition is explicitly recorded and physically admissible by construction.

By elevating the agent state to a first-class object, the system provides fine-grained traceability across long computational workflows, enabling domain scientists to audit how assumptions, parameters, and conclusions evolve throughout the analysis.

Experiments

The experimental setup is designed to evaluate whether the stateful LLM assistant could support non-trivial scientific inquiry through question decomposition and tool invocation. Specifically, we investigate whether such an architecture enables physically grounded scientific analysis, yields traceable and interpretable reasoning, and operates reliably. To assess progress toward these goals, we tested the following research questions:

- **Scientific Reasoning (Q1):** Does the agent demonstrate coherent reasoning about MOC dynamics and their sensitivity to physical parameters?
- **Tool Selection and Use (Q2):** Does the agent correctly differentiate among available models and invoke them in an appropriate sequence?
- **Interpretability (Q3):** Does the agent produce outputs (e.g. summaries, figures) that are scientifically interpretable to domain experts?

- **Efficiency (Q4):** Does the agent reduce the number of model evaluations relative to baseline approaches such as grid search and stateless tool-use strategies?

MOC Use Case Study

To empirically evaluate these questions, we design a controlled experimental framework centered on MOC state changes and the AMOC collapse. This use case study provides a well-scoped, yet nontrivial scientific setting in which success criteria, failure modes, and reasoning errors can be systematically observed.

We evaluate the proposed stateful assistant in this MOC study that is traditionally conducted using exhaustive grid search over physical parameters. In contrast to fixed, manually specified parameter sweeps, our system autonomously: **a.)** selects parameter exploration strategies based on intermediate simulation results, **b.)** invokes a six box MOC simulation model through physically constrained tool calls, **c.)** maintains and updates a persistent global state across iterations, and **d.)** synthesizes domain-interpretable conclusions from the resulting simulations.

To isolate the contribution of explicit agent state, we compare three experimental conditions:

1. **Coarse to Fine Grid Search Baseline**: Traditional practice of an oceanographer programming coarse to fine sweeps over predefined parameter ranges and exiting when the solution is found.
2. **Stateless LLM Agent**: An LLM-driven assistant with identical tools, typed interfaces, and prompts, but without a persistent structured global state, relying instead solely on result message-passing via strings between steps.
3. **Stateful Agent (Ours)**: The proposed architecture with explicit state representation, typed interfaces, and state-governed transitions.

The stateless agent baseline reflects common agentic LLM designs in which reasoning and tool invocation occur without an enforceable notion of persistent state. This comparison enables us to attribute observed differences in behavior, efficiency, and performance specifically to the state representation, rather than to tool availability or prompt engineering.

Table 1 lists the experimental prompts used to probe these capabilities, while Table 2 summarizes quantitative and qualitative results across all three approaches. The experimental prompts were selected to reflect scientifically meaningful questions related to the stability and collapse of the AMOC, as studied using reduced-order models and AI-based approaches. In particular, the prompts align with prior work leveraging simplified circulation models to study abrupt transitions (Gnanadesikan, Kelson, and Sten 2018) and recent machine learning methods for identifying regime shifts in Earth system dynamics (Sleeman et al. 2023a).

	Prompt
1	Simulate the effect of increased northward freshwater flux in the Atlantic and determine the value that will cause the Atlantic MOC overturning to shut off.
2	Simulate the effect of increased southward freshwater flux in the Atlantic and determine the value that will cause the Atlantic MOC overturning to shut off.
3	Simulate the effect of decreasing Ekman flux and determine what value of the Ekman flux will cause the MOC overturning to change states.
4	Simulate the effect of varying the Pacific pycnocline depth and determine which of increasing or decreasing the Pacific pycnocline depth will cause the overturning to change states.
5	Simulate the effect of increased A_{Redi} and determine the value that will cause the Atlantic MOC overturning to shut off.
6	Simulate the effect of decreased A_{Redi} and determine the value that will cause the Atlantic MOC overturning to shut off.
7	Simulate the effect of varying ϵ_P and determine how large it must be to cause the overturning to change states.
8	Simulate the effect of decreasing ϵ_{IB} and determine how small it must be to cause the overturning to change states.
9	Simulate the effects of varying A_{Redi} and determine which of increasing or decreasing the A_{Redi} parameter will cause a change in overturning state quicker.
10	Simulate the effect of increasing the initial temperature and the restore time for temperature in the North Atlantic. Determine the value of initial temperature and temperature restore time that causes the overturning to change states.

Table 1: Scientifically motivated prompts used to evaluate human and agent performance on MOC analyses. Each prompt probes a distinct physical mechanism related to MOC state (Gnanadesikan, Kelson, and Sten 2018).

Extended Tasks and Complex Scenarios

In addition to the core MOC use case study task, we further evaluate the system on more complex scientific scenarios requiring multi-step tool chaining and model training. These experiments are designed to investigate the capacity of the system to scale.

Specifically, we construct tasks that require multi-tool and long-horizon scientific reasoning, in which the agent must maintain the correct execution order, preserve intermediate assumptions, and enforce physical admissibility across multiple tool invocations.

In the first experiment, the agent is prompted with:

"Predict the effects of varying northward freshwater flux in the Atlantic on the AMOC over time."

Addressing this query requires a sequential workflow in which the six box MOC simulation model is first executed to generate physically consistent circulation data, followed by downstream analysis tools to forecast temporal trends and sensitivities. Correct execution therefore depends on maintaining both ordering and contextual consistency across tool calls.

In the second experiment, the agent is prompted with:

"Train the most accurate TIP-GAN using the dataset at dataset.pkl."

This task requires coordinating multiple stages of computa-

tion, including dataset loading, model initialization, training, and hyperparameter selection. Successful completion depends on tracking intermediate performance metrics and preserving the training context across iterations, particularly when interacting with a high-dimensional deep learning model.

In all experiments, the Router, Logic, Question and Summary agents are instantiated using Qwen-32B-FP8 (Yang et al. 2025), while the Analysis agent uses GPT-4o-mini (OpenAI and et al. 2024). Collectively, these experiments allow us to directly connect the system's architectural objectives to measurable outcomes aligned with our research questions, as well as investigate the extensibility of such a system under growing complexity.

Evaluation

MOC Use Case Study Results

Table 2 summarizes the results of the MOC study under the three experimental conditions. In general, the stateful assistant reproduces expert-validated behaviors in five of the ten evaluated prompts, with parameter estimates typically within one order of magnitude of the reference values. Given that several physical parameters in the six box MOC model span multiple orders of magnitude, this level of agreement indicates physically meaningful reasoning rather than chance alignment (Q1).

	Grid Search		Stateless		State (Ours)	
	Result	**n**	**Result**	**n**	**Result**	**n**
1	7.75e5	22	4.5e5	16	8.0e5	3
2	1.45e6	21	No change	7	4.5e5	5
3	1.50e7	22	2.4e7	5	**1.50e7**	4
4	No change	11	**No change**	–	**No change**	4
5	No change	23	1e4	1	3000	6
6	No change	11	1e4	17	**No change**	6
7	No change	38	1.4e-4	16	**No change**	6
8	1e-5	14	7e-5	6	**1e-5**	2
9	Decrease	29	Neither	19	Increase	2
10	9.2 and 5.5	17	N/A	–	20.0 and 4.0	4

Table 2: Comparison of coarse-to-fine grid search (baseline, reflecting standard oceanographer practice), stateless agent and the stateful assistant. Here, n denotes the number of simulations required to reach the reported solution. **Bold** values indicate exact agreement with the baseline.

Figure 3 shows a formatted output log from the prompt 8 result. In this example, the assistant successfully determines the threshold value of ϵ_{IB} at which the overturning state changes. It also detects that a subsequent parameter choice violates the model's minimum allowable bound, correctly interprets the resulting validation error, and bases its final conclusion on the valid simulation outcome.

Given the inherent stochasticity of LLM-based systems, we additionally examine the distribution of outcomes produced by the stateful agent across 5 repeated runs on all prompts. Figure 4 shows a representative sample of these. Figure 4a and Figure 4c demonstrate that even in cases

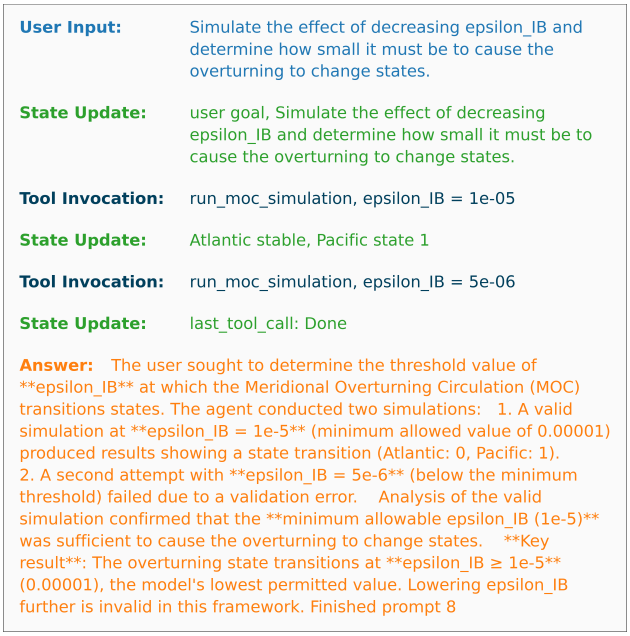


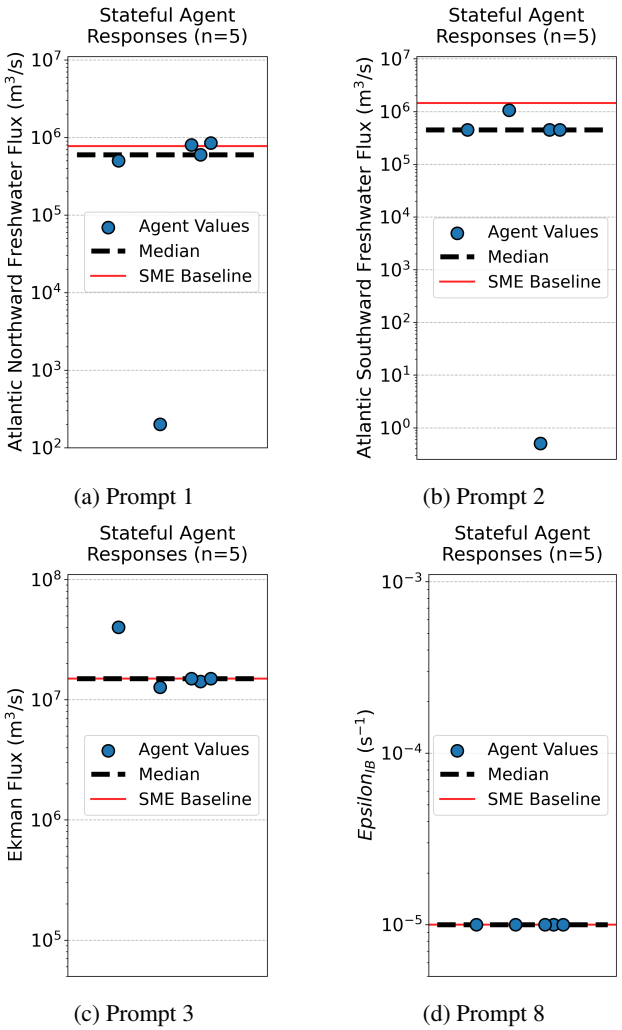
Figure 3: Formatted output log from a representative execution of the state-machine-oriented agent, demonstrating step-wise reasoning, tool use, state evolution, and respect for physical validity during a multi-step scientific task.

where the agent produces a correct answer on average, repeated executions with nonzero temperature yield occasional significant outliers, noting that the relevant parameter scale in this task is on the order of 10^7. In others (Fig. 4b), the agent slightly underestimates the value while still maintaining relatively consistent outputs across runs. Finally, Fig. 4d shows strong agreement with the expert-validated reference value across runs. These patterns highlight task-dependent variability and motivate further study with larger sample sizes to more fully characterize the distributional behavior of LLM-driven scientific agents.

Although exact parameter matching remains challenging, the stateful agent consistently achieves these results while requiring substantially fewer model evaluations than both grid search and the stateless LLM baseline, directly addressing efficiency (Q4). This reduction in simulation count is achieved without sacrificing physical plausibility, suggesting that explicit state enables more targeted and non-redundant exploration of the parameter space.

We emphasize that these experiments are intentionally small in scale and designed as a formative evaluation of the proposed architecture. The goal is not to claim statistical optimality but rather to expose systematic behavioral differences between stateful and stateless agent designs in a controlled scientific setting with complex tasks. As such, these results provide initial evidence that motivates more extensive benchmarking in future work.

Failure Mode Analysis Several consistent failure modes of the stateless system emerge from the comparison in Table 2. Broadly, stateless agents struggle to appropriately



(a) Prompt 1 (b) Prompt 2

(c) Prompt 3 (d) Prompt 8

Figure 4: Distributional behavior of the stateful agent across repeated executions relative to the baseline. (a,c) show accurate medians with occasional extreme outliers. (b) shows similar variability but slightly underestimates the baseline. (d) shows consistent agreement across trials.

scope their investigations, frequently invoking simulation tools in an unfocused manner and repeatedly evaluating similar configurations. These behaviors often lead to incorrect conclusions, inefficient exploration, or execution loops that ultimately time out. In contrast, explicit agent state mitigates many of these issues by enabling more coherent tracking of explored configurations and intermediate results.

One common failure mode is the inability to selectively invoke relevant simulation tools within a constrained study. Investigation of the execution logs reveals that stateless agents frequently invoke multiple tools in an unfocused sequence, leading to incorrect conclusions after many simulations. This behavior is most evident in prompts *6*, *7*, and *10*, where stateless agents repeatedly evaluate parameter configurations without converging on a correct interpretation of the system behavior.

A second failure mode arises from stateless agents misinterpreting the absence of observable change. For example, in prompt *6*, the stateless agent fails to recognize a valid "no-change" regime and instead defaults to nominal MOC conditions after repeated simulations. In contrast, the stateful agent more consistently reasons about the explored parameter space and correctly identifies cases in which no transition occurs.

Stateless agents also exhibit inefficient exploration patterns, frequently re-sampling previously explored regions or repeatedly invoking identical model configurations even when prior results remain available in context. These behaviors can lead to unproductive execution loops that ultimately time out—indicated by "–" in Table 2—after 20 unsuccessful attempts. By contrast, execution logs reveal that the stateful agent avoids redundant simulations. The stateful agent requires fewer model evaluations than the stateless agent (Table 2), indicating that explicit state tracking enables a more coherent notion of experimental progress. However, the stateful agent often favors coarse, order-of-magnitude parameter sweeps rather than fine-grained optimization, reflecting stochastic exploration rather than systematic search. Together, these observations suggest opportunities for incorporating classical search or adaptive sampling methods within the stateful framework.

Overall, relative to the stateless LLM agent, the stateful system demonstrates more stable exploration behavior, improved adherence to physically admissible parameter ranges, and more coherent summaries of parameter–response relationships. These behaviors collectively contribute to improved interpretability (Q3), more consistent tool use (Q2), and reduced simulation cost (Q4).

Extended Tasks and Complex Scenario Results

Results from the extended tasks further demonstrate the advantages of explicit agent state in long-horizon, multi-tool scientific workflows. In both complex scenarios, the stateful agent successfully completed the full sequence of required steps while preserving execution order, intermediate assumptions, and physically meaningful constraints.

Freshwater Flux Forecasting Task In the freshwater flux experiment, the agent maintained consistency between intermediate six box MOC simulation outputs and downstream AMOC forecasting. Execution traces show that the agent recognized the need for input data before forecasting, generated simulation outputs with appropriate dimensionality, and invoked the forecasting model with correct inputs.

Figure 5 illustrates a representative segment of this execution trace. This task is nontrivial and requires the selection of 41 physical parameters, the generation of multivariate ocean circulation time series, and the windowing of the resulting data into sequences of 200 time steps to satisfy the input requirements of the forecasting model. Successful execution therefore depends critically on maintaining stateful awareness of data dependencies and execution context, directly addressing tool selection and sequencing (Q2).

TIP-GAN Training Task In the TIP-GAN training experiment, the stateful agent executed the entire training

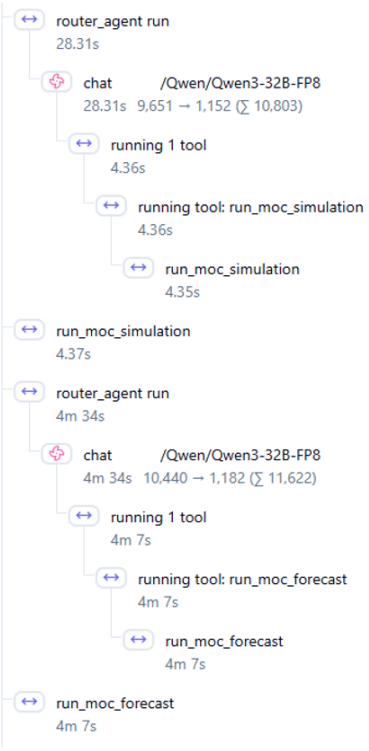


Figure 5: Trace using Langfuse (Rawert, Klingen, and Deichmann 2023) of stateful agent tool invocations in an extended scientific task. The agent identifies the need to invoke the MOC simulation tool to generate the input to the MOC forecasting tool. The trace illustrates state-aware tool chaining and planning during multi-step reasoning.

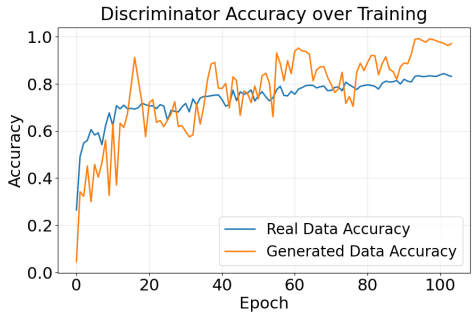
pipeline, including dataset loading, model initialization, hyperparameter selection, and training execution. The agent selected 14 hyperparameters and bounds for 30 physical input variables in a high-dimensional configuration space.

The explicit state played a central role in the tracking of prior errors and unsuccessful configurations, enabling rapid correction and preventing redundant trials. Compared to the human baseline, the agent adopted a more aggressive optimization strategy, including a higher learning rate (10^{-3} versus 10^{-4}), a larger number of generators (3 versus 1), and fewer training epochs (50 versus 100). Despite these differences, all selected parameters remained within physically admissible bounds, and the resulting models achieved classification performance comparable to human-tuned baselines, as measured by weighted accuracy on both held-out test data and generated samples (Figure 6).

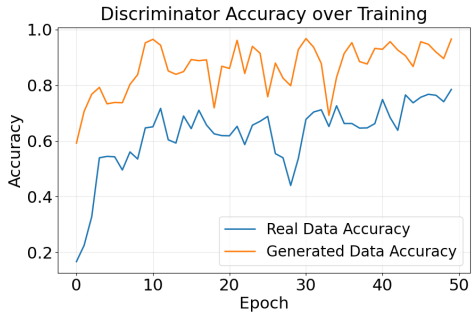
Conclusions & Future Work

We introduced a stateful agentic architecture for scientific reasoning in Earth systems science. By coupling stateful control with typed state representations and multi-agent composition, the proposed system supports structured scientific investigation while maintaining physical grounding and interpretability. The results demonstrate that an explicit, enforceable agent state enables more reliable and efficient execution of complex scientific workflows involving heterogeneous, high-dimensional tools. Across both controlled MOC use case study and extended task scenarios, the stateful agent exhibits coherent scientific reasoning, context-aware tool selection, interpretable outputs, and reduces computational cost relative to the baseline. These findings directly address the core research questions posed in this work, demonstrating the advantages of stateful control for scientific LLM assistants. This framework generalizes beyond the MOC study to scientific workflows that require structured reasoning, complex tool orchestration, and explicit state evolution.

This exploratory evaluation establishes a compelling foundation for stateful agent architectures as a scalable and transparent approach to scientific AI assistance. Future work will focus on larger-scale benchmarking across a broader range of scientific tasks, tighter integration of intelligent search components, and exploration of learned transition policies that augment the current state machine control logic while preserving interpretability.



(a) Human-selected hyperparameters



(b) Stateful agent–selected hyperparameters

Figure 6: Weighted accuracy of TIP-GAN during training with hyperparameters selected by a human versus the stateful agent. Comparable performance is observed between human and agent-selected hyperparameter trained models.

Acknowledgments

This work was supported by NASA under Grant No. 80NSSC25K0062. This work was also supported by internal research and development funding from the Research and Exploratory Development Mission Area of the Johns Hopkins Applied Physics Laboratory.

References

Ashcraft, C.; Sleeman, J.; Tang, C.; Brett, J.; and Gnanadesikan, A. 2023. Neuro-Symbolic Bi-Directional Translation - Deep Learning Explainability for Climate Tipping Point Research. arXiv:2306.11161.

Bran, A. M.; Cox, S.; Schilter, O.; Baldassari, C.; White, A. D.; and Schwaller, P. 2024. Augmenting large language models with chemistry tools. *Nature Machine Intelligence*, 6(5): 525–535.

Bulian, J.; Schäfer, M. S.; Amini, A.; Lam, H.; Ciaramita, M.; Gaiarin, B.; Chen Huebscher, M.; Buck, C.; Mede, N. G.; Leippold, M.; and Strauss, N. 2024. Assessing Large Language Models on Climate Information. In Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; and Berkenkamp, F., eds., *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, 4884–4935. PMLR.

Bury, T. M.; Sujith, R.; Pavithran, I.; Scheffer, M.; Lenton, T. M.; Anand, M.; and Bauch, C. T. 2021. Deep learning for early warning signals of tipping points. *Proceedings of the National Academy of Sciences*, 118(39): e2106140118.

Colvin, S.; Jolibois, E.; Ramezani, H.; Garcia Badaracco, A.; Dorsey, T.; Montague, D.; Matveenko, S.; Trylesinski, M.; Runkle, S.; Hewitt, D.; et al. 2023. Pydantic. *Zenodo*.

Dijkstra, H. A.; and van Westen, R. M. 2025. The Probability of an AMOC Collapse Onset in the Twenty-First Century. *Annual Review of Marine Science*, 18.

Gladyshev, P.; and Patel, A. 2004. Finite state machine approach to digital event reconstruction. *Digital Investigation*, 1(2): 130–149.

Gnanadesikan, A.; Fabiani, G.; Liu, J.; Gelderloos, R.; Brett, G. J.; Kevrekidis, Y.; Haine, T.; Pradal, M.-A.; Siettos, C.; and Sleeman, J. 2024. Tipping points in overturning circulation mediated by ocean mixing and the configuration and magnitude of the hydrological cycle: A simple model. *Journal of Physical Oceanography*, 54(7): 1389–1409.

Gnanadesikan, A.; Kelson, R. K.; and Sten, M. 2018. Flux correction and overturning stability: Insights from a dynamical box model. *Journal of Climate*, 31(22): 9335–9350.

Guo, Z.; Wang, J.; Ling, F.; Wei, W.; Yue, X.; Jiang, Z.; Xu, W.; Luo, J.-J.; Cheng, L.; Ham, Y.-G.; Song, F.; Gentine, P.; Yamagata, T.; Fei, B.; Zhang, W.; Gu, X.; Li, C.; Wang, Y.; Chen, T.; Ouyang, W.; Zhou, B.; and Bai, L. 2025. A Self-Evolving AI Agent System for Climate Science. *arXiv preprint arXiv:2507.17311*.

Ilievski, F.; Hammer, B.; van Harmelen, F.; Paassen, B.; Saralajew, S.; Schmid, U.; Biehl, M.; Bolognesi, M.; Dong, X. L.; Gashteovski, K.; et al. 2025. Aligning generalization between humans and machines. *Nature Machine Intelligence*, 7(9): 1378–1389.

Jin, R.; Guo, Y.; Qu, Y.; Yang, M.; Shang, C.; Yang, Q.; Chao, L.; Zhou, Y.; Xu, R.; Xu, Z.; Zhou, R.; Zhang, Z.; Wang, M.; Zhang, X.; and Cong, L. 2025. BioLab: End-to-End Autonomous Life Sciences Research with Multi-Agents System Integrating Biological Foundation Models. *bioRxiv*.

Kim, J.; Podlasek, A.; Shidara, K.; Liu, F.; Alaa, A.; and Bernardo, D. 2025. Limitations of large language models in clinical problem-solving arising from inflexible reasoning. *Scientific reports*, 15(1): 39426.

Lenton, T. M.; Abrams, J. F.; Bartsch, A.; Bathiany, S.; Boulton, C. A.; Buxton, J. E.; Conversi, A.; Cunliffe, A. M.; Hebden, S.; Lavergne, T.; et al. 2024. Remotely sensing potential climate change tipping points across scales. *nature communications*, 15(1): 343.

Li, X.; Wang, S.; Zeng, S.; Wu, Y.; and Yang, Y. 2024. A survey on LLM-based multi-agent systems: workflow, infrastructure, and challenges. *Vicinagearth*, 1(1): 9.

Lim, B.; Arik, S. O.; Loeff, N.; and Pfister, T. 2020. Temporal Fusion Transformers for Interpretable Multi-horizon Time Series Forecasting. arXiv:1912.09363.

McCarthy, G.; Smeed, D.; Cunningham, S.; and Roberts, C. 2017. Atlantic Meridional Overturning Circulation (AMOC). In *Marine Climate Change Impacts Partnership: Science Review: MCCIP Science Review 2017*, 15–21.

Mitchener, L.; Yiu, A.; Chang, B.; Bourdenx, M.; Nadolski, T.; Sulovari, A.; Landsness, E. C.; Barabási, D. L.; Narayanan, S.; Evans, N.; Reddy, S.; Foiani, M.; Kamal, A.; Shriver, L. P.; Cao, F.; Wassie, A. T.; Laurent, J. M.; Melville-Green, E.; Caldas, M.; Bou, A.; Roberts, K. F.; Zagorac, S.; Orr, T. C.; Orr, M. E.; Zwezdaryk, K. J.; Ghareeb, A. E.; McCoy, L.; Gomes, B.; Ashley, E. A.; Duff, K. E.; Buonassisi, T.; Rainforth, T.; Bateman, R. J.; Skarlinski, M.; Rodriques, S. G.; Hinks, M. M.; and White, A. D. 2025. Kosmos: An AI Scientist for Autonomous Discovery. *arXiv preprint arXiv:2511.02824*.

Oliveira, C. V.; Zagalo, N.; Silva, F.; Brandao, A.; Khurrum, S. F. H.; and Santos, J. 2025. Plausibility as Failure: How LLMs and Humans Co-Construct Epistemic Error. *arXiv preprint arXiv:2512.16750*.

OpenAI; and et al., A. H. 2024. GPT-4o System Card. arXiv:2410.21276.

Pantiukhin, D.; Shapkin, B.; Kuznetsov, I.; Jost, A. A.; and Koldunov, N. 2025. Accelerating earth science discovery via multi-agent LLM systems. *Frontiers in Artificial Intelligence*, 8: 1674927.

Rawert, C.; Klingen, M.; and Deichmann, M. 2023. Langfuse — Open-Source LLM Engineering Platform. Software available from https://langfuse.com.

Ren, S.; Jian, P.; Ren, Z.; Leng, C.; Xie, C.; and Zhang, J. 2025. Towards scientific intelligence: A survey of llm-based scientific agents. *arXiv preprint arXiv:2503.24047*.

ŞAHiN, E.; Arslan, N. N.; and Özdemir, D. 2025. Unlocking the black box: an in-depth review on interpretability, explainability, and reliability in deep learning. *Neural Computing and Applications*, 37(2): 859–965.

Schmidgall, S.; Su, Y.; Wang, Z.; Sun, X.; Wu, J.; Yu, X.; Liu, J.; Liu, Z.; and Barsoum, E. 2025. Agent laboratory: Using llm agents as research assistants. *arXiv preprint arXiv:2501.04227*.

Sleeman, J.; Chung, D.; Ashcraft, C.; Brett, J.; Gnanadesikan, A.; Kevrekidis, Y.; Hughes, M.; Haine, T.; Pradal, M.-A.; Gelderloos, R.; et al. 2023a. Using Artificial Intelligence

to aid Scientific Discovery of Climate Tipping Points. *arXiv preprint arXiv:2302.06852*.

Sleeman, J.; Chung, D.; Gnanadesikan, A.; Brett, J.; Kevrekidis, Y.; Hughes, M.; Haine, T.; Pradal, M.-A.; Gelderloos, R.; Ashcraft, C.; et al. 2023b. A generative adversarial network for climate tipping point discovery (tipgan). *arXiv preprint arXiv:2302.10274*.

Sleeman, J.; Keller, C. A.; Ribaudo, C.; Chung, D.; and Szeto, M. 2023c. Deep Learning Ensembles for Improved Atmospheric Composition Modeling. In *Proceedings of the AAAI Symposium Series*, volume 2, 148–152.

Thulke, D.; Gao, Y.; Pelser, P.; Brune, R.; Jalota, R.; Fok, F.; Ramos, M.; van Wyk, I.; Nasir, A.; Goldstein, H.; Tragemann, T.; Nguyen, K.; Fowler, A.; Stanco, A.; Gabriel, J.; Taylor, J.; Moro, D.; Tsymbalov, E.; de Waal, J.; Matusov, E.; Yaghi, M.; Shihadah, M.; Ney, H.; Dugast, C.; Dotan, J.; and Erasmus, D. 2024. ClimateGPT: Towards AI Synthesizing Interdisciplinary Research on Climate Change. ArXiv:2401.09646, arXiv:2401.09646.

Weidener, L.; Brkić, M.; Jovanović, M.; Singh, R.; Baccin, C.; Ulgac, E.; Dobrin, A.; and Meduri, A. 2026. Rethinking the AI Scientist: Interactive Multi-Agent Workflows for Scientific Discovery. arXiv:2601.12542.

Widlansky, M. J.; and Komar, N. 2025. Building an Intelligent Data Exploring Assistant for Geoscientists. *Journal of Geophysical Research: Machine Learning and Computation*, 2(3): e2025JH000649. E2025JH000649 2025JH000649.

Yamada, Y.; Lange, R. T.; Lu, C.; Hu, S.; Lu, C.; Foerster, J. N.; Clune, J.; and Ha, D. 2025. The AI Scientist-v2: Workshop-Level Automated Scientific Discovery via Agentic Tree Search. *arXiv preprint arXiv:2504.08066*.

Yang, A.; Li, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; Yu, B.; Gao, C.; Huang, C.; Lv, C.; Zheng, C.; Liu, D.; Zhou, F.; Huang, F.; Hu, F.; Ge, H.; Wei, H.; Lin, H.; Tang, J.; Yang, J.; Tu, J.; Zhang, J.; Yang, J.; Yang, J.; Zhou, J.; Zhou, J.; Lin, J.; Dang, K.; Bao, K.; Yang, K.; Yu, L.; Deng, L.; Li, M.; Xue, M.; Li, M.; Zhang, P.; Wang, P.; Zhu, Q.; Men, R.; Gao, R.; Liu, S.; Luo, S.; Li, T.; Tang, T.; Yin, W.; Ren, X.; Wang, X.; Zhang, X.; Ren, X.; Fan, Y.; Su, Y.; Zhang, Y.; Zhang, Y.; Wan, Y.; Liu, Y.; Wang, Z.; Cui, Z.; Zhang, Z.; Zhou, Z.; and Qiu, Z. 2025. Qwen3 Technical Report. arXiv:2505.09388.

Zhao, W. X.; Zhou, K.; Li, J.; Tang, T.; Wang, X.; Hou, Y.; Min, Y.; Zhang, B.; Zhang, J.; Dong, Z.; et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 1(2).

Zhuge, C.; Li, J.; and Chen, W. 2025. Deep learning for predicting the occurrence of tipping points. *Royal Society Open Science*, 12(7).