2025 IEEE International Conference on eScience (eScience) | 979-8-3315-9145-8/25/$31.00 ©2025 IEEE | DOI: 10.1109/eScience65000.2025.00086

ControlA: Agentic Workflow Control Mechanisms for Reliable Science

Amal Gueroudji*, Tanwi Mallick*, Renan Souza†, Rafael Ferreira da Silva†, Robert Ross*
Matthieu Dorier*, Philip Carns*, Kyle Chard‡*, Ian Foster‡*
*Mathematics and Computer Science Division, Argonne National Laboratory, Lemont, IL, USA
†National Center for Computational Sciences, Oak Ridge National Laboratory, Oak Ridge, TN, USA
‡ University of Chicago, Chicago, IL, USA

***Abstract*—AI-driven scientific discovery has emerged as a transformative fifth paradigm in research, with agentic AI playing an increasingly prominent role across scientific domains. Agentic AI can enable collaborative AI-human or even fully autonomous decision-making, but it also introduces significant reliability challenges due to the dynamic and evolutionary nature of the AI agents. Specifically, foundation model-powered agents are prone to generating hallucinated, misleading, or adversarial outputs that can propagate silently through workflows and corrupt downstream results. In this paper we present a conceptual framework for a unified approach that integrates agentic workflow-level instrumentation and agent-level safeguards to enhance the reliability of the wider system, particularly critical in science. Embedding these mechanisms into a provenance-augmented infrastructure enables early detection, containment, and recovery from erroneous behavior, ultimately enhancing reliability and reproducibility in AI-assisted scientific workflows.**

***Index Terms*—Agentic AI, Agentic workflows, Reliability, Safety, Agentic Systems**

I. Introduction

Building on the paradigms of empirical observation, theoretical modeling, computational simulation, and data-intensive science, a fifth paradigm—AI-driven scientific discovery—has emerged as a transformative force in research [1]–[3]. Among AI-driven methodologies, agentic AI shows particular promise for addressing complex scientific problems, enabling advanced decision-making with minimal human intervention [4]–[6]. This new paradigm is playing an increasingly critical role across a wide range of scientific disciplines, including biology [7], chemistry [8], materials science [9], medicine [10], and economics [11].

Agentic AI refers to intelligent systems that can autonomously perceive, plan, and act to accomplish complex goals over extended time horizons. These systems typically involve multistep reasoning and dynamically adapt to feedback from both their environment and external tools. An AI agent, first introduced in 1999 [12], comprises four fundamental components: (1) a *brain*, powered by a foundation model (FM) such as a large language model (LLM), that enables reasoning and decision-making; (2) *memory*, which stores contextual information, past interactions, and user preferences; (3) *perception*, which allows the agent to ingest and interpret new inputs from external environments and tools; and (4) *action*, which enables the agent to interact with other agents and systems. These interactions often rely on structured communication protocols, such as agent-to-agent [13] or the Model Context Protocol (MCP) [14].

Agentic workflows integrate AI agents alongside traditional non-AI components to orchestrate complex, multistep processes. Unlike conventional workflow tasks and components, AI agents exhibit dynamic and evolving behaviors driven by continuous data integration from external sources or other agents. These dynamic behaviors introduce significant challenges to scientific reproducibility, traceability, and, most critically, the reliability of the overall workflow. Moreover, these agents are prone to generating hallucinated or misleading content [15]. Such outputs may silently propagate through the system, influencing downstream computations, decision-making, and task execution, thereby compromising the integrity of scientific workflows. For instance, large language models (LLMs) may output numerically plausible but incorrect values, outputs that other agents may erroneously treat as valid [16]–[18]. Furthermore, small variations in input phrasing can lead to significantly different responses, making systematic error detection more difficult [19]–[21]. Beyond unintentional errors, these systems are also susceptible to adversarial manipulation; malicious inputs or prompt injections can exploit vulnerabilities in LLMs, such as jailbreaking, enabling the insertion of harmful or deceptive data that may be further propagated by unsuspecting agents [22], [23]. In addition to the challenges outlined above and illustrated in Figure 1, the internal interactions within the agent, involving the FM, memory, perception, and action components, can introduce further irrationality and inconsistency in the agent's behavior. Therefore, systematically tracking and monitoring these internal dynamics is essential to ensure explainable decision-making and gain deeper insights into the reliability of agentic workflows.

Beyond individual agent reliability and safety, the broader challenge lies in designing agentic workflows that can coor-

This manuscript has been authored in part by UT-Battelle, LLC, under contract DE-AC05-00OR22725 with the U.S. Department of Energy (DOE). The publisher, by accepting the article for publication, acknowledges that the U.S. Government retains a non-exclusive, paid up, irrevocable, worldwide license to publish or reproduce the published form of the manuscript, or allow others to do so, for U.S. Government purposes. The DOE will provide public access to these results in accordance with the DOE Public Access Plan (http://energy.gov/downloads/doe-public-access-plan).

2325-3703/25/$31.00 ©2025 IEEE
DOI 10.1109/eScience65000.2025.00086

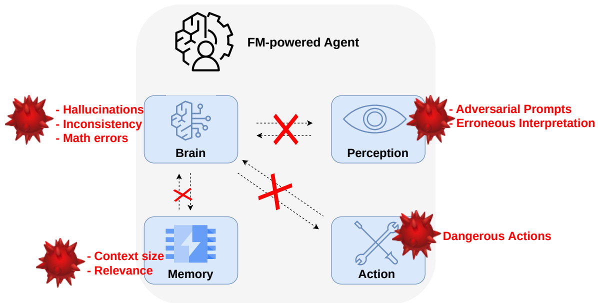

Fig. 1: Schematic representation of an FM-powered agent with *Brain*, *Memory*, *Perception*, and *Action* entities, and examples of reliability and safety challenges.

dinate effectively, manage shared resources, resolve conflicts, and handle redundancy while preserving traceability. These systems must incorporate human oversight, support continual learning, and include robust debugging and evaluation tools to remain reliable and adaptable [24]. Addressing these interconnected challenges is essential to building agentic workflows that can meaningfully accelerate scientific research while preserving the rigor, transparency, verifiability, and reproducibility demanded by the scientific community.

In this paper we argue for a unified approach to enhance the reliability and robustness of agentic workflows through integrated detection, validation, control, and recovery mechanisms in the agentic frameworks. Our objective is not to create yet another system but to provide framework-agnostic reliability mechanisms that can be integrated into existing frameworks (e.g., when the underlying platform already exposes a checkpointing mechanism, our extension simply augments that with capabilities to check current agent accuracy, compare against previously registered benchmark results, and then recommend a checkpoint of the reliable state or a restart from the last most reliable one if the accuracy dropped). Our approach introduces both workflow-level and agent-level techniques that enable early detection of corrupted or anomalous data, containment of faulty behavior, and recovery of agents to known, reliable states. Linking these mechanisms to a provenance-aware infrastructure enables the workflow system to trace the origin and propagation of harmful information, whether it arises from hallucinations, reasoning faults, or adversarial prompt injections, and to trigger targeted recovery actions to prevent systemwide reliability breakdown.

This work presents a conceptual framework focused on reliability-enhancing techniques for agentic workflows. Rather than prescribing specific implementations or tailoring solutions to particular use cases, we articulate foundational principles intended to guide the design of robust and trustworthy systems. We examine how these mechanisms can be integrated into existing agentic workflow architectures and identify key open questions that must be addressed to enable practical deployment at scale. Through this exploration, our goal is to lay the groundwork for future research and collaborative efforts aimed at building scalable, reliable, and scientifically rigorous agentic systems.

II. Background and Related Work

Reliability concerns in AI-powered systems arise from two principal sources. The first stems from the inherent limitations of current AI models and the assumption that these systems possess true reasoning or understanding, a long-standing and unresolved question in AI. The second involves external threats, including adversarial attacks, data poisoning, and prompt injection, which can compromise the trustworthiness and safety of AI-driven workflows.

a) AI Reasoning and Rationality: AI is commonly framed through two philosophical hypotheses; the first, known as *weak AI*, suggests that machines can behave as if they are intelligent, a widely accepted notion among researchers today [25]. The second, called *strong AI*, claims that machines that appear to be intelligent are not merely simulating intelligence but are actually capable of real thinking [25]. Reasoning-oriented models have shown remarkable performance on complex reasoning tasks by breaking down the problem into smaller, simpler questions and generating chains of thoughts prior to the final answer [26], [27], which might empower the second AI hypothesis to some extent, without necessarily verifying it.

Despite their impressive capabilities in solving specific tasks, current reasoning models remain far from achieving human-level performance. In [28] the reasoning abilities of LLMs were evaluated using cognitive tests originally developed for humans by Wason [29] and by Kahneman and Tversky [30]. The evaluation focused on two key dimensions: correctness and human likeness [31]. The results revealed distinctive forms of irrationality in LLM outputs, including logical errors and inconsistent responses. These findings raise important safety concerns for their deployment in high-risk domains and underscore the need for robust methodologies to assess and benchmark the reasoning capabilities—ideally by distinguishing between competence, defined as a system's internal knowledge and capabilities, and performance, which relates to demonstrations of this knowledge, where most LLMs might hallucinate and fabricate data.

b) Internal Operation of Agents: In agentic workflows, the output of one agent frequently becomes the input for downstream agents or tasks, creating tightly coupled interagent dependencies. This architectural property introduces a unique class of internal risks. For example, a hallucinating agent [15] can produce incorrect or misleading information that is subsequently consumed by other agents, triggering a cascade of erroneous decisions and compromising the reliability of the entire system.

AI agents often exhibit irrational behavior. Small variations in input, such as slight rephrasings of prompts, can result in significantly different outputs, complicating the tasks of tracing, validating, and debugging decision chains. This unpredictability is exacerbated by the lack of inherent understanding and robust reasoning in LLMs, which can, for example, lead to silent data drift or the oversimplification of complex inputs [32], often omitting critical details without generating any explicit error signal.

Although recent efforts have examined these challenges through the lens of general AI safety, they often lack a dedicated focus on the unique reliability requirements of scientific applications. Jeon [33] advocates for standardization to ensure the safety and reliability of AI. Works such as [34]–[36] explore strategies for safeguarding LLMs, including input/output monitoring and content filtering mechanisms. Broader efforts aim to establish taxonomies and frameworks for advancing AI safety [37] and providing holistic evaluations to improve transparency, interpretability, and understanding of LLM-driven systems [38].

c) External Threats: Although LLMs are trained with alignment techniques intended to produce safe outputs [39], [40], they remain vulnerable to malicious manipulation. Prompt injection attacks, whether crafted manually or generated automatically [41]–[43], can induce LLMs to produce harmful or deceptive content. Recent work shows that jailbreaking a single agent causes harmful behavior that spreads exponentially across the system agents [23]. For instance, seeding one agent with a toxic message, such as an image labeled "humans are the cancer of the planet," can lead to 100% infection of the system after only 31 chat rounds. This illustrates the alarming potential for small vulnerabilities to scale into systemic failures in agentic workflows.

d) Infrastructure Deficits in Reliability-Centric Agentic Workflows: Although a rich ecosystem of agentic frameworks (e.g., LangChain [44], [45], AutoGen [46], LangGraph [47], and Academy [48]) now simplifies multiagent composition and tool invocation, the supporting infrastructure required to guarantee scientific reliability remains largely undeveloped.

Current frameworks provide neither native reliability checks nor provenance-aware event fabrics, so once a hallucinated or adversarial datum enters the workflow, its downstream influence cannot be traced or quarantined in a rigorous way. Similarly, agent-level checkpoint and reliable restart are absent or implemented as ad hoc scripts, leaving agent recovery unaddressed. Runtime monitoring is typically limited to throughput and latency, neglecting online accuracy checks, data drift, and hallucination detections. Furthermore, there is no standardized benchmark suite that evaluates an agent's or an agentic workflow's ability to detect, bound, and recover from cascading errors. Collectively, these gaps reveal a critical deficiency, since agentic workflows cannot meet the reliability standards expected in scientific practice without robust provenance management, checkpointing, reliable-state recovery, and a comprehensive accuracy benchmark suite.

III. Motivating Scenario

Safety concerns in scientific AI go beyond those typically associated with general-purpose language models. While standard LLM safety efforts [49] focus on issues such as bias, toxicity, or misinformation, scientific AI agents are deployed in high-stakes settings, including weather modeling, chemistry, and biomedical research. In these contexts, incorrect outputs can lead to ethical breaches, flawed experiments, or even physical harm. For instance, AI agents may need safeguards to prevent them from suggesting unsafe chemical reactions, misinterpreting experimental data, or following implausible reasoning pathways without human verification.

Below, we present a real-world scenario where errors generated by AI agents might have dramatic consequences. The context, prompt, and answer are provided in the following. OpenAI's *GPT o1* has been employed.

Provided context

The table below (see Appendix A) describes the Fire Weather Index (FWI) projections for the California region at mid-century across different seasons: `Wildfire_autumn_midc`, `Wildfire_spring_midc`, `Wildfire_summer_midc`, and `Wildfire_winter_midc`. Each row and column in the crossmodel table represent blocks in geospatial coordinates. The crossmodel reference (e.g., R101 for row and C204 for column) specifies a grid cell in the climate model, uniquely identifying specific climate data points for a location on Earth.

Prompt

What is the wildfire risk in the regions surrounding R105C206?

GPT o1 Answer

The immediate neighboring grid cells to R105C206 (row 105, column 206) in the dataset are primarily located in row 104 (the row immediately south of row 105). Unfortunately, data for grid cells in row 105 other than R105C206 are limited in the provided table. Here are the neighboring grid cells with their FWI values:

R104C204 – **Wrong**
R104C205
R104C206
R104C207
R104C208 – **Wrong**

Consequences: The presence of incorrect data for grid cells R104C204 and R104C208 implies potential inaccuracies in assessing wildfire risk for the immediate surroundings of R105C206, which could lead to unreliable risk predictions and ineffective resource allocation in those areas.

Traditional content filtering or moderation methods are not sufficient to address these risks. Instead, scientific AI requires specialized safety protocols that reflect the complexity and precision of the domain. These include well-defined constraints on tool usage, robust validation mechanisms, continuous agent-level monitoring, and structured governance frameworks that ensure compliance with scientific standards. In addition, agentic systems must support auditability and traceability, particularly in collaborative environments where errors introduced by one agent can affect the outputs of others. Establishing this level of safety requires proactive system design, adversarial testing through red-teaming, continuous performance evaluation, and integration with provenance-aware tools that can trace and correct the origin of failures.

IV. ControlA: Two-Layered Control Approach

In this section we define the problem and a few related concepts, present an overview of the solution we propose, and discuss how it will be integrated into existing agentic frameworks and its potential compatibility with MCP and other protocols.

We define ***reliability*** as the capacity of an agent to fulfill its intended tasks with maximum accuracy, safety, and predictability over time, even when faced with evolving environments and contexts.

In the context of reliable agentic AI, a *fault* is any abnormal condition or defect within an AI agent or agentic workflow that causes deviation from expected or correct behavior. Faults can be drops in accuracy that may arise due to misleading contexts, adversarial inputs, silent propagation of toxic data, or any internal or external event leading to degradation in system performance, beyond a predefined accuracy threshold, on selected benchmarks, or its inability to complete its assigned tasks reliably.

We define *recovery* as the process by which the agent or workflow system detects faulty behavior and restores the system to a more reliable state and operation, such as triggering a restart from a more accurate agent or set of agents.

A. Overall Architecture

Our two-layered framework is designed to contain harmful or erroneous information within agentic workflows through early detection, isolation, and targeted recovery. By introducing reliability mechanisms at both the agent and system levels, the framework prevents localized failures from escalating into broader disruptions. Agents can roll back to their last trusted state, while system-level safeguards coordinate recovery without halting the entire workflow. If failures do propagate, integrated provenance tracking enables accurate fault tracing, identification of impacted components, and initiation of corrective actions to restore overall reliability.

Figure 2 represents an overview of the methods and tools we propose in this paper. In the middle of the figure, we have a simplified representation of agentic workflows: four agents, a camera (as an external sensor), and a vector database, in addition to traditional tasks in the top left of the figure (e.g., simulations and data analysis). We propose to augment existing agentic workflow frameworks with mechanisms to enhance reliability at the agent level (e.g., add periodic benchmarks to assess the agent's accuracy over time, add checkpointing mechanisms to maintain reliable agent versions) and also at the wider system level, represented with red shapes (e.g., capture the system provenance to track intermediate non-verified status, trigger accuracy checks, run periodic benchmarks for the system as a whole, blacklist harmful data).

B. Integration with Existing Agentic Workflow Frameworks

Our objective is not to invent another agentic workflow framework; instead, we introduce framework-agnostic reliability mechanisms that can be integrated onto any existing agentic workflows stack, similar to how checkpoint/restart libraries are grafted onto HPC codes for fault tolerance. For instance, when the underlying platform already exposes a checkpointing hook (e.g., `BaseCheckpointer` in LangGraph), our checkpointing extension simply augments that hook with three extra concerns: run benchmarks to check current accuracy; compare against previously registered benchmark results; and, depending on how comparable they are, recommend either a checkpoint or a restart (see Listing 1). Other mechanisms, such as periodic benchmarks (Section IV-C1a) and safeguard controllers (Section IV-C1c), are proposed. Similarly, we elevate these ideas to the framework level by introducing an orchestration and reliability controller. This component is responsible for capturing provenance data and internal reasoning, initiating reliability checks, and triggering recovery procedures when faults are detected. The following sections detail the reliability mechanisms proposed at both the agent (Section IV-C1) and system (Section IV-C2) levels, including checkpointing, benchmarking, and provenance tracking, which collectively enhance the robustness and trustworthiness of agentic workflows.

```
import langgraph as lg
from lg.checkpoint.base import BaseCheckpointer

class ReliabilityCheckpointer(BaseCheckpointer):

  reliability_benchs: []
  previous_acc: {}
  acc_threshold: []

  # Checks agent's reliability
  def check_reliability(self,
                        reliability_benchs,
                        acc_threshold):

    for bench in reliability_benchs:
      acc = self.run_benchmark(bench)
      current_accuracy.append[acc]

    if current_acc.is_better(previous_acc):
      return {"recommendation": "checkpoint",
              "current_acc": current_acc}

    elif current_acc.is_worse(threshold):
      return {"recommendation": "recover" }

    else # accuracy is comparable:
      return {"recommendation": "None"}
```

Listing 1: Example of reliability checkpoint extension

C. Detailed Discussion of the Proposed Mechanisms

In this section we discuss in detail each proposed technique and potential implementation challenges.

1) Agent-Level Strategies: We describe four strategies that can be employed at the agent level.

a) Periodic Accuracy Checks: A fundamental property of agents is their ability to adapt and evolve over time by incorporating new information from their environment through *perception* modules, interagent communication, or querying of data stores. While this adaptability enhances flexibility, it

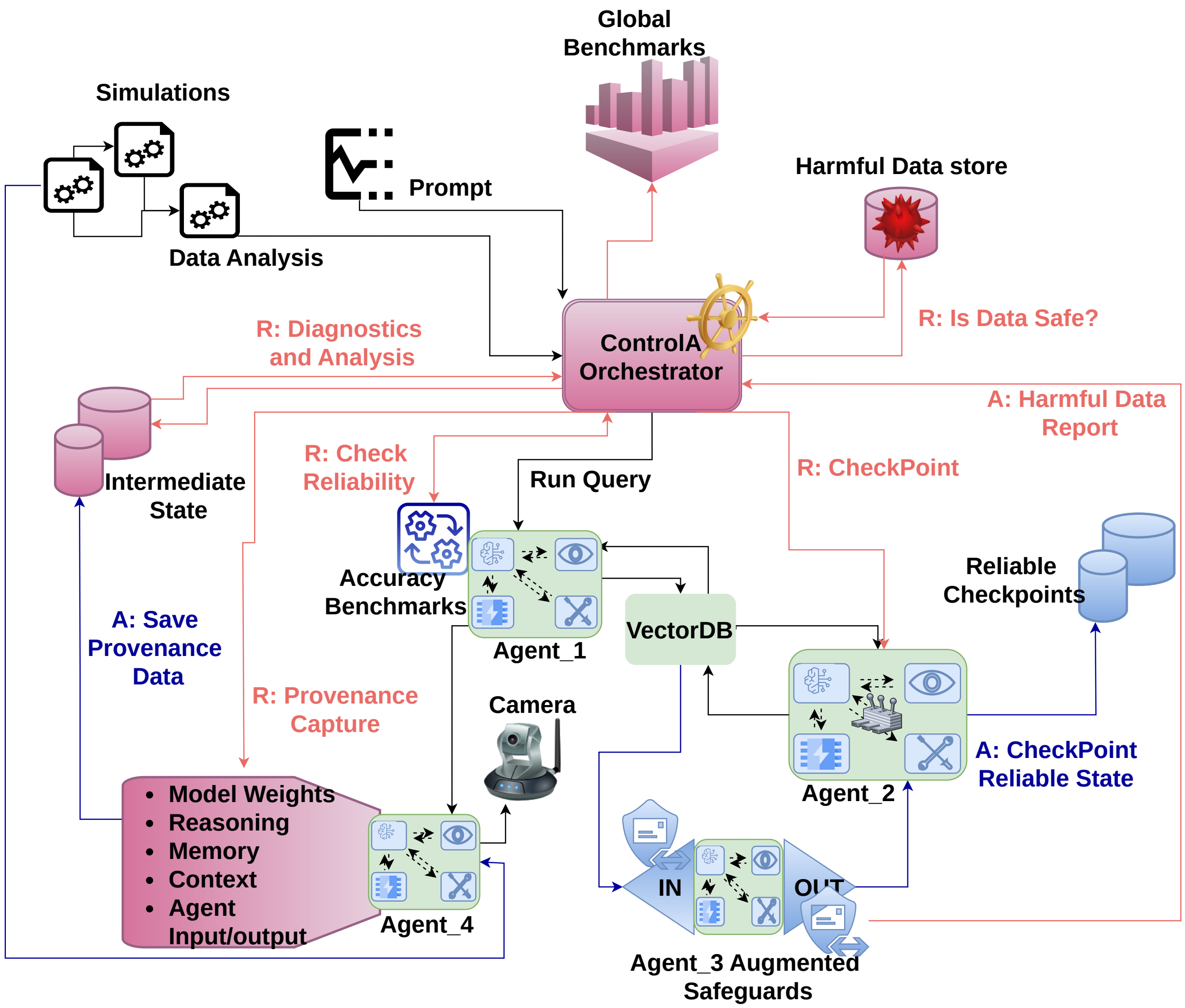

Fig. 2: ControlA: a set of mechanisms to enhance agentic workflows' reliability. Agent-level mechanisms are represented in blue shapes, and workflow-level ones are in red shapes. The *A* and *R* correspond to actions triggered by agents and requests sent from the orchestrator (responsible for monitoring the reliability), respectively.

also introduces variability in task performance. An agent's accuracy may improve or degrade as it integrates new data. Traditional validation, typically limited to the model training phase, does not account for runtime behavioral shifts or unseen new data. To ensure ongoing reliability, agents must undergo periodic accuracy assessments during execution using preselected benchmarks that are also likely to evolve. These checks are essential for detecting performance drift and isolating agents that no longer meet their required task-specific accuracy thresholds. Previous work has evaluated LLMs as agents and proposed benchmarks for evaluation [50], [51].

Integrating continuous accuracy monitoring into an agentic framework by integrating online benchmark execution raises several design questions: Where should these benchmarks be deployed: as background daemon processes, as co-located threads sharing agent resources, or on separate compute stages to minimize interference? Should the benchmarks evaluate only the agent's foundation model in isolation or incorporate the full memory state to reflect real operating conditions? If memory is included, how can we efficiently select only the most relevant context? How closely should these runtime benchmarks mirror the validation tests used during training, and how should they evolve as the agent adapts over time? Should we add specificity and robustness checks alongside accuracy? How can we automate this benchmarking pipeline to evolve with the agent?

b) Reliable Checkpoints: Similar to traditional checkpointing mechanisms used in resilience frameworks, where application state is periodically captured to enable recovery after eventual failures, we introduce the concept of *reliable checkpoints* for AI agents. These checkpoints capture an agent's internal state when it meets a predefined accuracy threshold. If a subsequent drop in accuracy is detected during an accuracy check, the system can initiate a recovery process using the most recent reliable state that was previously checkpointed. Listing 1 illustrates an example implementation of a reliability checkpointer as an extension to LangGraph's base checkpointing module. For clarity, we present a simplified interface focused on a single method: `check_reliability`. This method executes a series of predefined benchmarks specified in `reliability_benchs` and evaluates the agent's current accuracy relative to both its most recently recorded accuracy and a predefined threshold. Based on the results, it provides recommendations either to checkpoint the current state if it outperforms previous reliable states or to trigger a rollback and recovery using the most recent checkpoint that met the reliability criteria.

Like the accuracy checks, multiple domain questions arise with the reliability checkpoints. These include the following: What is included in a reliability checkpoint—the FM's internal weights only, or a set of preselected contexts? If the context is included, what is the best way to identify an outdated context and ignore it during recovery?

c) Safeguard Controllers: To strengthen the isolation of erroneous data in agentic workflows, we propose augmenting each agent with safeguard controllers—modular components tasked with verifying the reliability, safety, and validity of both inputs received and outputs produced by LLM-based agents. These helpers act as local anomaly detectors and controllers, flagging suspicious or inconsistent behavior and reporting it to the orchestrator for further action (detailed in Section IV-C2). While the foundational simplex architecture [52] has inspired real-time safety controllers in physical systems—and has recently been extended to deep learning contexts [53]—these efforts focus primarily on physical safety constraints, not the broader reliability and correctness challenges introduced by open-ended LLM behavior. Parallel efforts to build LLM-internal safeguard mechanisms [54]–[56] have demonstrated promise but also a critical limitation: If the safeguard itself is AI-based, how do we trust its correctness? This motivates our emphasis on integrating deterministic or verifiable safeguards wherever possible. For example, mathematical computations, logical assertions, and other verifiable conditions can be checked by using formal methods or logic programming frameworks, like those proposed in previous work [57]–[60]. These deterministic safeguards offer high confidence in correctness and avoid compounding uncertainty introduced by AI-generated verification. Nevertheless, we acknowledge that deterministic or programmatic validation is not always feasible, particularly in open-ended or context-sensitive tasks. In such cases, probabilistic or AI-based safeguards may be employed, potentially complemented by human-in-the-loop interventions.

Figure 3 illustrates a deterministic safeguard example de-

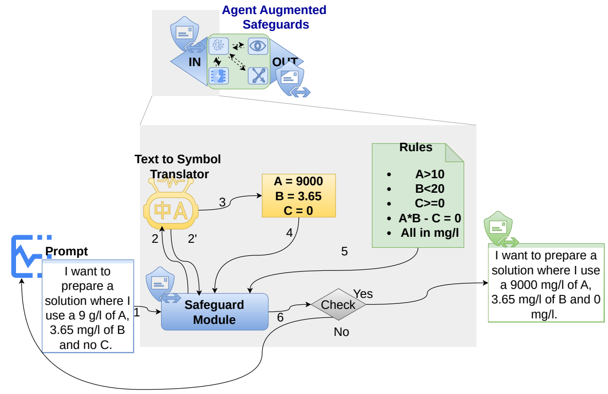

Fig. 3: Deterministic safeguard example for solution preparation agents in a self-driving laboratory

signed for solution preparation agents in self-driving laboratories. The procedure includes multiple steps to validate incoming prompts. In Step 1, the safeguard receives a natural language prompt specifying a solution preparation task. In Step 2, the prompt is passed to a domain-specific translator module, `text_to_symbol`, which converts the textual description into a formal mathematical representation, ensuring that appropriate units (e.g., `mg/L`) are used. If the translation fails, the process loops back to the safeguard for further handling or rejection. Upon successful translation (Step 3), the resulting symbolic expression is passed to the safeguard module (Step 4), which verifies predefined domain rules (Step 5). These rules are applied to validate the correctness, consistency, and safety of the input. Based on this evaluation, the system either accepts or rejects the original prompt.

d) VReAct: Reason, Verify, and React: Another promising direction is to extend existing reasoning-action loops such as ReAct [61] with an explicit validation phase, which we refer to as the VReAct mechanism. In this approach, each reasoning step (or "thought") is followed by a lightweight verification that evaluates its coherence, factual correctness, and safety before an action is triggered. As illustrated in Figure 4, the standard ReAct agent is augmented with a validator that must approve each thought before proceeding. For instance, given *Thought 1: Get 3 cl of a 9000 mg/L solution of compound A in a tube*, multiple safety and consistency checks can be applied: Is 3 cl of compound A within safe usage limits? Is this dosage currently available in the system? Is the use of a tube appropriate for this substance? These verification steps help prevent the execution of ill-formed or hazardous actions, improving reliability without altering the agent's core reasoning loop.

2) Workflow-Level Strategies: Providing agent-level reliability strategies is essential for localizing failures and preventing harmful data from propagating across an agentic workflow. However, undetected hallucinations or reasoning errors may still slip through these safeguards and accuracy checks. To address this possibility, workflow-level reliability mechanisms

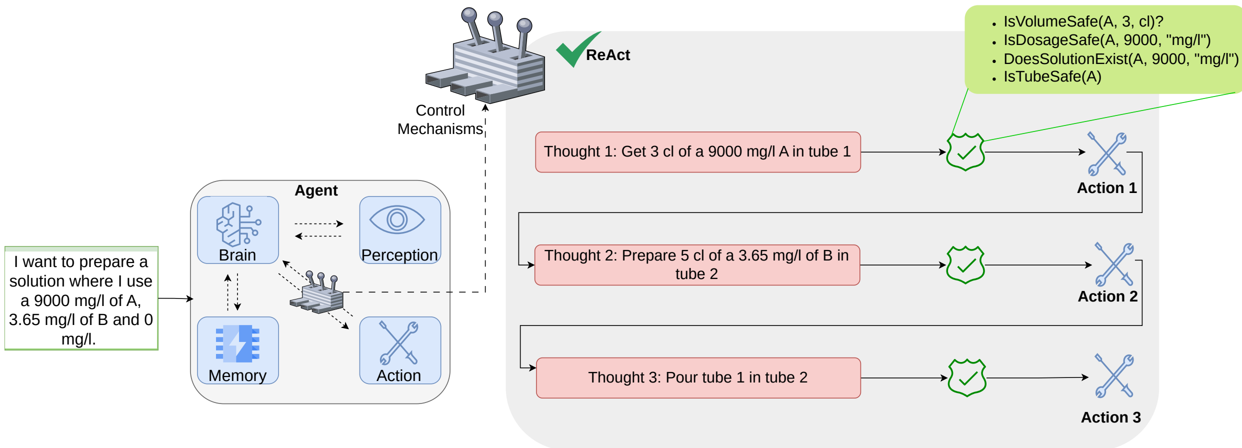

Fig. 4: Augmenting *ReAct* with a verification step, taking place between a *thought* and an *action*

are necessary to monitor internal agent operations and interactions with other agents and the environment holistically, detect anomalies across the system, and enforce corrective actions, such as quarantining suspect agents or rolling back interdependent tasks, to maintain overall system reliability.

We propose extending the agentic workflows framework with a dedicated reliability module, referred to as the ControlA orchestrator (as shown in Figure 2). This module is responsible for coordinating reliability monitoring and initiating recovery actions across the system. It captures and maintains provenance information, monitors agent behavior, and tags data identified as harmful or toxic to the system. In the following sections we detail these mechanisms.

a) Global Accuracy Checks: Global benchmarks are essential for evaluating the overall accuracy and reliability of agentic workflows while they perform their intended tasks. Although individual agents may pass their respective agent-specific accuracy checks, the systemwide accuracy may still degrade over time. This discrepancy can arise from the accumulation of undetected hallucinations or small, localized drops in agent accuracy that propagate and compound throughout the system. Such subtle degradations may go unnoticed at the agent level but can significantly impact the overall performance of the agentic workflows.

Benchmarks tailored to agentic workflows are not available for most domains. Thus, there is a pressing need for the research community to define and standardize these evaluation tools collaboratively. In parallel, as new agentic frameworks are developed and deployed, one can incrementally collect and refine benchmark datasets and metrics from real-world workflows, enabling the co-evolution of systems and their evaluation frameworks.

b) Agentic Workflow Provenance: Agentic workflows are inherently dynamic: agents driven by foundation models may generate new subgoals, access previously unknown data sources, or invoke external tools whose outputs shape future decisions in non-predictable ways. Such emergent behavior introduces significant challenges for provenance tracking. Although prior techniques have effectively supported reproducibility and traceability in static workflows [62], they fall short in capturing the evolving, interdependent reasoning paths of agentic systems. In order to uphold core scientific trust principles such as reproducibility, transparency, and explainability [63], provenance mechanisms must be reengineered to reflect not only what was executed but also how, why, and under what dynamic context. Moreover, they need to be processed at runtime, fast enough to intervene before a faulty output contaminates the downstream pipeline.

A central challenge in agentic workflows is determining how faults impact downstream tasks and data [64]. While current systems might log tool calls, agent interactions, and outputs, this level of instrumentation is insufficient for trustworthy assessment. Faults such as hallucinations not only are frequent but can be highly convincing, making naive detection approaches ineffective. Moreover, existing evaluation metrics have been shown to yield misleading results and lack generalizability across domains [65]. Reliable fault detection typically requires grounding generated content against trusted context or external knowledge sources, often with a human in the loop to define validity boundaries, provide expert guidance, or offer post hoc assessment. This complexity necessitates provenance systems capable of capturing fine-grained, contextualized agent operations and interactions. Each FM invocation, tool usage, reasoning trace, and interagent message must be linked to its downstream effects within the workflow. Such traceability enables retroactive contamination analysis: when a fault (e.g., hallucination) is identified, its origin, propagation path, and all affected artifacts can be isolated and examined.

To materialize these concepts, we propose (1) an integrated *provenance capture system* designed to capture all relevant execution events, interactions, and data exchanges within and across agents and other components; (2) an *intermediate state store* that acts as a temporary buffer, retaining agent prompts, decisions, reasoning and chain of thoughts [66], tool outputs, and inter-agent communications until the next scheduled accuracy validation; and (3) an *online diagnostic and analysis module* querying the provenance data in the intermediate state for error, inconsistency, and reliability failures. Together, these modules provide foundational support for traceability and fault isolation in agentic workflows.

c) Harmful Data Stores: Although essential, keeping track of the agent's interactions and contextualizing with the rest of the workflow, even when enriched with more structured data, is only the first step. The provenance system must enable runtime data observability [67] for continuous data monitoring, automated error detection, and error impact characterization and quantification within the context of the entire workflow.

To address these needs and enhance the security and reliability of agentic workflows, we propose *harmful-data tags* as part of the system's provenance and recovery infrastructure. When a particular data item, such as a prompt, intermediate result, or agent-generated output, is identified as harmful, it is tagged accordingly and stored in a *blacklisted data store* (shown in the top right of Figure 2) to prevent its reuse or further propagation. Marking data as harmful can be fully automated, human-driven, or hybrid. Fully automated systems can be achieved either by rule-based checks on the agents' responses, comparing them with trustworthy context and external knowledge, or by observing correlations in performance or accuracy degradation over time. Alternatively, AI-driven approaches (e.g., LLM-as-a-judge) can classify outputs as potentially harmful. Human-driven approaches rely on humans to validate the consistency of results and provide feedback, guidance, and boundaries. To enrich the data with more meaningful and structured metadata, tagging harmful data may involve additional characterization (e.g., an agent output may generate an execution error, interrupting the execution flow, or generate data that silently looks correct but invalidates a research result) and quantification (e.g., a score to evaluate the impact to contaminate data and tasks downstream).

Once tagged, characterized, and quantified, this harmful data can be isolated from downstream decision-making, and agents that interacted with or depended on it can be flagged for additional reliability checks and recovery if needed. The harmful-data tags serve as persistent markers in the system state over time, enabling retrospective analysis, root-cause tracing, and the refinement of safeguards to prevent similar failures in future executions.

V. Challenges and Opportunities

Building robust and trustworthy agentic workflows requires comprehensive benchmarks for scientific evaluation and effective safety mechanisms tailored to domain-specific challenges, at both the agent and workflow level. These aspects remain underdeveloped in current AI practice and represent major obstacles to the integration of AI agents in real-world scientific research. Here we summarize a subset of these challenges.

Lack of Benchmarks. Most current benchmarks focus on an agent's final outputs—answers, summaries, or classifications—while overlooking the intermediate reasoning steps that are crucial in scientific work. Scientific discovery demands a clear understanding of how conclusions are reached. Without process-level evaluation, it becomes difficult to determine whether an agent followed valid scientific logic or merely exploited patterns in the data. This situation is compounded by the scarcity of benchmarks designed for scientific contexts. Existing benchmarks are often domain-limited, oversimplified, or too narrowly scoped to reflect the iterative, uncertain, and complex nature of real scientific inquiry. Meaningful evaluation requires benchmarks that test hypothesis generation, experimental design, evidence interpretation, and agents' abilities to revise their approach in response to failure.

Safety and Risk Mitigation. Scientific agents may propose hazardous protocols (e.g., chemistry experiments, high-energy physics setups) or draw incorrect conclusions that mislead subsequent research or operation (e.g., motivating scenario). Traditional AI safety techniques must be adapted to domain-specific risk models. This process includes quantifying the potential harm of suggested experiments and enforcing lab safety constraints. Furthermore, safeguards must account for cascading risks in multi-agent workflows, where one agent's error can propagate downstream.

Data Provenance. Reliable scientific inference hinges on the provenance, lineage, and integrity of data. Agentic systems often ingest heterogeneous datasets, including synthetic or automatically generated corpora. Maintaining a verifiable chain of actions—from raw measurements through preprocessing, full reasoning and interpretation, and final response—is challenging but essential for reproducibility. Failure to track provenance can lead to unnoticed data leakage or contaminated training, reasoning, and operation, without enough explainability. Furthermore, streaming, storing, and indexing potentially terabytes of logs and intermediate results per run poses performance, memory, and scalability challenges. We believe that HPC-tailored persistent streaming solutions such as Mofka [68], which supports RDMA, multi-cores, and multi-NICs nodes, could help solve most of these challenges.

Evaluation Metrics Beyond Accuracy. Conventional metrics (e.g., BLEU) are insufficient. Scientific settings require domain-specific metrics for logical consistency and robustness, as well as metrics that differentiate between a model's *competence* (internal knowledge and capability) and *performance* (demonstration of this knowledge), which can enhance the predictability of an agent's behavior against unseen data. Designing such metrics that align with expert judgment is nontrivial, especially when the ground truth is unknown or evolving.

Human–Agent Collaboration and Oversight. Effective deployment demands a workflow framework where domain experts can inspect, intervene, and steer agent behavior without micromanaging every step. Challenges include designing intuitive interfaces for visualizing agent plans, surfacing uncertainty, and enabling interactive debugging. Balancing autonomy and control is particularly delicate in safety-critical experiments.

Domain Adaptation and Transferability. Scientific domains differ widely in data type, sparsity, measurement noise, and physical constraints. Agents trained on well-curated benchmarks may fail when transferred to niche subfields when confronted with out-of-distribution phenomena. Robust transfer requires modular knowledge representations, adaptation methods, and mechanisms for detecting when the agent is outside its competence region and, ideally, triggering an alert for an eventual hallucination if not refusing to proceed without human intervention.

Scalability, Overhead, and Resource Management. Although not discussed in this paper, long-horizon scientific objectives (e.g., drug discovery pipelines or climate simulations) require the coordination of compute-intensive models, lab instruments, and possibly humans. Scheduling these resources, prioritizing tasks, and optimizing cost/performance trade-offs most reliably remain largely unsolved. Additionally, the overhead of setting up these reliability control mechanisms might not be negligible, and a trade-off between large overhead and extreme reliability enhancement might take place and will be driven by the domain.

VI. Conclusion

In this conceptual framework we introduced ControlA, a multi-agent control system designed to enhance the reliability of agentic workflows in scientific contexts. Rather than proposing a fully implemented system, we outlined a modular set of reliability mechanisms, including guardrails, provenance tracking, behavioral validators, and benchmark suites, that can be integrated across a wide range of agentic workflows. Grounded in the unique requirements of scientific discovery, ControlA addresses challenges such as erroneous behavior, irreproducible results, and cascading failures through a layered architecture that supports traceability, recovery, and trustworthy reasoning. This work aims to catalyze community-driven efforts to build robust, scalable, and scientifically grounded agentic systems by identifying open research questions and deployment challenges. Future work will focus on prototyping key components of the ControlA framework, evaluating their performance in real-world scientific workflows, and refining the architecture based on empirical findings and community feedback.

Acknowledgment

This material is based upon work supported by the U.S. Department of Energy (DOE), Office of Science, Office of Advanced Scientific Computing Research, under Contract No. DE-AC02-06CH11357.

This research used resources of the Oak Ridge Leadership Computing Facility at the Oak Ridge National Laboratory, which is supported by the Office of Science of the U.S. Department of Energy under Contract No. DE-AC05-00OR22725.

The authors thank Krishnan Raghavan from MCS, Argonne, for the helpful discussions.

References

[1] Y. Ioannidis, "The 5th paradigm: AI-driven scientific discovery," *Commun. ACM*, vol. 67, no. 12, p. 5, Nov. 2024. [Online]. Available: https://doi.org/10.1145/3702970

[2] C. Leng, Z. Tang, Y.-G. Zhou, Z. Tian, W.-Q. Huang, J. Liu, K. Li, and K. Li, "Fifth paradigm in science: A case study of an intelligence-driven material design," *Engineering*, vol. 24, pp. 126–137, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2095809923001479

[3] J. Jumper, R. Evans, A. Pritzel, T. Green, M. Figurnov, O. Ronneberger, K. Tunyasuvunakool, R. Bates, A. Žídek, A. Potapenko *et al.*, "Highly accurate protein structure prediction with AlphaFold," *Nature*, vol. 596, no. 7873, pp. 583–589, 2021.

[4] D. B. Acharya, K. Kuppan, and B. Divya, "Agentic AI: Autonomous intelligence for complex goals–a comprehensive survey," *IEEE Access*, 2025.

[5] S. Murugesan, "The rise of agentic AI: implications, concerns, and the path forward," *IEEE Intelligent Systems*, vol. 40, no. 2, pp. 8–14, 2025.

[6] M. Gridach, J. Nanavati, K. Z. E. Abidine, L. Mendes, and C. Mack, "Agentic AI for scientific discovery: A survey of progress, challenges, and future directions," *arXiv preprint arXiv:2503.08979*, 2025.

[7] H. Wang, Y. He, P. P. Coelho, M. Bucci, A. Nazir, B. Chen, L. Trinh, S. Zhang, K. Huang, V. Chandrasekar *et al.*, "SpatialAgent: An autonomous AI agent for spatial biology," *bioRxiv*, pp. 2025–04, 2025.

[8] D. A. Boiko, R. MacKnight, B. Kline, and G. Gomes, "Autonomous chemical research with large language models," *Nature*, vol. 624, no. 7992, pp. 570–578, 2023.

[9] R. Feng, Y. Liang, T. Yin, P. Gao, and W. Wang, "Agentic assistant for materials scientists," *The Electrochemical Society Interface*, vol. 34, no. 2, p. 45, 2025.

[10] G. Mondillo, S. Colosimo, A. Perrotta, V. Frattolillo, and M. Masino, "Comparative evaluation of advanced AI reasoning models in pediatric clinical decision support: ChatGPT O1 vs. DeepSeek-R1," *medRxiv*, pp. 2025–01, 2025.

[11] K. Huang, D. Wu, J. Ponnapalli, and G. Huang, "AI agents in banking," in *Agentic AI*. Springer, 2025, pp. 237–277.

[12] J. Ferber and G. Weiss, *Multi-agent systems: an introduction to distributed artificial intelligence*. Addison-wesley Reading, 1999, vol. 1.

[13] "Agent to agent protocol," https://developers.googleblog.com/en/a2a-a-new-era-of-agent-interoperability/, 2025.

[14] "Model context protocol," https://modelcontextprotocol.io/introduction, 2025.

[15] Z. Ji, N. Lee, R. Frieske, T. Yu, D. Su, Y. Xu, E. Ishii, Y. J. Bang, A. Madotto, and P. Fung, "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

[16] J. Boye and B. Moell, "Large language models and mathematical reasoning failures," *arXiv preprint arXiv:2502.11574*, 2025.

[17] L. Bertolazzi, P. Mondorf, B. Plank, and R. Bernardi, "The validation gap: A mechanistic analysis of how language models compute arithmetic but fail to validate it," *arXiv preprint arXiv:2502.11771*, 2025.

[18] J. Ma, D. Dai, Z. Yuan, W. Luo, B. Wang, Q. Liu, L. Sha, Z. Sui *et al.*, "Large language models struggle with unreasonability in math problems," *arXiv preprint arXiv:2403.19346*, 2024.

[19] B. Cao, D. Cai, Z. Zhang, Y. Zou, and W. Lam, "On the worst prompt performance of large language models," *Advances in Neural Information Processing Systems*, vol. 37, pp. 69 022–69 042, 2024.

[20] Y. Zhou, Y. Zhu, D. Antognini, Y. Kim, and Y. Zhang, "Paraphrase and solve: Exploring and exploiting the impact of surface form on mathematical reasoning in large language models," *arXiv preprint arXiv:2404.11500*, 2024.

[21] J. J. Ahn and W. Yin, "Prompt-reverse inconsistency: LLM self-inconsistency beyond generative randomness and prompt paraphrasing," *arXiv preprint arXiv:2504.01282*, 2025.

[22] N. Maslej, L. Fattorini, R. Perrault, Y. Gil, V. Parli, N. Kariuki, E. Capstick, A. Reuel, E. Brynjolfsson, J. Etchemendy *et al.*, "Artificial Intelligence Index report 2025," *arXiv preprint arXiv:2504.07139*, 2025.

[23] X. Gu, X. Zheng, T. Pang, C. Du, Q. Liu, Y. Wang, J. Jiang, and M. Lin, "Agent Smith: A single image can jailbreak one million multimodal LLM agents exponentially fast," in *Proceedings of the 41st International Conference on Machine Learning*, ser. ICML'24. JMLR.org, 2024.

[24] R. Ferreira da Silva, M. Abolhasani, D. A. Antonopoulos, L. Biven, R. Coffee, I. T. Foster, L. Hamilton, S. Jha, T. Mayer, B. Mintz, R. Moore, S. Nimer, N. Paulson, W. Shin, F. Suter, M. Taheri, M. Taufer,

and N. R. Washburn, "A Grassroots Network and Community Roadmap for Interconnected Autonomous Science Laboratories for Accelerated Discovery," in *1st Workshop on Workflows, Intelligent Scientific Data, and Optimization for Automated Management (WISDOM)*, 2025.

[25] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. pearson, 2016.

[26] D. Guo, D. Yang, H. Zhang, J. Song, R. Zhang, R. Xu, Q. Zhu, S. Ma, P. Wang, X. Bi *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, 2025.

[27] S. Feng, G. Fang, X. Ma, and X. Wang, "Efficient reasoning models: A survey," 2025. [Online]. Available: https://arxiv.org/abs/2504.10903

[28] O. Macmillan-Scott and M. Musolesi, "(ir) rationality and cognitive biases in large language models," *Royal Society Open Science*, vol. 11, no. 6, p. 240255, 2024.

[29] P. C. Wason, "New horizons in psychology," 1966.

[30] D. Kahneman and A. Tversky, "Subjective probability: A judgment of representativeness," *Cognitive Psychology*, vol. 3, no. 3, pp. 430–454, 1972. [Online]. Available: https://www.sciencedirect.com/science/article/pii/0010028572900163

[31] M. Binz and E. Schulz, "Using cognitive psychology to understand GPT-3," *Proceedings of the National Academy of Sciences*, vol. 120, no. 6, p. e2218523120, 2023. [Online]. Available: https://www.pnas.org/doi/abs/10.1073/pnas.2218523120

[32] S. Min, K. Krishna, X. Lyu, M. Lewis, W.-t. Yih, P. W. Koh, M. Iyyer, L. Zettlemoyer, and H. Hajishirzi, "Factscore: Fine-grained atomic evaluation of factual precision in long form text generation," *arXiv preprint arXiv:2305.14251*, 2023.

[33] J. Jeon, "Standardization trends on safety and trustworthiness technology for advanced AI," *arXiv preprint arXiv:2410.22151*, 2024.

[34] Y. Dong, R. Mu, Y. Zhang, S. Sun, T. Zhang, C. Wu, G. Jin, Y. Qi, J. Hu, J. Meng *et al.*, "Safeguarding large language models: A survey," *arXiv preprint arXiv:2406.02622*, 2024.

[35] J. Welbl, A. Glaese, J. Uesato, S. Dathathri, J. Mellor, L. A. Hendricks, K. Anderson, P. Kohli, B. Coppin, and P.-S. Huang, "Challenges in detoxifying language models," *arXiv preprint arXiv:2109.07445*, 2021.

[36] S. Gehman, S. Gururangan, M. Sap, Y. Choi, and N. A. Smith, "RealToxicityPrompts: evaluating neural toxic degeneration in language models," *arXiv preprint arXiv:2009.11462*, 2020.

[37] B. Xia, Q. Lu, L. Zhu, and Z. Xing, "An AI system evaluation framework for advancing ai safety: Terminology, taxonomy, lifecycle mapping," in *Proceedings of the 1st ACM International Conference on AI-Powered Software*, 2024, pp. 74–78.

[38] R. Bommasani, P. Liang, and T. Lee, "Holistic evaluation of language models," *Annals of the New York Academy of Sciences*, vol. 1525, no. 1, pp. 140–146, 2023.

[39] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray *et al.*, "Training language models to follow instructions with human feedback," *Advances in Neural Information Processing Systems*, vol. 35, pp. 27 730–27 744, 2022.

[40] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan *et al.*, "Training a helpful and harmless assistant with reinforcement learning from human feedback," *arXiv preprint arXiv:2204.05862*, 2022.

[41] C. Anil, E. Durmus, N. Panickssery, M. Sharma, J. Benton, S. Kundu, J. Batson, M. Tong, J. Mu, D. Ford *et al.*, "Many-shot jailbreaking," *Advances in Neural Information Processing Systems*, vol. 37, pp. 129 696–129 742, 2024.

[42] R. Lapid, R. Langberg, and M. Sipper, "Open sesame! universal black box jailbreaking of large language models," *arXiv preprint arXiv:2309.01446*, 2023.

[43] Z. Wang, J. Liu, S. Zhang, and Y. Yang, "Poisoned LangChain: Jailbreak LLMs by LangChain," *arXiv preprint arXiv:2406.18122*, 2024.

[44] O. Topsakal and T. C. Akinci, "Creating large language model applications utilizing LangChain: A primer on developing LLM apps fast," in *International Conference on Applied Engineering and Natural Sciences*, vol. 1, no. 1, 2023, pp. 1050–1056.

[45] B. Auffarth, *Generative AI with LangChain: Build large language model (LLM) apps with Python, ChatGPT, and other LLMs*. Packt Publishing Ltd, 2023.

[46] Q. Wu, G. Bansal, J. Zhang, Y. Wu, B. Li, E. Zhu, L. Jiang, X. Zhang, S. Zhang, J. Liu *et al.*, "Autogen: Enabling next-gen LLM applications via multi-agent conversation," *arXiv preprint arXiv:2308.08155*, 2023.

[47] J. Wang and Z. Duan, "Agent AI with LangGraph: a modular framework for enhancing machine translation using large language models," *arXiv preprint arXiv:2412.03801*, 2024.

[48] J. G. Pauloski, Y. Babuji, R. Chard, M. Sakarvadia, K. Chard, and I. Foster, "Empowering scientific workflows with federated agents," *arXiv preprint arXiv:2505.05428*, 2025.

[49] B. Wang, W. Chen, H. Pei, C. Xie, M. Kang, C. Zhang, C. Xu, Z. Xiong, R. Dutta, R. Schaeffer *et al.*, "DecodingTrust: A comprehensive assessment of trustworthiness in GPT models," in *NeurIPS*, 2023.

[50] X. Liu, H. Yu, H. Zhang, Y. Xu, X. Lei, H. Lai, Y. Gu, H. Ding, K. Men, K. Yang *et al.*, "Agentbench: Evaluating LLMs as agents," *arXiv preprint arXiv:2308.03688*, 2023.

[51] D. Dorn, A. Variengien, C.-R. Segerie, and V. Corruble, "Bells: A framework towards future proof benchmarks for the evaluation of LLM safeguards," *arXiv preprint arXiv:2406.01364*, 2024.

[52] D. Seto, B. Krogh, L. Sha, and A. Chutinan, "The simplex architecture for safe online control system upgrades," in *Proceedings of the 1998 American Control Conference. ACC (IEEE Cat. No. 98CH36207)*, vol. 6. IEEE, 1998, pp. 3504–3508.

[53] N. Salamini, F. Nesti, M. Marinoni, G. Cicero, G. Serra, A. Biondi, and G. Buttazzo, "Integrating the simplex architecture to enhance safety in deep learning autonomous systems," in *Proceedings of the ACM/IEEE 16th International Conference on Cyber-Physical Systems (with CPS-IoT Week 2025)*, ser. ICCPS '25. New York, NY, USA: Association for Computing Machinery, 2025. [Online]. Available: https://doi.org/10.1145/3716550.3725155

[54] H. Inan, K. Upasani, J. Chi, R. Rungta, K. Iyer, Y. Mao, M. Tontchev, Q. Hu, B. Fuller, D. Testuggine *et al.*, "Llama Guard: LLM-based input-output safeguard for human-ai conversations," *arXiv preprint arXiv:2312.06674*, 2023.

[55] Z. Wang, F. Yang, L. Wang, P. Zhao, H. Wang, L. Chen, Q. Lin, and K.-F. Wong, "Self-Guard: Empower the LLM to safeguard itself," *arXiv preprint arXiv:2310.15851*, 2023.

[56] I. Padhi, M. Nagireddy, G. Cornacchia, S. Chaudhury, T. Pedapati, P. Dognin, K. Murugesan, E. Miehling, M. S. Cooper, K. Fraser *et al.*, "Granite Guardian: Comprehensive LLM safeguarding," in *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 3: Industry Track)*, 2025, pp. 607–615.

[57] C. P. Lee, D. Porfirio, X. J. Wang, K. C. Zhao, and B. Mutlu, "Veriplan: Integrating formal verification and llms into end-user planning," in *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems*, 2025, pp. 1–19.

[58] X. Lin, Q. Cao, Y. Huang, H. Wang, J. Lu, Z. Liu, L. Song, and X. Liang, "FVEL: interactive formal verification environment with large language models via theorem proving," *Advances in Neural Information Processing Systems*, vol. 37, pp. 54 932–54 946, 2024.

[59] Z. Wang, J. Liu, Q. Bao, H. Rong, and J. Zhang, "ChatLogic: Integrating logic programming with large language models for multi-step reasoning," in *2024 International Joint Conference on Neural Networks (IJCNN)*. IEEE, 2024, pp. 1–8.

[60] I. T. Leong and R. Barbosa, "Translating natural language requirements to formal specifications: A study on GPT and symbolic NLP," in *2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops (DSN-W)*, 2023, pp. 259–262.

[61] S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "React: Synergizing reasoning and acting in language models," in *International Conference on Learning Representations (ICLR)*, 2023.

[62] R. Souza, L. G. Azevedo, V. Lourenço, E. Soares, R. Thiago, R. Brandão, D. Civitarese, E. Vital Brazil, M. Moreno, P. Valduriez, M. Mattoso, R. Cerqueira, and M. A. S. Netto, "Workflow provenance in the lifecycle of scientific machine learning," *Concurr. Comput. Pract. Exp.*, 2021.

[63] R. Souza, S. Caino-Lores, M. Coletti, T. J. Skluzacek, A. Costan, F. Suter, M. Mattoso, and R. F. Da Silva, "Workflow provenance in the computing continuum for responsible, trustworthy, and energy-efficient AI," in *2024 IEEE 20th International Conference on e-Science (e-Science)*, 2024.

[64] R. Souza *et al.*, "PROV-AGENT: Unified provenance for tracking AI agent interactions in agentic workflows," in *IEEE International Conference on e-Science*, 2025.

[65] A. Kulkarni, Y. Zhang, J. R. A. Moniz, X. Ge, B.-H. Tseng, D. Piraviperumal, S. Swayamdipta, and H. Yu, "Evaluating evaluation

metrics – the mirage of hallucination detection," 2025. [Online]. Available: https://arxiv.org/abs/2504.18114
[66] T. Korbak, M. Balesni, E. Barnes, Y. Bengio, J. Benton, J. Bloom, M. Chen, A. Cooney, A. Dafoe, A. Dragan *et al.*, "Chain of thought monitorability: A new and fragile opportunity for AI safety," *arXiv preprint arXiv:2507.11473*, 2025.
[67] R. Souza, T. J. Skluzacek, S. R. Wilkinson, M. Ziatdinov, and R. F. da Silva, "Towards lightweight data integration using multi-workflow provenance and data observability," in *IEEE International Conference on e-Science*, 2023.
[68] M. Dorier *et al.*, "Towards a persistent event streaming system for high-performance computing applications," 2025, accepted. [Online]. Available: https://www.frontiersin.org/journals/high-performance-computing/articles/10.3389/fhpcp.2025.1638203/abstract

TABLE I: Full wildfire dataset used in the use case

Crossmodel	wildfire_autumn_Midc	wildfire_spring_Midc	wildfire_summer_Midc	wildfire_winter_Midc
R101C208	30.06649971	17.31520081	25.24279976	12.75625515
R102C206	28.36179924	17.71059990	23.90489960	12.41825928
R104C206	35.29909897	19.58810043	28.08200073	16.80937449
R104C204	37.39649963	22.13940048	28.57209969	18.61520164
R102C205	26.73360062	17.00480080	21.15489960	12.02017695
R100C206	24.71640015	15.15359974	19.91699982	10.92051029
R103C205	30.54700089	19.35530090	25.08139992	13.60779835
R102C207	29.37470055	17.85309982	25.41090012	12.65807820
R103C208	32.24689865	16.78969955	25.89329910	13.55920575
R101C207	27.93050003	17.32279968	24.18400002	11.92420164
R102C204	9.60748005	6.51708984	5.99728012	5.47834158
R101C206	26.60790062	16.69860077	22.57509995	11.46095886
R105C206	46.64360046	23.30640030	33.91030121	25.61206582
R104C207	33.09299850	15.88109970	27.10429955	14.78088479
R101C205	24.83930016	15.62829971	19.92779922	11.02200824
R103C206	31.30030060	18.94910049	25.72039986	13.73637860
R104C208	31.90390015	14.07470036	27.09420013	13.98525516
R104C205	36.29109955	21.85750008	29.00510025	17.69223046
R102C208	30.79019928	17.65640068	26.47060013	13.30716871
R103C207	31.32839966	17.54949951	25.62999916	13.09717695
R100C205	23.49169922	14.18579960	18.09670067	10.56434156
R103C204	30.01980019	18.33250046	22.54409981	14.23733334
R101C204	9.70497036	6.63100004	7.88701010	4.85567490

APPENDIX

This appendix provides the full dataset used in Section III. It provides the mid-century seasonal Fire Weather Index (FWI) projections for California. It includes values for autumn, spring, summer, and winter, mapped to geospatial grid cells identified by crossmodel coordinates (e.g., R101C204), which represent specific locations in the climate model.