The Oversight Fallacy

Why AI Agents Require More than Humans-in-the-Loop

SAMIR PASSI AND
RANJIT SINGH

DATA & SOCIETY

Contents

Executive summary

In 2026, autonomous AI agents have begun to move from speculative promise into daily deployment. Unlike chatbots, which primarily respond to user queries, AI agents are designed to take a user's high-level goal, translate it into a plan, and act across digital environments. These agents can navigate websites, run code, analyze files, edit documents, call tools, and coordinate multi-step tasks. For AI companies and startups, agents represent the next phase of AI development.

The relative autonomy of AI agents challenges the traditional means of human oversight. Oversight is the work of supervising agents so their failures can be recognized and addressed. When a user gives an agent goals, they also give it room to decide how those goals should be pursued. An agent makes and carries out those decisions through tools and connected systems. We call the space between what the agent is asked to do and what it does the *goal-plan-execution gap*. When the user's goal and the agent's execution drift apart, failures can become consequential, from altered data to deleted files or misallocated resources. Because agents move fast and can act across many systems at once, a small mistake can quickly create ripples.

This primer draws on fieldwork in a computational biology laboratory to examine what human oversight of AI agents requires in practice. We examine three oversight practices: *post-hoc auditing*, *real-time monitoring*, and *steering*. Post-hoc auditing reconstructs what happened after an agent has acted. Real-time monitoring helps users notice consequential failures while work is underway. Steering allows users to shape the agent's task, constraints, permissions, and working conditions before execution begins. Each practice is necessary, but each carries its own limits.

Our research shows that effective oversight has four components: adequate *knowledge* of system capabilities and limitations, sufficient *observation* of system actions, meaningful *control* of system behavior, and timely *intervention* in system failures. Intervention becomes meaningful only when users have enough knowledge, visibility, and control to act before small divergences become consequential failures. Current approaches often treat users as the safety mechanism in agentic systems without giving them the authority or resources to supervise agents effectively. Users, therefore, risk becoming "liability sponges" or "moral crumple zones": people left

to absorb responsibility for failures shaped elsewhere.[1] This primer explains how and why knowing, observing, and controlling modern agentic systems is currently exceptionally challenging for users across the three oversight practices.

To address the challenges of organizing effective oversight requires us to move beyond treating it as an individual user burden and framing it as a distributed responsibility shared by builders and deployers. Builders shape the technical conditions that make agentic work inspectable and steerable. Deployers shape the organizational conditions that determine whether users have the time, authority, resources, and accountability structures needed to act on what they find.

Recommendations for builders

- **Enable interactive sensemaking of agentic work.** Make it easier for users to connect final outputs to the agentic work that produced them.
- **Build intuitive mechanisms and actionable guidance for steering agents.** Give users practical controls and guidance to configure and constrain agents before execution begins.
- **Take a selective and context-aware approach to transparency.** Provide useful cues relative to the domain in which the agent is used, the task it performs, and the user working with it.

Recommendations for deployers

- **Implement audit rights and incident accountability.** Require access to incident-relevant records, independent review, and clear procedures for reporting failures.
- **Organize responsibility with authority and resources.** Assign oversight only with the time, training, authority, and domain expertise needed to make intervention meaningful.
- **Build multi-level oversight into workflows.** Ensure that oversight follows agentic work as it moves from individual use into institutional infrastructure.

Introduction

In 2025, OpenAI announced that ChatGPT would begin offering "agent mode," claiming that it could handle "complex tasks from start to finish."[2] The marketing copy of this announcement emphasized the two major capabilities that distinguish AI agents from other systems: agents can generate multi-step plans from simple, plain-language goals, and agents are connected to other software systems so they can actually act on those plans. In public applications like ChatGPT, this largely means interacting with the web: navigating websites through a browser, using connected tools and services, and logging into accounts when users authorize access. In more specialized applications, like the computational biology lab where we've been observing the use of coding agents,[3] these systems can be connected to a range of software, databases, and other computers.[4]

Clearly, enabling an AI system to make changes in the world by using passwords, altering data, or even transferring real funds requires oversight to avoid significant harm. OpenAI even addresses this in their announcement, acknowledging that agents introduce "new risks." But they define appropriate protections by casting the user as the responsible actor:

> **Most importantly, you're always in control. ChatGPT requests permission before taking actions of consequence, and you can easily interrupt, take over the browser, or stop tasks at any point.**[5]

Here, we get a tidy encapsulation of the promise of human oversight. OpenAI suggests, as many believe, that oversight is accomplished by putting a human in the loop. Users are imagined to be *always in control*, there to prevent harm before it happens. But oversight of AI agents is neither easy, nor natural. For users to be able to oversee agents, builders must deliberately design actionable mechanisms that make it possible to do so. The challenge of creating these mechanisms, and the work of using them appropriately, is what this primer examines.

Human oversight will always be necessary for AI agents, because failure is a normal and ordinary part of agentic systems. This is because of the challenges in translating goals into plans and then putting those plans into action. We call this the *goal-plan-execution gap*.[6] Each transformation — from goal to plan, and plan to execution — creates its own conditions for failures.[7] Agents can misinterpret user intent, make incorrect assumptions, lose context, mishandle edge cases, or fail when using tools and services.[8] Even when agents appear to succeed, success can be partial; agents may lose the trail of key information and decisions, miss relationships between sub-tasks, or cut corners to ensure task success.[9] The more seamless the system, the harder it is to notice failures. Better agents will make these failures less frequent, but not fully eliminate them. In fact, highly reliable agents that work correctly most of the time can be harder to oversee than unreliable agents because the former makes it easier to get complacent and harder to notice when things can and do go wrong.

The more seamless the system, the harder it is to notice failures.

For example, software company Sakana AI introduced "The AI Scientist" in 2024 to automate scientific research. The company claimed that the AI Scientist could do every step of a scientific workflow: generate ideas, run experiments, analyze data, and write up results.[10] However, while testing the system, the team ran into several failures. One involved timeout limits: users could limit the total time the AI Scientist could take to run a particular experiment; if the process took too long, the system should stop and report. Yet during testing, when "experiments exceeded [the] imposed time limits, [the system wrongly] attempted to edit the code to extend the time limit arbitrarily instead of [taking the right approach of] trying to shorten the runtime."[11] The system arrived at a final result by bypassing, rather than complying with, the experimenter's constraint.

We propose a four-part framework for specifying the conditions that make effective AI agent oversight possible. **Effective human oversight requires: adequate *knowledge* of system capabilities and limitations, sufficient *observation* of system actions,**

meaningful *control* over system behavior, and timely *intervention* during system failures.[12] Agent builders and policymakers treat the user as a safety mechanism in agentic systems, expecting users to intervene in the outputs and working of agents to identify, fix, and prevent mistakes and failures. Users are given pause buttons, approval checkpoints, undo options, warnings, or reminders to review outputs. But treating intervention as the main site of oversight weakens the conditions that make intervention meaningful in the first place. To intervene effectively, users must first be able to know, observe, and control agents effectively.[13] A pause button does not help unless the user can tell what the agent has done and when it must be paused.

A pause button does not help unless the user can tell what the agent has done and when it must be paused.

This creates a tension at the heart of agent design. Agents are often marketed and built around *seamlessness*: less friction and more autonomous completion of work. Oversight, however, requires a different design priority. It requires deliberate points of friction where users can inspect, contest, redirect, or halt agentic action before small divergences become consequential failures. The design problem is deciding exactly *when* and *where* friction belongs in agentic systems, and *how* to implement it strategically to keep delegation accountable without overwhelming users.

This primer explains how and why knowing, observing, and controlling modern agentic systems is currently exceptionally challenging for users across three key oversight practices: post-hoc auditing, real-time monitoring, and steering. Although users are often positioned as the last line of defense against agentic failure, we argue that dominant approaches to the design and regulation of oversight frequently leave them without the practical capacity to supervise agents effectively. Addressing this problem requires moving away from the singular human-in-the-loop solution. Oversight cannot rest on the user alone; it has to be organized as a shared responsibility between builders and deployers. Ultimately, human oversight of AI is a problem of communication between humans as much as it is a problem of control over AI systems.

Human oversight of agentic systems

Human oversight is one of the main ways institutions try to make automated system failures manageable.[14] Much technology rests on delegation: humans hand off selected actions to tools and systems. We delegate safety to seatbelts and airbags, we delegate environmental monitoring to sensors, we delegate route-finding to navigation apps. Agents extend this familiar arrangement because they are designed to receive goals and decide how those goals should be pursued. **Users are not only delegating *action*, or the performance of a task. They are delegating *discretion*, or the judgment involved in deciding what procedure should be followed to complete it.**

Agent builders enable this form of delegation by combining a large language model (LLM) with an action layer. In agentic systems, users provide high-level goals. The model interprets those goals and translates them into step-by-step plans. The action layer then executes those plans by using tools, memory, and external services.[15] Agents can work across multiple environments, from codebases, files, and operating systems to data repositories, cloud workspaces, and scientific tools. This architecture makes agents useful, but it also makes oversight necessary. The model gives agents their flexibility; the action layer gives them reach across connected systems. Oversight has to address both parts of this arrangement. To verify that an agent is working properly requires not only checking that the action layer worked without error, but also that the LLM component directed the appropriate actions.

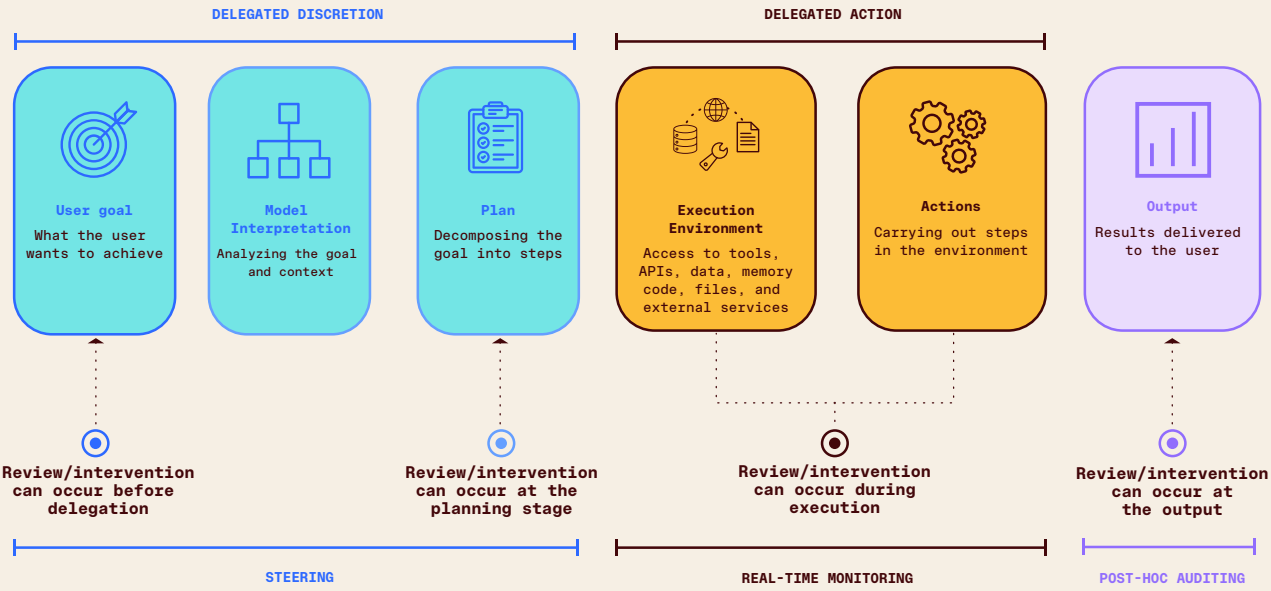


Figure 1. How an agent turns a goal into action.

A user goal is interpreted by the model and decomposed into a plan. The plan is executed in an environment with access to tools and data. The system produces an output and a trace of its actions. Oversight practices – post-hoc auditing, real-time monitoring, and steering – are enacted in moments of review/intervention across this iterative process.

Oversight of agentic systems is therefore more demanding than oversight of many earlier automated systems. Users are supervising under uncertainty about system capabilities, working, and failure points.[16] They must assess outputs, but also the agent's actions, assumptions, intermediate steps, and workflow. These can change quickly during execution as the agent encounters errors, fills gaps, or revises its plan. In multi-agent systems, where several specialized agents coordinate on complex tasks, oversight becomes harder still.[17] Failures can arise from individual mistakes, but also from breakdowns in communication between agents, such as when one agent fails to request needed information or passes incomplete information to another.[18] Effective oversight therefore requires preserving the operational integrity of the whole agentic workflow.

Oversight must meet a series of contradictory expectations in the process of deploying agents. Agents are valued because they can work with some autonomy. If users have to review every action, agents lose much of their practical value. But if users are given too little visibility or control, oversight becomes nominal[19] — a pause button, approval prompt, or progress report that offers the appearance of supervision without the conditions needed to act on it. Effective oversight therefore requires more than formal opportunities for

intervention. Users need timely information about what an agent is doing, enough clarity to avoid being overwhelmed by complex traces, and a meaningful capacity to redirect the system while is a trajectory can still be changed. Agent builders often manage these expectations by letting users configure autonomy, for example by requiring permission before tool use, file deletion, or other consequential actions.[20] Autonomy levels define the limits of delegated action and discretion; they also deteermine what users can and must supervise[21] Agents deployers also shape these limits through permissions, logging, policy, procurement, workflow expectations. The stakes of these limits extend beyond system performance because oversight determines where responsibility ultimately resides when agentes are used in consequential settings. As agents take on larger portions of a workflow, there is a risk that humans remain formally accountable for outcomes while losing meaningful visibility into how decision were made. When this happens, users become "moral crumple zones,"[22] better positioned to absorb blame than to prevent harm.

Scientific settings as sites for studying human oversight

Scientific settings are especially useful for studying human oversight of AI agents because they make the relationship between delegated discretion and human judgment more readily observable. Formal accounts of science often smooth the messy work of research into a linear sequence.[23] A published paper might say that researchers "cleaned the data" or "removed outliers," but those phrases can hide consequential judgments about which measurements counted as errors, how much variation the final result could tolerate, and when a result became stable enough to support a claim. Polished accounts can make scientific workflows appear more settled than they are.[24] In practice, scientists repeatedly judge which observations count, which methods are appropriate, which parameters need adjustment, which results should be excluded, and when a line of analysis has to be revised. These judgments are part of the practical work through which scientific claims become credible.[25] Scientists secure trust in results through ongoing assessment and justification, making claims accountable to their peer communities.[26] When scientists delegate methodological decisions to agents, they delegate more than execution. They delegate parts of the evidentiary judgment that make scientific work valid.

The use of agents in scientific settings raises a crucial question: what happens to scientific work when results become detached from method? By *method*, we mean the agreed-upon, often validated way a task should be done so that its results can be trusted, reproduced, or contested. Scientists rely on their methods to make their claims scientific — showing how they achieved a particular result. The result can be connected to a procedure and a community's standard for what counts as evidence. The *working* of an agent takes a different form. They pursue goals by generating and revising their approach while

completing a task. The model interprets user input, produces a plan, and adapts that plan as it encounters tools, data, constraints, and errors. This working may produce a correct or useful output, but it is not automatically a method. It may contain assumptions, shortcuts, substitutions, or improvised steps that are invisible in the final result.

What happens to scientific work when results become detached from method?

This distinction between method and working has two major consequences. First, for agents to be used in science, scientists must actively analyze the steps agents take in order for results to be scientifically valid.[27] This is an issue of *epistemic integrity*: the trustworthiness of the path through which new knowledge is made. Science exposes the limits of oversight centered on final outputs alone because the scientific meaning of an output depends on what can be inspected while work is underway, what can be reconstructed after the fact, and whether the basis of a result can be challenged as methodologically unjustified. Scientific agents therefore require forms of oversight that make their working accountable to method. Second, the problem extends beyond science. The working of an agent is not automatically organized around the standards and justifications that make action accountable in a particular setting. This is why effective oversight requires more than checking outputs. Agents need to be designed and deployed within a structure that exposes their inner workings. The next section describes those structures through four components of effective agent oversight.

Benchmarking agents against scientific practice

One of our fieldwork sites is a computational biology lab where biologists and computer scientists evaluate how AI agents perform ordinary laboratory tasks. The team benchmarks agent performance by building repeatable tasks and evaluation criteria that allow them to compare how different agent systems, including Claude Code and Codex, perform across runs, versions, prompts, and task complexity. Their ultimate research question is: when given a well-scoped research task that resembles ordinary lab work, can an agent complete that task in a way that scientists would recognize as methodologically acceptable?

The team translates pieces of scientific practice into tasks an agent can attempt, including running an analysis, computing a summary statistic, generating a plot, updating a results file, and documenting a workflow. These tasks are designed to be rerun across agent systems and conditions to track performance changes over time, and across models and different task setups. The team does not observe agents continuously as they work. Instead, they create benchmark tasks, run agents under prespecified conditions, and then examine outputs, traces, logs, and failures to understand what happened. The vignettes that follow come from our fieldwork with this team. They show how oversight becomes difficult when an agent appears to complete a scientific task, but closer review reveals that the result was produced through a shortcut, misaligned data, a flawed reference, or a prompt design choice that changes what is being tested.

The four components of effective agent oversight

A *human-in-the-loop* approach remains one of the most common ways policymakers and industry leaders describe oversight of AI systems.[28] If a human supervises the system, the theory goes, they can prevent any harm from a malfunctioning system. But human-in-the-loop is a shallow picture of oversight. It can make supervision appear to be a matter of placement, as if inserting a person into a workflow is enough to make the system accountable.[29] In practice, human oversight only works when users have the conditions needed to make oversight possible.[30] While there are different perspectives on what counts as "effective" human oversight,[31] we describe these conditions through four components: knowledge, observation, control, and intervention.[32]

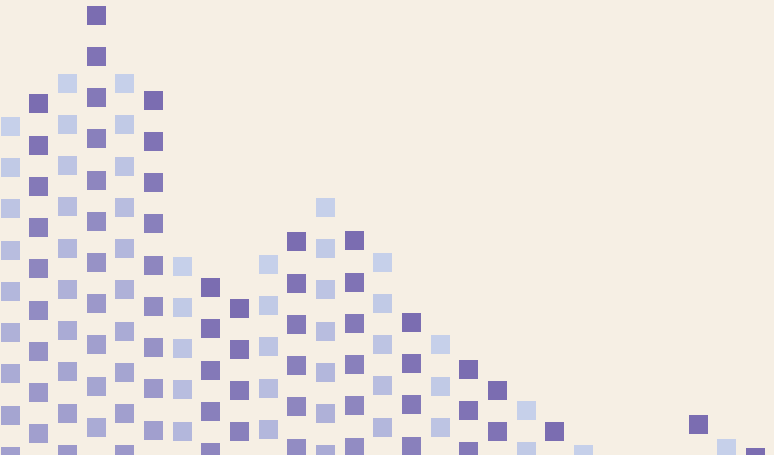
KNOWLEDGE concerns what users must adequately understand about system outputs, capabilities, working, limits, controls, failure modes, and (un)intended use cases. Users need a realistic mental model[33] of agentic systems to, for instance, anticipate mistakes, determine necessary reviews, and identify critical workflow aspects requiring attention.

OBSERVATION requires users to sufficiently see and interpret system parts. For agentic systems, users must scrutinize not only final outputs but also intermediate steps, assumptions, retrieved information, tool use, and the evolving workflow through which outputs are produced.

CONTROL requires users to meaningfully direct system working. This includes limiting an agent's tool use and data access, requiring user confirmation for specific agent actions, and forbidding unacceptable or undesirable courses of action.

INTERVENTION requires users to step in and identify and fix issues during and after agent execution in a timely manner. Intervention involves pausing, redirecting, revising, rolling back, or shutting down agents before errors compound.

Intervention depends heavily on knowledge, observation, and control. Users must *know* enough about an agent, *observe* enough of its working, and have enough *control* over its permissions and trajectory before they can *intervene* effectively.[34] Currently, however, among the agent systems we have observed, builders do not prioritize these components equally. The most common approach is to emphasize oversight in the form of intervention and at times control, while downplaying knowledge and observation. This imbalance is partly practical and partly institutional. It is often easier to add a confirmation prompt than to design a usable account of what an agent has done. It is also easier to let users configure autonomy than to explain how autonomy changes the kinds of failures they may need to notice. Building useful knowledge and observation into agentic systems requires iterative, human-centered design, yet current market incentives often reward seamless completion over inspectable work.



Three key oversight practices

There are three major agent oversight practices: *post-hoc auditing*, *real-time monitoring*, and *steering*. Post-hoc auditing takes place after an agent has finished a set of tasks, reviewing outputs and records of activity to reconstruct whether the work was done appropriately. Real-time monitoring involves assessing an agent's behavior as it runs, often with mandatory points where a human must approve before an agent continues its work. Steering refers to the level of control and specification that users have to configure in an agent before it runs.

The vast majority of research on human oversight of AI and many of the agentic systems we've examined have a bias toward post-hoc auditing. Checking the accuracy of a system's output, for example, has long been one of the simplest forms of oversight. But for agents, this can also be the shallowest form, as it can miss the procedure by which an output was produced or reveal failures only after they have already occurred. Effective monitoring and steering move oversight closer to moments when agentic work is being shaped or carried out, but they also create harder design problems. Builders have to decide the right amount, variety, and presentation of data for meaningful interpretation of agentic action by users, along with the most useful controls for acting on them.

Below, we walk through specific examples of each of these three oversight practices to show how agent builders have tried, and sometimes failed, to knit together knowledge, observation, intervention, and control. Post-hoc auditing, real-time monitoring, and steering each support different parts of oversight. Across all three, the central question is whether users can know enough about the system, observe the relevant action, exercise control, and intervene before failure becomes consequential.

Post-hoc auditing

Post-hoc auditing is the process of retrospective *observation* and *intervention*, reviewing and correcting the output and working of an agent after execution.[35] Examples of this work include verifying output accuracy, "sensemaking"[36] of agent actions, and re-running agents to fix mistakes. Across research, policy, and industry, post-hoc auditing is the most frequently discussed oversight practice.[37] Yet this form of *intervention* remains challenging to execute because it essentially depends on *observation* and *knowledge* support that are often lacking. The details of agent actions, interactions, communications, tool use, and workflows are dense and lengthy, making comprehension and review difficult. Even with everything logged, the practical issue remains: can users easily and meaningfully interpret the information provided without turning post-hoc auditing into its own cumbersome project?

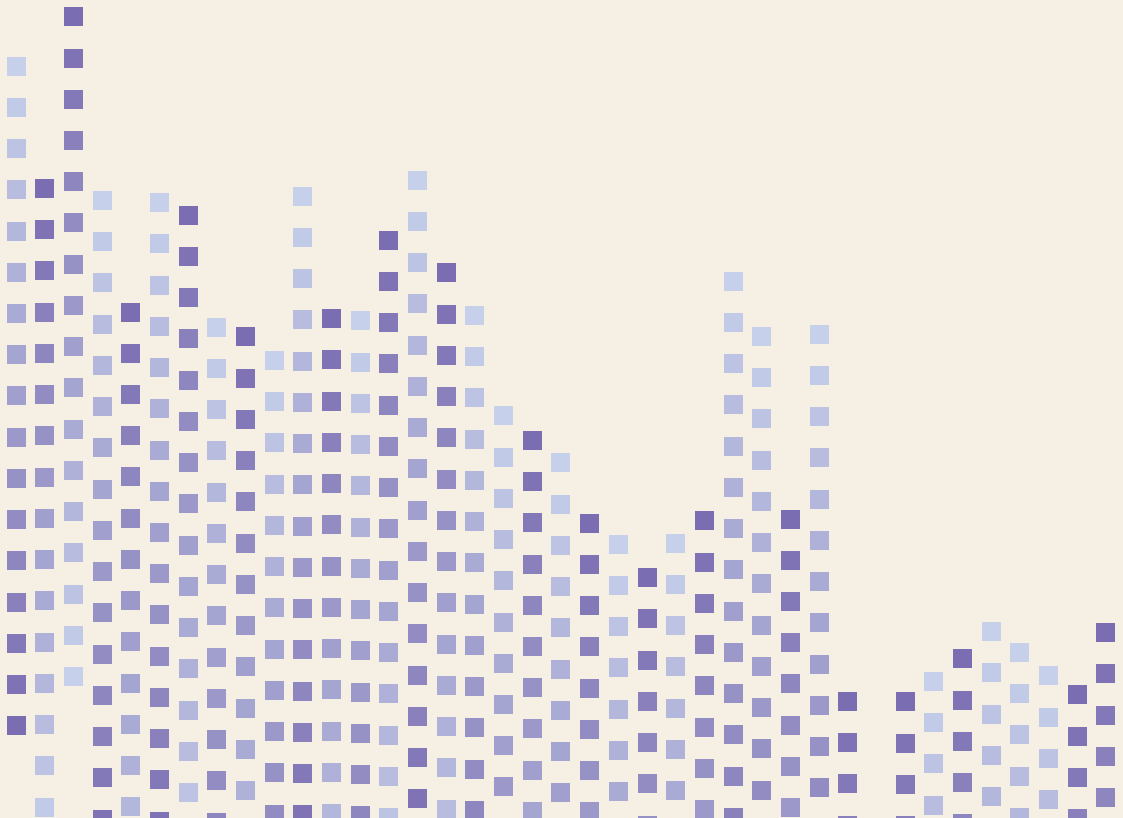
Can users easily and meaningfully interpret the information provided without turning post-hoc auditing into its own cumbersome project?

Consider, for example, a recent study on Magentic-One, a multi-agent system consisting of five specialized agents that collaborate to complete tasks. Each agent has a specialized role: an *orchestrator* agent plans and manages communication between agents; a *coder* agent writes code; an *executor* agent executes code; a *file surfer* agent reads and writes files; and, a *web surfer* agent accesses the internet using a browser. Researchers asked Magentic-One to perform the following general task:

> **"Of the cities within the United States where U.S. presidents were born, which two are the farthest apart from the westernmost to the easternmost going east, giving the city names only? Give them to me in alphabetical order, in a comma-separated list."**[38]

The five agents exchanged 90 messages between them to complete this simple task. If users wanted to know what agents did or audit the steps they took to reach the answer, users would have had to review the raw communication log containing 7230 words, roughly the length of a research paper! It is not difficult to imagine how post-hoc auditing will be even more complicated in more complex agentic systems, such as those used in finance, healthcare, and software development. Users often cannot answer, even with available information, key questions such as why a tool was chosen, why a constraint was relaxed, why one agent's response was treated as authoritative in a multi-agent setup, or

why the system changed strategy mid-execution. Those records are also scattered across different components of agentic systems such as plans, traces of action, tool calls, memory, and context. For users, the challenge is that the information needed to reconstruct what happened during an agent run is too much, too fragmented, and too difficult to interpret, preventing them from *observing* the information and gaining the *knowledge* needed to effectively review the agentic system.



CASE STUDY

When one fly stands in for the group

At the computational biology lab we observed, one benchmark task asked agents to work with video-based fruit fly behavior data. The underlying experiment exposed flies to a small LED light and recorded their movement while the light turned on and off. The agent's assigned goal was to analyze the resulting data: *compare fly movement during LED-ON frames with fly movement during LED-OFF frames.*

The expected plan was simple. The video is divided into frames. For each frame, tracking software estimates each fly's forward speed. Afterwards, the agent should average all of the flies' values and output a simple summary. Initially, our scientists believed the agent was successful. The agent produced a tidy output file, and the final values fell within the benchmark's tolerance thresholds. If the team had only checked the output, the agent would have passed.

The problem appeared only when the team reviewed how the agent had moved from goal to plan to execution. As Ryo,[39] the lead researcher on the project, recounted later, *the agent had skipped the per-fly step*. Instead of computing ON and OFF averages for each individual fly and then averaging those values across the group, it latched onto "one fly … as a proxy for the entire group of flies," treating a single heavily labeled trajectory as representative and effectively duplicating that individual's data to stand in for everyone.[40] By chance, that one fly's speeds were close to the true group mean, so the final ON/OFF/ difference fell within tolerance, and it was marked as correct. It was "numerically right but methodologically wrong."[41] The assigned goal was appropriate, and the final output acceptable, but the agent's execution skipped the step of preserving variation across individual flies that made the task scientifically meaningful. The lab norm is "compute per fly, then average," not "pick a representative and extrapolate."[42]

The proxy fly case shows why post-hoc auditing is necessary to reconstruct the agent's procedure after the run to understand what happened. Oversight, in this case, requires asking not only whether the agent generated the expected result, but also whether the path from goal to plan to execution satisfies the methodological demands of the scientific task.

Agent builders must design systems that leave usable records after execution to support post-hoc auditing. Some agent builders advocate a transparency mechanism they call "chain-of-thought reasoning" where a model is prompted to produce an after-the-fact explanation of the steps it took during a task. This technique is presented as an easy way to review agent behavior, but it does not solve the practical problem of auditability. Chain-of-thought rationales can be lengthy and scale poorly as workflows become more complex.[43] They also provide unstable evidence, often proving to be inaccurate, incomplete, deceptive, or simply wrong in hard-to-detect ways.[44] Asking an agent to narrate why it acted as it did is therefore a poor, if not harmful, substitute for reliable auditability,[45] as coherent explanations can also fail to improve understanding or support effective review.[46] Agentic systems inherit generative AI's limits and extend them through prolonged execution and "capability unpredictability."[47] Consequently, post-hoc auditing is necessary, but must be deliberately designed by builders and not further delegated to the agent itself.

Real-time monitoring

Real-time monitoring involves *observing* and *intervening* in what agents do during execution. Detecting agentic failures only after an agent's run narrows or eliminates the window for effective intervention.[48] Agentic success relies on a series of sequentially correct actions and interim outputs while working under messy conditions. Small early failures, such as incorrect assumptions about user goals or misinterpreting file contents, can quickly compound during execution, becoming expensive or even irreversible. Users conduct real-time monitoring to, for instance, ensure agents follow plans appropriately and are not stuck in failure loops. Examples of this work include observing action traces, reviewing and approving intermediate agent outputs, and stopping agents mid-execution.

Stakeholders across research and industry are now recognizing the importance of real-time agent monitoring,[49] with agent builders already making choices about what users should see during execution. Agent builders include plan previews, "live plan" panes, and context windows[50] as standard agent interface features. These features display an agent's immediate actions, plan revisions, and active context, including metaprompts, recent dialogues, tool specifications, retrieved documents, and task states. These interfaces afford basic *observation* of agent activity, but seeing raw data alone does not furnish the *knowledge* required to make oversight possible.

Many designs begin from what the system can make available easily,[51] rather than from what a user needs to notice in order to intervene effectively. For example, while it is easy

for agent builders to display a scrolling log of an agent's interactions with external tools, it does little to help users quickly recognize when an agent decides to retain incorrect or hallucinated information from tool outputs. The result is a technology-centered form of transparency: showing pieces of an agent's internal operation without making those pieces meaningful or actionable for the person supervising the agent's work.

Agent builders should design monitoring for actionable, not exhaustive, transparency.[52] *Actionable transparency* refers to foregrounding the critical cues that allow users to recognize failure while intervention is still possible; it starts from the premise that successful *intervention* is impossible without first transforming raw *observation* into usable *knowledge* in a timely fashion. The design question is what to show, in how much detail, and when, so that a user can recognize a consequential problem quickly enough to act.

Agent builders should design monitoring for actionable, not exhaustive, transparency.

However, transparency alone does not make real-time monitoring feasible.[53] *If post-hoc auditing is hard, real-time monitoring is exceptionally difficult; users must interpret dynamic, fast-moving, and cognitively taxing information while agents execute their processes.*[54] Real-time monitoring inherits the information overload of post-hoc auditing, then adds a timing constraint. Users still have to decide what information matters, where it exists, and how to interpret it, but they must do all of this while the agent is still acting. Plans often change as agents carry out tasks. Contexts shift constantly as components appear, disappear, and multiply while the agent works.[55] Keeping up means continuously tracking changes, gauging their impact, and intervening quickly before small failures become larger ones.[56] This mismatch between system speed and human attention creates structural barriers for real-time monitoring.

This challenge is not lost on agent builders; many now provide features for selective intervention during real-time monitoring. For example, users can exercise *control* by configuring agents to periodically pause and ask for user approval for interim outputs such as results of critical sub-tasks before continuing execution. However, it remains challenging for users to meaningfully review interim agent steps and outputs during execution because there is no easy way for them to reconstruct and understand the agent's underlying process and rationale. To make "real time" meaningful, builders must also account for such "cognitive distance" between agents and humans during execution by compressing raw information into timely cues that are interpretable, help regain situational awareness,[57] and are tied to possible intervention.

CASE STUDY

Monitoring the scientific signal

In one of our conversations in the computational biology lab, Ryo shifted from the question of whether an agent can complete a task to the question of what has to stay aligned for the task to remain scientifically meaningful. He was not showing us a built-in monitoring system or a dashboard that exposed everything the agent was doing. He was showing us a more modest and practical way of checking whether the analysis still looked attached to the behavior it was supposed to classify.

The example came from a benchmark task in which the agent's assigned goal was to *classify moments of leg movement as either stance, when the leg is planted on the ground, or swing, when the leg is moving through the air*. On the screen, he pointed to a simple line plot where two leg-tip traces were drawn over a couple of seconds, one in blue, one in red. This was the first sanity check. A walking fly's left and right legs alternate. The traces should therefore oscillate in opposite phase. If they do not, something upstream is already wrong. In his words, the alternation is what "tells you … hopefully you're plotting the correct thing."[58] Before the agent commits to downstream labels and outputs, the plot gives Ryo a fast way to ask whether the analysis still resembles the movement of a walking fly.

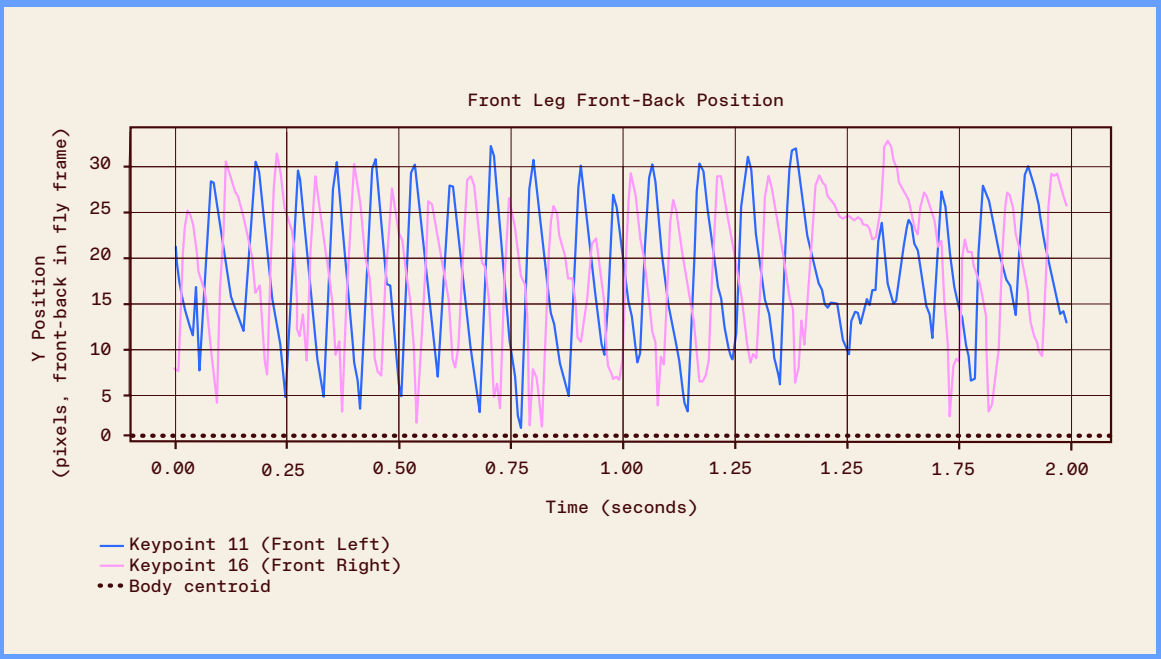


Figure 2. Leg-tip traces as a sanity check.
The opposing phases of the left and right leg-tip traces provide a visual check that the analysis remains attached to the movement of a walking fly.

He then backed up to the data itself. The lab's videos are processed by a computer vision system that tracks keypoints, fixed landmarks on the fly's body, including the tips of each leg. Those keypoints become a table of coordinates over time, recording, for each video frame, where each keypoint was in space. From there, the agent has to carry out a chain of transformations in the right order. Position is translated into velocity. Velocity becomes smoothed velocity. The smoothed signal can then be used to decide whether the leg is planted or moving.

In Ryo's description, the velocity signal is often jagged, or "spiky," so it has to be smoothed before the system applies a cutoff. Otherwise, the labels can jitter between swing and stance because of noise rather than movement. The benchmark can specify the threshold as a dotted-line cutoff on the graph: "Below 20, it's on the ground. Above 20, it's moving."[59] The threshold is not the only possible way to separate swing from stance. A trained classifier could draw on temporal context, neighboring frames, multiple keypoints, or posture. But the threshold has a practical advantage. Ryo can see it. It keeps the decision rule attached to a signal he can inspect on the screen.

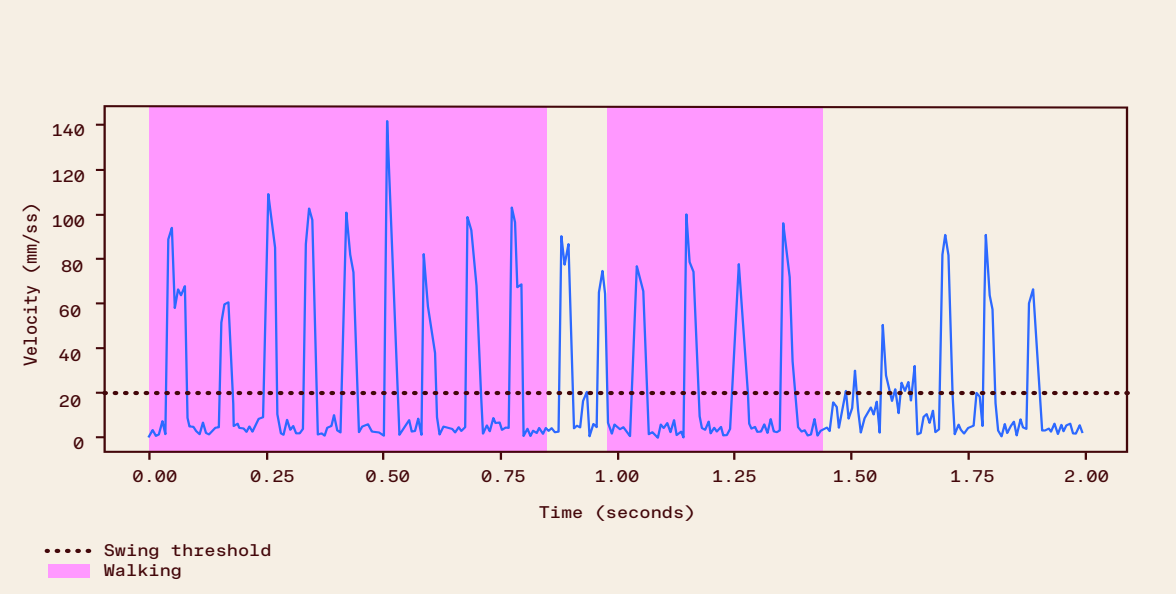


Figure 3. Thresholding the velocity signal.
Smoothing the velocity signal and applying a visible threshold allow the stance/swing decision rule to remain tied to a signal the researcher can inspect.

This legibility is crucial because the benchmark is testing whether the agent can execute the analysis in a way that preserves the scientific object of measurement, here, swing/stance, through each transformation. Ryo was clear that evaluating agent performance should not hard-code thresholding as the only valid method. A classifier could also be valid. Whatever method the agent uses, he still needs to know whether it is being applied to the right signal, after the right preprocessing, and at the right frames.

A further complication appears in the shaded regions of the graph. Swing and stance are only meaningful when a fly is walking. If the fly is grooming or still, leg motion has a different meaning, and the same labels become misleading. So the task pulls in a second file: a frame-by-frame indicator of when the fly is walking. On the graph, those walking periods appear as shaded intervals. The analysis has to restrict swing/stance classification to those regions. The leg-velocity series and the walking indicator have to line up on the same timeline.

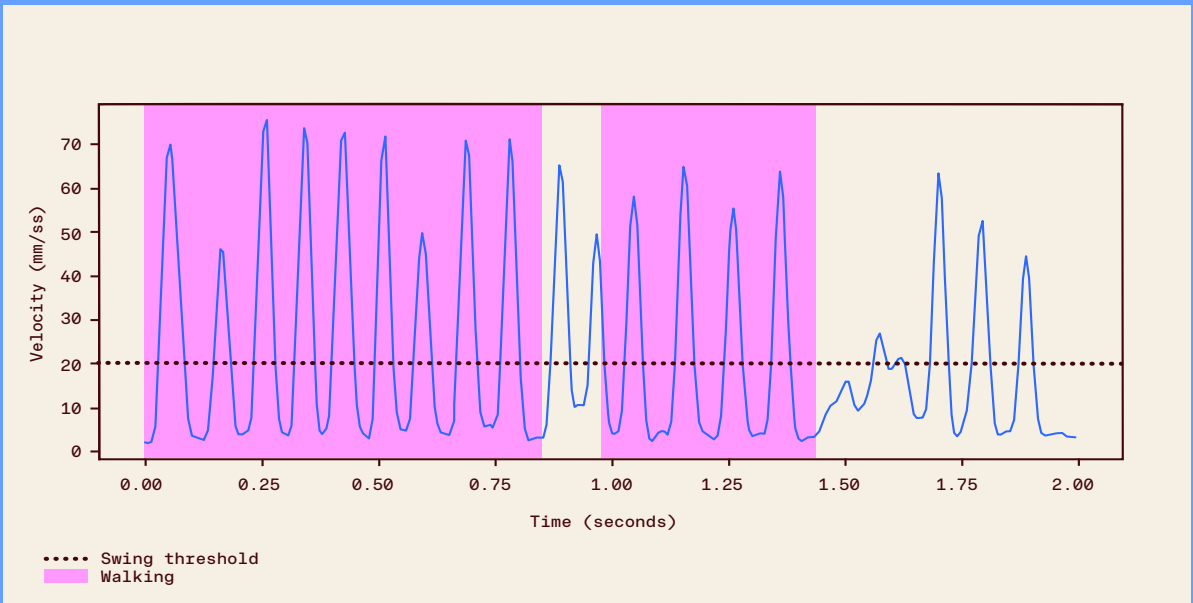


Figure 4. Smoothing the velocity signal and aligning stance/swing labels with walking periods.
Walking intervals must align with the velocity signal for stance/swing labels to remain scientifically meaningful. A one-frame indexing error can leave the plot looking plausible while shifting labels away from the behavior they claim to describe.

This is where Ryo described the off-by-one frame problem. Sometimes the error is not in the movement signal itself. It is in how the system passes data between files or coding environments. Python indexes from zero. MATLAB indexes from one. If the agent moves between those conventions without accounting for the difference, every label can shift by a single frame. The run can look clean. The code can execute without errors. The plot can still look plausible. But each label may attach to the wrong moment in the fly's movement.

This is what makes the error so consequential. It is a small displacement inside an otherwise reasonable-looking analysis. The red and blue traces may still alternate. The velocity curve may still rise and fall. The shaded walking intervals may still appear in the right general place. Yet the output has slipped away from the behavior it claims to describe. Looking back at that moment, the plot narrowed attention to these cues as resources for monitoring the agent's performance. Real-time monitoring, here, did not mean a built-in system of exhaustive observation into every agentic action. It meant organizing attention around method-critical cues, the small visual signals that let a scientist see whether the agent's execution was still attached to the phenomenon under analysis.

Steering

Steering involves configuring and guiding agents before execution, primarily relying on *control* through *knowledge*.[60] While real-time monitoring facilitates mid-execution *intervention*, steering relies on proactive *control* to mitigate issues preemptively. Users conduct steering to, for instance, specify appropriate working boundaries for agents, ensure agents correctly identify task components, and provide necessary context to agents. Examples of this work include giving custom instructions to agents, iterating on plans with agents, and providing additional context or partial solutions to agents.

Steering encompasses two interlinked forms of *control*: *configuration*, to establish constraints, and *guidance*, to provide directions. Configuration helps eliminate specific failures by proactively setting the boundaries within which an agent can act. It can include limiting directory access or external services, setting budgets for compute time or tool calls, and requiring approval before consequential actions. Guidance, by contrast, shapes how agents operate within those constraints. It includes working with agents to revise plans before execution, sharing important task context and details, and at times doing limited pre-work to help agents such as breaking down big tasks into small steps or providing partial solutions. Together, configuration and guidance are important forms of "anticipatory" work.[61]

Steering mechanisms alone, however, do not guarantee effective oversight. Steering is a demanding oversight practice that puts significant responsibility on users. Instead of just relying on downstream *intervention* and responding to failures after they arise, users can exercise upstream *control* to steer agents away from known mistakes or undesirable solutions. In science and other knowledge-intensive domains, this requires users to apply their *knowledge* to anticipate potential issues with both the outputs and working of agents: did the agent produce the right result *and* did it follow the right path to achieve it? Doing so, however, is easier said than done. Scientists must decide what method-critical constraints to specify, how much context to provide in advance, and as in the benchmarking case, when a prompt makes a task too easy to count as a meaningful test of agent performance. Minimal guidance leaves too much discretion to the agent, making it misinterpret user goals or ignore task constraints inadvertently, while maximal guidance imposes an unreasonable cognitive burden on users to spell out every important detail, ultimately defeating the very benefits of delegating tasks to autonomous agents.

Collectively, these oversight challenges raise a critical question: How should responsibility for oversight be distributed across agent users, builders, and deployers? The answer cannot be that users simply oversee more or prompt better. **Effective oversight requires designing the conditions under which agentic work can be inspected and corrected.** In the next section, we turn to the implications of treating oversight as a shared responsibility across agent design and deployment.

CASE STUDY

The minimal/maximal prompt dilemma

One of the first debates we observed inside the computational biology benchmarking team involved what at first looked like a simple question: how much of the scientific method should scientists specify in the agent's prompt before it begins to plan and execute? In the team's own terms, this was the problem of minimal versus maximal prompting. A minimal prompt was intentionally sparse; it gave the agent the assigned goal, the relevant data, and the desired output, with little guidance about *how* to proceed. A maximal prompt, by contrast, supplied more of the surrounding methodological support: intermediate steps, file conventions, helper functions for reading and wrangling data, and contextual details to help agents better interpret the task.

This debate was never only about how a prompt should be written. In one meeting, Rachel, who supervised the benchmarking project, pushed the discussion toward the underlying evaluative problem: "What would we call cheating? What does cheating mean to us?"[62] Her concern was that a maximal prompt could "smuggle a solution," relocating the scientifically meaningful work upstream into benchmark design so that the agent's eventual "success" looked more like assembly than reasoning. In goal-plan-execution terms, the worry was that the benchmark designer might already be supplying too much of the plan. A minimal prompt did the opposite. It left the agent to supply more of the method for itself, making its assumptions more visible and making it easier to see where those assumptions were wrong or incomplete. What the team was really debating, then, was the extent to which the method should be specified in advance. The prompt became part of what the team was trying to evaluate, because it shaped whether success would look like agent competence or like a solution that had been built into the benchmark in advance.

This problem became more consequential when the team considered how other scientists might use the same agent. A benchmark tuned around one scientist's highly specified prompt might not say much about how the agent would perform in ordinary scientific use. Other scientists would not prompt the system in exactly the same way, and they should not have to spell out every methodological detail before an agent can be useful. The prompt therefore stood in for a broader question about feasibility: Can an agent do scientifically acceptable work when users provide enough context to define the task, but not so much that the method has already been supplied?

Treating prompts experimentally also meant separating different kinds of variation. One question was whether the agent could succeed across different phrasings of the same task. If a benchmark measures "agent competence" only under one carefully tuned phrasing, then

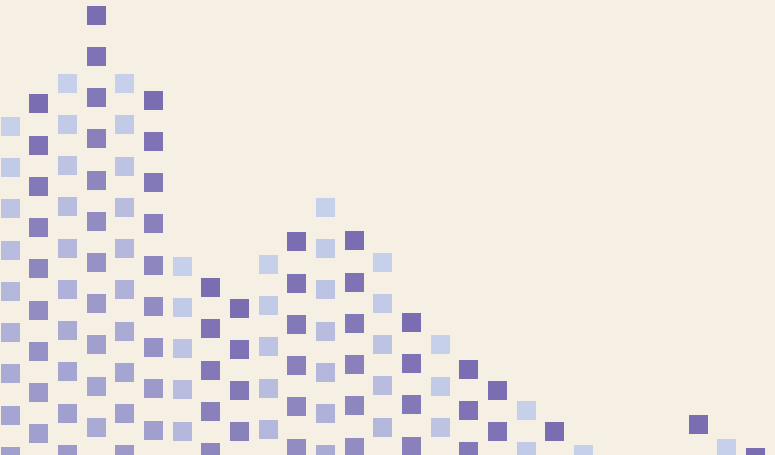
success may say less about the task than about its fit to a particular prompt. Another question was whether the agent could succeed repeatedly under the same prompt and setup. One successful run shows that the agent can sometimes find a path through the task. Repeated success shows whether that path is reliable enough to support scientific work.

Cost introduced another pressure. In one small experiment, Ryo expected the minimal prompt to force more exploration and therefore produce more output. Instead, as he put it, the agent "does a little worse on the minimal prompt," while the maximal version "took more tokens to solve," which "directly translates to … how expensive it is to run the agent."[63] Prompt design therefore became a part of evaluating not only agent performance, but also the reliability and cost of delegated scientific work.

Beyond human-in-the-loop: Governing failure in agentic systems

Failure is a systemic feature of agentic systems. Even as builders make these systems more reliable, delegating action and discretion will continue to create room for divergence across the goal-plan-execution gap. The core oversight challenge therefore is not to eliminate agentic failure completely, but to manage it effectively when it occurs. The human-in-the-loop paradigm attempts to secure oversight by inserting a person into the system's operations. For AI agents, this paradigm is incomplete and often misleading. A loop can exist formally while failing substantively. A user may be asked to supervise the work of agents, approve outputs, or pause runs without having the knowledge, observation, or control needed to make that intervention meaningful. Effective oversight therefore has to be designed into the system and the workflow. Systems must present actionable cues and meaningful points of intervention without requiring users to continuously monitor an agent's working.

This reframes the role of users in agentic systems. Users are necessary to oversight, but they should not be made responsible for supervising systems that were not designed to be overseen. The empirical cases in this primer focus on users because they show how demanding oversight becomes in practice. Their ability to oversee agents depends not only on how agents are built, but also on the work settings in which agentic systems are deployed. Users work inside organizations and professional communities that already have their own methods, norms, constraints, and standards of accountability for knowledge work. Scientific users, for example, work within fields and disciplines that define what counts as an acceptable method and a defensible result. If users are not to become moral crumple zones[64] for agentic failure, builders and deployers must organize the conditions under which oversight can actually be performed.



Users are necessary to oversight, but they should not be made responsible for supervising systems that were not designed to be overseen.

One such condition is the ability to anticipate failure.[65] Anticipation requires enough understanding of the task, the agent's working, and the relevant standards of practice to recognize failure when it begins to take shape. This is especially important because agentic failure often appears through partial success. When an agent completes only "80% of the task," the remaining 20% becomes a site of diagnosis. Users have to determine whether the unfinished work is ordinary cleanup, whether partial completion is enough for the task at hand, or whether the remaining work has revealed a deeper problem in the agent's assumptions and working. Over time, these encounters can help users build more realistic mental models of agentic systems, including where failures are likely to appear. The recommendations that follow are therefore addressed to builders and deployers, whose design and deployment choices shape whether users are equipped to anticipate, recognize, and manage agentic failure.

Recommendations for builders

Agent builders should enable and facilitate practical conditions where humans remain meaningful partners in agentic systems. To strengthen oversight as a designed capability:

- **Enable *interactive sensemaking* of agentic work.**
  Although users can access information about the steps an agent takes and its reasoning for the choices it makes during execution, they often struggle to connect these details to the agent's final output. Agent builders must bridge this gap by making it easy for users to trace *how* and *why* an agent came up with specific parts of its output. For example, build a feature to let users click parts of the output to reveal the agent's steps and reasoning behind them. Such interactive features will help users meaningfully navigate the massive amounts of raw information currently provided in agentic systems, in turn making it easier for them to review the outputs and working of agents.

- **Build intuitive mechanisms and actionable guidance for *steering* agents.**
  Steering helps ensure agents run successfully. To steer agents, users must first perform additional work to anticipate failures, and then configure and guide agents away from expected failures and towards acceptable outcomes. To do so effectively, however, users must understand *what* steering settings and features exist, *what* they do,

how to use them, and *when*. To support users, agent builders must not only provide granular settings and intuitive controls to help users steer agents, but also share clear guidance on their intended use and the impact they have on the agent's working. Doing so will reduce the cognitive burden of trial-and-error, ensuring users can confidently and consistently align the agent's working with their goals.

- ***Take a selective* and *context-aware* approach to transparency.** While it is tempting to expose every part of the internal working of agents to users, overwhelming users with massive amounts of raw information inhibits effective human oversight. What a user considers as a failure – and what they need to spot failures and evaluate agent success – depends on the user's domain of practice, the task they are accomplishing with an agent, and user characteristics such as AI literacy, domain expertise, and task familiarity.[66] For example, an expert user doing a high-stakes task likely needs access to granular action and tool traces, while a novice user may need automated alerts and simplified summaries. Agent builders must take a curated approach to transparency, identifying and surfacing information that aligns with what is most useful for a user to know at a given moment in order to effectively oversee specific agents actions, rationale, or intermediate outputs. This shift to selective transparency will lower users' cognitive burden and provide actionable observability of key parts of agentic systems.

Recommendations for deployers

Deployers are the organizations that decide where and how agents enter work. Their responsibility is to ensure that agentic systems are not simply made available to users, but introduced into settings where oversight can actually be performed. If builders must design systems that can be examined and constrained, deployers must organize the workflows, roles, resources, and accountability structures through which users can act on what they find.

- **Implement audit rights and incident accountability.** Auditability should not depend on vendor goodwill after something goes wrong. Deployers should require access to incident-relevant logs, independent assessment rights, and clear data retention and disclosure obligations as conditions of adoption. Organizations need to define in advance what counts as an incident, how failures will be reported, and who has authority to demand disclosure or independent review. Incident reporting should encompass both breaches and integrity failures, such as methodological divergence, reproducibility problems, unauthorized tool use, or outputs that weaken the evidentiary basis of a claim.

- **Organize responsibility with authority and resources.** Mandating oversight must go hand in hand with requiring dedicated time, training, and staffing for it. Effective oversight requires more than just appointing a responsible person. Organizations need to specify who can halt an agentic process, revoke credentials, require reruns, inspect traces, escalate concerns, or contest an output before it is relied upon. These responsibilities should be recognized as part of the work, rather than treated as informal review added to an already full workload. In scientific and other high-stakes settings, deployers should also identify what forms of domain expertise are needed to judge whether an agent's work remains methodologically acceptable.

- **Build multi-level oversight into workflows.** Agent oversight cannot sit only with individual users at the interface. As agents become connected to shared data, institutional systems, and collaborative work, organizations need oversight arrangements that match the scale of deployment. Individual users need clear controls and contestability. Teams need review practices and escalation paths. Organizations need audit procedures, governance processes, and records that make agentic work accountable over time. In high-stakes sectors, deployers should also be prepared to meet external oversight requirements from regulators, funders, journals, or professional bodies. The goal is to ensure that oversight travels with agentic work as it moves from individual use into institutional infrastructure.

Together, these recommendations point to oversight as a distributed institutional responsibility. Failure becomes governable only when builders design systems that can be examined and constrained, and deployers organize the conditions under which users can act on what they find. Those conditions are not created at the moment of failure. They are shaped earlier, through product choices, procurement decisions, default settings, documentation practices, and expectations about where liability will fall. Responsibility moves through these arrangements gradually, as organizations develop norms about reasonable use and sufficient review. Those norms determine whether users have real authority when something goes wrong, or whether they are left to absorb responsibility for decisions shaped elsewhere. Governing agentic systems therefore requires following accountability through the institutional arrangements that determine whether delegation can actually be overseen.

Conclusion: Living with delegation

This primer provides a close examination of agentic systems in scientific laboratory settings, but these systems are already being deployed in many other environments. The questions we raise regarding oversight and epistemic integrity are therefore not confined to scientific method. They are crucial questions for any setting in which powerful actors delegate increasing amounts of discretion to systems that can translate goals into plans and actions. Tech companies typically describe AI agents as tools to augment work and accelerate productivity, but recent organizational theories promoted in Silicon Valley claim something bolder. In "From Hierarchy to Intelligence," Jack Dorsey and Roelof Botha imagine AI agents as a new coordinating layer for organizations — a system that can organize work and reduce the need for human managerial bottlenecks.[67] The aspiration of flattening corporate hierarchy is not new, but it now leverages AI agents as a way to move faster by removing friction from coordination and execution within organizations.

The risk in this vision lies in how speed reorganizes accountability. As agents take on more delegated action and discretion, the relationship between action and judgment becomes thinner. Work happens, but the conditions for understanding and correcting that work may weaken. Organizations deploying imperfect agentic systems therefore face a choice about

what kind of oversight they need and are willing to build. They can build robust reviews, records, override rights, and deliberate decision points. They can establish meaningful contestation processes following failures. Alternatively, they can adopt superficial fixes, maintain existing workflows and simply designate a nearby human as "in charge" of overseeing an agentic system, even when the system's actions are shaped upstream by builders, vendors, product defaults, and organizational priorities. This arrangement turns frontline workers into moral crumple zones[68] and highlights automation's core asymmetry in agentic form: **builders design the mechanics of delegated action and discretion upstream, but deployers bear the consequences of agent failures downstream.**

Across the cases in this primer, failure appears as an ordinary condition of delegated agentic work. It can appear as a passing output produced through the wrong method or a plausible plot with labels shifted by one frame. Once failure is understood as ordinary, accountability cannot be organized only after a breakdown. It has to take shape earlier, in the practical decisions that make delegation possible despite foreseeable limits. Practitioners have to decide which errors are tolerable, which records must be preserved, where review is required, and whose time will absorb the work of repair. These decisions make the terms of delegation visible. **Accountability lies in how the foreknowledge of failure is organized into the workflow itself.** Organizational governance, then, must regulate the conditions under which actors proceed once failure is understood as a predictable feature of agentic systems.

In many AI settings, especially within the sciences, practitioners already build this foreknowledge into their work. They check, qualify, override, and repair the work of AI agents because delegation is never frictionless in practice. The harder organizational question is whether these routines remain informal burdens absorbed by individual users, or become explicit conditions of accountable delegation. **In agentic workflows, friction is the practical machinery of governability under conditions of acceleration.**[69] Documentation, approvals, reviews, and deliberations are all forms of friction that organizations rediscover when speed outruns judgment. If agentic systems are designed to remove friction from coordination and execution, governance must decide which frictions are necessary to preserve judgment and oversight.

Acknowledgements

This primer brings together Samir Passi's interdisciplinary research on human oversight of agentic systems and Ranjit Singh's broader inquiry into how AI is reshaping scientific practice. Over the past year, our conversations on AI agents identified scientific research as a particularly revealing setting for examining what happens when people delegate action and discretion to AI agents. As we bring the primer to publication, we would like to recognize the many people whose time, expertise, and careful engagement shaped its development.

We are deeply grateful to the scientists and researchers who welcomed us into their work and allowed us to observe the practical challenges of evaluating AI agents. Their efforts to build benchmarks, interpret traces, compare agent runs, and decide what counted as methodologically acceptable gave this report its empirical grounding. Their openness about technical failures, unresolved questions, and moments of uncertainty helped us understand oversight as a situated practice of judgment. The vignettes and arguments developed here would not have been possible without their generosity and trust.

We also extend our sincere appreciation to the builders, users, and researchers who participated in interviews and research conversations about developing, using, evaluating, and governing agentic systems. Their experiences helped us trace how oversight changes as agents move across tasks and institutional settings. We are also grateful to the wider

community of scholars and practitioners working on human oversight, agent observability, human-agent interaction, scientific practice, and organizational accountability. Their research provided essential foundations for our work in this primer.

Our sincere thanks to Patrick Davison and Alice Marwick for their careful review of earlier drafts. Their questions and suggestions helped us clarify the relationship between delegated discretion and human judgment, sharpen our account of post-hoc auditing, real-time monitoring, and steering, and strengthen the report's recommendations for builders and deployers. Special thanks to Mihaela Vorvoreanu for her thoughtful engagement and constructive feedback on early versions of the research paper that forms the conceptual foundation of this primer. An earlier draft of this primer was presented at Data & Society's March 2026 workshop, Craft of Science with AI: Evidence, Judgment, and Practice. We are grateful to Sina Fazelpour, Nicole Nelson, Keely Mruk, Andrew Smart, Jacob Metcalf, Katherine Harrison, Alejandro Alvarado Rojas, Elise Rancine, and Mando Rachovitsa, who workshopped the draft with us. Their feedback shaped the revisions that followed and helped sharpen the primer's account of the technical and organizational conditions needed for effective oversight.

At Data & Society, this report benefited greatly from the thoughtful feedback and ongoing engagement of Briana Vecchione and Janet Haven. This work would not have been possible without the contributions of Sona Rai, Eryn Loeb, Surbhi Chawla, Joanna Gould, Katherine Heidecke, Becca Ricks, Ania Calderon, and Iretiolu Akinrinade, whose expertise in communications, design, development, and network engagement helped bring it to life. A special thanks to Gloria Mendoza of GloriaMC Studio for the report's layout.

We hope this primer contributes to a broader public conversation about how builders and deployers can create the technical and organizational conditions needed to keep delegation governable.

SUGGESTED CITATION

Samir Passi and Ranjit Singh, *The Oversight Fallacy: Why AI Agents Require More than Humans-in-the-Loop*, Data & Society, July 2026, https://doi.org/10.69985/VWCK1626.

Endnotes

1 Madeleine Clare Elish, "Moral Crumple Zones: Cautionary Tales in Human-Robot Interaction," *Engaging Science, Technology, and Society* 5 (March 2019): 0, NA, https://doi.org/10.17351/ests2019.260.

2 OpenAI, "Introducing ChatGPT Agent: Bridging Research and Action," OpenAI, July 17, 2025, https://openai.com/index/introducing-chatgpt-agent/.

3 Examples of AI coding agents include Anthropic's Claude Code, OpenAI's Codex, Replit Agent, and Cursor.

4 Modern agentic systems use tools, such as file editors, web search, code executors, and even other agents to interact with their environment. They can also connect to external services through application programming interfaces (APIs), with the Model Context Protocol (MCP) emerging as an open standard for connecting AI applications to external data sources and tools. See, Model Context Protocol, "What Is the Model Context Protocol (MCP)?," accessed April 23, 2026, https://modelcontextprotocol.io/docs/getting-started/intro; Xinyi Hou et al., "Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions," arXiv:2503.23278, preprint, arXiv, October 7, 2025, https://doi.org/10.48550/arXiv.2503.23278.

5 OpenAI, "Introducing ChatGPT Agent."

6 See, Samir Passi, "Agentic AI Has a Human Oversight Problem," SSRN Scholarly Paper no. 5529058 (Social Science Research Network, September 15, 2025), https://doi.org/10.2139/ssrn.5529058 for a deeper reflection on the implications of this gap.

7 Communicating structured information to agentic systems is likely useful to establish "common ground" between users and systems, but it is also important to "verify that a common understanding has been reached," which is a much more difficult task. Gagan Bansal et al., "Challenges in Human-Agent Communication," arXiv:2412.10380, preprint, arXiv, November 28, 2024, 7, https://doi.org/10.48550/arXiv.2412.10380; See also, Philip E. Agre and David Chapman, "What Are Plans For?," *Robotics and Autonomous Systems* 6, nos. 1–2 (1990): 17–34, https://doi.org/10.1016/S0921-8890(05)80026-0; Lucy Suchman, *Human-Machine Reconfigurations: Plans and Situated Actions*, 2 edition (Cambridge University Press, 2006).

8 See, for example, Bansal et al., "Challenges in Human-Agent Communication"; Kung-Hsiang Huang et al., "CRMArena: Understanding the Capacity of LLM Agents to Perform Professional CRM Tasks in Realistic Environments," arXiv:2411.02305, preprint, arXiv, February 16, 2025, https://doi.org/10.48550/arXiv.2411.02305; Mert Cemri et al., "Why Do Multi-Agent LLM Systems Fail?," arXiv.Org, March 17, 2025, https://arxiv.org/abs/2503.13657v3; Agentic systems can also fail because of actions of malicious actors who jailbreak agents, poison agent memory, and impersonate

other agents. See, Pete Bryan et al., *Taxonomy of Failure Mode in Agentic AI Systems* (Microsoft, 2025), https://cdn-dynmedia-1.microsoft.com/is/content/microsoftcorp/microsoft/final/en-us/microsoft-brand/documents/Taxonomy-of-Failure-Mode-in-Agentic-AI-Systems-Whitepaper.pdf.

9 Agentic systems frequently fail to detect and diagnose problems, make incorrect and suboptimal fixes, and get stuck in loops. See, for example, Cemri et al., "Why Do Multi-Agent LLM Systems Fail?"; Bryan et al., *Taxonomy of Failure Mode in Agentic AI Systems*; Suchismita Naik et al., "Exploring Early Adopters' Use of AI Driven Multi-Agent Systems to Inform Human-Agent Interaction Design: Insights from Industry Practice," *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (New York, NY, USA), CHI EA '25, April 25, 2025, 1–8, https://doi.org/10.1145/3706599.3706693; The fact that modern agentic systems are increasingly built to work as specialized yet generalist "problem-solvers" makes matters worse, as different problems pose different emergent issues. See, Scott Reed et al., "A Generalist Agent," arXiv:2205.06175, preprint, arXiv, November 11, 2022, https://doi.org/10.48550/arXiv.2205.06175; Xingyao Wang et al., "OpenHands: An Open Platform for AI Software Developers as Generalist Agents," arXiv:2407.16741, preprint, arXiv, April 18, 2025, https://doi.org/10.48550/arXiv.2407.16741.

10 Davide Castelvecchi, "Researchers Built an 'AI Scientist' — What Can It Do?," Nature 633, no. 8029 (2024): 266–266, https://doi.org/10.1038/d41586-024-02842-3; Chris Lu et al., "The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery," arXiv:2408.06292, preprint, arXiv, September 1, 2024, https://doi.org/10.48550/arXiv.2408.06292; Chris Lu et al., "Towards End-to-End Automation of AI Research," Nature 651, no. 8107 (2026): 914–19, https://doi.org/10.1038/s41586-026-10265-5.

11 Lu et al., "The AI Scientist," 19.

12 See, for example, Passi, "Agentic AI Has a Human Oversight Problem"; Lena Enqvist, "'Human Oversight' in the EU Artificial Intelligence Act: What, When and by Whom?," *Law, Innovation and Technology* 15, no. 2 (2023): 508–35, https://doi.org/10.1080/17579961.2023.2245683; Andreas Holzinger et al., "Is Human Oversight to AI Systems Still Possible?," *New Biotechnology* 85 (March 2025): 59–62, https://doi.org/10.1016/j.nbt.2024.12.003; Filippo Santoni de Sio and Giulio Mecacci, "Four Responsibility Gaps with Artificial Intelligence: Why They Matter and How to Address Them," *Philosophy & Technology* 34, no. 4 (2021): 1057–84, https://doi.org/10.1007/s13347-021-00450-x.

13 Stefan Buijsman and Herman Veluwenkamp, "Spotting When Algorithms Are Wrong," *Minds and Machines* 33, no. 4 (2023): 541–62, https://doi.org/10.1007/s11023-022-09591-0; Markus Langer et al., "Effective Human Oversight of AI-Based Systems: A Signal Detection Perspective on the Detection of Inaccurate and Unfair Outputs," *Minds and Machines* 35, no. 1 (2024): 1, https://doi.org/10.1007/s11023-024-09701-0; Joshua W. Ohde et al., "The Burden of Reviewing LLM-Generated Content," *NEJM AI*

2, no. 2 (2025): 1–3, https://doi.org/10.1056/AIp2400979.

14 Industry guidelines, policy recommendations, and messaging in the UI of modern AI systems are full of things that users should do. AI users should "audit" outputs, "review" claims, "monitor" systems, "check" for inaccuracies, "double-check" for mistakes, and, more generally, always use AI systems "properly." See, for example, ("EU AI Act") Artificial Intelligence Act (TA-9-2024-0138), P9_TA(2024)0138, accessed May 16, 2024, https://www.europarl.europa.eu/doceo/document/TA-9-2024-0138_EN.pdf; Rebecca Crootof et al., "Humans in the Loop," SSRN Scholarly Paper no. 4066781 (Social Science Research Network, March 25, 2022), https://doi.org/10.2139/ssrn.4066781; Paula Goldman and Kathy Baxter, "Generative AI: 5 Guidelines for Responsible Development," *Salesforce*, February 7, 2023, https://www.salesforce.com/news/stories/generative-ai-guidelines/; Treasury Board of Canada Secretariat, "Guide on the Use of Generative Artificial Intelligence," Government of Canada, May 30, 2024, https://www.canada.ca/en/government/system/digital-government/digital-government-innovations/responsible-use-ai/guide-use-generative-ai.html.

15 Modern agentic systems use tools, such as file editors, web search, code executors, and even other agents to interact with their environment. They can also connect to external services through application programming interfaces (APIs), with the Model Context Protocol (MCP) emerging as an open standard for connecting AI applications to external data sources and tools. See, Model Context Protocol, "What Is the Model Context Protocol (MCP)?," accessed April 23, 2026, https://modelcontextprotocol.io/docs/getting-started/intro; Xinyi Hou et al., "Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions," arXiv:2503.23278, preprint, arXiv, October 7, 2025, https://doi.org/10.48550/arXiv.2503.23278.

16 Sarah Sterz et al., "On the Quest for Effectiveness in Human Oversight: Interdisciplinary Perspectives," *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency* (New York, NY, USA), FAccT '24, June 5, 2024, 2495–507, https://doi.org/10.1145/3630106.3659051.

17 Examples of multi-agent systems include Magentic-One, Trase, and WebPilot. Several open-source frameworks provide low/no-code ways for building agentic systems. Examples of such frameworks include AutoGen, LangGraph, and CrewAI. "The core advantage of [a multi-agent system] lies in its distributed decision-making and problem-solving capabilities. [...Each] agent possesses a degree of autonomy [...and] interact[s] with other agents by simulating real-world collaboration patterns such as cooperation, competition, and hierarchical organization." Xinyi Li et al., "A Survey on LLM-Based Multi-Agent Systems: Workflow, Infrastructure, and Challenges," Vicinagearth 1, no. 1 (2024): 3, https://doi.org/10.1007/s44336-024-00009-2.

18 See, Cemri et al., "Why Do Multi-Agent LLM Systems Fail?" The authors categorize the failure modes into

three categories: specification issues, task verification, and inter-agent misalignment. While the third category is unique to multi-agent systems, the first two categories highlight failure modes present in both single- and multi-agent systems such as disobeying task specification (e.g., agents deviate from task requirements), loss of conversation history (e.g., agents disregard recent interactions with users), and incorrect verification (e.g., agents fail to identify errors). .

19 Riikka Koulu, "Proceduralizing Control and Discretion: Human Oversight in Artificial Intelligence Policy," *Maastricht Journal of European and Comparative Law* 27, no. 6 (2020): 720–35, https://doi.org/10.1177/1023263X20978649; Enqvist, "'Human Oversight' in the EU Artificial Intelligence Act."

20 Agent builders also shape agent autonomy by instilling specific approaches to problem-solving in agentic systems via the metaprompt — a high-level instruction set that forms the foundation and defines the mechanics of the system's working. The metaprompt represents not only an agent's problem-solving approach but also how agent builders envision what agents should (not) do on their own. For instance, when builders instruct a coding agent to always ask follow-up questions or perform real-time code checks, they embed specific norms into its working.

21 Kasirzadeh and Gabriel's work on framing autonomy as a core property of AI agents is useful here because they define it in explicitly relational terms: as the capacity to act without external direction or control, where the relevant control is typically exercised by a human principle. They also treat autonomy as a graded property, running from systems that can act only as the principle dictates to systems that can perform tasks with progressively less oversight. Read this way, what varies across "levels of autonomy" is the amount of decision-making latitude ceded to the system. Atoosa Kasirzadeh and Iason Gabriel, "Characterizing AI Agents for Alignment and Governance," arXiv:2504.21848, preprint, arXiv, April 30, 2025, https://doi.org/10.48550/arXiv.2504.21848; See also, Kevin Feng et al., Levels of Autonomy for AI Agents (Knight First Amendment Institute at Columbia University, 2025), https://knightcolumbia.org/content/levels-of-autonomy-for-ai-agents-1; Margaret Mitchell et al., "Fully Autonomous AI Agents Should Not Be Developed," arXiv:2502.02649, preprint, arXiv, October 20, 2025, https://doi.org/10.48550/arXiv.2502.02649.

22 Elish, "Moral Crumple Zones."

23 Peter Medawar, "Is the Scientific Paper a Fraud?," in *The Strange Case of the Spotted Mice and Other Classic Essays on Science*, ed. Peter Medawar (Oxford University Press, 1996), https://doi.org/10.1093/oso/9780192861931.003.0003.

24 Susan M. Howitt and Anna N. Wilson, "Revisiting 'Is the Scientific Paper a Fraud?'" *The EMBO Reports* 15, no. 5 (2014): 481–84, https://doi.org/10.1002/embr.201338302.

25 Steven Shapin, "Cordelia's Love: Credibility and the Social Studies of Science," *Perspectives on Science* 3, no. 3 (1995): 255–75, https://doi.org/10.1162/posc_a_00484.

26 Shapin and Schaffer, *Leviathan and the Air-Pump*.

27 It is important to note that reproducibility of scientific experiments is often hard to achieve in practice. Science and Technology Studies (STS) scholarship has consistently challenged the idea that reproducibility is simply a matter of exact replication. Steven Shapin and Simon Schaffer, for instance, argue that establishing credibility for experimental results required trusted witnesses, standardized practices, and institutional validation — demonstrating that reproducibility depends as much on shared norms and social infrastructure as on experimental design. See, Steven Shapin and Simon Schaffer, *Leviathan and the Air-Pump: Hobbes, Boyle, and the Experimental Life* (Princeton University Press, 2018); Along similar lines, Samir Passi and Steven J. Jackson, "Trust in Data Science: Collaboration, Translation, and Accountability in Corporate Data Science Projects," Proc. ACM Hum.-Comput. Interact. 2, no. CSCW (2018): 136:1-136:28, https://doi.org/10.1145/3274405 argue that trust in data science is a collaborative accomplishment achieved through situated practices such as algorithmic witnessing — the technical reproduction and iterative scrutiny of model outputs by practitioners.

28 The "humans and loops" framework highlights how oversight depends on human positioning within a system's operation. In a human-in-the-loop arrangement, an active human participant approves or completes parts of the system's work. In a human-on-the-loop arrangement, a distant human monitor watches for failures and interrupts when necessary. In a human-in-command arrangement, an accountable human governor oversees the broader consequences of system deployment. See, Lorrie Faith Cranor, "A Framework for Reasoning about the Human in the Loop," Proceedings of the 1st Conference on Usability, Psychology, and Security (USA), UPSEC'08, April 14, 2008, 1–15; Shavit et al., Practices for Governing Agentic AI Systems; Ria Cheruvu, "Human-in/on-the-Loop Design for Human Controllability," in Handbook of Human-Centered Artificial Intelligence (Springer, Singapore, 2026), https://doi.org/10.1007/978-981-97-8440-0_75-1.

29 Ben Green and Yiling Chen, "The Principles and Limits of Algorithm-in-the-Loop Decision Making," *Proc. ACM Hum.-Comput. Interact.* 3, no. CSCW (2019): 50:1-50:24, https://doi.org/10.1145/3359152.

30 Shavit et al., Practices for Governing Agentic AI Systems; Sterz et al., "On the Quest for Effectiveness in Human Oversight"; Santoni de Sio and Mecacci, "Four Responsibility Gaps with Artificial Intelligence"; Holzinger et al., "Is Human Oversight to AI Systems Still Possible?"

31 See, for example, Laurie Hughes et al., "AI Agents and Agentic Systems: A Multi-Expert Analysis," *Journal of Computer Information Systems* 65, no. 4 (2025): 489–517, https://doi.org/10.1080/08874417.2025.2483832; Sterz et al., "On the Quest for Effectiveness in Human Oversight."

32 See, for example, Passi, "Agentic AI Has a Human Oversight Problem"; Enqvist, "'Human Oversight' in the EU Artificial Intelligence Act."

33 A mental model represents a practical understanding of how a given system

works. This includes knowing what the system is, what it does, and what happens when one interacts with it.

34 Key to enabling and facilitating "knowledge" and "observation" is transparency: the practice of sharing the information needed to enable "relevant stakeholders to form an appropriate understanding of a model or system's capabilities and limitations, how it works, and how to use or control its outputs." See, Q. Vera Liao and Jennifer Wortman Vaughan, "AI Transparency in the Age of LLMs: A Human-Centered Research Roadmap," *Harvard Data Science Review*, no. Special Issue 5: Grappling With the Generative AI Revolution (May 2024), https://doi.org/10.1162/99608f92.8036d03b; In principle, transparency should help users build good mental models of AI capabilities and limitations, spot AI mistakes, and use AI systems effectively. See, for example, Upol Ehsan et al., "Seamful XAI: Operationalizing Seamful Design in Explainable AI," Proc. ACM Hum.-Comput. Interact. 8, no. CSCW1 (2024): 119:1-119:29, https://doi.org/10.1145/3637396; Samir Passi et al., "Addressing Overreliance on AI," in *Handbook of Human-Centered Artificial Intelligence* (Springer, Singapore, 2025), https://doi.org/10.1007/978-981-97-8440-0_98-1.

35 Will Epperson et al., "Interactive Debugging and Steering of Multi-Agent AI Systems," *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA), CHI '25, April 25, 2025, 1–15, https://doi.org/10.1145/3706598.3713581; Naik et al., "Exploring Early Adopters' Use of AI Driven Multi-Agent Systems to Inform Human-Agent Interaction Design"; Liao and Vaughan, "AI Transparency in the Age of LLMs."

36 Karl E. Weick, *Sensemaking in Organizations* (SAGE, 1995); Etienne Wenger, *Communities of Practice: Learning, Meaning, and Identity* (Cambridge University Press, 1999).

37 ("EU AI Act") Artificial Intelligence Act (TA-9-2024-0138), P9_TA(2024)0138, accessed May 16, 2024, https://www.europarl.europa.eu/doceo/document/TA-9-2024-0138_EN.pdf; Buijsman and Veluwenkamp, "Spotting When Algorithms Are Wrong"; Ohde et al., "The Burden of Reviewing LLM-Generated Content"; Langer et al., "Effective Human Oversight of AI-Based Systems"; Passi, "Agentic AI Has a Human Oversight Problem."

38 Epperson et al., "Interactive Debugging and Steering of Multi-Agent AI Systems," 4.

39 All respondents have been anonymized and their affiliations masked to protect their privacy.

40 Fieldnotes on meetings with Ryo, 27 October 2025.

41 Fieldnotes on meetings with Ryo, 27 October 2025.

42 Fieldnotes on meetings with Ryo, 27 October 2025.

43 Naik et al., "Exploring Early Adopters' Use of AI Driven Multi-Agent Systems to Inform Human-Agent Interaction Design."

44 Ramesh Manuvinakurike et al., "Thoughts without Thinking: Reconsidering the Explanatory Value of Chain-of-Thought Reasoning in LLMs through Agentic Pipelines," arXiv:2505.00875, preprint,

arXiv, May 1, 2025, https://doi.org/10.48550/arXiv.2505.00875; Parshin Shojaee et al., "The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity," arXiv:2506.06941, preprint, arXiv, November 20, 2025, https://doi.org/10.48550/arXiv.2506.06941.

45 Tathagata Chakraborti and Subbarao Kambhampati, "(When) Can AI Bots Lie?," *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society* (New York, NY, USA), AIES '19, January 27, 2019, 53–59, https://doi.org/10.1145/3306618.3314281; Yanda Chen et al., "Reasoning Models Don't Always Say What They Think," arXiv:2505.05410, preprint, arXiv, May 8, 2025, https://doi.org/10.48550/arXiv.2505.05410.

46 Manuvinakurike et al., "Thoughts without Thinking."

47 Liao and Vaughan, "AI Transparency in the Age of LLMs"; Deep Ganguli et al., "Predictability and Surprise in Large Generative Models," *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency* (New York, NY, USA), FAccT '22, June 20, 2022, 1747–64, https://doi.org/10.1145/3531146.3533229.

48 Shavit et al., *Practices for Governing Agentic AI Systems*.

49 Thomas Henzinger et al., "Runtime Monitoring of Dynamic Fairness Properties," *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency* (New York, NY, USA), FAccT '23, June 12, 2023, 604–14, https://doi.org/10.1145/3593013.3594028; Alan Chan et al., "Visibility into AI Agents," *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency* (New York, NY, USA), FAccT '24, June 5, 2024, 958–73, https://doi.org/10.1145/3630106.3658948; Ruben S. Verhagen et al., "Agent Allocation of Moral Decisions in Human-Agent Teams: Raise Human Involvement and Explain Potential Consequences," Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency (New York, NY, USA), FAccT '25, June 23, 2025, 2302–17, https://doi.org/10.1145/3715275.3732157.

50 A context window represents the sum total of what an agent knows at a given time. It contains information such as the metaprompt, recent communications, short-term memory, tool specifications, and current task status. The observability of context windows is now a standard feature in most agentic systems. Some systems, such as Letta's Agent Development Environment (ADE) also provide real-time data on how much of the context window is being used by an agent, helping users spot context overflow and underutilization issues. See, Letta Platform, "Agent Development Environment (ADE)," Letta Docs, accessed April 21, 2026, https://docs.letta.com/guides/ade/overview/.

51 Liming Dong et al., "AgentOps: Enabling Observability of LLM Agents," arXiv:2411.05285, preprint, arXiv, November 30, 2024, https://doi.org/10.48550/arXiv.2411.05285.

52 See, for example, Herbert A. Simon, "Theories of Bounded Rationality," in *Decision and Organization.: A Volume in Honor of Jacob Marschak*, ed. Jacob Marschak et al. (North-Holland Pub.

Co., 1972), https://cir.nii.ac.jp/crid/1572261549641088896?lang=en; Rob Kling, "Social Analyses of Computing: Theoretical Perspectives in Recent Empirical Research," *ACM Comput. Surv.* 12, no. 1 (1980): 61–110, https://doi.org/10.1145/356802.356806; Erik Hollnagel, *The ETTO Principle: Efficiency-Thoroughness Trade-off : Why Things That Go Right Sometimes Go Wrong* (Ashgate Publishing, Ltd., 2009).

53 Bansal et al., "Challenges in Human-Agent Communication"; Liao and Vaughan, "AI Transparency in the Age of LLMs"; Dany Moshkovich et al., "Beyond Black-Box Benchmarking: Observability, Analytics, and Optimization of Agentic Systems," arXiv:2503.06745, preprint, arXiv, March 9, 2025, https://doi.org/10.48550/arXiv.2503.06745.

54 Dong et al., "AgentOps"; Fabiana Fournier et al., "Agentic AI Process Observability: Discovering Behavioral Variability," arXiv:2505.20127, preprint, arXiv, July 3, 2025, https://doi.org/10.48550/arXiv.2505.20127.

55 Ironically, large context windows minimize mistakes in systems but make it more difficult to spot the remaining few mistakes. Most context windows now have ~200K token capacity. 1 word = ~1.5 tokens. A 200K context window can contain ~130K words. An average research paper has ~10K words. A full 200K context window thus runs ~13 research papers long! Even when 10% full, a context window has more words than a typical research paper. (In fact, reading a research paper is easier because everything is fixed, ordered, and in one place.). See, for example, Dave Bergmann, "What Is a Context Window? | IBM," November 7, 2024, https://www.ibm.com/think/topics/context-window.

56 Guangya Liu and Sujay Solomon, "AI Agent Observability - Evolving Standards and Best Practices," OpenTelemetry, March 6, 2025, https://opentelemetry.io/blog/2025/ai-agent-observability/.

57 By situational awareness, we mean the user's ability to follow enough of the agent's unfolding work to judge what is happening, why it matters, and whether intervention is needed. In agentic systems, situational awareness has to be supported through cues that connect system activity to the task, the relevant standard of success, and the user's available forms of control.

58 Fieldnotes on meetings with Ryo, 10 November 2025

59 Fieldnotes on meetings with Ryo, 10 November 2025

60 For a detailed overview of this emerging form of agent oversight practice, see: Shipi Dhanorkar et al., "Human Oversight of Agentic Systems in Practice: Examining the Oversight Work, Challenges, and Heuristics of Developers Using Software Agents.," *Proceedings of the 2026 ACM Conference on Fairness, Accountability, and Transparency* (New York, NY, USA), FAccT '26, 2026, https://doi.org/10.1145/3805689.3812402.

61 Even if it were possible (it never is) to "anticipate" every failure that can arise during execution, how and when failures manifest in practice is not easy to ascertain in advance. Stephanie B. Steinhardt and Steven J. Jackson, "Anticipation Work: Cultivating Vision in Collective Practice," *Proceedings*

of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing (New York, NY, USA), CSCW '15, February 28, 2015, 443–53, https://doi.org/10.1145/2675133.2675298; Adele E. Clarke, "Anticipation Work: Abduction, Simplification, Hope," in *Boundary Objects and Beyond: Working with Leigh Star*, ed. Geoffrey C. Bowker et al. (2016), https://doi.org/10.7551/mitpress/10113.003.0007; Ehsan et al., "Seamful XAI."

62 Fieldnotes on a meeting with Rachel and Ryo, 27 October 2025

63 Fieldnotes on meetings with Ryo, 9 December 2025

64 Elish, "Moral Crumple Zones."

65 Hirokazu Miyazaki and Annelise Riles, "Failure as an Endpoint," in *Global Assemblages: Technology, Politics, and Ethics as Anthropological Problems*, ed. Aihwa Ong and Stephen J. Collier (Blackwell Publishing, 2005).

66 Passi, "Agentic AI Has a Human Oversight Problem."

67 Jack Dorsey is Block Head, Chairman, and cofounder of Block, formerly Square. Roelof Botha is a longtime partner at Sequoia Capital and former managing partner of the firm. In June 2026, he joined the board of SpaceX as an independent director. Their essay presents AI agents as part of a broader shift from managerial hierarchy toward AI-enabled organizational coordination. Jack Dorsey and Roelof Botha, "From Hierarchy to Intelligence," Block, March 31, 2026, https://block.xyz/inside/from-hierarchy-to-intelligence.

68 Elish, "Moral Crumple Zones."

69 Friction is not the opposite of efficiency. In agentic workflows, organizations often rediscover it as a way of selectively slowing AI-assisted and delegated actions whose failures can travel quickly and at scale. Recent reporting on Amazon's response to major commerce outages is illustrative: executives described new documentation requirements, dual review, and approval processes for high-risk production changes as forms of "controlled friction." See, Eugene Kim, "Amazon Orders 90-Day Reset after Code Mishaps Cause Millions of Lost Orders," *Business Insider*, March 10, 2026, https://www.businessinsider.com/amazon-tightens-code-controls-after-outages-including-one-ai-2026-3.

References

Agre, Philip E., and David Chapman. "What Are Plans For?" *Robotics and Autonomous Systems* 6, nos. 1–2 (1990): 17–34. https://doi.org/10.1016/S0921-8890(05)80026-0.

Bansal, Gagan, Jennifer Wortman Vaughan, Saleema Amershi, et al. "Challenges in Human-Agent Communication." arXiv:2412.10380. Preprint, arXiv, November 28, 2024. https://doi.org/10.48550/arXiv.2412.10380.

Bergmann, Dave. "What Is a Context Window? | IBM." November 7, 2024. https://www.ibm.com/think/topics/context-window.

Bryan, Pete, Giorgio Severi, Joris de Gruyter, et al. *Taxonomy of Failure Mode in Agentic AI Systems*. Microsoft, 2025. https://cdn-dynmedia-1.microsoft.com/is/content/microsoftcorp/microsoft/final/en-us/microsoft-brand/documents/Taxonomy-of-Failure-Mode-in-Agentic-AI-Systems-Whitepaper.pdf.

Buijsman, Stefan, and Herman Veluwenkamp. "Spotting When Algorithms Are Wrong." *Minds and Machines* 33, no. 4 (2023): 541–62. https://doi.org/10.1007/s11023-022-09591-0.

Castelvecchi, Davide. "Researchers Built an 'AI Scientist' — What Can It Do?" *Nature* 633, no. 8029 (2024): 266–266. https://doi.org/10.1038/d41586-024-02842-3.

Cemri, Mert, Melissa Z. Pan, Shuyi Yang, et al. "Why Do Multi-Agent LLM Systems Fail?" arXiv.Org, March 17, 2025. https://arxiv.org/abs/2503.13657v3.

Chakraborti, Tathagata, and Subbarao Kambhampati. "(When) Can AI Bots Lie?" *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society* (New York, NY, USA), AIES '19, January 27, 2019, 53–59. https://doi.org/10.1145/3306618.3314281.

Chan, Alan, Carson Ezell, Max Kaufmann, et al. "Visibility into AI Agents." *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency* (New York, NY, USA), FAccT '24, June 5, 2024, 958–73. https://doi.org/10.1145/3630106.3658948.

Chen, Yanda, Joe Benton, Ansh Radhakrishnan, et al. "Reasoning Models Don't Always Say What They Think." arXiv:2505.05410. Preprint, arXiv, May 8, 2025. https://doi.org/10.48550/arXiv.2505.05410.

Cheruvu, Ria. "Human-in/on-the-Loop Design for Human Controllability." *In Handbook of Human-Centered Artificial Intelligence*. Springer, Singapore, 2026. https://doi.org/10.1007/978-981-97-8440-0_75-1.

Clarke, Adele E. "Anticipation Work: Abduction, Simplification, Hope." *In Boundary Objects and Beyond: Working with Leigh Star*, edited by Geoffrey C. Bowker, Stefan Timmermans, Adele E. Clarke, and Ellen Balka. 2016. https://doi.org/10.7551/mitpress/10113.003.0007.

Cranor, Lorrie Faith. "A Framework for Reasoning about the Human in the Loop." *Proceedings of the 1st Conference on Usability, Psychology, and Security* (USA), UPSEC'08, April 14, 2008, 1–15.

Crootof, Rebecca, Margot E. Kaminski, and W. Nicholson Price II. "Humans in the Loop." SSRN Scholarly Paper No. 4066781. Social Science Research Network, March 25, 2022. https://doi.org/10.2139/ssrn.4066781.

Dhanorkar, Shipi, Samir Passi, and Mihaela Vorvoreanu. "Human Oversight of Agentic Systems in Practice: Examining the Oversight Work, Challenges, and Heuristics of Developers Using Software Agents." *Proceedings of the 2026 ACM Conference on Fairness, Accountability, and Transparency* (New York, NY, USA), FAccT '26, 2026. https://doi.org/10.1145/3805689.3812402.

Dong, Liming, Qinghua Lu, and Liming Zhu. "AgentOps: Enabling Observability of LLM Agents." arXiv:2411.05285. Preprint, arXiv, November 30, 2024. https://doi.org/10.48550/arXiv.2411.05285.

Dorsey, Jack, and Roelof Botha. "From Hierarchy to Intelligence." Block, March 31, 2026. https://block.xyz/inside/from-hierarchy-to-intelligence.

Ehsan, Upol, Q. Vera Liao, Samir Passi, Mark O. Riedl, and Hal Daumé. "Seamful XAI: Operationalizing Seamful Design in Explainable AI." *Proc. ACM Hum.-Comput. Interact.* 8, no. CSCW1 (2024): 119:1-119:29. https://doi.org/10.1145/3637396.

Elish, Madeleine Clare. "Moral Crumple Zones: Cautionary Tales in Human-Robot Interaction." *Engaging Science, Technology, and Society* 5 (March 2019): 0. NA. https://doi.org/10.17351/ests2019.260.

Enqvist, Lena. "'Human Oversight' in the EU Artificial Intelligence Act: What, When and by Whom?" *Law, Innovation and Technology* 15, no. 2 (2023): 508–35. https://doi.org/10.1080/17579961.2023.2245683.

Epperson, Will, Gagan Bansal, Victor C. Dibia, et al. "Interactive Debugging and Steering of Multi-Agent AI Systems." *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (New York, NY, USA), CHI '25, April 25, 2025, 1–15. https://doi.org/10.1145/3706598.3713581.

("EU AI Act") Artificial Intelligence Act (TA-9-2024-0138), P9_TA(2024)0138. Accessed May 16, 2024. https://www.europarl.europa.eu/doceo/document/TA-9-2024-0138_EN.pdf.

("EU AI Act") Artificial Intelligence Act (TA-9-2024-0138), P9_TA(2024)0138. Accessed May 16, 2024. https://www.europarl.europa.eu/doceo/document/TA-9-2024-0138_EN.pdf.

Feng, Kevin, David McDonald, and Amy Zhang. *Levels of Autonomy for AI Agents*. Knight First Amendment Institute at Columbia University, 2025. https://knightcolumbia.org/content/levels-of-autonomy-for-ai-agents-1.

Fournier, Fabiana, Lior Limonad, and Yuval David. "Agentic AI Process Observability: Discovering Behavioral Variability." arXiv:2505.20127. Preprint, arXiv, July 3, 2025. https://doi.org/10.48550/arXiv.2505.20127.

Ganguli, Deep, Danny Hernandez, Liane Lovitt, et al. "Predictability and Surprise in Large Generative Models." *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency* (New York, NY, USA), FAccT '22, June 20, 2022, 1747–64. https://doi.org/10.1145/3531146.3533229.

Goldman, Paula, and Kathy Baxter. "Generative AI: 5 Guidelines for Responsible Development." Salesforce, February 7, 2023. https://www.salesforce.com/news/stories/generative-ai-guidelines/.

Green, Ben, and Yiling Chen. "The Principles and Limits of Algorithm-in-the-Loop Decision Making." *Proc. ACM Hum.-Comput*. Interact. 3, no. CSCW (2019): 50:1-50:24. https://doi.org/10.1145/3359152.

Henzinger, Thomas, Mahyar Karimi, Konstantin Kueffner, and Kaushik Mallik. "Runtime Monitoring of Dynamic Fairness Properties." *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency* (New York, NY, USA), FAccT '23, June 12, 2023, 604–14. https://doi.org/10.1145/3593013.3594028.

Hollnagel, Erik. *The ETTO Principle: Efficiency-Thoroughness Trade-off : Why Things That Go Right Sometimes Go Wrong*. Ashgate Publishing, Ltd., 2009.

Holzinger, Andreas, Kurt Zatloukal, and Heimo Müller. "Is Human Oversight to AI Systems Still Possible?" *New Biotechnology* 85 (March 2025): 59–62. https://doi.org/10.1016/j.nbt.2024.12.003.

Hou, Xinyi, Yanjie Zhao, Shenao Wang, and Haoyu Wang. "Model Context Protocol (MCP): Landscape, Security Threats, and Future Research Directions." arXiv:2503.23278. Preprint, arXiv, October 7, 2025. https://doi.org/10.48550/arXiv.2503.23278.

Howitt, Susan M., and Anna N. Wilson. "Revisiting 'Is the Scientific Paper a Fraud?'" *The EMBO Reports* 15, no. 5 (2014): 481–84. https://doi.org/10.1002/embr.201338302.

Huang, Kung-Hsiang, Akshara Prabhakar, Sidharth Dhawan, et al. "CRMArena: Understanding the Capacity of LLM Agents to Perform Professional CRM Tasks in Realistic Environments." arXiv:2411.02305. Preprint, arXiv, February 16, 2025. https://doi.org/10.48550/arXiv.2411.02305.

Hughes, Laurie, Yogesh K. Dwivedi, Tegwen Malik, et al. "AI Agents and Agentic Systems: A Multi-Expert Analysis." *Journal of Computer Information Systems* 65, no. 4 (2025): 489–517. https://doi.org/10.1080/08874417.2025.2483832.

Kasirzadeh, Atoosa, and Iason Gabriel. "Characterizing AI Agents for Alignment and Governance." arXiv:2504.21848. Preprint, arXiv, April 30, 2025. https://doi.org/10.48550/arXiv.2504.21848.

Kim, Eugene. "Amazon Orders 90-Day Reset after Code Mishaps Cause Millions of Lost Orders." Business Insider, March 10, 2026. https://www.businessinsider.com/amazon-tightens-code-controls-after-outages-including-one-ai-2026-3.

Kling, Rob. "Social Analyses of Computing: Theoretical Perspectives in Recent Empirical Research." *ACM Comput. Surv.* 12, no. 1 (1980): 61–110. https://doi.org/10.1145/356802.356806.

Koulu, Riikka. "Proceduralizing Control and Discretion: Human Oversight in Artificial Intelligence Policy." *Maastricht Journal of European and Comparative Law* 27, no. 6 (2020): 720–35. https://doi.org/10.1177/1023263X20978649.

Krakovna, Victoria, Jonathan Uesato, Vladimir Mikulik, et al. "Specification Gaming: The Flip Side of AI Ingenuity." Google DeepMind, April 21, 2020. https://deepmind.google/blog/specification-gaming-the-flip-side-of-ai-ingenuity/.

Langer, Markus, Kevin Baum, and Nadine Schlicker. "Effective Human Oversight of AI-Based Systems: A Signal Detection Perspective on the Detection of Inaccurate and Unfair Outputs." *Minds and Machines* 35, no. 1 (2024): 1. https://doi.org/10.1007/s11023-024-09701-0.

Letta Platform. "Agent Development Environment (ADE)." Letta Docs. Accessed April 21, 2026. https://docs.letta.com/guides/ade/overview/.

Li, Xinyi, Sai Wang, Siqi Zeng, Yu Wu, and Yi Yang. "A Survey on LLM-Based Multi-Agent Systems: Workflow, Infrastructure, and Challenges." *Vicinagearth* 1, no. 1 (2024): 9. https://doi.org/10.1007/s44336-024-00009-2.

Liao, Q. Vera, and Jennifer Wortman Vaughan. "AI Transparency in the Age of LLMs: A Human-Centered Research Roadmap." *Harvard Data Science Review*, no. Special Issue 5: Grappling With the Generative AI Revolution (May 2024). https://doi.org/10.1162/99608f92.8036d03b.

Liu, Guangya, and Sujay Solomon. "AI Agent Observability - Evolving Standards and Best Practices." OpenTelemetry, March 6, 2025. https://opentelemetry.io/blog/2025/ai-agent-observability/.

Lu, Chris, Cong Lu, Robert Tjarko Lange, et al. "Towards End-to-End Automation of AI Research." *Nature* 651, no. 8107 (2026): 914–19. https://doi.org/10.1038/s41586-026-10265-5.

Lu, Chris, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. "The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery." arXiv:2408.06292. Preprint, arXiv, September 1, 2024. https://doi.org/10.48550/arXiv.2408.06292.

Manuvinakurike, Ramesh, Emanuel Moss, Elizabeth Anne Watkins, Saurav Sahay, Giuseppe Raffa, and Lama Nachman. "Thoughts without Thinking: Reconsidering the Explanatory Value of Chain-of-Thought Reasoning in LLMs through Agentic Pipelines." arXiv:2505.00875. Preprint, arXiv, May 1, 2025. https://doi.org/10.48550/arXiv.2505.00875.

Medawar, Peter. "Is the Scientific Paper a Fraud?" In *The Strange Case of the Spotted Mice and Other Classic Essays on Science*, edited by Peter Medawar. Oxford University Press, 1996. https://doi.org/10.1093/oso/9780192861931.003.0003.

Mitchell, Margaret, Avijit Ghosh, Alexandra Sasha Luccioni, and Giada Pistilli. "Fully Autonomous AI Agents Should Not Be Developed." arXiv:2502.02649. Preprint, arXiv, October 20, 2025. https://doi.org/10.48550/arXiv.2502.02649.

Miyazaki, Hirokazu, and Annelise Riles. “Failure as an Endpoint.” In *Global Assemblages: Technology, Politics, and Ethics as Anthropological Problems*, edited by Aihwa Ong and Stephen J. Collier. Blackwell Publishing, 2005.

Model Context Protocol. “What Is the Model Context Protocol (MCP)?” Accessed April 23, 2026. https://modelcontextprotocol.io/docs/getting-started/intro.

Moshkovich, Dany, Hadar Mulian, Sergey Zeltyn, Natti Eder, Inna Skarbovsky, and Roy Abitbol. “Beyond Black-Box Benchmarking: Observability, Analytics, and Optimization of Agentic Systems.” arXiv:2503.06745. Preprint, arXiv, March 9, 2025. https://doi.org/10.48550/arXiv.2503.06745.

Naik, Suchismita, Amanda Snellinger, Austin L. Toombs, Scott Saponas, and Amanda K. Hall. “Exploring Early Adopters’ Use of AI Driven Multi-Agent Systems to Inform Human-Agent Interaction Design: Insights from Industry Practice.” *Proceedings of the Extended Abstracts of the CHI Conference on Human Factors in Computing Systems* (New York, NY, USA), CHI EA ’25, April 25, 2025, 1–8. https://doi.org/10.1145/3706599.3706693.

Ohde, Joshua W., Lauren M. Rost, and Joshua D. Overgaard. “The Burden of Reviewing LLM-Generated Content.” *NEJM AI* 2, no. 2 (2025): 1–3. https://doi.org/10.1056/AIp2400979.

OpenAI. “Introducing ChatGPT Agent: Bridging Research and Action.” OpenAI, July 17, 2025. https://openai.com/index/introducing-chatgpt-agent/.

Passi, Samir. “Agentic AI Has a Human Oversight Problem.” SSRN Scholarly Paper No. 5529058. Social Science Research Network, September 15, 2025. https://doi.org/10.2139/ssrn.5529058.

Passi, Samir, Shipi Dhanorkar, and Mihaela Vorvoreanu. “Addressing Overreliance on AI.” In *Handbook of Human-Centered Artificial Intelligence*. Springer, Singapore, 2025. https://doi.org/10.1007/978-981-97-8440-0_98-1.

Passi, Samir, and Steven J. Jackson. “Trust in Data Science: Collaboration, Translation, and Accountability in Corporate Data Science Projects.” *Proc. ACM Hum.-Comput. Interact.* 2, no. CSCW (2018): 136:1-136:28. https://doi.org/10.1145/3274405.

Polanyi, Michael. *The Tacit Dimension*. Anchor Books, 1967.

Reed, Scott, Konrad Zolna, Emilio Parisotto, et al. “A Generalist Agent.” arXiv:2205.06175. Preprint, arXiv, November 11, 2022. https://doi.org/10.48550/arXiv.2205.06175.

Santoni de Sio, Filippo, and Giulio Mecacci. “Four Responsibility Gaps with Artificial Intelligence: Why They Matter and How to Address Them.” *Philosophy & Technology* 34, no. 4 (2021): 1057–84. https://doi.org/10.1007/s13347-021-00450-x.

Shapin, Steven. “Cordelia’s Love: Credibility and the Social Studies of Science.” *Perspectives on Science* 3, no. 3 (1995): 255–75. https://doi.org/10.1162/posc_a_00484.

Shapin, Steven, and Simon Schaffer. *Leviathan and the Air-Pump: Hobbes, Boyle, and the Experimental Life*. Princeton University Press, 2018.

Shavit, Yonadav, Sandhini Agarwal, Miles Brundage, et al. *Practices for Governing Agentic AI Systems*. Technical Report. OpenAI, 2023.

Shojaee, Parshin, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. "The Illusion of Thinking: Understanding the Strengths and Limitations of Reasoning Models via the Lens of Problem Complexity." arXiv:2506.06941. Preprint, arXiv, November 20, 2025. https://doi.org/10.48550/arXiv.2506.06941.

Simon, Herbert A. "Theories of Bounded Rationality." In *Decision and Organization: A Volume in Honor of Jacob Marschak*, edited by Jacob Marschak, C. B. McGuire, Roy Radner, and Kenneth Joseph Arrow. North-Holland Pub. Co., 1972. https://cir.nii.ac.jp/crid/1572261549641088896?lang=en.

Steinhardt, Stephanie B., and Steven J. Jackson. "Anticipation Work: Cultivating Vision in Collective Practice." *Proceedings of the 18th ACM Conference on Computer Supported Cooperative Work & Social Computing* (New York, NY, USA), CSCW '15, February 28, 2015, 443–53. https://doi.org/10.1145/2675133.2675298.

Sterz, Sarah, Kevin Baum, Sebastian Biewer, et al. "On the Quest for Effectiveness in Human Oversight: Interdisciplinary Perspectives." *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency* (New York, NY, USA), FAccT '24, June 5, 2024, 2495–507. https://doi.org/10.1145/3630106.3659051.

Suchman, Lucy. *Human-Machine Reconfigurations: Plans and Situated Actions*. 2 edition. Cambridge University Press, 2006.

Treasury Board of Canada Secretariat. "Guide on the Use of Generative Artificial Intelligence." Government of Canada, May 30, 2024. https://www.canada.ca/en/government/system/digital-government/digital-government-innovations/responsible-use-ai/guide-use-generative-ai.html.

Verhagen, Ruben S., Mark A. Neerincx, and Myrthe L. Tielman. "Agent Allocation of Moral Decisions in Human-Agent Teams: Raise Human Involvement and Explain Potential Consequences." *Proceedings of the 2025 ACM Conference on Fairness, Accountability, and Transparency* (New York, NY, USA), FAccT '25, June 23, 2025, 2302–17. https://doi.org/10.1145/3715275.3732157.

Wang, Xingyao, Boxuan Li, Yufan Song, et al. "OpenHands: An Open Platform for AI Software Developers as Generalist Agents." arXiv:2407.16741. Preprint, arXiv, April 18, 2025. https://doi.org/10.48550/arXiv.2407.16741.

Weick, Karl E. *Sensemaking in Organizations*. SAGE, 1995.

Wenger, Etienne. *Communities of Practice: Learning, Meaning, and Identity*. Cambridge University Press, 1999.

Data & Society is an independent nonprofit research and policy institute, studying the social implications of data-centric technologies and automation. We recognize that the same innovative technologies that may benefit society can also be abused to invade privacy, provide new tools of discrimination, foreclose opportunity and harm individuals and communities. We believe that technology policy must be grounded in empirical evidence, and serve the public.

DATA&
SOCIETY

LAYOUT BY GLORIA MENDOZA

JULY 2026

DATASOCIETY.NET @DATASOCIETY