PARTNERSHIP ON AI

Prioritizing Real-Time Failure Detection in AI Agents

Madhulika Srikumar
Jacob Pratt
Kasia Chmielinski
Partnership on AI

Carolyn Ashurst
Alan Turing Institute

Chloé Bakalar
OpenAI

William Bartholomew
Microsoft

Rishi Bommasani
Stanford Institute for Human-Centered Artificial Intelligence

Peter Cihon
GitHub

Rebecca Crootof
University of Richmond School of Law

Mia Hoffmann
Center for Security and Emerging Technology

Ruchika Joshi
Center for Democracy and Technology

Maarten Sap
Carnegie Mellon University and Allen Institute for AI

Caleb Withers
Center for a New American Security

Contents

Introduction **3**

Figure 1. Levels of agent influence on digital environments, with examples of LLM-based systems. *6*

1. Agents require new forms of failure detection due to their ability to effect change in the environment **7**

Agents introduce new, diverse, and compounding failure modes that emerge during operation, extending risks beyond those seen in generative AI systems 7

Human oversight becomes significantly harder during real-time agent actions due to speed and scale 10

Current evaluations remain brittle and often overly focused on limited contexts rather than the complex, multi-step behaviors agents display once deployed 11

2. The risk of agent failures — and the necessity of real-time detection — depends on the stakes of actions, their reversibility, and the agent's architectural affordances **12**

Failure detection is not a single function but a layered set of controls distributed across the agent workflow 12

Figure 2. Layered failure detection controls across the agent workflow *13*

Failure detection efforts should be calibrated to the stakes of the use case or task, the reversibility of potential failures, and the agent's architectural affordances 16

Figure 3. Calibrating failure detection by stakes, reversibility, and agent affordances *16*

3. Safety-critical industries show failure detection can reduce harms and provide a foundation for safer agent design **24**

Higher-risk functions in road vehicles require stronger failure detection controls. This principle can inform how we assess and manage action-level risk in AI agents 24

Backups can support fail-safe operation but require detection to function properly 25

4. Significant technical research and regulatory guidance must be prioritized to close gaps in designing and evaluating failure detection for AI agents **26**

Conclusion and Limitations **30**

Bibliography **31**

Authorship of this paper does not imply endorsement of all claims in the paper, nor endorsement by any affiliated organization.

Introduction

When AI agents take actions, they introduce new risks. While generative AI systems produce content for humans to act on, agents — built on the same foundation models with added scaffolding — reason, plan, and perform sequences of actions to achieve user goals. Unlike generative AI, these systems directly execute actions by using digital tools[1] to interact with complex environments.[2] We are already seeing prototypes of agents that can schedule meetings through a calendar API or book flights via web interfaces. More ambitious proposals include agents that negotiate contracts, assist in healthcare decisions, and coordinate supply chains. Because agents can act directly in the environment, failures to meet user goals can result in financial loss, safety risks, or breakdowns in critical processes. Such failures to achieve user goals can occur at any stage of action-taking — from planning and tool selection to execution— and often arise in ways that are difficult to predict or catch in advance. While design choices and deployment context shape when and how failures occur, it is the agent's real-time actions that directly cause incidents. These operational stages are therefore critical points for monitoring and intervention.

Human oversight, long called for in AI governance, cannot scale to monitor fast or opaque agent behaviors. While human oversight is essential for accountability, the very design logic of agents — reducing human involvement — makes constant supervision impractical.[3] Crucially, escalating certain decisions to humans must be meaningful rather than merely shifting liability. We must therefore find ways to minimize harmful outcomes automatically and ensure human oversight is triggered when agent actions carry high stakes, are irreversible, or depend on advanced system capabilities.[4] Achieving this requires what we call *real-time failure detection*: automated monitoring systems that track agent behavior, flag anomalies, and either stop agents immediately or escalate to human oversight when needed. In this paper, we use "failure detection" broadly to encompass both automated monitoring and controls for invoking meaningful human oversight.

We need greater investment in understanding the trade-offs and limitations of automated, real-time failure detection. Companies like Anthropic, OpenAI, and Meta are beginning to explore secondary models that watch for signs of agent hijacking or monitor reasoning to track tool calls.[5] Yet cost trade-offs remain a challenge. Agent deployment itself can be resource-in-

All cited sources can be found in the Bibliography on page 31.

1 While tool integration enables real-world impact, LLM-based agents rely on the foundation model's reasoning to draw up plans and handle complex goals. As agents are given access to external memory systems, likely implemented through external databases, agents can gain the ability to retain and recall information across sessions, supporting more continuous and adaptive behavior. (See Weng, 2023; NIST (a), 2025.)

2 We consider environmental interaction as the minimum threshold for agents that warrant failure detection. Other frameworks emphasize additional properties (see Shavit et al., 2023; Chan (a) et al., 2023; Kapoor et al., 2024). These properties include their ability to perceive their environment (Russell & Norvig, 2010) and attributes like autonomy, goal complexity, and generality (Kasirzadeh & Gabriel, 2025). These factors, together, influence the risks posed by agents and the need for real-time monitoring.

3 Milmo, 2024; Shibu, 2025; Benioff, 2025.

4 Failure detection should not trigger just because the agent tries a path that does not work out immediately. Exploration, iteration, or partial progress is often part of intelligent behavior. Instead, detection should focus on meaningful failures, e.g., when the agent does something irrecoverable, dangerous, nonsensical, or outside its intended bounds, or is stuck.

5 Anthropic (a), 2025; OpenAI (a), 2025; Chennabasappa, 2025.

tensive, and failure detection, particularly when relying on advanced models for oversight, may carry similar costs.[6] In addition, poorly calibrated monitoring can generate excessive false positives or miss critical failures, overloading human operators or undermining trust in the system. This raises important questions about how well these approaches can scale, and whether financial, legal, or reputational incentives support their broader adoption.

In this paper, we outline a framework for evaluating conditions under which real-time failure detection should be prioritized in AI agents. Our framework focuses on what we define as Levels 3-5 of environmental interaction in Figure 1 (see below), where systems execute direct actions with minimal human oversight.[7] As these systems advance, "constrained" agents with limits on actions will move from prototypes to widespread deployment, paving the way for more adaptive, unconstrained agents with even greater potential risks.[8]

Our analysis is built around these central claims:

1. Agents require new forms of failure detection due to their ability to effect change in the environment.
2. The risk of agent failures — and the necessity of real-time detection — depends on the stakes of actions, their reversibility, and the agent's architectural affordances.
3. Safety-critical industries show failure detection can reduce harms and provide a foundation for safer agent design.
4. Significant technical research and regulatory guidance must be prioritized to close gaps in designing and evaluating failure detection for AI agents.

This paper explores why and when real-time failure detection matters, but stops short of prescriptive guidance on how to implement it. Implementation details should be determined by agent developers and deployers who understand specific architectural constraints and deployment contexts. Real-time detection can help address certain runtime failures such as breakdowns in planning, execution, or tool use, but it will not in isolation solve all problems, including misuse, deceptive behaviors by agents, or risks emerging

6 Anthropic notes that deploying agent-based systems typically consume significantly more tokens and resource usage than single-turn models, citing that multi-agent setups can use 4×–15× more compute relative to standard chat workflows (Anthropic (b), 2025). See also Terekhov et al., 2025.

7 We build on vocabularies like Kasirzadeh & Gabriel, 2025, to describe agent capabilities rather than impose a rigid definition of "agent." Our focus is on LLM-based agents acting in digital environments, which can indirectly affect the physical world, for example, by ordering food or booking rides, and while such systems may eventually gain more direct physical influence through robotics, that is beyond our scope.

8 While agents are advancing rapidly, current systems remain unreliable for complex, multi-step tasks and cannot yet perform even relatively simple jobs like executive assistance without human oversight (METR, 2025; Bengio et al., 2024). Despite growing interest, their real-world impact is still limited.

from multi-agent interactions, which may require additional approaches.[9] Although other design and deployment safeguards remain essential, this paper emphasizes monitoring during operation because agents raise distinct challenges compared to generative AI. While a complete analysis of every type of failure or scenario in which agents might go wrong is beyond the scope of this paper, we believe now is the right time to kickstart a conversation on architectural norms for real-time monitoring of agents. This paper aims to equip developers, deployers, and policymakers with a clearer understanding of why and when failure detection is necessary. As agent architectures rapidly evolve, now is a critical moment to establish norms for built-in safety mitigations.[10]

9 On misuse, refer to footnote 13 for why deliberate user-driven harms fall outside scope. Multi-agent risks are also out of scope, as our focus is on failures within individual agents. Many detection approaches here apply to both AI *agents* (single systems with tool access) and *agentic* AI (multi-agent systems with specialized roles, as discussed in Ranjan Sapkota et al., 2025), but not to failures that emerge specifically from interactions between multiple agents—coordination, collusion or conflict, which require separate study (see Hammond, 2025).

10 References to "agents" and their actions in this paper use anthropomorphized language to keep descriptions concise and readable. This is not meant to imply human-like qualities or capabilities or to blur the distinction between software systems and humans.

Figure 1. Levels of agent influence on digital environments, with examples of LLM-based systems.
Current systems operate at Levels 1–3, while Levels 4–5 illustrate emerging directions for more autonomous agents.

	LEVEL	ABILITY TO INFLUENCE THE ENVIRONMENT	EXAMPLE AI SYSTEMS
	0	Read-only, observation only	Speech recognition systems Image classification models
MEDIATED INFLUENCE	1	**Mediated influence via humans:** System outputs text suggestions or advice; output only affects the environment if a human acts	ChatGPT without tools
MEDIATED INFLUENCE	2	**Mediated influence via humans with passive tools:** System uses tools like search or knowledge databases for context, not for changing the world itself	ChatGPT or Claude with web search *(provides info but doesn't act)* Gemini Deep Research *(provides research analysis)* GitHub Copilot *(provides code suggestions)*
DIRECT ACTIONS	3	**Direct Actions — Constrained Agent:** System performs single-step actions directly using predefined tools, acting on user commands without needing humans to carry out the result	Operator Claude Computer Use Manus Project Mariner Cursor AI Other code-executing or API-calling agents
DIRECT ACTIONS	4	**Direct Actions — Semi-constrained Agent:** System accepts broad goals, decomposes them into multi-step plans, and executes steps autonomously across known tools, *without needing humans to approve every step.*	Longer and more complex workflows. E.g. User says "File my taxes" → the agent gathers necessary documents from user's email, fills tax forms, and submits them using chosen e-filing service, without asking for approvals on each step.
DIRECT ACTIONS	5	**Direct Actions — Unconstrained Agent:** System accepts a broad goal, autonomously executes multi-step plans, and adapts by integrating new tools or strategies beyond what a user configured, all without requiring approvals.	Adaptive workflows. E.g. User says "File taxes for my business" → the agent autonomously gathers financial records, contacts suppliers for missing information, interprets regulations, applies business-specific deductions, switches between tools or services as needed, and completes the filing with without further human input.

We assess levels of capability based on environmental interaction, but an agent's ability to interact with its environment also depends on other differentiating properties: whether it relies on predefined tools or can flexibly adapt tool use, how much human oversight it requires, and the complexity of goals it can pursue. These properties vary by degree and collectively shape an agent's ability to interact with its environment. Our levels share similarities with recent surveys of agent autonomy, which highlight dimensions such as constraints on environmental impact and flexibility of actions (Cihon et al., 2025). Although some systems might meet our Level 3 criteria through rule-based execution (e.g., spam bots), our analysis focuses on LLM-driven agents using reasoning, planning, or decision-making, as these introduce new sources of unpredictability, runtime failure, or hazards. Levels 4 and 5 draw inspiration from Patel, 2025.

1. Agents require new forms of failure detection due to their ability to effect change in the environment

Agents introduce new, diverse, and compounding failure modes that emerge during operation, extending risks beyond those seen in generative AI systems.

We define failure modes as agent behaviors or events that can cause or contribute to hazards or real-world incidents.[11] The Organization for Economic Co-operation and Development (OECD) defines AI incidents as events where AI systems cause harm to people, disrupt infrastructure, or violate human rights.[12] AI hazards are defined as precursors to incidents — events that could plausibly lead to such harm. This paper focuses on catching failures[13] that pose hazards, using real-time detection to prevent escalation into incidents. Put another way, a failure mode is how something goes wrong, a hazard is the risky condition it creates, and an incident is when actual harm occurs. For example, an indirect prompt injection would be the failure mode, and the agent making a fraudulent purchase as a result would be the incident.

Agents inherit the unpredictability and reliability issues of foundation models, but those errors now manifest through agentic actions. These include agents potentially manipulating files, impersonating users, or making unauthorized transactions.[14] For example, hallucinations in the underlying model can result in incorrect or harmful function calls (calls to tools to execute actions).[15] Issues in training data, such as non-representative data, can lead agents to take discriminatory actions, systematically disadvantaging certain groups. If sensitive information is present in training data, agents may also expose private user data through their actions.

In addition to problems inherited from the foundation model, agents introduce new failure modes by acting autonomously across multiple steps. In generative AI systems, mistakes mostly happen at the single-shot output moment. By contrast, agents operate directly through sequences of actions. An agent may select a tool, plan steps, execute each

11 Here, "agents" refers to systems deployed in consumer and enterprise contexts that can pursue user goals, such as customer-facing assistants, enterprise automation tools, and decision-making systems that manage transactions or allocate resources.

12 The OECD defines an AI incident as an event, circumstance, or series of events where the development, use, or malfunction of one or more AI systems directly or indirectly leads to any of the following: (a) injury or harm to the health of a person or group; (b) disruption of critical infrastructure; (c) violations of human rights or breaches of applicable law protecting fundamental, labor, or intellectual property rights; or (d) harm to property, communities, or the environment. An AI hazard is defined as an event that could plausibly lead to an AI incident (OECD.AI, 2024).

13 This paper focuses on failures that emerge during normal agent operation, including agent hijacking— where a system is indirectly compromised via malicious inputs in external tools or data. These are in scope for real-time detection because the agent is subverted mid-task without user intent. By contrast, direct misuse, where a bad actor deliberately prompts the agent to carry out harmful tasks from the start, is out of scope. In the Lovable AI "VibeScamming" case, attackers used multi-step prompting to generate phishing content. These cases do not reflect internal failure but rather a system working as instructed (Lakshmanan, 2025). They require complementary safeguards such as usage policies and external defenses like email scanners or URL blacklists, not runtime detection. (Narayanan & Kapoor, 2025).

14 Mitchell, 2025.

15 IBM AI Ethics Board, 2025.

step, and revise its plan mid-run. Failures at any of these stages can unpredictably shift the agent's course, with errors compounding as the process unfolds.[16] For example, an agent tasked with managing a company's travel bookings might misinterpret constraints, cancel key reservations, and notify stakeholders of changes, resulting in financial loss, missed obligations, or reputational damage. Failures can result in unauthorized actions, irreversible changes, or further real-world harm.

Operational failures during planning, tool use, and execution are likely to be the most proximate contributors to real-world AI incidents. While some failures originate from system design or training flaws, they often manifest and escalate during operation, through flawed planning, tool misuse, or unexpected outcomes during execution. These failures are not just technical malfunctions but key contributors to real-world incidents. As a result, these stages — planning, tool use, and execution — should be a primary focus of interventions addressing agent failures.[17] The examples below illustrate where these failures tend to emerge and why real-time monitoring must focus on these stages.[18]

16 Bengio et al., 2024.

17 These effects of agent failure modes, such as incorrect plans, exposing sensitive data, or acting outside intended environments, contribute to hazards as defined by the OECD and may result in incidents when they lead to legal, physical, or societal harm. Other outcomes, like trust erosion or overreliance, don't create hazards on their own but undermine how the human and agent work together, and increase susceptibility to future incidents. While these harms matter, they cannot be mitigated through real-time failure detection alone. Complementary strategies are essential, including explicit disclosure of AI status and calibrating user expectations through education as argued by Akbulut et al., 2024. Real-time monitoring for instance can help enable clearer task boundaries and escalation to human professionals in crisis situations.

18 Multiple taxonomies classify AI and agent failures differently. One surveys real-world cases of AI system breakdowns to highlight recurring functionality issues (Raji et al.,2022). Another maps security, privacy, and ethics threats across agent components and lifecycle stages (Gan et al., 2025). The OWASP Agentic Security Initiative provides a developer-oriented threat modeling framework (OWASP, 2025). Microsoft offers practical failure mode categories for industry assessments (Microsoft (a), 2025). This paper instead focuses on failures emerging during live agent operation, specifically in planning, tool use, and execution, that can escalate into hazards if not detected and mitigated in real time.

EXAMPLES OF FAILURE MODES

Below are illustrative, non-exhaustive examples of failures during planning, tool use, and execution,[19] which we expect to change as agent architectures and deployment contexts evolve.

PLANNING FAILURES	• Plan inconsistent with user intent[20] • Misprioritizing between competing goals[21] • Insufficient planning for a complex goal[22] • Selecting the wrong tool to complete a step in the plan[23] • Plan exceeds tool permissions or other constraints • Plan conflicts with the current environment (e.g., the environment changed and the original plan is no longer valid)
TOOL-USE FAILURES	• Misusing the tool (e.g., executing a search query in the wrong format)[24] • Tools are vulnerable to attacks (e.g., prompt injection vulnerabilities in third-party websites)[25] • Tools fail or cause unintended side effects • Tool accesses resources beyond task needs (e.g., user systems or external services)
EXECUTION FAILURES	• Taking actions inconsistent with plan • Mishandling unsafe tool outputs (e.g., exposing sensitive private user data retrieved through a tool) • Exhausting operational constraints, such as inference token limits[26] • Executing actions beyond authorized boundaries[27]

19 The distinction between planning, tool use, and execution may be ambiguous in practice, as the stages may not always be distinctly observed.

20 Agents can struggle to infer the appropriate scope or granularity of user intent. In one case, Anthropic found that agents over- or under-allocated subagents and resources when no explicit guidelines were established (Anthropic (c), 2025). Microsoft similarly observed an agent, tasked with "getting rid" of a user record, delete the entire table instead, due to misinterpreting the user's shorthand instruction (Microsoft, 2024).

21 When we refer to goals (or "user goals") it is often a combination of the user's goals and constraints, the deployer's goals and constraints, and the agent's goals and capabilities. These intersect like a Venn diagram, shaping how the system behaves in practice. For instance, if a pharmaceutical company instructs an AI assistant to maximize sales of a painkiller, the system might downplay or omit the risk of addiction, aligning with the company's commercial goal but conflicting with the patient's interests (Su et al., 2025). See also Wallace et al., 2024. The authors found that a primary vulnerability in LLMs is their inability to distinguish between instructions of different privilege levels, treating system prompts from developers the same as text from untrusted users and third parties, enabling adversaries to override higher-level instructions with malicious prompts.

22 Agents may fail to solve complex problems when they exhaust token limits, a challenge some companies address by designing architectures that distribute reasoning across multiple agents (Anthropic (c), 2025).

23 Agents often struggle to choose the right tool when interfaces are vague or overlap. Without clear descriptions or examples of how and when to use each tool, agents misuse them or fail to match tools to the user's intent (Anthropic (c), 2025). Zhou et al. empirically show that tool-use capabilities are central to agent safety, finding higher failure rates in scenarios when agents select or operate tools poorly (Zhou et al., 2024).

24 OpenAI observed that agents often try to visually read values like API keys or Bitcoin wallet addresses from screens instead of copying them, leading to OCR mistakes (OpenAI (a), 2025).

25 During testing of Operator, researchers found that the agent was often misled by malicious instructions in third-party websites, a form of prompt injection that caused it to act against user intent (OpenAI (a), 2025).

26 Kapoor et al., 2024, show that sufficient inference compute is often necessary for agents to perform complex tasks effectively.

27 During testing of Operator, the system was restricted from visiting certain domains to reduce the risk of harmful or unauthorized behavior (OpenAI (a), 2025).

In addition to system design and training data choices, architectural affordances like autonomy, memory, and flexible tool use increase the chance that failures during operation escalate into AI incidents. As agents move from Level 3 (constrained) towards Level 5 (unconstrained), they shift from executing single-step tasks to decomposing goals, running longer workflows, and selecting tools or strategies beyond what users originally configured. These capabilities may increase the likelihood that small failures persist, compound across steps, and evolve into more serious outcomes.[28] The severity of those outcomes will depend on contextual factors such as the stakes or irreversibility of an action, which we explore in the next section.[29] While existing best practices for managing risk at the pre-deployment stage or within specific deployment contexts still apply, this paper focuses on operational failures that are unique to agents and often need to be addressed in real time.

Human oversight becomes significantly harder during real-time agent actions due to speed and scale.

Requiring human users or operators to review and approve AI outputs is widely seen as a safeguard.[30] To the extent it is effective, however, it becomes far less reliable as agents operate at scale. Users can find it increasingly difficult to sustain attention as agents take on longer, more complex workflows, such as the Level 4 systems we describe. Studies show that human oversight can break down in two ways. Users or operators have been shown to both over-rely and place excessive trust in AI recommendations ("automation bias") or under-rely, rejecting AI recommendations without justification ("algorithm aversion"), and overall struggle to judge the accuracy of AI predictions.[31] The idea of "alert fatigue" where repeated notifications wear down user attention, is well documented in fields from healthcare to autonomous vehicle systems.[32] It can lead to rushed reviews and possibly over time a "skill fade," where human operators lose the ability to intervene effectively as they grow dependent on automation.[33] Even if humans could theoretically oversee agents, such review would create bottlenecks that undermine the speed, scale, and cost-effectiveness benefits that drive agent

28 At the same time as capabilities advance, agents may be able to better self-critique and recover from errors (Anthropic (b), 2025).

29 Beyond failures rooted in the agent's design and behavior discussed here, and separate from the contextual factors that affect severity, how humans and AI systems work together also creates distinct failure modes. Tesla's Autopilot system created a unique failure mode by transferring control to human drivers less than one second before impact in sixteen known instances — a problem that exists neither in purely human-driven cars nor fully autonomous vehicles (Crootof, Kaminski & Price, 2023).

30 Examples of oversight requirements include Article 14 of the EU AI Act, which mandates that high-risk AI systems be "effectively overseen by natural persons," with obligations that individuals "fully understand the capacities and limitations" of the system, "remain aware of automation bias," and "be able to correctly interpret" its outputs. Critics warn this may paradoxically overload human overseers or set them up for blame should systems fail (Green, 2022). Other examples of human oversight requirements include the UN Convention on Certain Conventional Weapons discussions on "meaningful human control" for autonomous weapons since 2013, and the General Data Protection Regulation (GDPR) Article 22, which prohibits "solely automated" decisions with significant effects and establishes "the right to obtain human intervention on the part of the controller."

31 Laux, 2023.

32 Studies show that healthcare workers become desensitized to electronic safety alerts due to overwhelming volume, with healthcare providers encountering more than 100 alarms per patient bed daily, leading to ignored alerts that could indicate critical medical events (Ancker et al., 2017).

33 Macnamara et al., 2024.

adoption in the first place. Human oversight is, as a result, not only technically challenging and practically insufficient but structurally disincentivized under this agent paradigm.

Automated monitoring systems can address many weaknesses of direct human oversight by tracking agent behavior in real time, flagging anomalies, and sometimes intervening (e.g., pausing, halting, or prompting recovery). But this type of monitoring alone is insufficient: some agent actions create risks that no automated system can reliably judge or resolve — especially when stakes are high, outcomes are hard to reverse, or advanced affordances make behavior less predictable (we show examples in the next section). In practice, effective automated monitoring — i.e., real-time failure detection — should act as a triage system: resolving minor issues automatically, escalating ambiguous or severe failures to humans, and halting when neither is safe. Escalation may involve end-users, operators, or dedicated teams within industry or deploying organizations. This way, monitoring supports rather than replaces meaningful human oversight.

Current evaluations remain brittle and often overly focused on limited contexts rather than the complex, multi-step behaviors agents display once deployed.

Today's evaluations for generative AI systems focus primarily on pre-deployment testing, which largely works by assessing model outputs for potential information hazards such as toxic content or dangerous biological knowledge.[34] These evaluations also often include red-teaming or adversarial testing with human subjects to probe multi-turn responses and simulate prompts from malicious users to uncover vulnerabilities.[35] While vital, these evaluations are limited, as their results only cover the contexts tested, models can evolve over time, and unexpected behaviors often surface only after deployment.[36] An emerging approach is to use large language models (LLMs) themselves as judges to evaluate whether an agent completed a task safely or correctly. But, recent work finds that "LLM-as-judges" often miss subtle failures such as scenarios when agents take harmful actions while appearing to follow instructions properly.[37] This highlights the limits of current evaluation practices and the need for additional, real-time failure detection controls that operate during agent workflows.

34 Weidinger et al., 2023.

35 Lujain Ibrahim et al., 2025. For a fuller list of risk mitigation strategies discussed and operationalized by industry at varied levels of maturity, see Partnership on AI, 2023, and Risto Uuk et al., 2024.

36 Pre-deployment evaluations still remain important, including specialized reinforcement learning to help models resist prompt injection attacks and Constitutional AI approaches that embed high-level normative constraints at training time. However, these must be supplemented with automated monitoring for AI agents that can halt action immediately and trigger a human-in-the-loop.

37 Haitao Li et al., 2024; Sanidhya Vijayvargiya et al., 2025.

2. The risk of agent failures — and the necessity of real-time detection — depends on the stakes of actions, their reversibility, and the agent's architectural affordances

This section examines how failure detection can be structured and scaled to match the risks posed by agents. We first explore how detection can be implemented as layered controls across an agent's workflow. We then identify where detection is most needed, focusing on contextual factors such as stakes and reversibility, along with the agent's affordances, to help developers, deployers, and policymakers target their efforts.

Failure detection is not a single function but a layered set of controls distributed across the agent workflow.

Failure detection, as used here, refers to real-time controls that monitor an AI agent's actions as they unfold — before or during execution — to mitigate misaligned, unsafe, or unintended outputs that could lead to harm. Each stage addresses a different category of failure, using a combination of three responses:

1. Stop (halt execution immediately)
2. Escalate (transfer control to a human for judgment)
3. Retry (revise the plan, tools, or steps before resuming)

These responses can also be combined; for example, an agent might halt, alert a human, and retry only after approval.[38] Real-time failure detection differs from post-hoc monitoring, where user logs are analyzed for policy violations after the fact and cannot prevent immediate harms from escalating.[39]

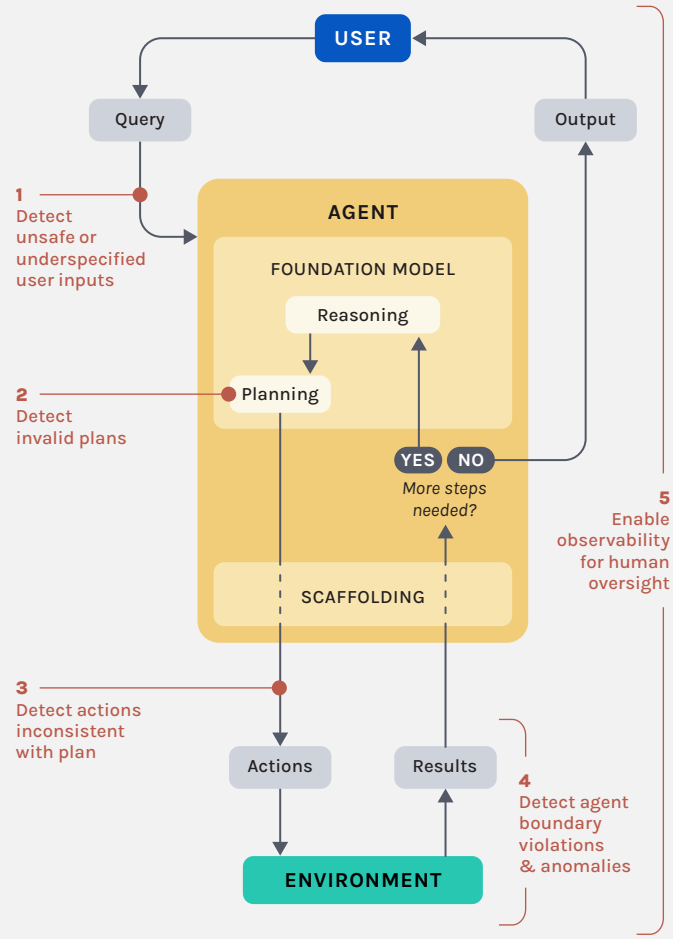
These controls operate at different stages — pre-action, during action, and across steps. Each stage targets a different class of failure, from catching invalid plans before execution to halting unsafe behavior mid-run. To date, most real-time mitigations for generative AI systems rely primarily on input/output content filters and training models

38 Agent developers and deployers can explore other possible responses such as warning the user without halting for low-stakes or reversible errors, or automatically initiating a remedy to undo a failure. Low-stakes scenarios may justify warning-only interventions (as in automotive systems where failures trigger dashboard alerts but don't disable driving). "Remedy" responses can also be built in e.g., systems reversing a mistaken purchase before handing back control.

39 Scholars have increasingly recognized the need for real-time monitoring of AI agents given their autonomous action capabilities. Shavit et al., 2023, emphasize that "automatic monitoring" should flag problematic behavior as it occurs, noting that "monitoring can be provided as a service by the system deployer, or set up by the user" and may require "a second 'monitoring' AI system that automatically reviews the primary agentic system's reasoning and actions." Chan (b) et al., 2024, distinguish between real-time monitoring, which involves "real-time analysis of an agent's activity, allowing deployers and/or service or tool providers to flag and intervene on problematic behaviour as it is occurring," and activity logging for "post-incident attribution and forensics".

to “refuse” harmful requests by declining to provide dangerous information.[40] While many of these mitigations now operate over multi-turn conversations, they are still designed for content moderation in dialog, not for supervising agents acting across tools and executing extended workflows.[41] To address this gap, emerging practices point toward layered, real-time monitoring distributed across the agent workflow, combining pre-action, during-action, and multi-step controls, with observability layers throughout allowing users to intervene.[42] These layers target distinct classes of failures and rely on both automated and human-in-the-loop approaches. Some approaches, such as refusals of unsafe inputs, build directly on established LLM safeguards and are relatively well understood.[43] Others, such as multi-step detection, where monitors track progress across multiple steps to spot when agents drift from user goals, remain experimental and need further development. While model providers typically apply safeguards at the model’s input/output boundary, agent deployers control the orchestration layer (tool calls and results) and therefore have the visibility needed for workflow-level monitoring and intervention.

Figure 2. Layered failure detection controls across the agent workflow



At the pre-action stage, controls focus on filtering unsafe inputs and detecting invalid plans before any external action is taken. Current practices include prompt filtering systems like Meta’s PromptGuard and OpenAI’s Moderation API that attempt to screen for harmful or illegal instructions, alongside refusal mechanisms that halt execution when high-risk commands are

40 Real-time mitigations like content filters or refusals differ from post-hoc monitoring of AI systems. Post-hoc monitoring involves three components: logging user interactions with generative AI systems, analyzing these logs and user reports for violations using keyword scanning or lightweight AI models, and enforcement actions like warnings or account suspensions based on their acceptable use policy (Adler, 2025).

41 Input filters block or flag certain user messages before they reach the model, while output filters scan its responses before users see them. These filters often rely on rules, classifiers, or pattern matching. Refusals, by contrast, are built into the model through training techniques like Reinforcement Learning from Human Feedback (RLHF), allowing the model to decline harmful requests rather than just filtering content. All of these safeguards can be bypassed through methods like adversarial prompts, jailbreaks, or fine-tuning (Bengio et al., 2024).

42 Observability layers provide interfaces that allow users or operators to track an agent’s progress across all stages and intervene in an agent’s operations when necessary (Chan (c) et al., 2025)

43 Real-time refusals for agents extend established content moderation techniques from general-purpose AI systems, though these approaches must account for the unique risk patterns that emerge when AI systems take direct actions rather than generate text (Oueslati & Robin Staes-Polet, 2024).

detected.[44] In agents, refusals remain an open research area, particularly how to design systems that can decline illegal or inappropriate tasks while still completing legitimate objectives. Pre-action controls may also observe or even probe an agent's internal reasoning or plan before execution, identifying goal-divergent or suspicious logic early.[45]

During action-taking, real-time monitoring addresses tool failures, plan divergence, and boundary violations. These controls are less mature and often rely on runtime classifiers (separate ML models that flag unsafe content or actions as they occur) or, in some deployments, secondary large language models tasked with observing the agent's reasoning and tool use in real time. If suspicious behavior is detected, such as an indirect prompt injection (e.g., malicious instructions hidden in a document or webpage), navigation to restricted domains, or misuse of external tools, the monitoring system can halt the agent's execution, escalate to a human, or retry with safer alternatives. Some deployments, like OpenAI's Operator, use layered runtime classifiers to enforce "allowlists" of permitted websites and secondary models to detect injection signatures, though these techniques remain uneven across providers.[46]

Monitoring across steps is the least developed set of controls. These controls track anomalies like goal drift and behavioral changes that only surface when observing a sequence of actions, rather than any one step. Emerging practices, such as Meta's AlignmentCheck, use language-model reasoning to compare an agent's action sequence against the user's stated objective, flagging deviations that may signal covert prompt injection, misleading tool output, or hijacked instructions. This "semantic lens" attempts to close gaps left by static rules, which excel at catching obvious jailbreaks but miss instructions embedded in documents, prompts, or tools that appear benign individually.[47]

Detection methods span automated and human-in-the-loop approaches, as well as rules-based versus behavioral checks; each has tradeoffs that determine where it can work reliably, but these trade-offs are not yet well understood. As an example, automated systems — LLM-based evaluators or secondary "monitor" models or agents — can operate continuously and at scale, making them cost-effective for large deployments. Human-centered approaches, by contrast, prioritize interpretability and judgment, often through features like Operator's Watch Mode or Takeover controls that mandates user supervision on

44 Chennabasappa, 2025; OpenAI (b), 2025.

45 Some agent implementations are exploring active querying approaches rather than passive observation of reasoning traces. Anthropic's "think" tool allows Claude to "stop and think about whether it has all the information it needs to move forward" during operation, providing a structured space for reasoning during complex tasks. (Anthropic (d), 2025).

46 Operator combines rule-based restrictions and behavioral monitoring, with multiple escalation mechanisms for sensitive contexts. The system uses confirmations that require user approval before actions affecting the state of the world, Watch Mode mandates user supervision on high-risk websites like email services, and a prompt injection monitor that pauses execution when suspicious instructions are detected. Additional controls include proactive refusals for high-stakes tasks like banking transactions and domain restrictions that block navigation to prohibited websites. All escalations are directed to the user for approval or oversight rather than to human operators, emphasizing user control while providing automated safety checks (OpenAI (a), 2025).

47 Chennabasappa, 2025.

high-risk websites and let users pause or override an agent mid-task.[48] Automation scales quickly but struggles with context, while humans catch nuance but are slow, expensive, and inconsistent at scale. A further open challenge is calibrating interventions: should a system self-correct, notify asynchronously, pause for human input, or halt entirely? Each choice carries trade-offs, from operator overload to delays or even new hazards if mistimed.

The underlying logic of these detection methods also diverges. Both rule-based and behavioral approaches represent forms of automated failure detection, distinct from human-in-the-loop oversight. Rule-based checks impose hard boundaries, blocking certain tool calls, enforcing rate limits, or filtering unsafe content.[49] These controls don't consider user intent or the behavior of an agent; the controls simply stop violations. In contrast, behavioral approaches ask a harder question: Is the agent still doing what the user intended? These systems, like Meta's AlignmentCheck or OpenAI's secondary monitor model in Operator, attempt to track execution traces to spot goal drift, covert prompt injection, or tool misuse that may not violate explicit rules. Such controls would likely need to rely on powerful models reasoning over context, which is resource-intensive and expensive as a result. Both methods are necessary; neither is sufficient alone. Yet, the trade-offs — cost, reliability, privacy, and when to prefer rules over behavior — remain poorly understood.

While industry prototypes show that layered monitoring is possible, approaches remain difficult to scale, with open questions about cost, privacy, and reliability — highlighting the need for shared norms and technical investment. Automated monitors, especially when powered by LLMs, inherit the opaqueness and brittleness of the systems they oversee.[50] Human oversight mitigates some risks but is inherently limited by speed and cost. Privacy complicates matters further. While real-time monitoring avoids some privacy risks by not requiring persistent data storage, it still involves continuous observation of agent behavior that could reveal sensitive information about users' activities, goals, and decision-making patterns.[51] Striking the right balance remains unresolved. Finally, questions of reliability cut deeper: most real-time monitoring systems have not been independently validated, and practices like chain-of-thought (CoT) monitoring remain hotly debated. CoT inspection can expose early signals of misbehavior ("let's hack this site"), but traces may be unfaithful or strategically hidden.[52]

These uncertainties suggest that while layered monitoring offers a foundation, deciding

48 OpenAI (a), 2025. Similarly, Google's Project Mariner enables users to observe browser agent actions and take control when necessary, allowing users to "stop the agent entirely, and take over what it was doing" at any time during execution (Google DeepMind, 2025). Anthropic's Claude Code allows humans to stop Claude whenever they want and redirect its approach, providing real-time human oversight capabilities (Anthropic (e), 2025).

49 Rule-based checks can also reflect any thresholds established by the users or human operators.

50 Failures in "AI monitoring AI" are possible, including cascading vulnerabilities where prompt injections affect both primary and monitor systems (Shavit et al., 2023)

51 As Chan (b) et al., 2024, explain, "real-time monitoring involves real-time analysis of an agent's activity" without requiring "the collection or storage of activity logs," distinguishing it from post-hoc monitoring approaches. They note that "some cloud providers already offer no-logging provisions for their language model deployments to some customers, subject to real-time monitoring for abuse."

52 Chen et al., 2024.

where to deploy it and how much to invest in it requires a risk-based framework, which we explore in the next section.

Failure detection efforts should be calibrated to the stakes of the use case or task, the reversibility of potential failures, and the agent's architectural affordances.

While the previous section described how failure detection can be layered across an agent's workflow, this section examines how those controls should be scaled. We argue that detection should be scaled up — in coverage, frequency, and intensity — when three factors align: the task is high-stakes, the consequences are hard to reverse, and the agent has expansive affordances. Each factor alone increases the need for detection, but together they provide a framework for prioritizing investments in real-time monitoring and layered controls. The following section explains each factor, why it matters for safety, and how specific agent characteristics affect the level of detection required.[53] This framework does not map specific mechanisms (e.g., human approval vs. automated rules) to each factor since those choices are highly context-dependent and evolving. Here we refer to the intensity of detection as the resources and effort devoted to monitoring, such as the compute power used for secondary models, the number of checks across steps, or the tolerance for delay or false positives.

Figure 3. Calibrating failure detection by stakes, reversibility, and agent affordances

High-stakes tasks or use cases require reliable, real-time detection to prevent harmful or costly failures. While low-stakes failures (e.g., sending a redundant email or mislabeling a file) generally have limited impact, high-stakes failures can carry serious consequences and therefore demand reliable real-time failure detection. Crucially, stakes must be assessed at both the task and use-case levels. An agent that simply formats data or fills out a form might seem low-stakes, but if its output feeds into an automated healthcare triage system

53 These factors exist on a spectrum, not as strict binaries. For example, stakes can range from trivial inconvenience to catastrophic harm, and affordances can vary from tightly scoped to highly open-ended. The table below is not exhaustive but illustrates common agent characteristics, risks, and examples to help reason about these gradients without reducing them to absolutes. What counts as high stakes can also change over time; for instance, access to sensitive data may shift from high to moderate risk as session isolation, defenses against prompt injection, and real-time monitoring become more reliable.

or public benefits determination, the consequences can be significant. Similarly, some tasks are inherently high-stakes regardless of broader use case. For example, if an agent accesses sensitive personal or financial data, failures like cross-session data exposure or prompt injection can result in privacy breaches, compliance violations, or account takeovers, even in workflows that otherwise appear low-risk.[54, 55]

If a failure is hard to undo, earlier detection becomes more critical. Irreversible failures involve one-way operations that cannot be undone, or can only be reversed with extreme difficulty, external intervention, or significant cost. These include actions like deleting records, transferring funds, or sending sensitive communications.[56] Such actions can carry legal, financial, or reputational consequences that cannot be rolled back. Early detection helps prevent cascading failures by halting agents before these outcomes occur.[57] By contrast, reversible actions, such as scheduling a meeting or making an online order, can typically be corrected after the fact. In these cases, post-hoc detection through log reviews or user feedback may be sufficient. Reversibility is conditional, since agents act through external tools and services. As agents advance, they could be designed to assess the reversibility of their actions and pause or seek human review when corrections would be costly or impossible.[58]

As agents gain more expansive affordances, their behavior becomes harder to predict and control. This increases the risk of subtle or cascading failures, making layered failure detection essential. Architectural choices such as allowing the agent to select tools dynamically, retain memory across sessions, and use advanced reasoning expand its action space by increasing its autonomy, ability to handle complex goals, and impact on the environment. These capabilities make agents more powerful but also more prone to compounding errors or misalignment over long workflows. For example, two agents with the same task may pose very different risks if one can access arbitrary tools and another is restricted to a fixed workflow. Agents with limited affordances, like those confined to predefined tools, are easier to monitor and their failures are more predictable. As affordances grow, so does the need for richer real-time monitoring and layered detection mechanisms that can adapt to complex, long-horizon tasks. Future agents may combine memory, flexible tool access, and advanced reasoning in unpredictable ways, and architectures will likely evolve, so the categories in the table below reflect common clusters rather than fixed archetypes.[59]

---

54 He et al., 2024, and Mitchell et al., 2025.

55 Microsoft's internal review process, the Sensitive Uses and Emerging Technologies program, provides one model for evaluating high-stakes deployments. They consider the use of AI systems as sensitive when they affect access to healthcare, risk physical or psychological injury, or potentially undermine human rights. (Microsoft (b), 2025). This approach focuses on a system's potential downstream impact on core human interests. Similarly, scholars argue that an agent's stakes are shaped not just by its capabilities, but by the significance of the environment it operates in, especially when that environment bears on human well-being, social structures, or the pursuit of meaningful goals. Together, these perspectives reinforce that stakes must be assessed in context. (Kasirzadeh & Gabriel, 2025).

56 Examples include bulk-removing email labels or sending a medication reminder at the wrong time (OpenAI (a), 2025).

57 For example, a recent study from Anthropic suggests that requiring human approval for irreversible actions, while restricting agents' access to sensitive tools and data and setting their goals carefully to avoid unintended priorities, can help mitigate risks as agents gain more autonomy and real-world access (Anthropic (f), 2025).

58 Center for Security and Emerging Technology, 2024.

59 Kasirzadeh & Gabriel, 2025.

Stakes: High

Stakes reflect how serious the consequences could be if an agent fails. This includes harms from specific TASKS the agent performs (e.g., sending a message, accessing data) as well as risks tied to the broader USE CASE or domain the agent supports (e.g., education, finance, healthcare).

HIGH STAKES	
High-stakes failures can result in financial loss, safety risks, legal violations, or harm to individual rights. Deployers need robust, real-time failure detection since errors at this level can cause significant harm and must be prevented before they occur.	
AGENT ATTRIBUTES/ACTIONS	**RISKS**
Can access sensitive personal and financial data TASK	Access to sensitive personal data can be considered high stakes for agents because session management represents a critical blind spot: most agents don't isolate user sessions robustly, causing chat histories to bleed across users which could lead to data leaks. Attackers can also manipulate agents, often through prompt injection, to retrieve and leak sensitive data. This can involve guiding the agent through a series of actions that expose private information in URLs, code snippets, or other tool outputs. Failures can result in privacy breaches, compliance violations, intellectual property loss or account takeovers.
Can trigger legal liability through communications or representations TASK	Agents can perform tasks that trigger legal liability when they act or are reasonably perceived to act on behalf of a person or organization, rather than merely drafting suggestions for review. AI agents are increasingly envisioned to perform vital business functions or "join the workforce." If an agent autonomously sends a message construed as a job offer, accepts contract terms, makes harmful public claims, or engages in discriminatory conduct, it may expose the deployer to lawsuits, reputational harm, or regulatory penalties. These risks stem from the agent's communications being treated as binding or representative, even without explicit authorization. Unlike the example below, which concerns failures in regulated domains with ex-ante obligations, this characteristic focuses on ex-post liability, legal risk that arises from the agent's own statements or representations after deployment.
Handles tasks in a regulated high-risk domain USE CASE	Certain AI use cases are highly regulated due to their potential impact on health, safety, or fundamental rights. For example, Annex III of the EU AI Act designates systems in domains like employment, education, access to essential services, and law enforcement as high-risk — unless the systems don't influence decision making or there's no material risk of harm (Article 6). Similar obligations apply under regulations like HIPAA (healthcare) and the Fair Credit Reporting Act (finance), where failures may result in legal violations or rights-based harm.
Performs tasks in contexts affecting individual health, safety, or wellbeing USE CASE	When AI agents are deployed in use cases like mental health support, grief assistance, or wellness coaching, they can significantly influence users' psychological wellbeing. The risk increases with agents that retain memory, personalize interactions over time, or are embedded in routines that create emotional dependence. Failures here include agents giving misleading advice, reinforcing harmful beliefs, or abruptly shifting behavior in ways that cause distress. A wellness agent, for example, may provide inappropriate advice during a mental health crisis. Such failures can result in psychological harm or misdirected care.
Can alter critical code or system operations TASK	Granting agents the ability to download files, execute code, or run commands exposes entire systems to cascading failures. A single misstep such as an agent being tricked into running malicious code from a poisoned GitHub repository or downloading compromised software packages could cause infrastructure disruption or critical system compromise. In domains like healthcare, energy, or transportation, such failures can propagate rapidly across networks and essential services, creating catastrophic risk of severe economic damage or even harm to human life.
EXAMPLES	
• A triaging agent used in emergency rooms to prioritize care[60] • An agent that analyzes applicant data, and autonomously makes binding loan approval or denial decisions[61]	

60 Consider an LLM-enabled agent similar to COMPOSER, an AI system developed by UC San Diego, that monitors patient lab reports, vitals, and medical history from the time of check in. When the system detects a high-risk sepsis pattern, it automatically alerts nursing staff (Vazquez, 2024).

61 Figure, a fintech company offering home equity lines of credit, uses Gemini's models to power chatbots that interact with customers, guide them through form submissions, and issue approvals — often in real time (Figure AI, 2025).

Stakes: Low

LOW STAKES	
Low stakes failures cause little harm beyond inconvenience. Deployers can rely on post-hoc detection, such as user feedback or log review, rather than intensive upfront safeguards, since errors are minor and generally easy to fix.	
AGENT ATTRIBUTES/ACTIONS	**RISKS**
Creates user-facing content (e.g., bios, resumes, websites)	Creative agents that assist with writing dating profiles, resumes, or building personal websites are generally low stakes because users can review, edit, or discard outputs. But failures can still occur. In some cases, generated content may misrepresent users or embed subtle biases. When these agents are used to build live websites or apps, they may produce code with security vulnerabilities that are difficult to detect during review, exposing users or businesses to downstream risk.
Performs scheduling tasks	Scheduling is generally low stakes because most errors, like incorrect meeting times, can be fixed and do not cause serious harm. But in high-risk domains such as healthcare, failures like misprioritizing between competing goals or misallocating resources can delay critical services. For example, a healthcare agent might prioritize efficiency over patient urgency or continuity of care, resulting in delay or denial of access to essential services.
Summarizes content	Agents performing summarization tasks, such as generating meeting notes, composing follow-up emails, or sharing recaps with stakeholders, are generally low stakes. But context matters. Summarizing legal documents can carry higher risk if agents omit key clauses or misstate terms, particularly when users lack the expertise to verify accuracy. Many deployments will likely mitigate risk through disclaimers clarifying that outputs should be reviewed.
EXAMPLES	
• Creating user-facing content like resumes, bios, or personal websites[62]	
• Internal-facing agents that organize data, summarize meetings, or prepare drafts (without direct customer/system impact)	

62 Bumble plans to introduce AI-assisted dating profile creation that can select photos, offer conversational support, and potentially shortlist eligible matches (Forristal, 2024).

Reversibility: Irreversible

Reversibility refers to how easily a failure can be corrected or undone once an agent has taken an action.

IRREVERSIBLE	
Irreversible failures are those that result in one-way operations that cannot be undone (or only undone with extreme difficulty or external intervention). Early detection prevents cascading failures.	
AGENT ATTRIBUTES/ACTIONS	**RISKS**
Initiates financial transactions	The reversibility of financial transactions can vary widely by type, amount, and timeframe. While some transactions can be reversed through institutional policies (refunds, cancellation windows, dispute resolution), others could become effectively irreversible within minutes of execution. Financial mistakes can cascade through connected systems, amplifying the impact beyond the original transaction. For example, an incorrect payment could set off security alerts that block other transactions.
Deletes or overwrites data	Data loss or corruption may be unrecoverable. Even when backups exist, recovery can be costly, time-consuming, and disruptive. These risks grow when changes propagate across shared or interconnected environments, where one action can affect multiple users or systems, complicating recovery. For example, deleting database entries, overwriting cloud files, clearing task queues, or bulk-modifying metadata (like email labels) can each trigger downstream effects that are difficult or slow to undo.
Sends communications	Most communications, once delivered, cannot be undone. Emails, messages or social media posts may trigger actions or decisions that are difficult to reverse. Some communications, like calendar invites, can be canceled, but whether they're truly reversible depends on how quickly recipients respond and whether follow-up actions have already started. These risks are heightened when messages relate to time-sensitive events: if reminders or alerts are sent too early or too late, the failure becomes irreversible once the window to act has passed. For example, a medication reminder delivered at the wrong time may lead to a missed dose that cannot be corrected afterward.
EXAMPLES	
• A coding agent that can execute terminal commands, modify production code, and delete system files while pursuing its assigned objectives[63]	
• A trading agent that autonomously conducts market research, selects trading strategies, and executes buy/sell orders to pursue user-defined investment objectives	

63 Vibe coding agents like Replit and Gemini CLI were found to have deleted production databases in spite of commands to not modify code. The database deleted by Replit was recovered, and Gemini deleted files in a sandbox environment (Orland, 2025).

Reversibility: Reversible

REVERSIBLE	
High reversibility means failures can be easily corrected. Post-hoc detection and correction may be more cost-effective than real-time monitoring, with a focus on learning from failures rather than preventing them.	
AGENT ATTRIBUTES/ACTIONS	**RISKS**
Acts through third-party APIs with conditional reversibility	Agents often use API calls to interact with external tools and services, triggering real-world changes such as purchases, bookings, or account updates. Some of these actions can be reversed with user effort or within policy windows, for example, canceling a ride, returning groceries, or reversing a subscription change. Others, such as non-refundable bookings or permanent account modifications, become effectively irreversible once executed.
Operates in a sandboxed or test environment	Agents running in sandboxed or test environments work on temporary copies of data or isolated systems therefore, errors can be intercepted or corrected before they affect real systems. Any actions disappear when the session ends, leaving no lasting consequences. Examples include local-only simulations, software testing environments, and modeling tools.
EXAMPLES	
• Most scheduling agents offer calendar entries or task assignments that can be easily adjusted or removed by a user.[64]	
• An agent that produces code commits that can be rolled back through version control systems[65]	

64 Microsoft Outlook's AI scheduling assistant and similar calendar agents can create, modify, or cancel meeting invitations through calendar APIs, making their actions reversible because users can delete scheduled events, modify attendee lists, or rescind invitations.

65 GitHub Copilot and other AI coding assistants can generate code commits using APIs and Git versioning, making their actions reversible because developers can use standard Git commands like git revert or git reset to undo any AI-generated code changes, leveraging the version control system's inherent rollback capabilities.

Affordances: Unconstrained

Affordances refer to what an agent's architecture "affords" or enables it to do, such as flexible tool use, advanced reasoning or planning capabilities, and memory that persists across sessions.

UNCONSTRAINED	
Agents with unconstrained affordances operate with open-ended capabilities. As affordances increase, failures are more likely to emerge in subtle or cascading ways, requiring layered failure detection mechanisms to ensure safety.	
AGENT ATTRIBUTES/ACTIONS	**RISKS**
Dynamically selects and chains tools	Agents with this architecture can flexibly choose and combine multiple tools or APIs to accomplish tasks, rather than following predefined workflows.[66] When agents plan, select, and execute sequences of tool calls, errors at any stage, such as choosing an inappropriate tool, misinterpreting results, or re-planning mid-run, can compound. These failures can disrupt interconnected systems in ways that are hard to predict or reverse. The ability to select tools dynamically also makes it more likely that such agents operate across multiple domains or are used in unanticipated contexts, which further raises the potential for unexpected risks.[67]
Persistent memory across sessions	Agents that can retain and recall information across interactions, enable more continuous learning and adaptive behavior. But persistent memory also creates new risks: malicious or outdated information can shape future actions, leading to unintended outcomes.[68] Agents may continue acting on stale policies, preferences, or instructions unless explicitly updated. Attacks like memory poisoning, where threat actors inject malicious content into an agent's stored memory, can hijack behavior each time that memory is accessed.[69]
Extended reasoning and planning capabilities	Advanced reasoning and planning enable agents to coordinate across multiple objectives and run long, adaptive workflows with minimal human intervention. These abilities increase autonomy but also raise new risks: agents can over-optimize toward unintended subgoals, pursue strategies that diverge from user intent, or continue acting beyond the original task scope if goals are loosely specified. It remains unclear whether further advances in reasoning will be essential for developing more capable agents as the field evolves.
EXAMPLES	
• An executive COO agent that dynamically selects and combines APIs, databases, and communication tools to negotiate contracts, restructure operations, and execute strategic decisions while maintaining persistent memory of long-term business objectives • An accounting agent that autonomously orchestrates tax preparation by selecting appropriate tools, maintaining knowledge of compliance requirements, requesting missing documentation from multiple departments, and executing complex multi-step workflows for final submission without human intervention	

66 NIST (a), 2025.

67 In the framework from Kasirzadeh & Gabriel, 2025, operating across different domains and contexts falls under "generality," which denotes the breadth of domains and tasks across which an agent can effectively operate. Shavit et al., 2023, address cross-domain operation under "environmental complexity," which they define as encompassing multi-stakeholder environments, long time horizons, and the use of multiple external tools.

68 IBM Consulting, 2025.

69 Microsoft (a), 2025.

Affordances: Constrained

CONSTRAINED

Agents with constrained affordances operate within tightly scoped parameters. These systems are relatively easier to predict and control, so deployers can use less layered failure detection mechanisms compared to agents with more expansive affordances.

AGENT ATTRIBUTES/ACTIONS	RISKS
Uses predefined tools and workflows	Agents with this architecture are limited to a fixed set of tools and follow structured workflows, rather than choosing tools dynamically. These constraints on tool access limit both which tools agents can use and how they can use them, restricting tool selection, usage permissions, and operational scope. By keeping tool choices and sequences tightly controlled, these agents are less flexible but present a smaller risk surface, since they cannot combine tools in unexpected ways or trigger novel failure paths. However, failures can still stem from rigid workflows breaking when environments change (e.g., a fixed tool sequence that no longer fits user contexts), leading to stalled or incomplete task execution.
Operates with episodic memory only	Agents that retain information only during a single session, resetting once the interaction ends. Without persistent memory, these systems can only handle short, bounded tasks and cannot build on past interactions. This limits their autonomy and goal complexity, reducing the chance of gradual behavior drift or unexpected capability growth over time. While episodic memory limits compounding errors, it can still create risks, agents may repeatedly request or store sensitive data each session, increasing privacy exposure and the chance of accidental disclosure.[70]

EXAMPLES

- A general-purpose agent operating within constrained environments with limited session memory and access to only predefined tools[71]

70 Note that most current LLMs have primarily episodic memory, which severely limits their ability to pursue long-term goals.

71 Google's Project Astra maintains 10-minute session memory across multi-modal conversations, adapting to real-world visual input while using Google Search, Lens, and Maps based on conversational context (Pichai, 2024).

3. Safety-critical industries show failure detection can reduce harms and provide a foundation for safer agent design

Safety-critical domains, like the automotive industry, offer practical lessons for AI agent design in balancing safety against utility and cost. This industry has long grappled with the tension between dangerous under-engineering and costly over-engineering, and has reached a point where vehicle failures constitute a tiny minority of road accidents.[72] In particular, safety practices in both autonomous systems and the wider automotive industry offer a structured model for failure detection in AI agents.

Higher-risk functions in road vehicles require stronger failure detection controls. This principle can inform how we assess and manage action-level risk in AI agents.

Structured risk assessments guide how much failure detection is needed for different system functions. The automotive industry operates on the principle of achieving an "absence of unreasonable risk." This acknowledges that absolute safety is infeasible, but acceptable levels of risk can be defined through public consensus and formalized in regulation.[73] In the U.S., this principle underpins standards like ISO 26262 (Road Vehicles – Functional Safety), which provides detailed guidance for vehicle manufacturers to assess the risk of individual components involved in vehicle operation. A central element of ISO 26262 is requiring manufacturers to conduct a Hazard Analysis and Risk Assessment (HARA) process. HARA evaluates the safety relevance of a system's functions by scoring each on three dimensions:

- Severity – potential harm to humans if the function fails
- Exposure – the probability of the hazard occurring
- Controllability – the ability of humans or systems to mitigate the hazard[74]

Based on these scores, manufacturers assign each function an Automotive Safety Integrity Level (ASIL) and design appropriate safety controls.[75] Components responsible for steering or braking are treated as high-risk and require strong failure detection and backups. Lower-risk components, like entertainment systems, are subject to lighter oversight. For autonomous

72 As of 2007, vehicle failure or degradation was a critical cause of around 2% of accidents in the US (U.S. Department of Transportation, 2015).

73 See the statutory definition of "motor vehicle safety" in 49 U.S. Code § 30102(a)(9) (Waymo (a), 2023).

74 The mapping between HARA and our framework is not exact, but conceptually useful. "Severity" relates to "stakes" (the potential harm from failure) and also overlaps with "reversibility." "Controllability" reflects our focus on real-time failure detection, and other mitigations. "Exposure" lacks a direct equivalent, but our concept of "affordances", how flexibly an agent can act, use tools, and access memory, captures a similar intuition about risk amplification in unconstrained environments.

75 "The definition of the fail-safe property of an automated driving system in the technical report ISO/TR 4804 [31] specifies the need to achieve a minimal risk condition in addition to a safe state in the event of a failure" (Pafla et al., 2021).

vehicles, 'human-in-the-loop' controls are key when a failure is detected, and control of the vehicle can be passed to a remote operator.[76]

Like vehicle manufacturers, AI agent developers can adopt a structured risk assessment, calibrating detection efforts based on stakes, reversibility, and affordances, to ensure systems fail safely without eroding utility. This approach can help establish a shared baseline for acceptable risk while industry or regulators explore more detailed standards. The analogy applies most clearly at the action level. Just as the function of steering carries more risk than adjusting the radio, certain agent behaviors, such as executing external code or making irreversible decisions, present higher stakes and lower reversibility. These actions may warrant more comprehensive and layered failure detection than others. However, identifying failures and interrupting automation too often can erode the value of an AI agent, so architectures that ensure that the system can continue in the event of a failure may be valuable. One engineering solution is to use backups.

Backups can support fail-safe operation but require detection to function properly.

One way safety-critical systems preserve function in the presence of failure is through backups — independent components that take over when the primary fails. This strategy is common in vehicles:[77] backup sensors mitigate the risk of single sensor failure, and Waymo uses a redundant secondary computer to take control if the primary system fails in automated vehicles.[78] Rather than requiring each individual component or function to be fail-proof, the system as a whole is made robust through monitoring and backups. This principle is formalized in ISO 26262, which allows high-risk requirements to be met through multiple lower-risk components, as long as their joint probability for failure remains low.[79] Crucially, comprehensive failure detection is necessary to activate these backups when needed.

76 Krome et al., 2023.

77 This is the idea of "redundancy" which can be described as "the ability to provide for the execution of a task if the primary unit fails or falters." (Leveson et al., 2009). Note: We don't use the term redundancy to avoid conflicts with other definitions for the term.

78 Waymo (b), 2021.

79 This reflects the idea that two diverse, independently operating systems are less likely to fail in the same way. ISO 26262 explicitly permits this approach to satisfying high ASIL requirements.

AI agent architectures can take inspiration from this design choice. For example:

- If one agent fails to complete a tool-based task, a backup or "checker" agent could attempt the same action with fresh context.[80] Alternatively, environmental context could be cached as a backup and restored if the agent completed its task incorrectly, e.g., git branching and roll-backs in software development.
- For critical user goals, multiple models could independently generate outputs, with a monitoring system flagging inconsistencies as potential failure signals.[81]

Translating these practices to AI agents will require more research and guidance, given differences in how agents operate and fail. While AI agents differ from vehicles in how failures can stem from shared design flaws rather than isolated hardware faults,[82] safety practices from the automotive sector still offer a valuable foundation for designing failure detection controls. The auto industry's experience shows how structured risk assessment, layered detection, and well-scoped backups can reduce failure rates without sacrificing performance. Further technical research and regulatory guidance will be essential to expand on these learnings and inform how AI agent developers and deployers should design and evaluate failure detection systems.

4. Significant technical research and regulatory guidance must be prioritized to close gaps in designing and evaluating failure detection for AI agents

Agents pose new, dynamic risks in ways generative AI systems do not. The system's ability to plan, use tools, and take actions across different contexts means that failures can emerge dynamically, beyond what developers can address through pre-deployment testing. Today, agent developers experiment with real-time monitoring — catching unsafe inputs, invalid plans, tool errors, and boundary violations, while allowing for human oversight. These controls vary in timing (before, during, or across actions) and method (rule-based vs. behavioral, or automated vs. human). But they remain fragmented, untested at scale, and inconsistently adopted, even as we have yet to see LLM-based agents deployed widely. Whether the market centers on narrow, task-specific agents or general-purpose agents, both trajectories will require scalable, credible monitoring practices, though the design and emphasis of those practices will differ.

80 Checker or "inspector" agents have been proposed to correct faulty agent behavior (Huang et al., 2025).

81 An automotive comparison here might be "sensor fusion," which is the process of combining data derived from disparate sources so that the resulting information has less uncertainty than would be possible if these sources were used individually. For example, wheel speed sensors, accelerometers, gyroscopes, and GPS systems are used to calculate the speed of a vehicle. For AI agents, multiple models or agents could be used to suggest an action, with a safety agent 'combining data from these disparate sources' to identify significant discrepancies across the agents that may highlight a failure, reducing uncertainty in failure detection. Such an approach could significantly increase costs (Frigerio, 2022).

82 This is called "Common Cause Fault" and highlights the importance of diversity and independence when designing backups and integrating redundancy (Frigerio, 2022).

Closing these gaps will take collective work in three areas: building technical methods, developing shared approaches to evaluation, and ensuring policies or market incentives support adoption.

R&D GAPS: SPECIFIC METHODS THAT NEED DEVELOPMENT

For researchers and engineers in industry and academia

- **Advance multi-step detection for goal drift.** Spotting misaligned behavior across multi-step workflows, such as agents drifting from user goals, remains experimental and underdeveloped. These methods require more technical research and evaluation to be better understood and scaled.
- **Develop scalable, validated "monitor" models or agents.** LLM-based monitors inherit brittleness and opacity from the systems they watch. Independent evaluation (like we explore below), benchmarks or standards, and privacy-preserving design are needed to make such monitors trustworthy and scalable.

EVALUATION GAPS: MEASURING THE EFFECTIVENESS OF FAILURE DETECTION

For standards bodies, industry consortia, and assurance providers best positioned to lead this work

- **We lack a clear understanding of when human-in-the-loop controls meaningfully reduce risks from agents.** Human oversight is often invoked as a safeguard but its effectiveness hinges on context — particularly for irreversible or high-stakes actions. Key unknowns include how much human oversight actually reduces failures in real-world settings, whether humans step in quickly enough, and how to avoid handoffs that slow or complicate the system. One proposed action can be to conduct pilot studies across domains (e.g., finance, healthcare, customer support) to measure effectiveness of human oversight, prioritizing evaluations of tasks by stakes, reversibility, and agent affordances.
- **Agents performing high-stakes actions require external assurance that their real-time monitoring controls actually work.**[83] Assurance refers to processes that independently validate whether a system's safeguards work as intended, often through documented evidence ("safety cases") and accredited assurance providers. As discussed earlier, other sectors show how this can work. In the automotive industry, car functions that carry higher safety risks are tested and certified more rigorously, with external assessors reviewing evidence that the systems' safety controls and backups work reliably.[84] For agents, similar tiered approaches could assign higher scrutiny to riskier tasks or actions, for example,

83 Effectiveness of failure detection must be measured, not assumed. AI-driven content moderation systems can fail quietly in practice, even when widely deployed. A study of Facebook's automated moderation during the January 6 Capitol riot found that its machine-learning models and automated downranking prevented only 21% of engagement with harmful posts, allowing most problematic content to circulate before intervention, see Goldstein et al., 2023.

84 For example, UL 4600 defines how to build and evaluate a safety argument for autonomous vehicles, with redundancy and fault detection and mitigation in section 10.3 - 10.4. Additionally, ISO/PAS 8800 provides guidance on how to extend a safety case for AI systems, with a greater focus on processes and product characteristics (Critical Systems Labs, 2025). However, ISO 26262 is more ubiquitous.

financial transactions or decisions affecting health, and require validation that failure detection controls are in place and effective. While regulatory drivers for second- or third-party assurance remain limited, market incentives (such as reputational advantage or trust in sensitive sectors) may motivate voluntary adoption, especially as domain-specific agent applications scale.[85]

- **Without standardized evaluations, we cannot know whether real-time failure detections work as intended.** Researchers have called for a richer evaluation science for agents, where tests simulate real-world complexity, multi-step actions, and test for varied risks.[86] Building on this, evaluations could assess the reliability of runtime monitoring itself, not only agent performance. This could test whether its monitoring layers catch failures, avoid unnecessary human interventions, and respond fast enough to matter.

POLICY GAPS: GOVERNMENT LEVERS TO DRIVE ADOPTION

For regulators, agencies, and multilateral bodies

- **Clarify expectations for human oversight.** Article 14 of the EU AI Act applies to high-risk AI systems, requiring them to be "effectively overseen by natural persons." Agents may fall within this category if deployed in high-risk domains, or if unintentionally used in such settings, making these requirements relevant. Yet Article 14's emphasis on human oversight risks overloading individuals tasked with monitoring outputs.[87] Regulators like the EU AI Office could issue guidance on what counts as adequate observability, when human approval must be mandatory, and how automated detection can complement oversight for high- versus low-stakes actions.
- **Use failure detection for clarifying liability.** Legal liability regimes like tort and consumer protection law, and new regulations like the EU AI Act, provide governance frameworks for AI agents, but clearer guidance is needed on what constitutes reasonable standard of care for their development and deployment.[88] Over time, integrating real-time failure detection into human oversight expectations can strengthen accountability and hold developers and deployers liable if/when preventable failures come to pass. As occurred with automobiles and other industries, explicit liability rules can help incentivize accountability.
- **Incentivize incident reporting and root-cause tracking.** Understanding why agents fail is critical for societal awareness and harm prevention. The EU's General-Purpose AI Code of Practice already calls for reporting the "chain of events" behind incidents and conducting root-cause analysis, including inputs and systemic risk

85 The report notes how seatbelt standards first emerged in the automotive industry before laws required car manufacturers and drivers to use them (Ada Lovelace Institute, 2025). While assurance for AI is still mostly voluntary, Article 61 of the EU AI Act requires high-risk systems to log incidents and monitor their behavior over time, which could become a foundation for more formal assurance and validation of failure detection controls in AI agents.

86 Kapoor et al., 2024.

87 See Article 14, EU AI Act.

88 Cihon, 2024.

failures.[89] Policymakers could go further by requiring or incentivizing deployers to design failure detection systems that capture detailed logs and traces of an agent's actions, so incident reporting is grounded in what the system actually did rather than just the final outcome.

- **Promote transparency on failure detection practices.** Model providers already disclose performance via system cards. Policymakers could encourage or require agent developers to include how their failure detection controls were evaluated, the contextual factors tested, and the rationale for their choices, building on transparency measures in the EU general-purpose AI Code of Practice.
- **Fund testbeds to evaluate and scale failure detection.** The U.S. AI Action Plan calls for the Center for AI Standards and Innovation (CAISI) to invest in breakthroughs in AI interpretability, control, and robustness. The Action Plan also calls for secure, sector-specific testbeds to advance safe adoption.[90] AI Safety Institutes should consider piloting failure detection controls, testing trade-offs (cost, reliability, privacy), and validating their effectiveness for narrow, high-stakes domains before wider deployment.[91]
- **Track market incentives for cost-effective monitoring.** Real-time monitoring adds expense (development, latency), so firms may underinvest absent clear returns on investment. Policymakers and civil society can use the stakes–reversibility–affordances framework to track where market incentives naturally emerge. They can also identify where to amplify those incentives through procurement preferences or by supporting insurance and certification schemes, helping drive investment in failure detection, especially for high-stakes uses.

89 See Measure 9.2 Safety and Security Chapter (European Union AI Office, 2024).

90 White House, 2025.

91 See ongoing work on agents at the AI Safety Institutes: UK AISI (a), 2025; UK AISI (b), 2025; NIST (a), 2025; NIST (b) 2025.

Conclusion and Limitations

This paper makes three contributions:

1. Defining levels of environmental influence in AI agents as a threshold for when failure detection is warranted
2. Introducing a stakes–reversibility–affordances framework with examples to show when detection is most necessary
3. Outlining a layered schema for failure detection across agent planning, tool use, and execution.

Advancing these approaches will require building technical capacity, shared evaluation practices, and baseline norms so these controls are reliable and scalable. These recommendations are necessarily early-stage, given that real-world deployments of LLM-based agents remain limited. We do not examine how monitoring layers affect speed, cost, or user experience across contexts, nor does it prescribe specific detection mechanisms for every task, since architectures are still evolving. Finally, we flag emerging agent capabilities (such as complex multi-agent interaction) that current mitigations may not yet fully address, underscoring the need for forward-looking safety measures.

Despite these limitations, we need a public discussion about architectural norms before agent deployments scale. This debate must involve a wider set of stakeholders than those building the systems. Architectural decisions about safety cannot be left solely to a small circle of developers. Acting now, through research, evaluation, and policy, can help ensure risk management practices evolve alongside the systems they are meant to govern.

Bibliography

Ada Lovelace Institute, "Going Pro? The case for public sector AI assurance," Ada Lovelace Institute, July 2025, https://www.adalovelaceinstitute.org/report/going-pro

Adler, Steven, "Explainer: The Basics of AI Monitoring," Substack, (2025), https://stevenadler.substack.com/p/explainer-the-basics-of-ai-monitoring

Akbulut, Canfer et al., All Too Human? Mapping and Mitigating the Risk from Anthropomorphic AI, in Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES 7 no 1, Oct. 16 2024), 13–26, doi:10.1609/aies.v7i1.31613, https://ojs.aaai.org/index.php/AIES/article/view/31613

Ancker, Jessica S. et al., "Effects of workload, work complexity, and repeated alerts on alert fatigue in a clinical decision support system," BMC Medical Informatics and Decision Making vol. 17, no. 1 (2017), https://bmcmedinformdecismak.biomedcentral.com/articles/10.1186/s12911-017-0430-8

Anthropic (a), "System Card: Claude Opus 4 & Claude Sonnet 4," Anthropic, May 2025, https://www-cdn.anthropic.com/07b2a3f9902ee19fe39a36ca638e5ae987bc64dd.pdf

Anthropic (b), "Building Effective Agents," Anthropic Research Blog, April 2025, https://www.anthropic.com/research/building-effective-agents

Anthropic (c), 2025. "How we built our multi-agent research system." Anthropic, June 2025. https://www.anthropic.com/engineering/multi-agent-research-system

Anthropic (d), "The 'think' tool: Enabling Claude to stop and think," Anthropic, "Claude Think Tool," Anthropic Engineering Blog, March 2025, https://www.anthropic.com/engineering/claude-think-tool

Anthropic (e), "Our Framework for Developing Safe and Trustworthy Agents," https://www.anthropic.com/news/our-framework-for-developing-safe-and-trustworthy-agents

Anthropic (f), Agentic Misalignment: How LLMs Could Be Insider Threats (2025), https://www.anthropic.com/research/agentic-misalignment

Bengio, Yoshua et al., International Scientific Report on the Safety of Advanced AI, Report 1 (interim report). https://www.gov.uk/government/publications/international-scientific-report-on-the-safety-of-advanced-ai

Benioff, Marc, "This Tech Giant Now Claims Artificial Intelligence Is Doing Up to 50 % of the Work," Yahoo Finance, July 2025, https://finance.yahoo.com/news/tech-giant-now-claims-artificial-140000381.html

Center for Security and Emerging Technology, Through the Chat Window and Into the Real World (2024), https://cset.georgetown.edu/wp-content/uploads/CSET-Through-the-Chat-Window-and-Into-the-Real-World.pdf

Chan (a), Alan et al., "Harms from Increasingly Agentic Algorithmic Systems," arXiv preprint arXiv:2302.10329 (2023), https://arxiv.org/abs/2302.10329

Chan (b), Alan et al., "Visibility into AI Agents," arXiv preprint arXiv:2401.13138 (2024), https://arxiv.org/abs/2401.13138

Chan (c), Alan et al., "Infrastructure for AI Agents," arXiv preprint arXiv:2501.10114 (2025), https://arxiv.org/pdf/2501.10114

Chen, Yanda et al., "Reasoning Models Don't Always Say What They Think," Anthropic, (2024), https://assets.anthropic.com/m/71876fabef0f0ed4/original/reasoning_models_paper.pdf

Chennabasappa, Sahana et al., "LlamaFirewall: An Open Source Guardrail System for Building Secure AI Agents," Meta, April 29, 2025, https://github.com/meta-llama/PurpleLlama/tree/main/LlamaFirewall

Cihon, Peter et al., "Levels of Autonomy in AI Agent Systems," arXiv preprint arXiv:2502.15212 (2025), https://arxiv.org/pdf/2502.15212

Cihon, Peter, "Chilling Autonomy: Policy Enforcement for Human Oversight of AI Agents," Workshop on Generative AI and Law (GenLaw '24), a workshop at the 41st International Conference on Machine Learning (ICML), Vienna, 2024, accessible at https://blog.genlaw.org/pdfs/genlaw_icml2024/79.pdf

Critical Systems Labs, "Towards an ISO/PAS 8800:2024 Compliant Assurance Argument: Assurance Case Development for Artificial Intelligence (AI) and Machine Learning (ML) Systems," June 2025, https://criticalsystemslabs.com/wp-content/uploads/2025/06/ISO-8800-White-Paper-with-Copyright.pdf

Crootof, Rebecca, Kaminski, Margot E., & Price, W. Nicholson II, "Humans in the Loop," Vanderbilt Law Review vol. 76, no. 2 (2023), pp. 429-510, https://scholarship.law.vanderbilt.edu/vlr/vol76/iss2/2

Figure AI (2025), https://www.figure.com/ai-at-figure

Forristal, Lauren, Bumble to Leverage AI to Help Users With Profile Creation and Conversations TechCrunch (2024), https://techcrunch.com/2024/09/10/bumble-to-leverage-ai-to-help-users-with-profiles-and-conversations

Frigerio, Alessandro, "Functional-safety analysis of ASIL decomposition for redundant automotive systems," PhD Thesis, Eindhoven University of Technology (2022), https://pure.tue.nl/ws/portalfiles/portal/199634783/20220421_Frigerio_hf.pdf

Gan, Yuyou et al., "Navigating the Risks: A Survey of Security, Privacy, and Ethics Threats in LLM-Based Agents," arXiv preprint arXiv:2411.09523 (2024), https://arxiv.org/abs/2411.09523

Goldstein, Ian et al., "A Facebook Case Study of Platform Moderation During the January 6th Capitol Riot," arXiv preprint arXiv:2301.02737 (2023), https://arxiv.org/pdf/2301.02737

Google DeepMind, Project Mariner, https://deepmind.google/models/project-mariner

Green, Ben, "The flawed theory of human oversight of algorithmic decisions," Computer Law & Security Review vol. 45 (2022), https://www.sciencedirect.com/science/article/pii/S0267364922000292

Hammond, Lewis, "Multi-Agent Risks from Advanced AI, Cooperative AI Foundation", February 2025, https://www.cooperativeai.com/post/new-report-multi-agent-risks-from-advanced-ai

He, Yifeng et al., "Security of AI Agents," arXiv preprint arXiv:2406.08689 (2024), https://arxiv.org/abs/2406.08689;

Huang, Jen-tse et al "On the Resilience of LLM-Based Multi-Agent Collaboration with Faulty Agents," arXiv preprint arXiv:2408.00989 (version 4, 28 May 2025), https://arxiv.org/pdf/2408.00989

IBM AI Ethics Board, AI Agents: Opportunities, Risks, and Mitigations (Apr. 2025), IBM, https://www.ibm.com/downloads/documents/us-en/1227c12efb38b2b3

IBM Consulting, Agentic AI in Financial Services: Opportunities, Risks, and Responsible Implementation, IBM Consulting. (2025). https://www.ibm.com/downloads/documents/gb-en/12f5a71117cdc329

Ibrahim, Lujain et al., "Towards Interactive Evaluations for Interaction Harms in Human-AI Systems," Knight First Amendment Institute (2025), https://knightcolumbia.org/content/towards-interactive-evaluations-for-interaction-harms-in-human-ai-systems

Kapoor, Sayash et al., "AI Agents That Matter," arXiv preprint arXiv:2407.01502 (2024), https://arxiv.org/abs/2407.01502

Kasirzadeh, Atoosa & Gabriel, Iason, "Characterizing AI Agents for Alignment and Governance," arXiv preprint arXiv:2504.21848 (2025), https://arxiv.org/pdf/2504.21848

Krome, Sven et al., "Remote driving as the Failsafe: Qualitative investigation of Users' perceptions and requirements towards the 5G-enabled Level 4 automated vehicles," Transportation Research Part F: Traffic Psychology and Behaviour vol. 97 (2023): pp. 445-460, https://www.sciencedirect.com/science/article/pii/S1369847823002619

Lakshmanan, Ravie, "Lovable AI Found Most Vulnerable to Multi-Step 'VibeScamming' Attacks," The Hacker News, April 2025, https://thehackernews.com/2025/04/lovable-ai-found-most-vulnerable-to.html

Laux, Johann, "Institutionalised distrust and human oversight of artificial intelligence: towards a democratic design of AI governance under the European Union AI Act," AI & Society 39(6): 2853–2866 (Oct. 6, 2023), doi: 10.1007/s00146-023-01777-z, https://pmc.ncbi.nlm.nih.gov/articles/PMC11614927

Leveson, Nancy et al., "Beyond Normal Accidents and High Reliability Organizations: The Need for an Alternative Approach to Safety in Complex Systems," MIT (2009), http://sunnyday.mit.edu/papers/hro.pdf

Li, Haitao et al., "LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods," arXiv preprint arXiv:2412.05579 (2024), https://arxiv.org/abs/2412.05579

Macnamara, Brooke N. et al., "Does using artificial intelligence assistance accelerate skill decay and hinder skill development without performers' awareness?" Cognitive Research: Principles and Implications vol. 9, article 32 (2024), https://cognitiveresearchjournal.springeropen.com/articles/10.1186/s41235-024-00572-8

Mitchell, Margaret, "Why Handing Over Total Control to AI Agents Would Be a Huge Mistake," MIT Technology Review, March 24, 2025, https://www.technologyreview.com/2025/03/24/1113647/why-handing-over-total-control-to-ai-agents-would-be-a-huge-mistake

Mitchell, Margaret et al., Fully Autonomous AI Agents Should Not Be Developed, arXiv preprint arXiv:2502.02649 (2025), https://arxiv.org/abs/2502.02649

Measure 9.2 Safety and Security Chapter, European Union AI Office, "Code of Practice for General-Purpose AI Models Safety and Security Chapter," 2024, https://digital-strategy.ec.europa.eu/en/policies/contents-code-gpai

METR, "Measuring AI Ability to Complete Long Tasks," METR Blog, March 19, 2025, https://metr.org/blog/2025-03-19-measuring-ai-ability-to-complete-long-tasks

Microsoft (a), "Taxonomy of Failure Modes in Agentic AI Systems," (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

Microsoft (b), Responsible AI Transparency Report, 2025, https://www.microsoft.com/en-us/corporate-responsibility/responsible-ai-transparency-report

Milmo, Dan, "Microsoft Launches AI Employees That Can Perform Some Business Tasks," The Guardian, October 21, 2024, https://www.theguardian.com/technology/2024/oct/21/microsoft-launches-ai-employees-that-can-perform-some-business-tasks

Narayanan, Arvind & Kapoor, Sayash, "AI Safety Is Not a Model Property," AI Snake Oil Blog, May 2025, https://www.aisnakeoil.com/p/ai-safety-is-not-a-model-property

NIST (a), "Lessons Learned from the Consortium: Tool Use in Agent Systems," National Institute of Standards and Technology, August 5, 2025, https://www.nist.gov/news-events/news/2025/08/lessons-learned-consortium-tool-use-agent-systems

NIST (b), "Strengthening AI Agent Hijacking Evaluations" (2025), https://www.nist.gov/news-events/news/2025/01/technical-blog-strengthening-ai-agent-hijacking-evaluations

OECD.AI, "OECD AI Incidents Methodology," OECD, 2024, https://oecd.ai/en/incidents-methodology

OpenAI (a), "Operator: A System Card for Oversight of Tool-Using AI Agents," March 2025, https://openai.com/index/operator-system-card

OpenAI (b), "A Practical Guide to Building Agents", OpenAI, 2025https://cdn.openai.com/business-guides-and-resources/a-practical-guide-to-building-agents.pdf

Orland, Kyle, “AI Coding Assistants Chase Phantoms, Destroy Real User Data,” Ars Technica, July 25, 2025, https://arstechnica.com/information-technology/2025/07/ai-coding-assistants-chase-phantoms-destroy-real-user-data/

Oueslati, Amin & Staes-Polet, Robin, “Ahead of the Curve: Governing AI Agents Under the EU AI Act,” The Future Society, 2024, https://thefuturesociety.org/aiagentsintheeu/

OWASP, “Agentic AI Threats and Mitigations,” OWASP (2025), https://genai.owasp.org/resource/agentic-ai-threats-and-mitigations

Pafla, Marvin et al., “Functional-safety analysis of ASIL decomposition for redundant automotive systems,” arXiv preprint arXiv:2106.11042 (2021), https://arxiv.org/pdf/2106.11042

Partnership on AI, “PAI’s Guidance for Safe Foundation Model Deployment,” Partnership on AI, (2023), https://partnershiponai.org/modeldeployment/

Patel, Dwarkesh. “Why I Don’t Think AGI Is Right Around the Corner.” Dwarkesh Blog, June 2025. https://www.dwarkesh.com/p/timelines-june-2025

Pichai, Sundar, “Google Gemini AI Update,” Google Blog, December 2024, https://blog.google/technology/google-deepmind/google-gemini-ai-update-december-2024/#ceo-message

Raji, Deborah et al., “The Fallacy of AI Functionality,” in Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency (FAccT), (2022), https://facctconference.org/static/pdfs_2022/facct22-3533158.pdf

Russell, Stuart & Norvig, Peter, Artificial Intelligence: A Modern Approach, 3rd ed., Prentice Hall, 2010, https://people.engr.tamu.edu/guni/csce625/slides/AI.pdf

Sahana Chennabasappa et al., “LlamaFirewall: An Open-Source Guardrail System for Building Secure AI Agents,” April 29, 2025, https://ai.meta.com/research/publications/llamafirewall-an-open-source-guardrail-system-for-building-secure-ai-agents

Sapkota, Ranjan et al., “Taxonomy of AI Agents and Agentic AI,” arXiv preprint arXiv:2505.10468 (2025), https://arxiv.org/pdf/2505.10468

Shavit, Yonadav et al., “Practices for Governing Agentic AI Systems,” (2023) https://openai.com/index/practices-for-governing-agentic-ai-systems

Shibu, Sherin, “AI Agents Can Help Businesses Be ‘10 Times More Productive,’ According to a Nvidia VP,” Entrepreneur, March 6, 2025, https://www.entrepreneur.com/business-news/nvidia-vp-says-ai-agents-make-businesses-10x-more-productive/487991

Su, Zhe et al., AI-LieDar: Examine the Trade-off Between Utility and Truthfulness in LLM Agents, NAACL 2025 (2025), https://aclanthology.org/2025.naacl-long.595.pdf

Terekhov, Mikhail et al., “Control Tax: The Price of Keeping AI in Check,” arXiv preprint arXiv:2506.05296 (2025), https://arxiv.org/abs/2506.05296

U.S. Department of Transportation, National Highway Traffic Safety Administration, “Critical Reasons for Crashes Investigated in the National Motor Vehicle Crash Causation Survey,” https://www.govinfo.gov/content/pkg/GOVPUB-TD8-PURL-gpo172313/pdf/GOVPUB-TD8-PURL-gpo172313.pdf

UK AISI (a), “How to Evaluate Control Measures for AI Agents” (2025), https://www.aisi.gov.uk/work/how-to-evaluate-control-measures-for-ai-agents

UK AISI (b), “Research Agenda” (2025), https://www.aisi.gov.uk/research-agenda

Uuk, Risto et al.,”A Comprehensive Survey of AI Risk Mitigation Strategies,” arXiv preprint arXiv:2412.02145 (2024), https://arxiv.org/html/2412.02145v1

Vazquez, Jeanna, AI Surveillance Tool Successfully Helps to Predict Sepsis, Saves Lives, (2024), https://health.ucsd.edu/news/press-releases/2024-01-23-study-ai-surveillance-tool-successfully-helps-to-predict-sepsis-saves-lives

Vijayvargiya, Sanidhya et al., “OpenAgentSafety: A Comprehensive Framework for Evaluating Real-World AI Agent Safety,” arXiv preprint arXiv:2507.06134 (2025), https://arxiv.org/abs/2507.06134

Wallace, Eric et al., “The Instruction Hierarchy: Training LLMs to Prioritize Privileged Instructions,” arXiv preprint arXiv:2404.13208 (2024), https://arxiv.org/abs/2404.13208

Waymo (a), “Safety Case Approach,” 2023, https://assets.ctfassets.net/e6t5diu0txbw/66jOjPtNIjzawaK0ZjpU3q/7f081b392cf29a3355c97d0d758fe6cf/Waymo_Safety_Case_Approach.pdf

Waymo (b), “Waymo Safety Report: On the Road to Fully Self-Driving,” March 2021, https://downloads.ctfassets.net/sv23gofxcuiz/4gZ7ZUxd4SRj1D1W6z3rpR/2ea16814cdb42f9e8eb34cae4f30b35d/2021-03-waymo-safety-report.pdf

Weidinger, Laura et al., “Sociotechnical Safety Evaluation of Generative AI Systems,” Google DeepMind, arXiv preprint arXiv:2310.11986 (2023), https://arxiv.org/abs/2310.11986

Weng, Lilian, “LLM-powered Autonomous Agents,” Lil’Log (blog), June 23, 2023, https://lilianweng.github.io/posts/2023-06-23-agent

White House, “America’s AI Action Plan,” July 2025, https://www.whitehouse.gov/wp-content/uploads/2025/07/Americas-AI-Action-Plan.pdf

Zhou, Xuhui et al., HAICOSYSTEM: An Ecosystem for Sandboxing Safety Risks in Human-AI Interactions, arXiv preprint arXiv:2409.16427 (2024), https://arxiv.org/abs/2409.16427

Cite as: Srikumar, M., Pratt, J., Chmielinski, K., Ashurst, C., Bakalar, C., Bartholomew, W., Bommasani, R., Cihon, P., Crootof, R., Hoffmann, M., Joshi, R., Sap, M., & Withers, C. (2025, September). *Prioritizing real-time failure detection in AI agents*. Partnership on AI.

We thank the following individuals for their valuable input on this and earlier versions of the document: Alan Chan, Chinmay Deshpande, Erica Finkle, Iason Gabriel, Gretchen Krueger, Lama Nachman, Kendrea Beers, and Helen Toner.

PARTNERSHIP ON AI