Practices for Governing Agentic AI Systems

Yonadav Shavit* Sandhini Agarwal* Miles Brundage* Steven Adler
Cullen O'Keefe Rosie Campbell Teddy Lee Pamela Mishkin
Tyna Eloundou Alan Hickey Katarina Slama Lama Ahmad
Paul McMillan Alex Beutel Alexandre Passos David G. Robinson

Abstract

Agentic AI systems—AI systems that can pursue complex goals with limited direct supervision—are likely to be broadly useful if we can integrate them responsibly into our society. While such systems have substantial potential to help people more efficiently and effectively achieve their own goals, they also create risks of harm. In this white paper, we suggest a definition of agentic AI systems and the parties in the agentic AI system life-cycle, and highlight the importance of agreeing on a set of baseline responsibilities and safety best practices for each of these parties. As our primary contribution, we offer an initial set of practices for keeping agents' operations safe and accountable, which we hope can serve as building blocks in the development of agreed baseline best practices. We enumerate the questions and uncertainties around operationalizing each of these practices that must be addressed before such practices can be codified. We then highlight categories of indirect impacts from the wide-scale adoption of agentic AI systems, which are likely to necessitate additional governance frameworks.

Table of Contents

*Lead authors. Correspondence should be directed to yonadav@openai.com.

1 Introduction

AI researchers and companies have recently begun to develop increasingly agentic AI systems: systems that adaptably pursue complex goals using reasoning and with limited direct supervision.[1] For example, a user could ask an agentic personal assistant to "help me bake a good chocolate cake tonight," and the system would respond by figuring out the ingredients needed, finding vendors to buy ingredients, and having the ingredients delivered to their doorstep along with a printed recipe. Agentic AI systems are distinct from more limited AI systems (like image generation or question-answering language models) because they are capable of a wide range of actions and are reliable enough that, in certain defined circumstances, a reasonable user could trust them to effectively and autonomously act on complex goals on their behalf. This trend towards agency may both substantially expand the helpful uses of AI systems, and introduce a range of new technical and social challenges.

Agentic AI systems could dramatically increase users' abilities to get more done in their lives with less effort. This could involve completing tasks beyond the users' skill sets, like specialized coding. Agentic systems could also benefit users by enabling them to partially or fully offload tasks that they already know how to do, meaning the tasks can get done more cheaply, quickly, and at greater scale. So long as these benefits exceed the cost of setting up and safely operating an agentic system, agentic systems can be a substantial boon for individuals and society [1]. In this paper, we will primarily focus on agentic systems with language models at their core (including multimodal models), as these have driven recent progress. [2]

Society will only be able to harness the full benefits of agentic AI systems if it can make them safe by mitigating their failures, vulnerabilities, and abuses [3]. [3] This motivates our overarching question: what practices could be adopted to prevent these failures, vulnerabilities, and abuses, and where in the life-cycle of creating and using agents are they best implemented? There are often many different stages at which harm could have been prevented. For example, consider a hypothetical agentic AI assistant whose user (not based in Japan) directs it to purchase supplies for baking a Japanese cheesecake. Instead of purchasing supplies locally, the agent purchases an expensive plane ticket to Japan, which the user only notices when it is too late to refund. In this hypothetical scenario, several parties could have prevented this outcome. The model developer could have improved the system's reliability and user-alignment[4], so that it wouldn't have made this

[1] See Section 2 for elaboration on this definition.

[2] This is in contrast to earlier generations of agentic AI systems, which did not explicitly reason through language, such as the Deep Blue chess playing program from IBM that defeated Garry Kasparov [2].

[3] In this context, a failure is when the agent fails to achieve some objective or does so in an unsatisfactory or harmful manner; a vulnerability is when the agent can be co-opted or undermined by an attacker, and an abuse is when an agent is used for harmful purposes.

[4] In this paper, we will refer to user-alignment as the propensity of an AI model or system to follow the goals specified by a user.

mistake. The system deployer could have disabled the agent from taking action without explicit approval. The user could have simply never agreed to delegate purchasing authority to an AI system that was commonly known to not be fully reliable. The airline company could have even instituted policies or technologies that required affirmative human consent for purchases. Given that multiple parties could have taken steps to mitigate the damages, every party can arguably cast blame on the other, and in the worst case a party can be held responsible even when they could not have reasonably prevented the outcome[4, 5].

A key goal of allocating accountability for harms from agentic AI systems should be to create incentives to reduce the likelihood and severity of such harms as efficiently as possible [6]. In order to make sure that *somcone* is incentivized to take the necessary measures, it is important that at least one human entity [5] is accountable for every uncompensated direct harm caused by an agentic AI system. Other scholarship has proposed more radical or bespoke methods for achieving accountability, such as legal personhood for agents coupled with mandatory insurance [7, 8], or targeted regulatory regimes [9]. These all appear to address the same problem: in order to create incentives to reduce or eliminate harms from agentic AI systems, society needs to agree on baseline best practices [6] that prudent model developers, system deployers, and users are expected to follow. Given such a baseline, when an agentic AI system causes harm, we can identify which parties deviated from these best practices in a way that failed to prevent the harm.

In this white paper, we lay out several practices that different actors can implement to mitigate the risk of harm from agentic AI systems, which could serve as building blocks for a set of agreed baseline best practices. We also highlight the many areas where operationalizing these practices may be difficult, especially where there could be tradeoffs among safety, usability, privacy, and cost. AI developers cannot answer these questions alone, nor should they, and we are eager for further research and guidance from the wider world.

In Section 2, we define agentic AI systems and the human parties in the agentic AI life-cycle. In Section 3, we briefly describe the potential benefits of agentic systems. In Section 4, we provide an initial seven practices that could be part of a set of agreed best practices for parties in the agent life-cycle and highlight open questions. Finally, in Section 5, we consider more indirect impacts from the introduction of AI agents that may not be addressable by a focus on individual harms.

We hope that the best practices we outline can serve as building blocks for a society-wide discussion about how to best structure accountability for risks from agentic AI systems. For example, they may inform discussion around what regulation of AI agent development might look like, or how parties structure contracts regarding agents (e.g. insurance for harms caused by agents, terms of use regarding agents), or how courts could think of various actors' standards of care. Given the nascent state of agents and their associated scholarship, we do not yet have strong recommendations on how accountability ought to be structured, and would like to see a more robust public discussion of possible options. We hope that this paper will help catalyze such conversations, without anchoring or biasing them too strongly in any particular direction.

[5] That is, an individual, corporation, or other legal entity, but not (solely) an AI system itself.

[6] We refer to baseline best practices here rather than, e.g., the legal concept of a professional "standard of care"the set of actions a reasonable and prudent party is expected to take, such that deviating from this standard opens them up to legal responsibility from the resulting harmsince the former could provide a foundation for the latter, and may also inform policymaking outside of courtrooms (e.g., through legislation and regulation).

2 Definitions

2.1 Agenticness, Agentic AI Systems, and "Agents"

Agentic AI systems are characterized by the ability to take actions which consistently contribute towards achieving goals over an extended period of time, without their behavior having been specified in advance. In the cultural imagination, an AI agent is a helper that accomplishes arbitrary tasks for its user, like Samantha from Her or HAL 9000 from 2001: A Space Odyssey. Such agents are very different from current AI systems like GPT-4, which, while surprisingly knowledgeable and clever in some ways, can thus far only complete a limited range of real-world tasks. Yet there is no clear line along which to draw a binary distinction between "agents" and current AI systems like GPT-4. Instead, an AI system's agenticness is best understood as involving multiple dimensions, along each of which we expect the field to continue to progress.

We define the *degree of* ***agenticness*** in a system as *"the degree to which a system can adaptably achieve complex goals in complex environments with limited direct supervision."* Agenticness as defined here thus breaks down into several components: [7]

- **Goal complexity**: How challenging would the AI system's goals[8] be for a human to achieve and how wide of a range of goals could the system achieve? Properties of the goal may include target levels of reliability, speed, and safety.
  - Example: An AI system that can correctly answer users' analytical questions across programming and law would have greater goal-complexity than a text classifier that can only classify the same inputs as belonging to law or programming.
- **Environmental complexity**: How complex are the environments under which a system can achieve the goal? (E.g., to what extent are they cross-domain, multi-stakeholder, require operating over long time-horizons, and/or involve the use of multiple external tools.)
  - Example: An AI system that can play any board game expertly has greater environment-complexity than an AI system that can only play chess, because the first system can succeed under a far greater range of environments (including chess) than the second.
- **Adaptability**: How well can the system adapt and react to novel or unexpected circumstances?
  - Example: Automated rule-based customer-service systems have lower adaptability than human customer-service representatives, since humans can address unexpected or unprecedented customer requests.
- **Independent execution**: To what extent can the system reliably achieve its goals with limited human intervention or supervision?
  - Example: Cars capable of level 3 autonomous driving [10], which can operate without human intervention under certain circumstances, have greater independent execution than traditional cars that require continuous human operation.

[7] We recognize that a variety of definitions of agenticness, agents, and agency are used by various people for various purposes. In our assessment many or all of the practices we discuss in the remainder of the paper are applicable across many alternative definitions of these terms.

[8] We will assume that an agentic AI system can be modeled as having goals, including externally-defined goals such as following a set of provided instructions.

Following recent literature [3], we will generally refer to systems exhibiting high degrees of agenticness as "agentic AI systems," to emphasize that agenticness as we use it here is a property rather than a category/classification, though we will sometimes use "agents" as it is the prevailing term of art in some contexts. This work will focus on the range of effects and best practices that may become relevant as systems' agenticness increases.[9] We emphasize that agenticness is a distinct concept from consciousness, moral patienthood, or self-motivation, and distinguish a system's degree of agenticness from its anthropomorphism.[10] Indeed, we will generally conceptualize agentic AI systems as operating in pursuit of goals defined by humans and in environments determined by humans (and often in cooperation with human "teammates"), rather than fully-autonomous systems that set their own goals. Agenticness as we define it is also not tied to physicality (i.e., many digital systems are more agentic in the sense above than most robots), but certain kinds of "independent execution" that have physical consequences (e.g. in a driverless car) can increase the risks and opportunities of agenticness in particular applications. Lastly, agenticness is conceptually distinct from an AI system's level of performance on a given task or the generality of its capabilities, though improvements in performance and generality may "unlock" the ability of a system to act as an agent in certain contexts [12].

2.2 The Human Parties in the AI Agent Life-cycle

We provide a simplified overview of the agentic AI life-cycle, though there are many different configurations of these roles in the AI industry [13] and we hope further taxonomies emerge. In our taxonomy,[11] the three primary parties that may influence an AI agent's operations are the model developer, the system deployer, and the user. The **model developer** is the party that develops the AI model that powers the agentic system, and thus broadly sets the capabilities and behaviors according to which the larger system operates. The **system deployer** is the party that builds and operates the larger system built on top of a model, including by making calls to the developed model (such as by providing a "system prompt"[14]), routing those calls to tools with which the agent can take actions, and providing users an interface through which they interact with the agent. The system deployer may also tailor the AI system to a specific use case, and thus may frequently have more domain-specific knowledge than the model developer or even the user. Finally, the agent's **user** is the party that employs the specific instance of the agentic AI system, by initiating it and providing it with the instance-specific goals it should pursue. The user may be able to most directly oversee certain behaviors of the agentic system through its operation, during which it can also interact with third parties (e.g. other humans, or the providers of APIs with which the agent can interact).

Sometimes, the same entity will fulfill multiple roles, such as the same company both developing a model and then deploying it via an API (making them both the model developer and one of the system deployers). Other times, multiple entities will share a role, such as when one company trains a model and a second company fine-tunes it for their application, making them share the

[9] For example, LLMs are being augmented with tools/scaffolding to increase their scores on the dimensions of agenticness, including "chain-of-thought" to help with strategic reasoning, "code execution" to help with independent execution, and "browsing" to help with adaptability, etc.)[11].

[10] Agenticness does not imply or require a human-like appearance or human-like behavior, though anthropomorphic appearances and behavior may increase the likelihood of humans perceiving such systems as agentic and have other implications for responsible design and deployment.

[11] We use this taxonomy as a useful mental model for enabling division of practices across the agent lifecycle and to better highlight open questions. These are not intended to establish a prescriptive framework for allocation of responsibility. Such responsibility may vary depending on the context. For example, for an agent that performs medical diagnoses, if the agent is deployed in a hospital more responsibility may fall on the user (a doctor), whereas if the agent is a consumer app marketed as a personal diagnostic tool, more perhaps more responsibility should fall on the system deployer (the app developer).

responsibilities of a "model developer."[12] We will also occasionally mention other relevant actors, including the **compute provider** (which operates the chips and other infrastructure on which agentic AI systems run) and **third-parties** which interact with the user-initiated AI system.

We illustrate with the specific example of a scheduling assistant built on OpenAI's Assistants API. OpenAI developed the GPT-4 model, making it the model developer.[13] OpenAI deployed the infrastructure (including serving the model and connecting it to tools such as a code execution environment), and the application developer builds an app on top of it (e.g., by building a user-interface, choosing a system prompt, and supplying an email template for the system to use when sending invites), meaning they both share the role of system deployer. Finally, a customer initiates a session with the scheduling assistant and specifies which goals (e.g. scheduling requirements) they'd like the system to satisfy, making them the user.

3 Potential Benefits of Agentic AI Systems

In this section, we take stock of the ways that agentic AI systems have the potential to benefit society. First, we consider the ways that a more agentic version of a particular AI system might be more beneficial than a less agentic version (agenticness as a helpful property). Second, we consider the ways in which agenticness can enable wider diffusion of AI in beneficial applications in society, and is often implicit in many definitions of and visions for AI (agenticness as an impact multiplier). While our discussion in this section is brief, this should not be read as an indication that the list of possible benefits is necessarily short, or that the magnitude of those benefits is small. Nor do we make claims that the benefits clearly outweigh the risks or vice versa.

3.1 Agenticness as a Helpful Property

Specific AI systems may in many cases be more beneficial in proportion to the extent to which they are agentic, provided they are designed safely and that appropriate best practices for safety and accountability are applied. Agenticness can make a particular system more beneficial in ways such as the following:

- *Higher quality and more reliable outputs*: for example, a language model that is capable of browsing the Internet autonomously, and revising its queries in response to the results it receives, may be capable of providing much more accurate answers to questions than a system that is not able to do so. This may be particularly true in instances involving topics that are dynamic in nature or events that occurred after the underlying model was trained.

- *More efficient use of users' time*: for example, if a user provides high level instructions to an AI system regarding code they want the system to produce, it may be smoother for the user if the system performs several steps autonomouslye.g. translating the instructions into code, running the code, displaying the results, assessing those results, and making edits to the code in order to improve outcomes.

- *Improved user preference solicitation*: For example, personal assistant AI that is capable of interactively sending messages to its users in order to ask clarifying questions in natural language, and that does so at strategically appropriate times, may provide a better experience

[12] The important question of how to split the responsibility for different best practices across the multiple entities that may share a single agent-life-cycle role is beyond the scope of this current whitepaper.

[13] If the application developer fine-tuned the model on their custom data, they may share the "model developer" responsibilities.

than an app with numerous complex configurations that is difficult for users to leverage effectively.

- *Scalability*: An agentic AI system may allow a single user to take many more actions than they could otherwise, or be capable of benefiting a much larger number of people than a less agentic version of the same system. Consider the example of radiology. A non-agentic radiology image classification tool may be helpful for making a radiologist slightly more efficient, but an agentic radiology tool that was capable of completing certain patient-care tasks without human supervision (e.g. compiling reports on the scan, asking patients basic follow-up questions) could potentially increase a radiologist's efficiency substantially and leave more time for seeing many more patients.[15].

3.2 Agenticness as an Impact Multiplier

In addition to analyzing the implications of agenticness in the context of particular AI systems, one can also view agenticness as a prerequisite for some of the wider systemic impacts that many expect from the diffusion of AIsome of which have significant potential to benefit society. Insofar as agenticness is a definitional or practical prerequisite for that diffusion, the impacts of agenticness may be closely related to the impacts of AI more generally. In this sense, the impacts of AI generally are likely to be more frequent and more pronounced, and to happen sooner, to the extent that agenticness increases, making agenticness an "impact multiplier" of the field of AI as a whole.

Sometimes agenticness is implicitly assumed when people talk about current or future AI capabilities. OpenAI's Charter defines artificial general intelligence (AGI) as "highly autonomous systems that outperform humans at most economically valuable work," and canonical textbooks such as Russell and Norvig's Artificial Intelligence: A Modern Approach emphasize agenticness in their conception of AI. Given these considerations, we briefly review several commonly expected impacts of AI as an overall technological field.

Even without significant further advances in agenticness, AI is likely to already constitute a general-purpose technology. Historically, the widespread adoption of general purpose technologies such as the steam engine and electricity has vastly increased the global standard of living over time (though also brought about significant harm for many, and in particular for less powerful or privileged groups, living through those periods). Highly capable and agentic AI systems that are widely deployed could even improve economic productivity so much that they fundamentally change the nature of work, potentially and perhaps more speculatively enabling a "leisure society" or "post-work" world, though this is by no means guaranteed and would carry risks [16]. Additionally, AI could accelerate progress on various non-economic measures of societal wellbeing, such as those encapsulated in the Sustainable Development Goals, and by accelerating scientific progress and understanding. The economic and other productivity gains some expect from AI may be greater to the extent that agentic AI systems are able to take actions autonomously [17].

4 Practices for Keeping Agentic AI Systems Safe and Accountable

Below, we suggest a range of practices different parties can adopt to ensure agentic AI systems operate safely and in accordance with users' intents, and to create accountability when harm does occur. When implemented together, the practices outlined in this section are intended to provide a "defense-in-depth" approach to mitigating risks from agentic AI systems. Though many of these practices are employed in some form today, we highlight many open questions around how they should be operationalized. We also discuss how additional precautions may be needed as AI systems

become more agentic. We emphasize that these practices alone are insufficient for fully mitigating the risks from present day AI systems, let alone mitigating catastrophic risks from advanced AI. For example, none of the principles below covers methods for ensuring the cybersecurity of agents so as to prevent them from being hijacked by attackers, even though we expect this to be a significant challenge that requires new practices. The practices discussed here are intended as an initial outline of approaches and relevant considerations.

We avoid discussion of what technical best practices to use in order to build capable and user-aligned agentic AI systems. These are both rapidly evolving fields, and practices are changing rapidly, such that we do not expect the fields to converge on "best practices" for guaranteeing particular AI capabilities or user-alignment in the near term. In addition, the science required to predict the capabilities/user-alignment of an AI model given training choices is in its infancy [18]. This means that it is currently not possible for a model developer to deterministically guarantee a model's expected behavior to downstream system deployers and users. There are exceptions, such as how fully excluding a training sample from the training data will mean that the model cannot regurgitate it. Still, given the limited degree to which model behavior can be delimited in advance, we will focus on designing a set of best practices that is agnostic to the particular model's method of training.

Open Question:

- What harm mitigations, if any, are primarily attainable via technical choices in the model's training process? What might corresponding best practices be?

4.1 Evaluating Suitability for the Task

Either the system deployer or the user should thoroughly assess whether or not a given AI model and associated agentic AI system is appropriate for their desired use case: whether it can execute the intended task reliably across the range of expected deployment conditions (or, to the extent reliability is not necessary or expected given the low stakes of the task and the nature of the user interface, that user expectations are suitably established via that interface). This raises the question of how to properly evaluate an agentic AI system, and what failure modes can and cannot be foreseen by sufficient testing.

The field of agentic AI system evaluation is nascent, with more questions than answers, so we offer only a few observations. Evaluating agentic AI systems raises new challenges on top of the already significant challenges with evaluating current language models [19]. This is in part because successful agents may often need to execute long sequences of correct actions, so that even if individual actions would only fail infrequently, these rare events could compound and make failure in deployment likely. One solution is for system deployers to independently test the agent's reliability in executing each subtask. For example, when an early system deployer was building an AWS troubleshooting agent on top of OpenAI's GPT-4 API, they broke down the agent's needed subtasks into "information gathering," "calculations," and "reasoning," and created evaluations for each independently. Breaking down all the subtasks that could be encountered in a complex real-world operating domain may sometimes be too difficult for system deployers; one approach could be to prioritize doing such evaluations for agents' use of high-risk actions, like financial transactions.

Even if the system is shown to do individual subtasks reliably, this still raises the problem of how to evaluate whether the agent will reliably chain these actions together. Finally, agentic systems may be expected to succeed under a wide range of conditions, but the real world contains a long tail of tasks which are difficult to define and events which are hard to anticipate in advance (including those that emerge from human-agent or agent-agent interactions). Similar difficulties with evaluating reliability under unanticipated conditions have significantly slowed the deployment of self-driving

cars [20], and one might expect a similar effect for agentic AI systems. Ultimately, there are currently few better solutions than to evaluate the agent end-to-end in conditions (whether simulated or real) as close as possible to those of the deployment environment.

So long as our ability to bound and evaluate the behaviors of agentic AI systems remains immature, system deployers and users may need to lean more heavily on other practices (such as human-approval for high-stakes actions) in order to bound the behavior of these systems.

A separate evaluation challenge for model developers and system deployers is how to determine what scale of harm their agentic system could enable, whether by a user intentionally, or by accident due to failures of user-alignment. For example, frontier model developers could test their models for capabilities that would facilitate harm such as generating individualized propaganda or assisting in cyberattacks.[14] It may be important to require system deployers (or model developers operating on their behalf) to do such evaluations in order to determine what other measures they should take to mitigate misuse of the agentic AI system services they provide. Such guidance is currently under development by the US government [21] and the international community [22].

Open Questions:

- How can system deployers and users effectively evaluate the agentic system's level of reliability in their use case? What constitutes "sufficient" evaluation?
- How can system deployers effectively evaluate the combination of agent and user, and identify behaviors and potential failures that only emerge through human-agent interaction?
- Given the heterogeneous nature of real-world deployment, what failure modes cannot be expected to be detected in advance via evaluation?
- What evaluations of agents' capabilities should be expected to be done by the model developer, rather than the system deployer? (E.g. universally useful checks, such as the system's propensity to act in alignment with the user's goals.)
- How can system deployers communicate to the user the intended conditions under which the agentic system can be used reliably, and at what point does a user's unintended usage of a system make them responsible for resulting harms?
- What misusable agentic system capabilities should model developers and system deployers be obligated to test for, both for specific sectors and for agents in general?

4.2 Constraining the Action-Space and Requiring Approval

Some decisions may be too important for users to delegate to agents, if there is even a small chance that they're done wrong (such as independently initiating an irreversible large financial transaction). Requiring a user to proactively authorize these actions, thus keeping a "human-in-the-loop" [23], is a standard way to limit egregious failures of agentic AI systems.[15] This raises the key challenge of how a system deployer should ensure that the user has enough context to sufficiently understand the implications of the action they're approving. This is also made harder when the user must approve many decisions and thus must make each approval quickly, reducing their ability to meaningfully consider each one [24].

[14] OpenAI has committed to testing for these and other model capabilities as part of its Preparedness work.

[15] As noted by Crootof et al, [23], a human-in-the-loop may serve various roles beyond simply improving the reliability of the human-machine system (e.g., assigning liability, preserving human dignity).

In some cases, agentic AI systems should be prevented from taking certain actions entirely, to better bound the system's operational envelope and thus enable safe operation [25]. For instance, it may be prudent to prevent agentic AI systems from controlling weapons. Similarly, to mitigate accidents resulting from agents running for unintentionally long periods without supervision, system deployers can cause agents to periodically "time out" until a human reviews and reauthorizes them. Of course, the risk-mitigation benefits of these measures must be carefully balanced with the reduction in agentic AI systems' capabilities and utility. Such operational limitations could be placed by the model developer at the model-level, by the system deployer via hardcoding or a monitoring system (Section 4.5), and/or proactively set by the user themselves.

As the agenticness of AI systems increases, hard-coded restrictions may cease to be as effective, especially if a given AI system was not trained to follow these restrictions, and thus may seek to achieve its goals by having the disallowed actions occur. An AI agent could circumvent a hard-coded restriction by causing another party to take the action on the system's behalf, while hiding the resulting potential impact from the user. For instance, an agent could send an email—an allowed action—to a non-user human that convinces said human to take the disallowed action. System deployers can bound this problem by ensuring strong network controls to limit agents' interactions with the outside world except through monitored channels. Agentic systems could also be sandboxed in order to prevent systems with cybersecurity capabilities from escaping these controls (especially during development when a system's capabilities are uncertain), but current sandboxing systems may not be well-suited to effectively bound highly-capable AI agents.

Open Questions:

- How should a user or system deployer determine and enforce which interactions need human approval, as a function of a model's empirical reliability on that task and other factors? Are there certain interactions, such as formation of legal contracts, where a user's approval should always be required?
- What design practices should system deployers follow to ensure users have sufficient context before approving an action?
- What are the best practices for users reviewing approvals for high-cost actions (such as minimum review times) to avoid their turning into a "rubber stamp" for agentic AI systems that cannot catch harmful actions?
- What are the best practices for system deployers to test that agentic AI systems follow the action constraints the system deployer specified?
- How can the effectiveness and meaningfulness of human oversight be evaluated?
- How can existing sandboxing mechanisms be modified to work for agentic AI systems? How can model developers or system deployers test the reliability of a sandbox for a given agentic system without incurring the actual risk of the agentic system breaking that sandbox, or a wider nested sandbox test environment?

4.3 Setting Agents' Default Behaviors

Model developers could significantly reduce the likelihood of the agentic system causing accidental harm by proactively shaping the models' default behavior according to certain design principles. For instance, user interactions with agentic AI systems may be designed to begin with a prompt to the

user to communicate their goals and preferences to the system. This preference information will almost always be unclear or incomplete: users don't want to have to tell their life story just to get help baking a cake. It is still valuable for the agent to have a set of default common-sense background preferences that allow it to "fill in the gaps" without a user's guidance, such as "users prefer if I don't spend their money." In the absence of user-specific information, one common-sense heuristic could be to err toward actions that are the least disruptive ones possible, while still achieving the agent's goal [26]. It should often still be possible for the user to overrule these default preferences if requested specifically, though it may also be important to have agents themselves refuse to execute user-intended harm (Section 4.2).

To avoid agentic systems being overconfident about users' objectives, model developers and system deployers may be advised to build in features that cause agents to be aware of their own uncertainty about users' intended goals [27]. Agents can be trained or prompted to proactively request clarifications from the user to resolve this uncertainty, especially when it may change their actions [28, 29]. However, better understanding of users alone does not guarantee the agent will pursue the right objectives. For example, instead of producing truthful outputs with which the user may disagree, certain AI systems have been found to pander to users based on what beliefs they think a given user holds [30, 31], which may reflect a deficiency of current techniques to align AI systems with their user's true goals. Having agents request information too frequently can also raise issues with usability and privacy (if the preference information is sensitive).

Open Questions:

- What other default behaviors could model developers and system deployers instill in agentic AI systems that could mitigate the possibility of errors and harms?
- How should these default behaviors be balanced, when in conflict?
- How is responsibility allocated between the model developer (who may not have intended for their model to be used in a particular agentic system) and the system deployer, when it comes to instilling certain behaviors in AI systems?

4.4 Legibility of Agent Activity

The more a user is aware of the actions and internal reasoning of their agents, the easier it can be for them to notice that something has gone wrong and intervene, either during operation or after the fact.

Revealing an agent's "thought process" to the user enables them to spot errors (including identifying when a system is pursuing the wrong goal), allows for subsequent debugging, and instills trust when deserved. Conveniently, current language model-based agentic systems can produce a trace of their reasoning in natural language (a so-called "chain-of-thought"[32]), which provides a convenient source of truth for how the system reached a conclusion on which action to take. It could be useful for system deployers to expose all details of the agents' interactions, such as any inputs it receives from tool-use API calls or interactions with other agents. This could have the added benefit of enabling users to detect when a malicious third party (such as a third-party agent) is attempting to manipulate the primary agent's operations [33].

However, "chain-of-thought" transparency comes with challenges and cannot yet be fully relied on. Early work has shown that sometimes models do not actually rely on their chains-of-thought when reasoning [34], so relying on these may create a false sense of security in the user. An alternative approach could instead be to expose different reasoning-related data to users to help them understand

models' behavior, such as probes of models' internal representations [35, 36]. (Such methods are still nascent.)

A separate challenge is that chains-of-thought are growing longer and more complicated, as agents produce thousands of words per action or are integrated into more complex architectures (e.g. "Trees-of-Thought"[37] or "Graphs-of-Thought"[38]). The length and complexity of these "internal monologues" may balloon beyond a user's ability to feasibly keep up. (One potential solution may be to leverage AI-based monitoring, as discussed in Section 4.5.) Other challenges arise when agents integrate new modalities like images, which may contain hidden messages that a human user cannot parse [39]. Future AI systems may also perform better if they can reason in compressed, more efficient ways, which may not necessarily be readable to humans, and these inscrutable architectures may bring significant safety and accountability costs. It is unfortunately not possible to simply "ask" the agent to retroactively justify its behavior, as this is likely to produce confabulated reasoning [40].

It may be particularly important that system deployers provide the user with a ledger of actions taken by the agent. This is a lighter-touch method than requiring human approval (Section 4.2), and can thereby give users visibility into a wider range of actions without substantially slowing down the system's operation. Actions that can only be reviewed after the fact should be more easily reversible than those that require approval. Much like when requiring action approvals, it may be important to also provide an agent's reasoning to the user to properly contextualize the action.

Open Questions:

- Should system deployers be required to provide their agents' "internal monologue" to the user?
- Should all AI agent reasoning, including inter-agent communication, be required to occur in natural language, in order to make subsequent auditing by humans easier?
- What methods can model developers and system deployers take in order to produce accounts of agents' thoughts that are more faithful to the logic the system truly employed?
- How can model developers make their models reason in ways that are more legible to the user or system deployer?
- What user-experience practices should system deployers follow in order for users to be able to easily understand and act on agents' internal reasoning?
- How can users easily identify the actions that are most likely to be mistaken or ill-informed, among many actions taken by their agent?

4.5 Automatic Monitoring

In practice, human users may not always have the time to go through the agent activity logs exposed by the system deployer at the speed or scale they desire. To address this, users or system deployers can set up a second "monitoring" AI system that automatically reviews the primary agentic system's reasoning and actions (made legible as in Section 4.4) to check that they're in line with expectations given the user's goals. This monitoring AI system could be a classifier, or a generative AI system capable of producing its own chains-of-thought [41]. Such automated monitors operate at a speed and cost that human monitoring cannot hope to match, and may be able to parse modalities (such as detecting adversarially-perturbed images) that a human could not. Monitoring can be provided as a service by the system deployer, or set up by the user in case they wish to exercise additional control.

Monitoring, however, comes with substantial costs and can be a double-edged sword. If sensitive agent session data needs to be retained by the system deployer, it may create privacy risks for the user (and by extension the system deployer). Depending on which party carries out monitoring, even monitoring intended solely to benefit the user could give rise to concerns about how that information will be used, and push users to self-censor [42, 43]. A separate challenge is how to prevent monitoring functions from being over-extended and being used as a means of centralized control, especially by authoritarian governments [44].

Additionally, monitoring poses some practical and technical challenges. One challenge of such monitoring is that it adds costs: if the same foundation model (prompted differently) is used to classify all text outputted by the primary model, the total inference cost may be doubled, or multiplied even further if the monitor model produces its own chain-of-thought for each primary model output. Smaller AI models may be useful for doing monitoring more cheaply, but doing so raises the risk that the primary AI system could misbehave in ways the smaller model can't reliably catch. (On the other hand, recognizing incorrect behavior may be easier than generating full behaviors that are correct, making savings possible.) This may create pressure for system deployers in competitive markets to minimize monitoring measures in order to undercut each other on price.

There is also a recursive issue with ensuring the reliability and user-alignment of monitor systems: if an input causes the primary agent to malfunction, it may cause the monitoring system to malfunction, too. For example, a malicious prompt injection that hijacks the primary system, when read by the monitoring system, could similarly apply to and hijack the monitoring system and prevent it from reporting the incident (meaning that the harm will go entirely undetected) [45].

While the ultimate value of automated monitoring remains uncertain, given the potential utility of AIs-monitoring-AIs as a means of overseeing future high-risk AI systems, it may be important to investigate methods that allow us to carry out monitoring effectively while appropriately balancing concerns related to privacy and centralized control.

Open Questions:

- How can system deployers verify that a monitor is capable of successfully rooting out mistakes from the primary AI system, particularly when they may have been trained in a similar manner, or when the agent is substantially more intelligent than the monitor?
- What types of human oversight of a primary AI system may still be useful when AI-based monitoring is also in use?
- What key behaviors should monitoring systems monitor for that would otherwise break implicit safety practices, such as agents undermining a user's "blank slate" assumption by storing information somewhere on the internet so that they can retrieve it in a future session?
- How can automatic monitoring systems be made to monitor for as-yet-unknown harms?

4.6 Attributability

In cases where preventing intentional or unintentional harms at the level of the user or system deployer is infeasible (such as a criminal operating an AI agent to scam a third party), it may still be possible to deter harm by making it likely that the user would have it traced back to them. With the creation of reliable attribution, it could become possible to have reliable accountability. One idea for such a system of attribution is to have each agentic AI instance assigned a unique identifier, similar to business registrations, which contains information on the agent's user-principal

and other key accountability information.[16] [17] It may be valuable to keep such agent identification optional and allow anonymity in many circumstances, so as to limit potentially harmful surveillance of AI usage. But in high-stakes interactions, such as those involving private data or financial transactions, third parties (including external tool providers) interacting with a user's agent could demand such identification before starting the interaction, to ensure they know a human user can be held accountable if something goes wrong. Given the substantial incentives for bad actors to spoof such a system (similar to the pressures that exist for identity-verification protocols in the financial industry[47]), making this system robust may be an important challenge.

Such attribution for individual interactions does not cover everything: in some cases AI agents may be used to cause harm to individuals who never had a chance to identify them (e.g. agents assisting a hacker in developing an exploit), for which alternative accountability approaches may be needed.

Open Questions:

- How can society practically enable AI agent identity verification? What existing systems, such as internet certificate authorities, can be adapted to facilitate such verification?
- What other ideas exist for practically enabling agentic AI system attributability?

4.7 Interruptibility and Maintaining Control

Interruptibility (the ability to "turn an agent off"), while crude, is a critical backstop for preventing an AI system from causing accidental or intentional harm. System deployers could be required to make sure that a user can always activate a graceful shutdown procedure for its agent at any time: both for halting a specific category of actions (revoking access to, e.g., financial credentials) and for terminating the agent's operation more generally. This graceful fallback is also useful in the event that agents crash, such as due to internet outages. There may be some cases where shutting a system down may cause more harm than good (e.g. a malfunctioning agent that is nonetheless assisting in a life-threatening emergency), but by investing in interruptibility, one can minimize the scenarios in which users are stuck with only poor options.

Ensuring graceful interruptibility is challenging when agents are terminated mid-action-sequence (e.g., while scheduling a five-person business meeting, when only two invites had been sent so far). An important principle for addressing this could be to have agents always pre-construct a fallback procedure if they're turned off (e.g., pre-launching a script that would automatically notify the two invitees that the agent has terminated and thus the meeting may not occur). A significant challenge is how to maintain such fallback procedures as agents' action-sequence complexities increase. In certain circumstances it may even be that any graceful fallback procedure would itself require significant agentic behavior (though perhaps by a separate AI agent).

A second important principle could be that an agent should not be able to halt or tamper with the user's attempt to shut them down, as might be the case if the agent is malfunctioning, or if the agent or its surrounding system has an instrumental goal of self-preservation [30]. It may be important for model developers or system deployers to deliberately design agentic systems to place "shut down gracefully when requested by the user" as a primary goal, above whatever other goals

[16] For example, it could reference the AI model powering the agent, and any certifications it has received [46]. It could even include information about the datacenter and chip powering the agent, for purposes of interruptibility (Section 4.7)., if there were a way for the user to be able to verify this information (e.g., by way of datacenters signing agent outputs, or even hardware-level signing).

[17] Similarly, if an AI system is associated with a bank account into which it deposits (or extracts) funds, authorities could track access to that account as a basis for identifying human principals.

the system was provided with, though this may be infeasible in certain situations (such as those in which an agent is assisting in a life-threatening emergency). Interruptibility requirements should likely also extend recursively to any other sub-aents the agent has initiated: all spawned sub-aents should be gracefully terminatable by the original user.

Sometimes, a user may be unwilling or unable to shut down their AI system, even as it causes harm. This raises important challenges around how third parties communicate to a user that their agent is causing harm, and in what circumstances an external party has the right or obligation to terminate a user's agent. In cases where it is merited, to be able to stop an agent quickly during an incident, society could encourage redundancy in the number of human parties that can turn off an AI agent instance. The two relevant parties are the system deployer and the data center operator or chip owner on whose hardware the AI system is running. If an agentic AI system causes significant ongoing harm that they could have halted, these parties could themselves bear some of the responsibility. In order for such shutdowns to be viable, the system deployer or chip operator may need to maintain awareness of roughly what agentic AI jobs they are running, though this must be done with significant care to avoid harms to privacy. It may even be desirable to automatically trigger such shutdowns if risk indicators cross a certain threshold (like an influx of new jobs from unknown accounts), similar to stock market circuit breakers that are triggered at a given threshold drop in price.

As AI systems' levels of agenticness increase, there is a risk that certain model developers, system deployers, and users would lose the ability to shut down their agentic AI systems. This could be because no viable fallback system exists (e.g., in a similar sense that no one can "shut down" the global banking system or the electric grid without very significant costs), or because the agent has self-exfiltrated its code to facilities beyond its initiator's grasp. We can begin to take steps that make this worst case scenario less likely, by establishing the degree to which model developers, system deployers, and users will be held accountable for the harms caused by the agent even after human control has been lost. This could incentivize them to develop stronger methods of control, making the worst case scenario less likely.

Open Questions:

- How can model developers and system deployers design their systems to ensure that agentic systems have graceful fallbacks in case they're shut down or interrupted, for the broad range of actions an agent might take? Are there principles by which a second agentic AI system could be used as the fallback, and where might this approach fail?
- In what settings is interruptibility users' responsibility, rather than model developers' or system deployers'? For instance, should users be considered responsible for only approving an agent's action if it is coupled with a fallback procedure?
- How can system deployers ensure that agents only spawn sub-aents that can be similarly turned off?
- Under what circumstances, if any, should an agent ever be able to (or be incentivized to) prevent its own termination?
- What information should system deployers or compute providers keep track of (such as agent IDs, as in Section 4.6), in order to help determine that a system they're hosting has caused significant harm, and needs to be turned off? How can such information be minimized to satisfy the strong need for user privacy?

- What restrictions should exist on such shutdowns, to prevent them from being abused to police harmless or low-stakes usage of agents?
- How realistic is it for agentic AI systems to resist being shut down in the near-term? How realistic is it for an agentic AI system to be integrated into a social process or critical infrastructure (including unintentionally) such that the cost of shutting it down would become prohibitive? If either scenario did happen, what are the likeliest pathways, and what signals might be observed in the run-up (by the system deployer and user, or by outside parties) that can be used to trigger intervention ahead of time?
- How should different parties' responsibilities be allocated in the event of the non-interruptibility of an AI system that causes harm?

5 Indirect Impacts from Agentic AI Systems

In addition to direct impacts from individual agentic AI systems, there will be indirect impacts that result collectively from the usage of many different AI systems and society's reaction to their usage [48]. Just as it would have been difficult to anticipate the full range of societal readjustments from previous general-purpose technologies like electricity and computers, one should "expect the unexpected." Still, we do think there are several categories of indirect impacts from agentic AI systems that are likely to require active mitigation by society, which we list below.

These indirect impacts may be addressed at least in part by adopting best practices for users, system deployers, and model developers, such as those outlined in Section 4. However, fully addressing these complex challenges will likely require additional strategies beyond this paper's proposals, including through industry-wide collaborations and society-wide mitigations. Some strategies towards this end may be domain or risk-specific, while others may involve placing general requirements on the usage of certain types of agentic AI systems.

5.1 Adoption Races

Given the advantages that agents may confer in competitive environments, such as competition between private firms or governments [49, 50, 51, 52], there may be significant pressure for competitors to adopt agentic AI systems without properly vetting those systems' reliability and trustworthiness [53]. A key observation driving such premature reliance is that agentic AI systems may succeed at a task on average, while being unreliable in rare but important cases which can be missed or ignored by competitors under pressure.

For example, consider a hypothetical class of agentic AI code-generation systems that can rapidly write new code, but whose code occasionally contains serious security flaws. If a software development company thinks their competitor has been using these coding systems without human supervision as a way to quickly build new features, they may feel pressured to do the same without doing proper due diligence, as they might otherwise lose market-share to their competitor. As a result, all firms' codebases would now be vulnerable to serious cyberattacks, even if each individual firm would've preferred to go slower and thereby avoid this outcome [54]. This trend toward overrapid adoption, even in high-risk domains, can lead to over-reliance, whereby humans trust agentic AI systems without fully understanding their limitations. This could create the conditions for widespread use of unsafe AI systems that in the worst case may prove catastrophic [55, 56].

5.2 Labor Displacement and Differential Adoption Rates

Agentic AI systems appear likely to have a more substantive impact on workers, jobs, and productivity than static AI systems. Traditional AI systems excel at some routine work, but increasing agenticness could expand what tasks are "routine" enough to be assisted or automated by AI (such as by adapting to unexpected conditions, gathering relevant context, and calibrating to a user's preferences). This means they may expose a greater number of jobs and tasks to augmentation and automation, similar to other axes of AI system improvement like tool use[17]. This could result in a range of different economic effects. These could lead to substantive boosts in worker productivity and economic growth, but could also result in the displacement of a large number of workers, either because their jobs are fully automated or because their skills are made less rare and thus their jobs become more precarious. At the same time, agentic AI systems may improve education and enable workers to upskill into new jobs. It is also possible that agentic AI systems can increase the agency and productivity of individual workers or small firms more than traditional AI systems have done, such as by increasing the availability of previously rare expertise. This may or may not offset large firms' advantages in capital (e.g. their ability to run more agents) and preexisting market position (such as firms with access to proprietary data that can be used to train bespoke agents).

Even similarly-positioned individuals and firms may differ in their ability to leverage agentic AI systems. Different individuals' jobs and firms' business strategies may be more or less amenable to AI agent automation, depending on the particular order in which each AI agent capability is unlocked and becomes reliable. Individuals who lack digital literacy, technology access, or representation in design decisions around agentic AI systems may find themselves less able to participate in an agentic-AI-system-fueled world. However, AI agents could also reduce the technology access gap, much like smartphones increased internet access to underserved populations [57] (though some gaps remain[58]). All these effects may alter the job landscape and business environment unevenly, and increase the importance of taking active policy measures to ensure the benefits of increasingly agentic AI systems are in fact shared broadly.

5.3 Shifting Offense-Defense Balances

Some tasks may be more susceptible to automation by agentic AI systems than others. This asymmetry is likely to undermine many current implicit assumptions that undergird harm mitigation equilibria in our society (known as "offense-defense balances" [59]), with unclear consequences. For example, in the cyber domain, human monitoring and incident response is still key to cyber-attack mitigation. The feasibility of such human monitoring is predicated on the fact that the volume of attacks is similarly constrained by the number of human attackers. Consider the hypothetical where agentic AI systems can substantially automate cyber-attacker responsibilities and thus dramatically expand the volume of attacks, but cyber-defender responsibilities such as monitoring are much harder to automate. In such a world, the overall effect of agentic AI systems would be to make cyberdefense less viable and make information systems less secure. Conversely, if agentic AI systems make monitoring and response cheaper than producing new cyberattacks, the overall effect would be to make cyberdefense cheaper and easier.[18]

While it is very difficult to anticipate the net effect of agent adoption dynamics in a particular domain in advance, one can be confident that some processes will be much more amenable to automation than others, and that numerous societal equilibria may shift as a result. It behooves actors to pay close attention in identifying which equilibrium assumptions no longer hold, and to

[18] Any such offense-defense analysis should also include the extent to which agents themselves represent a new attack surface, and thus could create new vulnerabilities that need to be secured.

quickly respond, such as by investing in differential technological development towards defender-oriented technologies [60].

5.4 Correlated Failures

Agentic AI systems may bring unexpected failure modes, and a particular risk arises when a large number of AI systems all fail at the same time, or all fail in the same way. These correlated errors can occur due to "algorithmic monoculture": the observation that AI systems trained using the same or similar algorithms and data can make them malfunction in similar ways[61, 62]. There is already evidence that language models trained on similar data distributions suffer from similar vulnerabilities, such as adversarial prompts that corrupt one system generalizing to corrupting other similarly trained systems[63]. Similarly, biases in common training datasets, when used by many different model developers, could expand the biased behavior of individual AI systems into a society-wide harm (such as by all agents suppressing the same news article in recommendations, or reinforcing stereotyped representations against the same social group). More broadly, AI systems may be vulnerable to disruption in shared infrastructure (e.g. power or internet outages).

Such correlated failures may be more dangerous in agentic AI systems as they could be delegated more power by humans, and thus the potential consequences of their failure could be greater. They may also be exacerbated because agentic AI systems may shape each others' information environments and even directly communicate with each other, allowing for much more direct and even deliberate propagation of certain failures. It is particularly challenging to guard against such correlated failures because they are a joint function of the individual AI system and its constantly-changing environment. One initial path forward is to create visibility and monitoring in the agentic AI ecosystem, to catch such wide-scale issues as they emerge.

Correlated failures may be particularly hard to deal with because they may overtax the fallback systems intended to remedy individual agents' failures, but which are unprepared for large-scale failures. This may be especially acute in cases where the fallback plan is to have humans manually take over for each malfunctioning agent. For example, if a company's loan approval chatbot generally fails 1% of the time and has a small number of staffers to handle those failures, then a rare correlated failure that takes down 100% of the chatbots would bring the loan-approval system to a halt. However, the rarity of this risk may make it difficult to discern from routine operation alone, and thus could make it challenging for concerned employees inside the company to justify the cost of retaining adequate staff for such a seemingly hypothetical failure. In the longer term, if/as certain human tasks are entirely replaced by agentic AI systems, human expertise in certain domains may atrophy and make us entirely dependent on agentic AI systems (and their attendant failure modes). It may be particularly important for policymakers and the AI ecosystem to find ways to ensure that fallback mechanisms for agentic AI systems are robust to these sorts of correlated failures.

6 Conclusion

Increasingly agentic AI systems are on the horizon, and society may soon need to take significant measures to make sure they work safely and reliably, and to mitigate larger indirect risks associated with agent adoption. We hope that scholars and practitioners will work together to determine who should be responsible for using what practice, and how to make these practices reliable and affordable for a wide range of actors and affordable. Agreeing on such best practices is also unlikely to be a one-time effort. If there is continued rapid progress in AI capabilities, society may need to repeatedly reach agreement on new best practices for each more capable class of AI systems, in order to incentivize speedy adoption of new practices that address these systems' greater risks.

7 Acknowledgements

We would like to thank Seth Lazar, Tim Hwang, Chris Meserole, Gretchen Krueger, Rebecca Crootof, Dan Hendrycks, Gillian Hadfield, Rob Reich, Meredith Ringel Morris, Josh Albrecht, Matt Boulos, Laura Weidinger, Daniel Kokotajlo, Jason Kwon, Artemis Seaford, Michael Kolhede, Michael Lampe, Andrea Vallone, Christina Kim, Tejal Patwardhan, Davis Robertson, Hannah Rose Kirk, Ashyana-Jasmine Kachra, and Karthik Rangarajan for their helpful advice, feedback, and comments, which were integral in the development of this white paper.