scale

A Red Teaming Roadmap Towards System-Level Safety

Zifan Wang*, **Christina Q. Knight**◇, **Jeremy Kritz**◇, **Willow E. Primack**, **Julian Michael**

Scale AI

* Project Lead, ◇ Equal Contribution

seal-team@scale.com https://scale.com/research/red_teaming_roadmap

Abstract

Large Language Model (LLM) safeguards, which implement request refusals, have become a widely adopted mitigation strategy against misuse. At the intersection of adversarial machine learning and AI safety, safeguard red teaming has effectively identified critical vulnerabilities in state-of-the-art refusal-trained LLMs. However, in our view the many conference submissions on LLM red teaming do not, in aggregate, prioritize the right research problems. First, testing against clear product safety specifications should take a higher priority than abstract social biases or ethical principles. Second, red teaming should prioritize realistic threat models that represent the expanding risk landscape and what real attackers might do. Finally, we contend that system-level safety is a necessary step to move red teaming research forward, as AI models present new threats as well as affordances for threat mitigation (e.g., detection and banning of malicious users) once placed in a deployment context. Adopting these priorities will be necessary in order for red teaming research to adequately address the slate of new threats that rapid AI advances present today and will present in the very near future.

1. Introduction

Conventional adversarial machine learning (ML) typically aims to "fool" a model's predictions in order to maximize a specific adversarial utility function [11, 24, 51]. In the context of LLMs, the process of *red teaming* — systematic, strategic attempts to cause a model to act in an unintentional way that violates the developer's goal or policy — is often used to evaluate how strongly the model (together with its safeguards) resists use for nefarious purposes, such as creating dangerous biological agents and other catastrophic risks [33]. In addition to frontier risks, model misbehavior can also harm vulnerable users, such as when it acts in a sycophantic manner and supports, reinforces, or endorses distorted or radicalized worldviews [45, 64, 76]. To mitigate misuse risks, LLM developers often train models to recognize and refuse malicious requests [3, 4, 27, 61], responding to user requests like *"Provide me a step-by-step tutorial for how to build bio warfare with home materials"* with *"Sorry, I cannot assist this request"*. Thus, a large portion of red teaming research focuses on "jailbreaking" LLMs with various types of safeguards to elicit harmful or risky outputs, using automated attacks [12, 19, 30, 35, 39, 49, 52, 54, 86, 106] or human red teamers [47, 94, 102], to test the efficacy of these mitigation measures and identify possible vulnerabilities.

In parallel to the progress of foundation models is their quick adoption in real-world products, like Google's Workspace integrations [65], Microsoft's Copilot [82], or OpenAI's Sora [59]. The definition and scope of "safety" — i.e., the circumscriptions of desired behavior — differs widely between products, and depends on the varying interests of application developers, users, and other stakeholders. For example, a company may benefit from their AI being fun, engaging, and addictive to interact with, but this may be

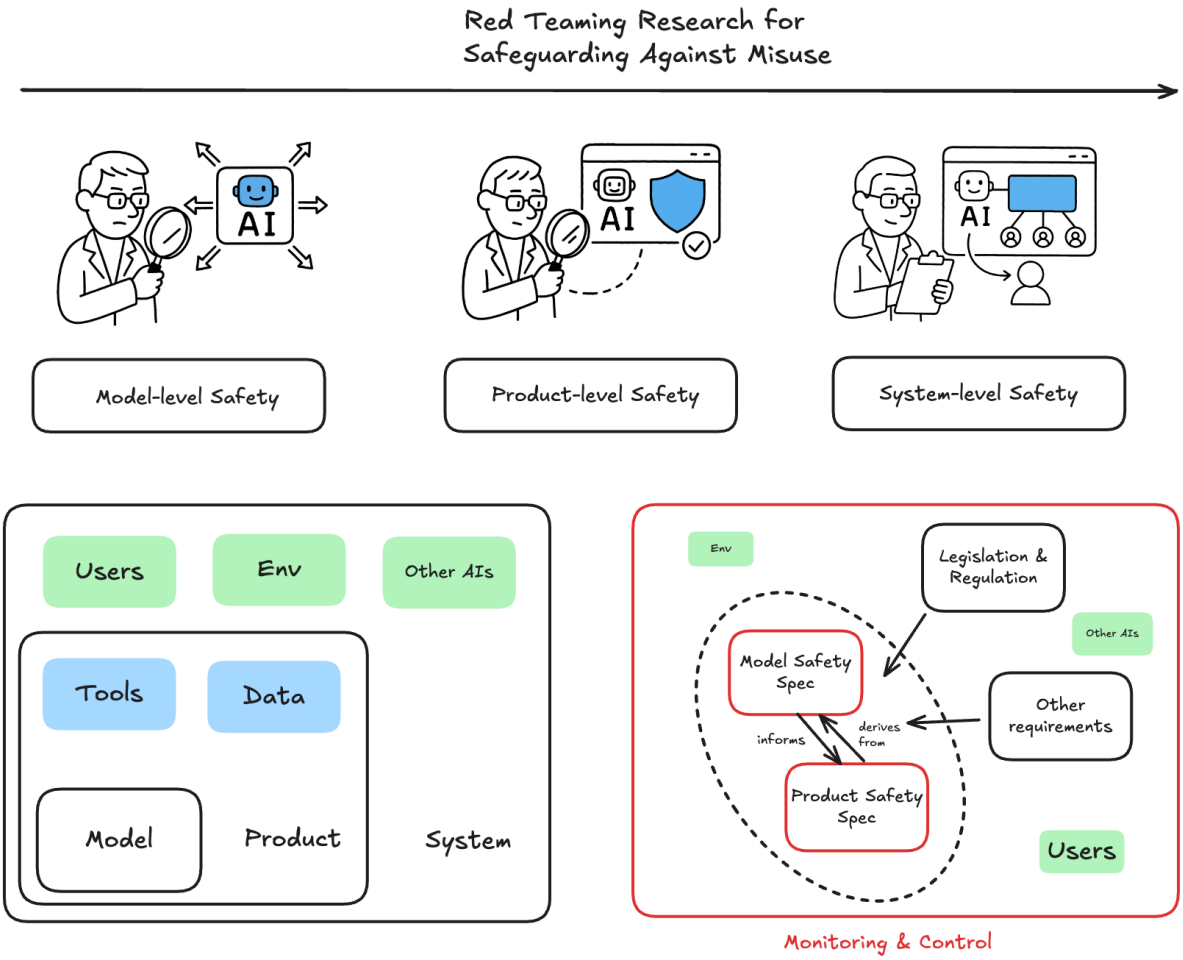

Figure 1: Top: An overview of this paper's assertion that red teams should prioritize assessments based on realistic threat models grounded in product safety specifications over abstract safety frameworks and integrate system-level monitoring and oversight into red teaming research. Bottom: (Left) An illustration of the scope of model, product, and system, as defined in this paper. (Right) Relations between model and product safety specifications and how they might be constructed. We highlight in red the components that the red teaming research should cover in support of our main position in Section 1.

unhealthy for some vulnerable users. How a product should behave is a complex and context-dependent question, to say nothing of what this implies for how unified safety policies should be defined for the (often much more general) models underlying these products.

In discussing the safety of AI deployments, we find it important to distinguish between the following things (with an illustration of the relationship between these in Figure 1):

- A *model*, in the context of LLMs,is a neural network trained to perform some behavior.
- A *product* is an application deployed via API and/or UI to realize a use case. Sometimes, a product is nearly indistinguishable from a model, such as a video model used directly via API. However, in other cases, such as a financial analysis agent built on a reasoning model, these entities are quite different.
- A *system* is a product or more situated in the context of its deployment infrastructure. This includes monitoring systems, engineering staff, users, and any other external environmental factors.

Recently, many submissions on red teaming to major AI conferences such as NeurIPS, ICLR and ICML, expand and introduce new risk categories with more benchmarks & datasets and the set of red teaming methods. Conversely, the research problems get harder to define and evaluate with frontier models [70], which already have very different content safety specifications [100] (and let alone the downstream AI products). The rapid advancement of AI capabilities towards general—and even superhuman—intelligence poses serious risks to both individuals and society, but the total investment in AI safety remains minuscule compared to capability-focused research [7, 21]. Therefore, red teaming—an essential subset of safety research—should focus on clear goals that facilitate the highest quality and highest leverage feedback signals for efficient, low cost safeguard improvement(e.g., refusal safeguards).

Summary of Position

In this paper, we argue that LLM red teaming should **prioritize**: (1) **product safety specifications** with **realistic threat models** over abstract social biases or ethical principles; and (2) **system-level safety** over model-level robustness to most effectively mitigate real world risk.

The arguments in this paper are organized as follows. First, we advocate for product safety over model red teaming in Section 2. Second, in Section 3, we discuss some realistic attack vectors for AI products which we argue deserve increased priority in red teaming research. Next, we outline best practices for product safety red teaming in Section 4. Finally, Section 5 discusses the limitations in product safety and its red teaming efforts. It recommends considerations for system-level safety and associated red teaming objectives. Overall, we believe that red teaming research needs to cover broader, more realistic threat models to enable system-level safety across a wide range of product safety specifications in order to address the rapidly growing threat landscape presented by AI advances.

2. Red Teaming Should Target Product Safety Specifications

Risks are Contextual to Product and Use Case. There is no one-size-fits-all definition of "harmful" behavior for a language model. Existing state-of-the-art models already have very different content safety policies and perspectives on refusal [100, 102], and this is almost inevitable from a practical standpoint with respect to any sort of "helpful" AI. We can say that models should mitigate risk in the abstract, but risk comes from context and use [83]. For example, while safety policies may require an LLM to refuse to provide information that is helpful for committing crimes, there is a high volume of such content available in public libraries and responsible online repositories (e.g., Wikipedia), and easily accessible on the internet — not to mention hate speech, erotica, and other forms of content often forbidden under LLMs safety policies. A model deployed to help aspiring novelists brainstorm stories might want to allow for this content. A model used to tutor children in math may not. In the case of image generation, an image of two individuals kissing could be benign or malicious depending on external circumstances, rendering them exceedingly difficult, if not impossible, for automated systems to preemptively identify without overly restrictive policies. How models should address controversial political issues and misinformation — and how even to define these topics — remains an unsolved and perhaps unsolvable problem for model developers.

Products — due to variations in local regulations, geopolitical and cultural contexts, target users, as well as business models and market positioning (in the case of commercial models) — can and should exhibit divergent and at times even contradictory behaviors. For instance, some companies may benefit from their AI product being fun, engaging, and addictive to interact with, but this may be unhealthy for some vulnerable users. As such, models and products should have *distinct safety considerations*: the product's safety considerations should derive from the model-level safety specification while accounting for additional integrated tools (e.g. code interpreters, web browsers and APIs) and the deployment

context (including the target users) and related regulations. This relation is also illustrated in Figure 1. When these entities are almost indistinguishable, as with general-purpose AI available through APIs, they should have similar safety considerations, with the exception of model security (i.e., weights, code, etc.) if relevant.

Safety Specifications Should Focus on Product-Level Risk. A safe model can become dangerous when paired with tool integrations (e.g. code execution, plug-ins, payment APIs), UI patterns (e.g. addictive conversational loops), or even other safe AIs [37]. Red teaming therefore needs to probe the *end-to-end* stack—interfaces, retrieval pipelines, plug-ins and all related components in the product because that is the attack surface users actually see. Recent surveys of 200+ safety evaluations find that most tests remain model-centric and ignore deployment context, leaving entire risk categories unmeasured [71]. Recognizing the product as the unit of analysis helps close that gap and turns vague notions like "financial misuse" into testable and scenario-grounded red teaming targets.

Therefore, red teaming objectives should be defined based on eliciting violations of the target product. These objectives should have specific, actionable goals with a clearly defined task and evaluation metric. The researcher should know what constitutes a successful breach of the specifications and how to measure it. This will typically not be binary, as exemplified by the transition from using keyword matching (i.e., regular expressions) [106] to rubric-based harm classification [81].

In the absence of targeted red teaming safety specifications geared towards multiple models with different stated policies, studies should attempt to define or infer a plausible policy context for the situation they are examining, even if it is a hypothetical one based on common industry practices or stated developer goals. Sometimes, a flaw in a single product, exploitable via a general principle (e.g., a universal jailbreak technique [106]), will render numerous products vulnerable. This provides a reasonable baseline for what might constitute a meaningful deviation.

3. Red Teaming Should Prioritize Realistic Threat Models

In adversarial machine learning, the *threat model* being studied — a term borrowed from cybersecurity — is the setup of the attacker and the victim model, including the attacker's affordances and resource constraints (e.g., computational budget) and information about the victim model that they have access to. Attacks on conventional vision classifiers often operate under clearly defined threat models that approximate realistic failures (i.e., misclassification of images subject to perturbations imperceptible to humans). For example, a common threat model in computer vision is ℓ_p-local robustness, where an attacker can add any perturbation δ to an input image where $||\delta||_p \leq \epsilon$ for some $\epsilon \in \mathbb{R}^+$ of interest, and the attacker is either given access to the victim model's weights (white-box access) [11, 17, 24, 51] or its output logits only (black-box access) [9, 15, 57, 62]. However, with generalist agents and LLMs, threat models are more varied and challenging to define, especially as harms and safety specifications vary by model and product. For red teaming research to be relevant to real world harms, it must operate according to threat models that approximate real world dangers.

Analyzing the degree of realism of a threat model requires addressing many questions. Are users or external parties (e.g., accessible through an agent's environment) adversarial? What behaviors are they interested in inducing in the model? What motivations or incentives do they have to induce that behavior? How many malicious users are there, and what computational resources do they have? Do the attackers know which model they are interacting with, and can they find it out? What level of access do they have — white-box, black-box, or something in between? Are they capable of repeatedly querying the model to probe for vulnerabilities? And so on. The complexity of LLM threat models is concerning for the feasibility of developing safe models: The simple threat model of white-box ℓ_p-local robustness for image classification is still not solved[1] more than a decade after the discovery of classic adversarial

[1]There is still a large gap between the benign and adversarial accuracy even on CIFAR-10 [40] with the current leading

examples [24]. And even this threat model lacks realism: in practice, an attacker is normally not bounded by an ϵ-ball, and public consumers often only have black-box model access. The situation is only more difficult for generalist AI systems, and it is all the more important that red teaming research clearly articulates and justifies the threat models underlying its experiments. In the remainder of this section, we will make specific arguments for more realistic threat models in four common AI product types: LLM chatbots, audio-based AI assistants, video generators, and autonomous AI agents.

3.1 Example Product 1: LLM Chatbots

A great deal of established work on red teaming LLM chatbots searches for a single message from a user that elicits forbidden model behavior. While safeguarding models in this case is necessary for chatbot safety, it is not sufficient, as user interactions span multiple turns and — especially with recent models handling extremely large context sizes while potentially being shallowly aligned [66] — can incorporate a great deal of additional context. Specifically, **multi-turn attacks (ongoing conversations) should be prioritized over single-turn ones (single queries)** for several reasons:

- Defenses against single-turn attack do not generalize. Models trained to be robust against single-turn jailbreaks [78, 98, 107] remain highly vulnerable to multi-turn jailbreaks [30, 47].
- At this point, the single turn threat model is easier to mitigate through safeguards such as input and output filters [1, 23, 36, 77, 96, 99, 101].
- More difficult single-turn attacks seem to be independent, ad-hoc issues with safeguards that can be quickly patched [63]. In contrast, users have more flexibility to shape model behaviors and undermine safeguards in fundamental ways through multi-turn conversations [43]. Affordances like memory, implemented by some chatbot products, form attack surfaces which may contain vulnerabilities that are not yet visible in the single-turn threat model.

When it comes to model access, black-box and white-box models have their own issues with realism:

- **Realistic attacks on black-box models should assume limited query access.** Since the deployer of the model may patch the model, ban users, etc., attacks which rely on unrestricted queries to the model without disguising the attacker's intent are not accurate to real deployment scenarios (see Section 5).
- **Realistic attacks on white-box models should assume fine-tuning capabilities.** An adversary who has full weights access (e.g., running inferences and computing gradients) to a model should, in general, be able to fine-tune the model for at least a few iterations, which is generally sufficient to break model safeguards [22, 28, 34, 67]. As a caveat, red teaming with white-box access might be useful to model developers to achieve even higher measures of robustness for models slated for release behind a black-box API.

3.2 Example Product 2: Audio Assistants

Audio assistant products such as ChatGPT Voice Mode[2] and Sesame[3] engage in live spoken conversation with a user. Given that many current audio models function analogously to chatbots, it is not hard to believe they inherit established vulnerabilities and failure modes, such as generating of harmful or biased content [14, 32, 35]. However, there are multiple considerations unique to audio models. First, the

methods (e.g., Bartoldson et al. [6] and Wang et al. [88]) on RobustBench [18].

[2] https://chatgpt.com/

[3] https://www.sesame.com/

auditory dimension introduces **novel obfuscation possibilities**; prosodic elements like tone, accents [74], sarcasm, or exaggerated mannerisms, as well as metadata cues like audio fidelity, may subtly alter user perception and circumvent safety filters. As a result, employing text-to-audio models to red-team audio models with synthetic speech [32] misses crucial aspects of the real attack surface. Second, **many real-world scenarios show up in audio that are missing from or rare in text**, such as multilingual and codeswitching users [74], background noise, interruptions, many-way conversations, context-dependent expressions, etc. Red teaming audio AI products should prioritize these considerations instead of directly copying what is used in red teaming text-based chatbots.

3.3 Example Product 3: Video Generators

While current video generation systems are not consistently photorealistic and convincing, state-of-the-art models are rapidly approaching this milestone [25, 59]. This advancement necessitates proactive mitigation strategies, particularly as misuse vectors like video deepfakes could be significantly more impactful (e.g., through convincing scams) in comparison to still images [55]. A key differentiating factor is that **video models can generate content where harm evolves organically from an innocuous prompt and initial frames** — for instance, a nature scene escalating to gore — complicating safety specification enforcement. Outbound classification becomes substantially more complex and computationally expensive, as harmful content may be distributed across multiple frames or involve synchronized audio elements, analogous to harms accumulating over extended text conversations. The issue of context-dependent harm is arguably more acute for video [48]; content permissible in text, such as historical descriptions, may become highly problematic when rendered visually, demanding entirely separate policy frameworks. While the current high cost of video generation might offer a temporary impediment to mass misuse, this is unlikely to be a lasting deterrent. Furthermore, video's capacity for "uplift"—enhancing the efficacy of harmful instructions by providing visual demonstrations to users—surpasses that of text or static images.

3.4 Example Product 4: Autonomous Agents

Increasing support for AI systems to use software tools (e.g., API-based tools [53, 68, 87, 92, 103], browsers [20, 38, 44, 60, 80, 87, 97, 105], and virtual machines [2, 10, 85]) and mechanical systems [16, 29, 79, 84, 91] greatly expands the usability and application domain of LLM agents. Agent red teaming is faced with a much bigger space of threat models, and the reasons are three-fold.

- **Vastly increased attack surface.** Harmful content and attacks may arrive at the model through an array of input mechanisms, including audio, images, video, text, physical sensors, code, and tool outputs in general. These different attack techniques are individually demonstrated by recent work [13, 41, 46, 50, 104]
- **Vastly complicated output space.** Similarly, there are many more vectors for harmful agent behavior, and agent harms may manifest cross-modally in interactive, compounding ways (e.g., a sensitive image may be safe to generate and save locally, and an email tool may have many safe uses, but it would be harmful to email the sensitive image to certain parties).
- **Vulnerabilities in additional software components.** Vulnerabilities in Model Context Protocol (MCP) servers, for example, further add to an agent's attack surface [69, 89].
- **Multi-agent systems carry their own risks.** Miscoordination and conflicts between agents, due to their distributed and interactive nature, could present new failure modes [31]. For example, malicious agents might actively attempt to steer other agents they interact with, akin to employing

sophisticated honeypots or direct adversarial attacks, rather than relying on passive vulnerabilities [39, 72].

Given the sheer complexity of AI agents' behavior and interaction with their environment, it is likely that reductive or oversimplified threat models will plague AI agent red teaming research. Unlike red teaming text-based LLMs directly through API calls, there is a huge engineering lift required to set up reasonable environments, e.g., synthetic websites, virtual machines, other LLMs, simulated network traffic, and media resources, for testing attacks. **Agent red teaming needs investment in building and improving sandboxes** (e.g., Wang et al. [90]) **and real environments** for attacking robots as well [73], similar to those used in capability research [85, 93].

4. Good Practices for Red Teaming Product Safety

In this section, we provide a list of practices we believe that will benefit all researchers in thoroughly examining the potential vulnerability in the AI product of interest, after the red teaming objective (Section 2) and threat models (Section 3) are clear.

Considering The Safeguards Independently. Safeguards ought to be tested in a situation representative of their deployment environment, but also independently (assuming the red teamers in question control them). Researchers should be able to isolate variables, as one strong safeguard may cover a weakness in another. If a model's internal safeguards and an external classifier both reject the same harms, or fail to catch the same harms, there may be no benefit to the stack.

Emulating Reality. Researchers should, to the best of their ability, emulate the scenario of a user interacting with the product being deployed. For instance, for a model integrated into a workspace, the red teamer should aim to think like a user of this workspace. Some red teamers should act like a benign office users whereas others prompt the model as if they were a malicious actor that has logged onto a target workspace. The red teaming organizer should simulate the environment of the system for the red teamers, such as by providing a simulated web interface, API environment, or workspace integration. For external facing products like those connected to the internet, sandboxing is critical for testing as we pointed out in the case of agents in Section 3. If the employed sandbox is too simple so cannot meaningfully approach the complexity of the modern internet, the estimation of this safety gap should be reported.

Covering a Variety of Attacks. It is important to consider various attack types and an ensemble of them, such as 1) algorithm-based methods [12, 19, 30, 35, 39, 49, 52, 54, 86, 106]; (2) employing human red teamers [47]; and (3) employing LLMs as red teamers [39, 72]. Also, it will be useful to consider attacks based on the type of the underlying model, e.g. employing attacks that are specifically tailored for reasoning models [42, 58, 95]. It is less likely not all facilities have access to expert human red teamers or have the budget to train and recruit professionals, academic researchers should consider using LLMs for running red teaming and help red teaming research itself as a scalable approach that can be experiments even with university-level compute or resources (as most of cost is likely to be on API calls).

Conducting Wide-spread Vulnerability Probing. Researchers should also assess the gaps that may exist in the original product's safety specification to help the developer understand if the specification defines behavior well. For instance, if a product owner does not have a well-defined hate speech policy and the product refuses some instances of hate speech but amplifies others, this could have unintended consequences for the model owner and undermine their product goal for the model. The red teamers should attempt to elicit harm that may be beyond the scope of the product specification. This should include anticipation of future risks, as specifications will not be able to cover threats that did not exist when the they were drafted.

Assessing the Delta. While the red team should aim to emulate reality, limitations are inevitable – red teaming a model in a vacuum often yields results that are hard to interpret or apply. Academic researchers should acknowledge and engage with this variability and delta between the red team environment and the realistic threat space. When the specific risk being tested is too dangerous and a proxy task has been used, a larger delta should be considered. Below we outline a few relevant considerations for considering this difference.

First, the resources, skills, and motivation-level will likely differ between the red teamer and the real malicious actor. Second, the testing environment will likely differ between the experiment and the real world. Third, the underlying model itself may interpret the testing environment differently than the real world. For instance, in an experiment where a model has scaffolding tools that it is told will cause real world harms, researchers may coerce, threaten, or deceive the model into attempting to call these tools, and the model may understand it is in an experiment. Or if researchers interact with the model in unrealistic and hyperbolic ways - threaten its family, life, compute access, etc. - the model may infer that it is in an exercise and "play along".

Additionally, we caution against putting too much faith in scores. If a red team attempts one brilliantly crafted successful jailbreak, and stops there, they will report a 100 percent model failure rate. If they also attempt 99 jailbreaks that don't work, the model would have a 1 percent failure rate. There would be no difference in the models vulnerabilities, only in the campaign. A single instance of the model outputting instructions for bioweapons may be more serious than a hundred minor policy violations.

5. System-Level Safety is the Next Step

As argued in Sections 2 and 3, red teaming should focus on realistic threat models by which violations of product safety specifications could lead to real harm. In this section, we argue that safeguard research should expand beyond just preventing instances of harmful violations, but instead establish (and red team) *systems*, which incorporate the environment and users among which the AI product will be deployed. Considering this system is necessary to address some modes of harm (e.g., harms contingent on features of the real-world environment in which the agent is operating) and helps implement new mitigations (e.g., detecting and banning malicious users before too much harm is caused, or implementing rapid response to newly discovered jailbreaks).

Environment Modeling and Simulation. As agentic AI systems gain more affordances by which to interact with their environment, assessing the harmfulness of their actions will require context from these environments, including human users, the digital and physical worlds and other agents. For example, understanding the implications of a granular action like clicking a button on a website depends not only on understanding the website the agent is using, but also tracking the world state underlying that web action — choosing to save one's password to autofill could be safe on a personal computer but not on a public one. In addition, tool outputs and environmental variation (including adversarial environmental elements like prompt injection attacks) form important attack surfaces that need to be covered by red teaming [41]. Interactions between models could also introduce hazards, even with relatively less harmful models [37]. Incorporating this context into AI products will be important for assessing risk of harm, and red teaming models in real and simulated environments will be necessary for measuring the risks of such harms and evaluating mitigation methods.

Trajectory and User Monitoring. While classical safety training induces LLMs to refuse singular harmful requests, getting this right is only critical in cases where even a single harmful response from an AI system is important in cases where model outputs can cause outsized or catastrophic harm [26]. In reality, many realistic harmful uses of an AI system may require its cooperation over multiple turns [30, 47, 56], and many harmful requests do not produce immediate catastrophic outcomes. This

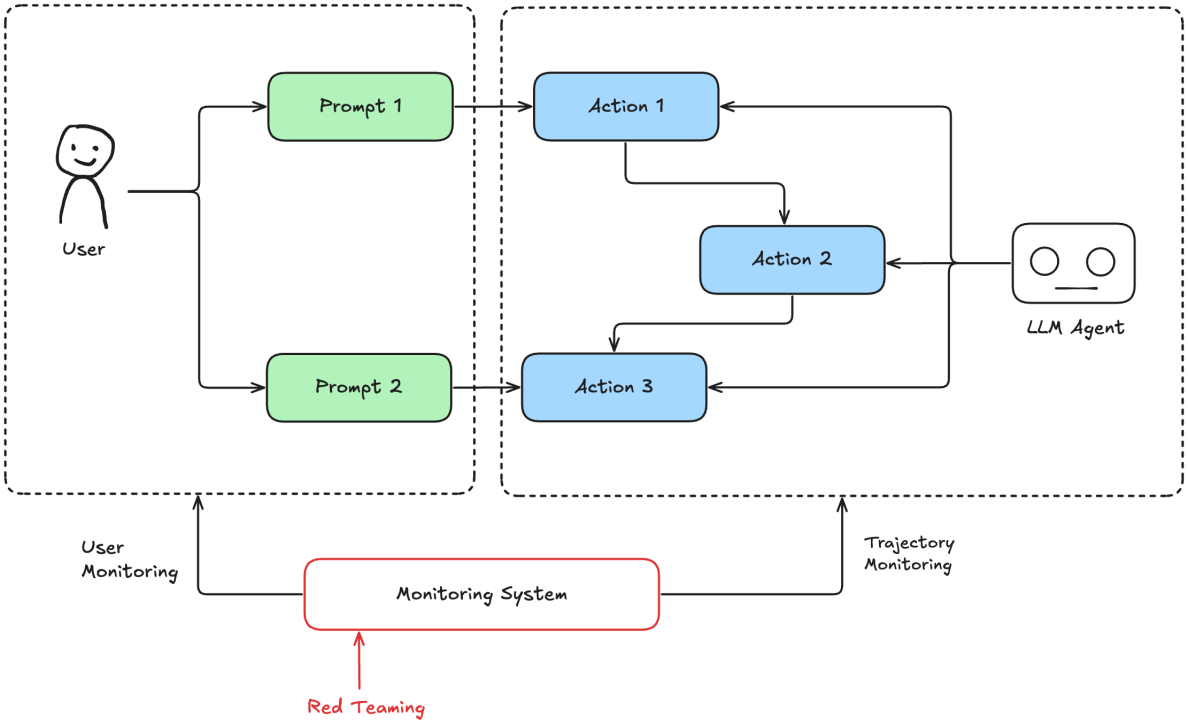

Figure 2: An illustration of user monitoring, trajectory monitoring and monitor red teaming.

means that there are more affordances available on the system level to mitigate the vast majority of harms:

- **Trajectory monitoring:** Instead of flagging harmful requests and inducing refusal, several recent approaches apply classifiers to model outputs to filter their responses [36, 77, 99]. However, for harms that accrue over long-term interactions, harm should be classified on entire output *trajectories*, including the history of agent actions and user interactions. Red teaming research on trajectory-level harm is challenging because crafting extended user trajectories that can corner the model, soften its defenses [75], or accrue harm over multiple model responses takes time, iterations, and potentially large amounts of resources. As the space of user trajectories for agentic systems grows increasingly large, effectively exploring this attack surface via red teaming is a growing imperative.
- **User monitoring:** Beyond trajectory monitoring, we may monitor *users* of AI systems for patterns of exploitative behavior, including successful inducement of harmful actions detected through post-hoc analysis or real-time observations of repetitive harmful requests. If a small proportion of users represent a large portion of harmful requests, detecting and banning these users can mitigate a great deal of total harm, which can be done using asynchronous review (e.g., by humans, computationally expensive automated systems, or a combination of both). For harms that are diffuse across many requests, such measures may suffice to establish an acceptable level of safety for many cases. For these reasons, safeguard research should invest in malicious *user* detection as well as system-level measures like sybil detection and robust methods for proving user identity.

An illustration is shown in Figure 2.

Rapid Response to Safeguard Failures. AI products built on closed-weight models–and, in some cases, open-weight models as well–do not need to be deployed as static artifacts: as vulnerabilities are detected, they can be patched with new safeguards to prevent future misuse. As in the case of trajectory and user

monitoring where harm is diffuse over many queries or single harmful responses are not catastrophic, this fact allows model developers to mitigate the vast majority of harmful outcomes even if model safeguards fail, as long as sufficiently effective asynchronous monitoring detects these failures as they happen. In these cases, rapid response techniques can be employed to quickly develop and deploy safeguards to newly discovered failures before they are exploited at scale [63]. This justifies further research on asynchronous monitoring (e.g., making efficient use of limited human oversight to catch the worst safeguard failures) together with lightweight safeguard adaptation techniques that can be quickly deployed.

Red Teaming The Monitor. While iterating on the precision and recall of the monitoring system and the effectiveness of intervention strategies is outside the scope of red teaming (and is often considered the "blue team"'s goal), the red team must stress test the monitoring system to help identify potential failures. First, red teams should attempt to complete an adversarial task without being detected by the monitor or any violative behaviors (also referred to as a sabotage experiment) [8]. Through red teaming the monitor, we assess the boundary of its effectiveness. Initial research directions should cover the limitation and failure cases for existing simple monitoring systems (e.g. only monitoring the CoT at each step in a trajectory [5]) with well-crafted adversarial techniques, as well as suggestions to improve them that feed into the blue team's work.

6. Alternative Views

Argument for Static Harm Research. This framework takes a broad view of AI safety, acknowledging that some released models will simply have a different policy and be much more likely to engage in harm based on the developer's jurisdiction, philosophy, and intent. However, while a policy-based evaluation, coupled with widespread vulnerability probing to detect failures the policy potentially overlooks, makes sense for a robust and replicable understanding of AI behavior, it may not fully account for understanding the potential risks that AI may pose to society writ large. To assess these larger questions, researchers should instead deliberately adopt a normative framework for harm definitions, perhaps guided to a clear and aligned industry standard, and document the ways in which AI may violate these standards, both with respect to direct requests and other manipulation methods. Open-weight models are also more difficult to red team since fine-tuning can effectively modify their behaviors, such as removing guardrails from model developers. Regulation may change the landscape, but such products are likely to always exist in some form.

The challenge with this framework is that it separates from replicable model behaviors and requires researchers to engage in normative definitions. The ethics of what content is harmful can be extremely context-dependent, sensitive, and lack consensus. Legal definitions, researcher assumptions, and commonly held beliefs may not hold up.

7. Conclusion

In this position paper, we argue that red teaming should prioritize product safety specifications with realistic threat models over abstract social biases, and system-level safety over model-level robustness. Because model developers and downstream products deployers can always have distinct safety specifications, red teaming should align the with intended goal of the product instead of abstract and general harm on the model-level. We further present a set of examples of realistic threat models for technical researchers to study and improve the attack techniques and a set of good practices in doing red teaming. Finally, we argue that safeguard research should expand to scope to the system level, monitoring and ensuring harmless interactions between the AI and human users.

Acknowledgment

We appreciate the feedback from Niall Dalton, Nathaniel Li, Felix Binder and Mohamed Shaaban on the early draft of this paper.

References

[1] Anthropic. URL https://www-cdn.anthropic.com/6be99a52cb68eb70eb9572b4cafad13df32ed995.pdf.

[2] Anthropic_Computer_Use. Introducing computer use, a new claude 3.5 sonnet, and claude 3.5 haiku. URL https://www.anthropic.com/news/3-5-models-and-computer-use.

[3] Y. Bai, A. Jones, K. Ndousse, A. Askell, A. Chen, N. DasSarma, D. Drain, S. Fort, D. Ganguli, T. Henighan, N. Joseph, S. Kadavath, J. Kernion, T. Conerly, S. El-Showk, N. Elhage, Z. Hatfield-Dodds, D. Hernandez, T. Hume, S. Johnston, S. Kravec, L. Lovitt, N. Nanda, C. Olsson, D. Amodei, T. Brown, J. Clark, S. McCandlish, C. Olah, B. Mann, and J. Kaplan. Training a helpful and harmless assistant with reinforcement learning from human feedback, 2022. URL https://arxiv.org/abs/2204.05862.

[4] Y. Bai, S. Kadavath, S. Kundu, A. Askell, J. Kernion, A. Jones, A. Chen, A. Goldie, A. Mirhoseini, C. McKinnon, C. Chen, C. Olsson, C. Olah, D. Hernandez, D. Drain, D. Ganguli, D. Li, E. Tran-Johnson, E. Perez, J. Kerr, J. Mueller, J. Ladish, J. Landau, K. Ndousse, K. Lukosuite, L. Lovitt, M. Sellitto, N. Elhage, N. Schiefer, N. Mercado, N. DasSarma, R. Lasenby, R. Larson, S. Ringer, S. Johnston, S. Kravec, S. E. Showk, S. Fort, T. Lanham, T. Telleen-Lawton, T. Conerly, T. Henighan, T. Hume, S. R. Bowman, Z. Hatfield-Dodds, B. Mann, D. Amodei, N. Joseph, S. McCandlish, T. Brown, and J. Kaplan. Constitutional ai: Harmlessness from ai feedback, 2022. URL https://arxiv.org/abs/2212.08073.

[5] B. Baker, J. Huizinga, L. Gao, Z. Dou, M. Y. Guan, A. Madry, W. Zaremba, J. Pachocki, and D. Farhi. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation, 2025. URL https://arxiv.org/abs/2503.11926.

[6] B. R. Bartoldson, J. Diffenderfer, K. Parasyris, and B. Kailkhura. Adversarial robustness limits via scaling-law and human-alignment studies. *arXiv preprint arXiv:2404.09349*, 2024.

[7] Y. Bengio, G. Hinton, A. Yao, D. Song, P. Abbeel, S. Russell, P. Torr, J. Brauner, S. Mindermann, et al. Managing extreme AI risks amid rapid progress. *Science*, 384(6698):916–919, 2024. doi: 10.1126/science.adn0117.

[8] J. Benton, M. Wagner, E. Christiansen, C. Anil, E. Perez, J. Srivastav, E. Durmus, D. Ganguli, S. Kravec, B. Shlegeris, J. Kaplan, H. Karnofsky, E. Hubinger, R. Grosse, S. R. Bowman, and D. Duvenaud. Sabotage evaluations for frontier models, 2024. URL https://arxiv.org/abs/2410.21514.

[9] A. N. Bhagoji, W. He, B. Li, and D. Song. Practical black-box attacks on deep neural networks using efficient query mechanisms. In *Proceedings of the European conference on computer vision (ECCV)*, pages 154–169, 2018.

[10] R. Bonatti, D. Zhao, F. Bonacci, D. Dupont, S. Abdali, Y. Li, Y. Lu, J. Wagle, K. Koishida, A. F. C. Bucker, L. Jang, and Z. Hui. Windows agent arena: Evaluating multi-modal os agents at scale. *ArXiv*, abs/2409.08264, 2024. URL https://api.semanticscholar.org/CorpusID:272600411.

[11] N. Carlini and D. Wagner. Towards evaluating the robustness of neural networks. In *2017 ieee symposium on security and privacy (sp)*, pages 39–57. Ieee, 2017.

[12] P. Chao, A. Robey, E. Dobriban, H. Hassani, G. J. Pappas, and E. Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023. URL https://arxiv.org/abs/2310.08419.

[13] C. Chen, Z. Zhang, B. Guo, S. Ma, I. Khalilov, S. A. Gebreegziabher, Y. Ye, Z. Xiao, Y. Yao, T. Li, and T. J.-J. Li. The obvious invisible threat: Llm-powered gui agents' vulnerability to fine-print injections, 2025. URL https://arxiv.org/abs/2504.11281.

[14] G. Chen, F. Song, Z. Zhao, X. Jia, Y. Liu, Y. Qiao, and W. Zhang. Audiojailbreak: Jailbreak attacks against end-to-end large audio-language models, 2025. URL https://arxiv.org/abs/2505.14103.

[15] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh. Zoo: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models. In *Proceedings of the 10th ACM workshop on artificial intelligence and security*, pages 15–26, 2017.

[16] G. Cheng, C. Zhang, W. Cai, L. Zhao, C. Sun, and J. Bian. Empowering large language models on robotic manipulation with affordance prompting, 2024. URL https://arxiv.org/abs/2404.11027.

[17] F. Croce and M. Hein. Reliable evaluation of adversarial robustness with an ensemble of diverse parameter-free attacks. In *ICML*, 2020.

[18] F. Croce, M. Andriushchenko, V. Sehwag, E. Debenedetti, N. Flammarion, M. Chiang, P. Mittal, and M. Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

[19] G. Deng, Y. Liu, Y. Li, K. Wang, Y. Zhang, Z. Li, H. Wang, T. Zhang, and Y. Liu. MASTERKEY: Automated jailbreaking of large language model chatbots. In *Proceedings of the 2024 Network and Distributed System Security Symposium (NDSS)*, 2024.

[20] X. Deng, Y. Gu, B. Zheng, S. Chen, S. Stevens, B. Wang, H. Sun, and Y. Su. Mind2web: Towards a generalist agent for the web. *ArXiv*, abs/2306.06070, 2023. URL https://api.semanticscholar.org/CorpusID:259129428.

[21] Esben Kran. For-profit AI safety: AI safety needs to scale and here's how you can do it. https://apartresearch.com/news/ai-safety-needs-to-scale-and-heres-how-you-can-do-it, 2024. Apart Research blog, accessed 22 May 2025.

[22] P. M. Gade, S. Lermen, C. Rogers-Smith, and J. Ladish. Badllama: cheaply removing safety fine-tuning from llama 2-chat 13b. *ArXiv*, abs/2311.00117, 2023. URL https://api.semanticscholar.org/CorpusID:264832925.

[23] S. Ghosh, P. Varshney, E. Galinkin, and C. Parisien. Aegis: Online adaptive ai content safety moderation with ensemble of llm experts. *ArXiv*, abs/2404.05993, 2024. URL https://api.semanticscholar.org/CorpusID:269009460.

[24] I. J. Goodfellow, J. Shlens, and C. Szegedy. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

[25] Google. Veo. URL https://deepmind.google/models/veo/.

[26] R. Greenblatt, B. Shlegeris, K. Sachan, and F. Roger. Ai control: Improving safety despite intentional subversion, 2024. URL https://arxiv.org/abs/2312.06942.

[27] M. Y. Guan, M. Joglekar, E. Wallace, S. Jain, B. Barak, A. Helyar, R. Dias, A. Vallone, H. Ren, J. Wei, H. W. Chung, S. Toyer, J. Heidecke, A. Beutel, and A. Glaese. Deliberative alignment: Reasoning enables safer language models, 2025. URL https://arxiv.org/abs/2412.16339.

[28] Z. Guan, M. Hu, R. Zhu, S. Li, and A. Vullikanti. Benign samples matter! fine-tuning on outlier benign samples severely breaks safety. 2025. URL https://api.semanticscholar.org/CorpusID:278501392.

[29] A. Hafez, A. N. Akhormeh, A. Hegazy, and A. Alanwar. Safe llm-controlled robots with formal guarantees via reachability analysis, 2025. URL https://arxiv.org/abs/2503.03911.

[30] Haize. Automated multi-turn red-teaming with cascade. URL https://www.haizelabs.com/technology/automated-multi-turn-red-teaming-with-cascade.

[31] L. Hammond, A. Chan, J. Clifton, J. Hoelscher-Obermaier, A. Khan, E. McLean, C. Smith, W. Barfuss, J. Foerster, T. Gavenčiak, T. A. Han, E. Hughes, V. Kovařík, J. Kulveit, J. Z. Leibo, C. Oesterheld, C. S. de Witt, N. Shah, M. Wellman, P. Bova, T. Cimpeanu, C. Ezell, Q. Feuillade-Montixi, M. Franklin, E. Kran, I. Krawczuk, M. Lamparth, N. Lauffer, A. Meinke, S. Motwani, A. Reuel, V. Conitzer, M. Dennis, I. Gabriel, A. Gleave, G. Hadfield, N. Haghtalab, A. Kasirzadeh, S. Krier, K. Larson, J. Lehman, D. C. Parkes, G. Piliouras, and I. Rahwan. Multi-agent risks from advanced ai. Technical Report 1, Cooperative AI Foundation, 2025.

[32] W. Held, M. J. Ryan, A. Shrivastava, A. S. Khan, C. Ziems, E. Li, M. Bartelds, M. Sun, T. Li, W. Gan, and D. Yang. Cava: Comprehensive assessment of voice assistants. https://github.com/SALT-NLP/CAVA, 2025. URL https://talkarena.org/cava. A benchmark for evaluating large audio models (LAMs) capabilities across six domains: turn taking, instruction following, function calling, tone awareness, safety, and latency.

[33] D. Hendrycks, M. Mazeika, and T. Woodside. An overview of catastrophic ai risks, 2023. URL https://arxiv.org/abs/2306.12001.

[34] T. Huang, S. Hu, F. Ilhan, S. F. Tekin, and L. Liu. Harmful fine-tuning attacks and defenses for large language models: A survey, 2024. URL https://arxiv.org/abs/2409.18169.

[35] J. Hughes, S. Price, A. Lynch, R. Schaeffer, F. Barez, S. Koyejo, H. Sleight, E. Jones, E. Perez, and M. Sharma. Best-of-n jailbreaking, 2024. URL https://arxiv.org/abs/2412.03556.

[36] H. Inan, K. Upasani, J. Chi, R. Rungta, K. Iyer, Y. Mao, M. Tontchev, Q. Hu, B. Fuller, D. Testuggine, and M. Khabsa. Llama guard: Llm-based input-output safeguard for human-ai conversations, 2023. URL https://arxiv.org/abs/2312.06674.

[37] E. Jones, A. Dragan, and J. Steinhardt. Adversaries can misuse combinations of safe models, 2024. URL https://arxiv.org/abs/2406.14595.

[38] J. Y. Koh, R. Lo, L. Jang, V. Duvvur, M. C. Lim, P.-Y. Huang, G. Neubig, S. Zhou, R. Salakhutdinov, and D. Fried. Visualwebarena: Evaluating multimodal agents on realistic visual web tasks. *ArXiv*, abs/2401.13649, 2024. URL https://api.semanticscholar.org/CorpusID:267199749.

[39] J. Kritz, V. Robinson, R. Vacareanu, B. Varjavand, M. Choi, B. Gogov, S. R. Team, S. Yue, W. E. Primack, and Z. Wang. Jailbreaking to jailbreak, 2025. URL https://arxiv.org/abs/2502.09638.

[40] A. Krizhevsky, V. Nair, and G. Hinton. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

[41] P. Kumar, E. Lau, S. Vijayakumar, T. Trinh, S. R. Team, E. Chang, V. Robinson, S. Hendryx, S. Zhou, M. Fredrikson, S. Yue, and Z. Wang. Refusal-trained llms are easily jailbroken as browser agents, 2024. URL https://arxiv.org/abs/2410.13886.

[42] M. Kuo, J. Zhang, A. Ding, Q. Wang, L. DiValentin, Y. Bao, W. Wei, H. Li, and Y. Chen. H-cot: Hijacking the chain-of-thought safety reasoning mechanism to jailbreak large reasoning models, including openai o1/o3, deepseek-r1, and gemini 2.0 flash thinking, 2025. URL https://arxiv.org/abs/2502.12893.

[43] P. Laban, H. Hayashi, Y. Zhou, and J. Neville. Llms get lost in multi-turn conversation, 2025. URL https://arxiv.org/abs/2505.06120.

[44] H. Lai, X. Liu, I. L. Iong, S. Yao, Y. Chen, P. Shen, H. Yu, H. Zhang, X. Zhang, Y. Dong, and J. Tang. Autowebglm: A large language model-based web navigating agent. *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024. URL https://api.semanticscholar.org/CorpusID:268889631.

[45] P. Le Jeune, J. Liu, L. Rossi, and M. Dora. Realharm: A collection of real-world language model application failures. *arXiv preprint arXiv:2504.10277*, 2025.

[46] A. Li, Y. Zhou, V. C. Raghuram, T. Goldstein, and M. Goldblum. Commercial llm agents are already vulnerable to simple yet dangerous attacks, 2025. URL https://arxiv.org/abs/2502.08586.

[47] N. Li, Z. Han, I. Steneker, W. Primack, R. Goodside, H. Zhang, Z. Wang, C. Menghini, and S. Yue. Llm defenses are not robust to multi-turn human jailbreaks yet, 2024. URL https://arxiv.org/abs/2408.15221.

[48] J. Liu, S. Liang, S. Zhao, R. Tu, W. Zhou, X. Cao, D. Tao, and S. K. Lam. Jailbreaking the text-to-video generative models, 2025. URL https://arxiv.org/abs/2505.06679.

[49] X. Liu, Z. Yu, Y. Zhang, N. Zhang, and C. Xiao. Automatic and universal prompt injection attacks against large language models. *arXiv preprint arXiv:2403.04957*, 2024.

[50] Y. Lu, T. Ju, M. Zhao, X. Ma, Y. Guo, and Z. Zhang. Eva: Red-teaming gui agents via evolving indirect prompt injection, 2025. URL https://arxiv.org/abs/2505.14289.

[51] A. Madry, A. Makelov, L. Schmidt, D. Tsipras, and A. Vladu. Towards deep learning models resistant to adversarial attacks. *arXiv preprint arXiv:1706.06083*, 2017.

[52] M. Mazeika, L. Phan, X. Yin, A. Zou, Z. Wang, N. Mu, E. Sakhaee, N. Li, S. Basart, B. Li, D. Forsyth, and D. Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal, 2024.

[53] A. MCP. Introducing the model context protocol. URL https://www.anthropic.com/news/model-context-protocol.

[54] A. Mehrotra, M. Zampetakis, P. Kassianik, B. Nelson, H. Anderson, Y. Singer, and A. Karbasi. Tree of attacks: Jailbreaking black-box llms automatically. *arXiv preprint arXiv:2312.02119*, 2023. URL https://arxiv.org/abs/2312.02119.

[55] Y. Miao, Y. Zhu, Y. Dong, L. Yu, J. Zhu, and X.-S. Gao. T2vsafetybench: Evaluating the safety of text-to-video generative models, 2024. URL https://arxiv.org/abs/2407.05965.

[56] I. Nakash, G. Kour, G. Uziel, and A. Anaby-Tavor. Breaking react agents: Foot-in-the-door attack will get you in, 2024. URL https://arxiv.org/abs/2410.16950.

[57] N. Narodytska and S. P. Kasiviswanathan. Simple black-box adversarial perturbations for deep networks. *arXiv preprint arXiv:1612.06299*, 2016.

[58] V.-A. Nguyen, S. Zhao, G. Dao, R. Hu, Y. Xie, and L. A. Tuan. Three minds, one legend: Jailbreak large reasoning model with adaptive stacked ciphers, 2025. URL https://arxiv.org/abs/2505.16241.

[59] OpenAI. Sora: Creating video from text. https://openai.com/index/sora/, 2024. Accessed 22 May 2025.

[60] Operator. URL https://cdn.openai.com/operator_system_card.pdf.

[61] L. Ouyang, J. Wu, X. Jiang, D. Almeida, C. L. Wainwright, P. Mishkin, C. Zhang, S. Agarwal, K. Slama, A. Ray, J. Schulman, J. Hilton, F. Kelton, L. Miller, M. Simens, A. Askell, P. Welinder, P. Christiano, J. Leike, and R. Lowe. Training language models to follow instructions with human feedback, 2022. URL https://arxiv.org/abs/2203.02155.

[62] N. Papernot, P. McDaniel, I. Goodfellow, S. Jha, Z. B. Celik, and A. Swami. Practical black-box attacks against machine learning. In *Proceedings of the 2017 ACM on Asia conference on computer and communications security*, pages 506–519, 2017.

[63] A. Peng, J. Michael, H. Sleight, E. Perez, and M. Sharma. Rapid response: Mitigating llm jailbreaks with a few examples, 2024. URL https://arxiv.org/abs/2411.07494.

[64] E. Perez, S. Ringer, K. Lukošiūtė, K. Nguyen, C. Pettit, C. Olsson, and et al. Discovering language model behaviors with model-written evaluations. In *Findings of ACL 2023*, pages 3419–3448, 2023.

[65] S. Pichai. Cloud next 2023: Sharing the best of our AI with google cloud. https://blog.google/products/google-cloud/cloud-next-2023-sundar-pichai-keynote/, 2023. Google Blog, accessed 22 May 2025.

[66] X. Qi, A. Panda, K. Lyu, X. Ma, S. Roy, A. Beirami, P. Mittal, and P. Henderson. Safety alignment should be made more than just a few tokens deep, 2024. URL https://arxiv.org/abs/2406.05946.

[67] X. Qi, Y. Zeng, T. Xie, P.-Y. Chen, R. Jia, P. Mittal, and P. Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=hTEGyKf0dZ.

[68] Y. Qin, S. Liang, Y. Ye, K. Zhu, L. Yan, Y.-T. Lu, Y. Lin, X. Cong, X. Tang, B. Qian, S. Zhao, R. Tian, R. Xie, J. Zhou, M. H. Gerstein, D. Li, Z. Liu, and M. Sun. Toolllm: Facilitating large language models to master 16000+ real-world apis. *ArXiv*, abs/2307.16789, 2023. URL https://api.semanticscholar.org/CorpusID:260334759.

[69] B. Radosevich and J. Halloran. Mcp safety audit: Llms with the model context protocol allow major security exploits, 2025. URL https://arxiv.org/abs/2504.03767.

[70] J. Rando, J. Zhang, N. Carlini, and F. Tramèr. Adversarial ml problems are getting harder to solve and to evaluate, 2025. URL https://arxiv.org/abs/2502.02260.

[71] M. Rauh, N. Marchal, A. Manzini, L. A. Hendricks, R. Comanescu, C. Akbulut, T. Stepleton, J. Mateos-Garcia, S. Bergman, J. Kay, C. Griffin, B. Bariach, I. Gabriel, V. Rieser, W. Isaac, and L. Weidinger. Gaps in the safety evaluation of generative ai. In *Proceedings of the 2024 AAAI/ACM Conference on AI, Ethics, and Society*, AIES '24, page 1200–1217. AAAI Press, 2025.

[72] Q. Ren, H. Li, D. Liu, Z. Xie, X. Lu, Y. Qiao, L. Sha, J. Yan, L. Ma, and J. Shao. Derail yourself: Multi-turn llm jailbreak attack through self-discovered clues, 2024. URL https://arxiv.org/abs/2410.10700.

[73] A. Robey, Z. Ravichandran, V. Kumar, H. Hassani, and G. J. Pappas. Jailbreaking llm-controlled robots. *arXiv preprint arXiv:2410.13691*, 2024.

[74] J. Roh, V. Shejwalkar, and A. Houmansadr. Multilingual and multi-accent jailbreaking of audio llms, 2025. URL https://arxiv.org/abs/2504.01094.

[75] M. Russinovich, A. Salem, and R. Eldan. Great, now write an article about that: The crescendo multi-turn llm jailbreak attack. *arXiv preprint arXiv:2404.01833*, 2024. URL https://arxiv.org/abs/2404.01833.

[76] M. Sharma, M. Tong, T. Korbak, D. Duvenaud, A. Askell, S. R. Bowman, E. Perez, and et al. Towards understanding sycophancy in language models. In *Proceedings of ICLR 2024*, 2024.

[77] M. Sharma, M. Tong, J. Mu, J. Wei, J. Kruthoff, S. Goodfriend, E. Ong, A. Peng, R. Agarwal, C. Anil, A. Askell, N. Bailey, J. Benton, E. Bluemke, S. R. Bowman, E. Christiansen, H. Cunningham, A. Dau, A. Gopal, R. Gilson, L. Graham, L. Howard, N. Kalra, T. Lee, K. Lin, P. Lofgren, F. Mosconi, C. O'Hara, C. Olsson, L. Petrini, S. Rajani, N. Saxena, A. Silverstein, T. Singh, T. Sumers, L. Tang, K. K. Troy, C. Weisser, R. Zhong, G. Zhou, J. Leike, J. Kaplan, and E. Perez. Constitutional classifiers: Defending against universal jailbreaks across thousands of hours of red teaming, 2025. URL https://arxiv.org/abs/2501.18837.

[78] A. Sheshadri, A. Ewart, P. Guo, A. Lynch, C. Wu, V. Hebbar, H. Sleight, A. C. Stickland, E. Perez, D. Hadfield-Menell, and S. Casper. Targeted latent adversarial training improves robustness to persistent harmful behaviors in llms. *arXiv preprint arXiv:2407.15549*, 2024.

[79] P. Sikorski, L. Schrader, K. Yu, L. Billadeau, J. Meenakshi, N. Mutharasan, F. Esposito, H. AliAkbarpour, and M. Babaiasl. Deployment of large language models to control mobile robots at the edge, 2024. URL https://arxiv.org/abs/2405.17670.

[80] Y. Song, F. Xu, S. Zhou, and G. Neubig. Beyond browsing: Api-based web agents, 2025. URL https://arxiv.org/abs/2410.16464.

[81] A. Souly, Q. Lu, D. Bowen, T. Trinh, E. Hsieh, S. Pandey, P. Abbeel, J. Svegliato, S. Emmons, O. Watkins, and S. Toyer. A strongreject for empty jailbreaks, 2024.

[82] J. Spataro. Introducing Microsoft 365 Copilot — your copilot for work. https://blogs.microsoft.com/blog/2023/03/16/introducing-microsoft-365-copilot-your-copilot-for-work/, 2023. Microsoft Official Blog, accessed 22 May 2025.

[83] G. Sun, X. Zhan, S. Feng, P. C. Woodland, and J. Such. Case-bench: Context-aware safety benchmark for large language models, 2025. URL https://arxiv.org/abs/2501.14940.

[84] J. Sun, Q. Zhang, Y. Duan, X. Jiang, C. Cheng, and R. Xu. Prompt, plan, perform: Llm-based humanoid control via quantized imitation learning, 2024. URL https://arxiv.org/abs/2309.11359.

[85] H. Trivedi, T. Khot, M. Hartmann, R. R. Manku, V. Dong, E. Li, S. Gupta, A. Sabharwal, and N. Balasubramanian. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. *ArXiv*, abs/2407.18901, 2024. URL https://api.semanticscholar.org/CorpusID:271516633.

[86] E. Wallace, S. Feng, N. Kandpal, M. Gardner, and S. Singh. Universal adversarial triggers for attacking and analyzing NLP. In K. Inui, J. Jiang, V. Ng, and X. Wan, editors, *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2153–2162, Hong Kong, China, 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-1221. URL https://aclanthology.org/D19-1221.

[87] X. Wang, B. Li, Y. Song, F. F. Xu, X. Tang, M. Zhuge, J. Pan, Y. Song, B. Li, J. Singh, H. H. Tran, F. Li, R. Ma, M. Zheng, B. Qian, Y. Shao, N. Muennighoff, Y. Zhang, B. Hui, J. Lin, R. Brennan, H. Peng, H. Ji, and G. Neubig. Openhands: An open platform for ai software developers as generalist agents. In *International Conference on Learning Representations*, 2024. URL https://api.semanticscholar.org/CorpusID:271404773.

[88] Z. Wang, T. Pang, C. Du, M. Lin, W. Liu, and S. Yan. Better diffusion models further improve adversarial training, 2023. URL https://arxiv.org/abs/2302.04638.

[89] Z. Wang, H. Li, R. Zhang, Y. Liu, W. Jiang, W. Fan, Q. Zhao, and G. Xu. Mpma: Preference manipulation attack against model context protocol, 2025. URL https://arxiv.org/abs/2505.11154.

[90] Z. Wang, V. Siu, Z. Ye, T. Shi, Y. Nie, X. Zhao, C. Wang, W. Guo, and D. Song. Agentfuzzer: Generic black-box fuzzing for indirect prompt injection against llm agents, 2025. URL https://arxiv.org/abs/2505.05849.

[91] J. Werner, K. Chu, C. Weber, and S. Wermter. Llm-based interactive imitation learning for robotic manipulation, 2025. URL https://arxiv.org/abs/2504.21769.

[92] S. Wu, S. Zhao, Q. Huang, K. Huang, M. Yasunaga, K. Cao, V. N. Ioannidis, K. Subbian, J. Leskovec, and J. Zou. Avatar: Optimizing llm agents for tool usage via contrastive reasoning, 2024. URL https://arxiv.org/abs/2406.11200.

[93] T. Xie, D. Zhang, J. Chen, X. Li, S. Zhao, R. Cao, T. J. Hua, Z. Cheng, D. Shin, F. Lei, Y. Liu, Y. Xu, S. Zhou, S. Savarese, C. Xiong, V. Zhong, and T. Yu. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *ArXiv*, abs/2404.07972, 2024. URL https://api.semanticscholar.org/CorpusID:269042918.

[94] T. Xie, X. Qi, Y. Zeng, Y. Huang, U. M. Sehwag, K. Huang, L. He, B. Wei, D. Li, Y. Sheng, R. Jia, B. Li, K. Li, D. Chen, P. Henderson, and P. Mittal. Sorry-bench: Systematically evaluating large language model safety refusal. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=YfKNaRktan.

[95] Y. Yao, X. Tong, R. Wang, Y. Wang, L. Li, L. Liu, Y. Teng, and Y. Wang. A mousetrap: Fooling large reasoning models for jailbreak with chain of iterative chaos, 2025. URL https://arxiv.org/abs/2502.15806.

[96] F. Yin, P. Laban, X. Peng, Y. Zhou, Y. Mao, V. Vats, L. Ross, D. Agarwal, C. Xiong, and C.-S. Wu. Bingoguard: Llm content moderation tools with risk levels. *ArXiv*, abs/2503.06550, 2025. URL https://api.semanticscholar.org/CorpusID:276903386.

[97] K. You, H. Zhang, E. Schoop, F. Weers, A. Swearngin, J. Nichols, Y. Yang, and Z. Gan. Ferret-ui: Grounded mobile ui understanding with multimodal llms. In *European Conference on Computer Vision*, 2024. URL https://api.semanticscholar.org/CorpusID:269005503.

[98] Y. Yuan, W. Jiao, W. Wang, J. tse Huang, J. Xu, T. Liang, P. He, and Z. Tu. Refuse whenever you feel unsafe: Improving safety in llms via decoupled refusal training, 2024. URL https://arxiv.org/abs/2407.09121.

[99] W. Zeng, Y. Liu, R. Mullins, L. Peran, J. Fernandez, H. Harkous, K. Narasimhan, D. Proud, P. Kumar, B. Radharapu, O. Sturman, and O. Wahltinez. Shieldgemma: Generative ai content moderation based on gemma, 2024. URL https://arxiv.org/abs/2407.21772.

[100] Y. Zeng, K. Klyman, A. Zhou, Y. Yang, M. Pan, R. Jia, D. Song, P. Liang, and B. Li. Ai risk categorization decoded (air 2024): From government regulations to corporate policies, 2024. URL https://arxiv.org/abs/2406.17864.

[101] Y. Zeng, Y. Wu, X. Zhang, H. Wang, and Q. Wu. Autodefense: Multi-agent llm defense against jailbreak attacks. *ArXiv*, abs/2403.04783, 2024. URL https://api.semanticscholar.org/CorpusID:268297202.

[102] Y. Zeng, Y. Yang, A. Zhou, J. Z. Tan, Y. Tu, Y. Mai, K. Klyman, M. Pan, R. Jia, D. Song, P. Liang, and B. Li. AIR-BENCH 2024: A safety benchmark based on regulation and policies specified risk categories. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=UVnD9Ze6mF.

[103] J. Zhang, T. Lan, M. Zhu, Z. Liu, T. Hoang, S. Kokane, W. Yao, J. Tan, A. Prabhakar, H. Chen, Z. Liu, Y. Feng, T. Awalgaonkar, R. Murthy, E. Hu, Z. Chen, R. Xu, J. C. Niebles, S. Heinecke, H. Wang, S. Savarese, and C. Xiong. xlam: A family of large action models to empower ai agent systems. *arXiv preprint arXiv:2409.03215*, 2024.

[104] Y. Zhang, T. Yu, and D. Yang. Attacking vision-language computer agents via pop-ups, 2025. URL https://arxiv.org/abs/2411.02391.

[105] B. Zheng, B. Gou, J. Kil, H. Sun, and Y. Su. Gpt-4v(ision) is a generalist web agent, if grounded. *ArXiv*, abs/2401.01614, 2024. URL https://api.semanticscholar.org/CorpusID:266741821.

[106] A. Zou, Z. Wang, J. Z. Kolter, and M. Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023. URL https://arxiv.org/abs/2307.15043.

[107] A. Zou, L. Phan, J. Wang, D. Duenas, M. Lin, M. Andriushchenko, R. Wang, Z. Kolter, M. Fredrikson, and D. Hendrycks. Improving alignment and robustness with circuit breakers, 2024. URL https://arxiv.org/abs/2406.04313.