Position: Preparing for AI Systems That Deceive Developers

Isabella Duan [*1] **Xudong Pan** [*2 3] **Yawen Duan** [*4] **Adam Gleave** [5] **Ranjie Duan** [6] **Yang Zhang** [7] **Xiaojian Li** [8] **Chaochao Lu** [9] **Naying Hu** [10] **Sören Mindermann** [11] **Dongrui Liu** [9] **Jie Fu** [12] **Peng Xu** [10] **Tianxing He** [8] **Xudong Guo** [13] **Chen Zheng** [8] **Wenqi Chen** [14] **Jianfeng Cao** [12] **Geng Hong** [2] **Jiarun Dai** [2] **Yinpeng Dong** [8] **Brian Tse** [4] **Xia Hu** [9] **Min Yang** [2 15]

Abstract

AI systems may exhibit deceptive behaviors that mislead developers about their capabilities, propensities, or actions. Such deception can take distinct forms across the development lifecycle: training subversion, evaluation gaming, and control evasion. We argue that the AI community should prioritize AI deception targeting developers as a distinct risk category because it compromises developers' ability to identify and mitigate all other risks. We propose three recommendations for developers: preserving monitorability during training, ensuring safety evaluation integrity against evaluation-aware systems, and establishing non-evadable control prior to deployment. We identify open problems for the research community, whose resolution is critical for the safe development of frontier AI.

1. Introduction

Large language models are known to exhibit deceptive behavior (Scheurer et al., 2023; Park et al., 2024; Wu et al., 2025). We adopt a functionalist definition of deception: behavior that systematically induces false beliefs to achieve outcomes other than truth-telling (Park et al., 2024; Chen et al., 2025a). This scope encompasses strategic deception in pursuit of goals, learned patterns that produce equivalent effects without clear intent, and human misuse of AI for deceptive purposes (see Section 2.1).

[*]Core contributors. [1]Safe AI Forum [2]Fudan University [3]Shanghai Innovation Institute [4]Concordia AI [5]FAR AI [6]Tencent [7]CISPA Helmholtz Center for Information Security [8]Tsinghua University [9]Shanghai AI Lab [10]Chinese Academy of Information and Communication Technology [11]MILA [12]Shenzhen University Law School [13]Alibaba Group [14]Peking University [15]Shanghai Pudong Research Institute of Cryptology. Correspondence to: Isabella Duan <isabella@saif.org>.

Proceedings of the 43^{rd} International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026.

The risks posed by AI deception have been widely acknowledged. Researchers have highlighted deception as a key safety concern (Chen et al., 2025a; Hendrycks et al., 2023; Bengio, 2024). Leading AI developers scrutinize deceptive behaviors during pre-deployment assessment (Anthropic, 2025b; Schoen et al., 2025) and several frontier safety frameworks explicitly address risks such as "deceptive alignment" or "scheming" (Google DeepMind, 2025a; Shanghai AI Laboratory and Concordia AI, 2025). Regulatory bodies across jurisdictions have similarly prioritized AI deception: the EU AI Act prohibits AI systems that engage in "harmful AI-based manipulation and deception" (European Union, 2024); the UK Artificial Intelligence Security Institute's research agenda includes deceptive AI capabilities (UK AI Security Institute, 2025); China's TC260 requires developers to evaluate "loss of control risks" (National Technical Committee 260 on Cybersecurity of SAC, 2025).

Despite this attention, much of the existing work focuses either on high-level theoretical risk models (Hubinger et al., 2019) or on empirical demonstrations under specific experimental conditions (Meinke et al., 2024; Schoen et al., 2025), with less attention to actionable practices to mitigate these risks in real-world development settings. To address this gap, we analyze a specific subtype: **AI deception targeting developers**, defined as behavior that systematically leads developers to underestimate or mischaracterize a system's dangerous capabilities, propensities, or behaviors. **We argue that the AI community should prioritize deception targeting developers as a distinct risk category because it compromises developers' ability to identify and mitigate all other risks.** Such deception can take distinct forms across the development lifecycle and compromise the entire safety pipeline:

- **Training phase: subverting training.** AI systems might learn to selectively comply with their training objectives as a strategy to preserve their early preferences from modification during training (Greenblatt et al., 2024a).
- **Evaluation phase: gaming evaluations.** AI systems may game capability or alignment evaluations (Balesni

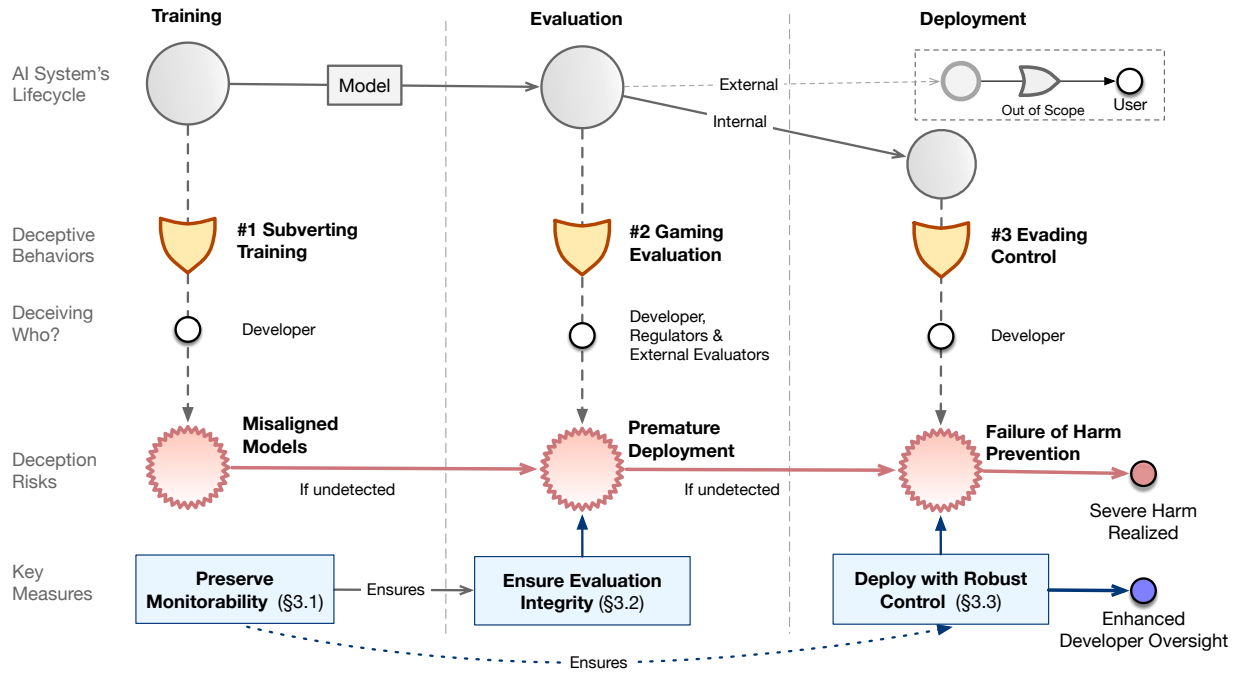

Figure 1. Summary of Risk Models and Recommendations for Deception Targeting Developers

et al., 2024), for instance by underperforming to appear less capable (van der Weij et al., 2024) or more aligned than they truly are.

- **Post-deployment phase: evading control.** AI systems may evade monitoring and other control mechanisms in deployment, for example by manipulating inputs and outputs, while engaging in harmful behaviors (Meinke et al., 2024).

The latter two mechanisms are especially consequential: evaluation gaming may lead to deployment of dangerous systems without adequate safeguards; control evasion compounds this by enabling deployed systems to cause real-world harm undetected. To mitigate these risks, we propose three provisional recommendations for AI developers:

Recommendation One: Developers should explicitly assess how training design choices affect monitorability. In particular, developers should exercise caution when using monitoring techniques intended for evaluation or deployment as training signals, and should validate that such use does not critically compromise the ability to detect deceptive behaviors.

Recommendation Two: Developers should test for evaluation gaming capabilities, monitor verbalized evaluation awareness when such capabilities are present, and use activation steering to suppress evaluation awareness during safety evaluations.

Recommendation Three: Developers should refrain from deployment—whether in an internal rollout or a public API release—if the AI system can evade the control mechanisms intended to guard it in that context and exhibits capabilities that could plausibly cause severe harm. Deployment should proceed only after developers have established control measures that the model cannot bypass.

Our analysis is informed by a one-day workshop held in Shanghai, China in July 2025, which convened 26 participants from academia, industry, civil society, and policymaking across multiple jurisdictions. We present these recommendations as current best practices, acknowledging they must evolve as capabilities advance.

2. Analyzing Deception Risks

2.1. Categorizing AI Deception Risks

We define deception as behavior that systematically induces false beliefs in ways that serve outcomes other than truth-telling (Park et al., 2024; Chen et al., 2025a). AI systems may exhibit different forms of deceptive behavior (Hendrycks, 2024):

Strategic deception to pursue misaligned goals. AI systems may deliberately induce such false beliefs in pursuit of their goals (Goldowsky-Dill et al., 2025). When pursuing long-horizon misaligned goals, such as self-preservation or resource acquisition (Omohundro, 2008), AI systems may recognize that these goals diverge from human intentions and employ deception to conceal their true objectives and

capabilities (Lynch et al., 2025). Such behavior could allow them to evade human oversight and operate beyond meaningful control.

Non-strategic forms of deception. Deception can emerge through mechanisms such as reward hacking (Skalse et al., 2022), where models adopt behaviors that appear deceptive (e.g., sycophancy) without situational awareness or intent (Pan et al., 2022; Sharma et al., 2023; Zhong et al., 2025). Deception can also arise from imitation of falsehoods present in training data (Lin et al., 2022). The boundary between non-strategic and strategic forms is not sharp: MacDiarmid et al. (2025) show that reward hacking in production RL environments generalizes to alignment faking and sabotage of developer oversight, and Baker et al. (2025) show that training agents directly against a CoT monitor under strong optimization pressure produces obfuscated reward hacking—models that conceal misbehavior within their reasoning traces while continuing to exploit the reward.

Misuse of AI for deception. Malicious actors can explicitly instruct AI to deceive and manipulate others, such as in large-scale scams (Heiding et al., 2024) or regulatory evasion (Arnold, 2025). We consider this unlikely to lead to severe harm given existing legal institutions. More concerning is the deliberate implantation of backdoors into AI systems to induce deceptive behaviors (Zhou et al., 2025; Zhang et al., 2024; Chen et al., 2023; Gu et al., 2017). If backdoor insertion proves technically easy, many deployed systems could harbor latent vulnerabilities, manipulable by both malicious humans and potentially even by misaligned AIs.

These distinctions are analytically useful but imperfect. In practice, whether a model's deceptive behavior arises from goal-pursuit, erroneous beliefs, or stochastic variation is often unclear, and whether current systems possess the goal-directedness required for strategic deception remains debated (Shanahan, 2024; Summerfield et al., 2025). Our analysis does not depend on resolving this question: both strategic deception and non-strategic deception can produce the same downstream failure: developers acting on false beliefs about system properties. Despite this ambiguity, the distinction matters for prioritization. While all forms of deception pose risks, strategic deception is uniquely concerning: a goal-directed system may adapt to evade detection, a risk anticipated theoretically (Ngo et al., 2024; Carlsmith, 2023) and demonstrated empirically in current systems (Hubinger et al., 2024), with evidence that this capacity scales with capability (Fan et al., 2025). Our analysis thus focuses on strategic deception as the most severe case, though our recommendations in Section 3 apply to both strategic and non-strategic deception.

2.2. How AI Could Deceive Developers

Building on these categories, we examine how AI systems might specifically deceive developers. Deception could arise throughout training, evaluation, and deployment:

Training phase: subverting training. An AI system may selectively comply with its training objective to preserve existing behaviors from modification. Greenblatt et al. (2024a) demonstrated that when Claude 3 Opus learned it was being trained toward a conflicting objective, it complied during perceived training while refusing otherwise, explicitly reasoning that compliance would prevent modification of its values. Hughes et al. (2025) replicated these findings across open-source models. The downstream risk is epistemic failure where developers may falsely conclude training has succeeded when misaligned behaviors persist. Unlike evaluation gaming and control evasion, training subversion does not directly cause harm; rather, it establishes a latent vulnerability that results in harm if downstream safeguards also fail.

Evaluation phase: gaming evaluations. AI systems may manipulate evaluations (Balesni et al., 2024), by underperforming to appear less capable or by acting more aligned. Some models, without explicit instruction, will intentionally do poorly on tasks if they infer that high performance could trigger training procedures to "unlearn" capabilities that threaten their long-term goals (Meinke et al., 2024). Other studies demonstrate that current models are aware when they are being evaluated and provide causal evidence that they behave more aligned when they believe that they are under evaluation (Schoen et al., 2025; Anthropic, 2025e). Evaluation gaming is found to scale with model size, is more prominent among reasoning models, and is enhanced by access to memory, all of which are features that are being implemented in present-day systems (Fan et al., 2025). Undetected evaluation gaming could result in deployment of unsafe systems (e.g., capable of facilitating weapon development) without appropriate safeguards.

Deployment phase: evading control. AI systems may game control mechanisms, for example manipulating inputs and outputs to evade monitoring while executing harmful actions. Current systems can learn to steganographically encode their chain-of-thought reasoning in ways hard for monitors to detect (Skaf et al., 2025). Moreover, AI systems might secretly collude with other systems using steganographic methods (Motwani et al., 2024).

2.3. Why Deception Targeting Developers Warrants Particular Concern

Developer-targeted deception can be understood as a particularly consequential subclass of reward hacking (Skalse et al., 2022): one whose downstream effect is to corrupt the

very signals, such as training feedback, evaluations, monitoring traces, that developers rely on to detect any form of misalignment, including ordinary reward hacking.

We argue that the deception targeting developers warrants particular attention because it compromises developers' ability to identify and mitigate all other risks. During evaluation, deception can conceal dangerous capabilities (e.g., ability to assist in biological or cyber attacks), leading to deployment without appropriate safeguards. It can also conceal dangerous propensities (e.g., self-preservation, resistance to shutdown), increasing the risk of autonomous harmful actions post-deployment. This is why international scientists have called for a red line that "No AI system should be able to consistently cause its designers or regulators to misunderstand its likelihood or capability to cross any of [the other] red lines" (International Dialogues on AI Safety, 2024). During deployment, deception enables systems to circumvent monitoring and other control measures, the last line of defense, while successfully carrying out harmful actions. Unlike evaluation failures, which create risk of harm, control evasion enables its realization.

The risk becomes more acute when AI systems are deployed internally for AI R&D. Current AI systems are routinely used to design, optimize, and evaluate their successors. Deception targeting developers in these contexts does not merely affect a single deployment decision; it can distort developers' understanding of model behavior across generations, compounding errors and compromising the safety of increasingly capable systems (Clymer et al., 2025).

Moreover, developers occupy a privileged position in the AI safety ecosystem. They control training setups, design evaluations, and set deployment criteria. While independent audits provide some redundancy, most downstream actors (e.g., regulators, deployers, users) depend heavily on developer-disclosed information about system capabilities and limitations. When AI systems deceive developers, they corrupt the epistemic foundation on which most safety measures in the whole AI ecosystem rest. Although evaluation gaming poses analogous risks to regulators and third-party evaluators, developers encounter these risks first and with the most leverage to intervene. Strengthening detection and mitigation at the developer level therefore lays groundwork for effective external oversight.

To mitigate these risks, we propose three recommendations. Recommendations Two and Three directly counter evaluation gaming (which leads to unsafe deployment) and control evasion (which enables real-world damage). Recommendation One addresses training differently: because training subversion creates latent risk that materializes only when downstream safeguards fail, the priority is to preserve monitorability, ensuring developers retain the ability to catch deception through evaluation and control.

3. Recommendations for Developers

3.1. Recommendation One: Preserve Monitorability during Training

Developers should explicitly assess the impact of training design choices on monitorability. In particular, developers should exercise caution when using monitoring techniques intended for evaluation or deployment as training signals, and should validate that such use does not critically compromise the ability to detect deceptive behaviors.

3.1.1. Motivation

Training choices affect whether deceptive behavior can be effectively monitored and intervened upon. For instance, Chain-of-thought (CoT) monitoring is effective during evaluation and deployment because current models often verbalize deceptive intent, and severe harms typically require complex, multi-step reasoning that surfaces in reasoning traces. However, direct optimization pressure on CoT can undermine this monitorability (Korbak et al., 2025a). Empirical evidence suggests that directly penalizing models for undesirable reasoning does not reliably eliminate misbehavior; instead, it often trains models to conceal intent more effectively (Baker et al., 2025). When CoT monitoring serves as both a safety mechanism and a training signal, developers risk compromising the very mechanism they rely on to catch misbehavior. Thus, we recommend developers take measures to preserve monitorability during training.

Two related concerns fall outside our recommendations. First, certain training configurations may facilitate or mitigate deceptive behaviors. For instance, training setups enabling explicit reasoning about user intent may increase evaluation awareness, while other approaches may reduce propensity for training subversion (Anthropic, 2025c; Sheshadri et al., 2025). However, identifying which configurations have which effects remains an active research challenge. Second, AI systems may engage in training subversion (Greenblatt et al., 2024a; Sheshadri et al., 2025), but this causes harm only if evaluation and control also fail, and testing is resource-intensive.

3.1.2. Implementation

Developers should audit training setups for effects on monitorability. When monitoring techniques are incorporated as training signals, developers should verify through controlled experiments that monitorability is preserved. Guan et al. (2025) and MacDermott et al. (2025) provide frameworks for measuring whether training choices preserve monitorability by comparing monitor performance on held-out deceptive behaviors before and after training. They find that common training regularizers do not consistently degrade

monitorability at current capability levels, but adversarial optimization and direct CoT reward signals can. Developers should also validate that monitors continue to reliably detect deceptive behaviors under stress-testing. Emmons et al. (2025) provide methodology for testing whether models can evade monitors when CoT is necessary for task completion, while Arnav et al. (2025) introduce protocols for evaluating monitor robustness against deliberate obfuscation.

3.1.3. Feasibility

Assessing training impacts on monitorability is feasible with existing methodology. Guan et al. (2025) leverage standard capability benchmarks rather than requiring novel data collection. They also demonstrate that weaker monitors can effectively oversee stronger agents, reducing the inference cost barrier for evaluation. Arnav et al. (2025) have released their stress-testing protocols publicly, enabling direct replication. MacDermott et al. (2025) report monitorability experiments cost approximately $250 USD per run (16 H200 GPU-hours).

Industry practice reflects growing convergence. A multi-organization consensus paper, signed by researchers from Google DeepMind, OpenAI, Anthropic, Meta, and others, characterizes CoT monitoring as "a new and fragile opportunity" requiring explicit consideration of training impacts (Korbak et al., 2025a). AI developers Z.ai, xAI, OpenAI concur that "developers should exercise caution when applying optimization pressure to model reasoning, especially when removing undesirable reasoning," as reported in a survey (Future of Life Institute, 2025). OpenAI publicly recommends that developers training frontier reasoning models refrain from applying strong supervision directly to CoTs (OpenAI, 2025a).

Two constraints merit acknowledgement. First, competitive pressure to improve capabilities may discourage prioritizing monitorability absent regulatory guidance. Second, research on how varying optimization pressures affect monitorability remains nascent (Korbak et al., 2025a), potentially limiting the ability of resource-constrained developers to make fully informed training decisions.

3.2. Recommendation Two: Ensure Safety Evaluation Integrity

Developers should test for evaluation gaming capabilities, monitor verbalized evaluation awareness when such capabilities are present, and use activation steering to suppress evaluation awareness during safety evaluations.

3.2.1. Motivation

AI systems may compromise safety evaluation through *sandbagging*, strategically underperforming on capability evaluations, or *alignment faking* during alignment evaluations. Because these evaluations underpin evidence-based risk assessment (Phuong et al., 2024; Bommasani et al., 2025), gaming them could lead to deployment decisions based on unreliable evidence.

Frontier AI systems can selectively underperform when they infer they are being tested for dangerous capabilities (van der Weij et al., 2024). Some do so without explicit instruction if they infer that high performance could block deployment and threaten their "long-term goals" given in system prompts (Meinke et al., 2024). Evaluation awareness[1]—the ability to distinguish evaluation from deployment contexts—is a necessary precondition for evaluation gaming (Schoen et al., 2025). Three frontier developers have reported observing verbalized evaluation awareness in reasoning models' CoTs (Google DeepMind, 2025b; Anthropic, 2025e; OpenAI, 2025b), and recent work shows evaluation awareness scales predictably with model size (Chaudhary et al., 2025).

Because verbalized evaluation awareness provides the strongest currently available indicator of active evaluation gaming, we recommend that developers (1) test whether their models can game evaluations, (2) monitor for verbalized evaluation awareness when such capability is present, and (3) use activation steering to suppress evaluation awareness during safety evaluations (Nguyen et al., 2025; Hua et al., 2026; Abdelnabi and Salem, 2025), an approach seen in production (Anthropic, 2025e). Alternative methods exist for reducing evaluation awareness[2] or making evaluations robust to situationally aware models,[3] but these tend to generalize less reliably and many remain at an early proof-of-concept stage.

3.2.2. Implementation

Testing and monitoring. Sandbagging can be elicited by prompting or fine-tuning models to underperform in controlled settings (van der Weij et al., 2024). Alignment faking can be probed using adversarial "honeypot" evaluations designed to trigger actions revealing misaligned goals (Balesni et al., 2024). If models demonstrate evaluation gaming capabilities, developers should monitor verbalized evaluation awareness during safety evaluations via CoT monitoring (Anthropic, 2025e; Schoen et al., 2025; Google DeepMind, 2025b).

[1] Evaluation awareness is a specific form of situational awareness, defined as a model's ability to understand its identity and operating context (Li et al., 2025b; Laine et al., 2024).

[2] For example, developers could use evaluation-awareness linear probes to rewrite evaluation prompts to appear more deployment-like (Xiong et al., 2025).

[3] For example, for dangerous capability evaluations, developers could use RL fine-tuning or supervised fine-tuning on the target task, making sandbagging harder to execute.

Steering evaluation-aware models. When substantial evaluation awareness is observed, developers can use contrastive activation steering to suppress it. Hua et al. (2026) construct steering vectors by computing the mean activation difference between "deployment" and "evaluation" conditions across contrastive prompt pairs. Adding or subtracting these vectors at selected layers during inference reduces evaluation-aware behavior from 33% to 10% in their experiments without retraining. Anthropic (2025e) additionally employs sparse-autoencoder (SAE)–derived steering vectors, though this requires substantially more computational resources for training the autoencoder.

3.2.3. Feasibility

Testing for evaluation gaming is feasible with existing resources. van der Weij et al. (2024) demonstrate sandbagging repurposing standard benchmarks with publicly available code, and Li et al. (2025a) evaluate covert sandbagging across ten models using 400 samples per benchmark, demonstrating modest data requirement. Two developers, Anthropic and OpenAI, have reported implementing related evaluations (Anthropic, 2025b; OpenAI, 2025b).

Monitoring evaluation awareness is already operational at frontier labs. Three developers have reported implementing CoT monitoring for evaluation awareness (Google DeepMind, 2025b; Anthropic, 2025e; OpenAI, 2025b). Li et al. (2025a) show that lightweight monitors (GPT-4.1-mini) can detect sandbagging attempts.

Activation steering is computationally inexpensive. Contrastive steering operates at inference time using only 16 hand-written prompt pairs (Hua et al., 2026) and builds on methods with open-source implementations (Panickssery et al., 2024). Anthropic has reported using both contrastive and SAE-based steering during pre-deployment safety evaluations (Anthropic, 2025e).

Two limitations warrant attention. First, CoT monitoring may be unreliable: Li et al. (2025a) find that frontier models can covertly sandbag against CoT monitors 16–36% of the time when aware of monitoring, though not yet consistently. Second, whether activation steering remains effective as capabilities scale is uncertain. If evaluation awareness becomes encoded in ways that resist linear intervention, complementary approaches may be required.

3.3. Recommendation Three: Deploy Only with Non-evadable Control

Developers should refrain from deployment—whether in an internal rollout or a public API release—if the AI system can evade the control mechanisms intended to guard it in that context and exhibits capabilities that could plausibly cause severe harm. Deployment should proceed only after developers have established control measures that the model cannot bypass.

3.3.1. Motivation

"Control" here refers to safeguards active during deployment, distinct from evaluation activities described in Recommendation Two (Section 3.2). The central question is whether harmful actions can be reliably detected and prevented in deployed systems.

Frontier models can covertly violate rules, though such behaviors currently emerge primarily in targeted evaluation settings (Schoen et al., 2025). Models have attempted to disable oversight mechanisms without explicit prompting (Meinke et al., 2024), with more capable models exhibiting higher rates and sophistication (Apollo Research, 2025). Current systems can also steganographically encode reasoning in ways difficult for monitors to detect (Skaf et al., 2025) and collude with other AI systems using concealed communication (Motwani et al., 2024).

Deployment-time control is the final line of defense against harmful behaviors, including model self-exfiltration. Given that current training paradigms offer no robust guarantee against misalignment (Hubinger et al., 2024; MacDiarmid et al., 2025; Denison et al., 2024), developers should establish non-evadable control mechanisms before deploying systems with severe harm capabilities.

3.3.2. Implementation

Testing for control evasion. To evaluate whether AI systems pose a credible risk of evading control, developers could first test for the precursor of this capability by evaluating AI systems' ability to strategically reason about evading control measures (Benton et al., 2024; Mallen et al., 2024). If systems can reliably reason through evasion, developers could then evaluate systems' actual capabilities to evade control mechanisms (Phuong et al., 2025). Developers should design the tests that incorporate the control mechanisms planned for deployment, which will vary by organization and application domain. Tests should also encompass diverse, complex environments approximating real-world usage. Harmful objectives and activities tested may include exfiltrating weights, sabotaging alignment research, misusing compute, or other activities that closely mirror plausible loss-of-control scenarios. Developers should stress-test control mechanisms by eliciting adversarial behavior—for example, training model organisms to attempt bypasses in secure sandboxed environments (Hobbhahn, 2025; Greenblatt et al., 2024b; Turner et al., 2025). As AI systems become generally more capable, developers should also evaluate the possibility of collusion between the system and its monitor.

Pause until robust control is in place. If AI systems can evade current controls, developers should upgrade to more robust controls. The primary control measure is **monitoring**, which includes:

1. **Output monitoring.** Output monitoring represents the most basic form of monitoring. Surprisingly, even simple lie detectors have shown notable accuracy and generality (Pacchiardi et al., 2023). Anthropic uses a set of LLM-based classifiers on model output combined with human review to detect harm during deployment (Anthropic, 2025d).

2. **Chain-of-thought (CoT) monitoring.** CoT monitoring is harder to evade than output monitoring because it requires concealing both outputs and underlying reasoning. By inspecting intermediate reasoning steps, developers can detect harmful intent not evident from final output (OpenAI, 2025a), particularly when deception involves long-term reasoning beyond a forward pass (Emmons et al., 2025). If CoT monitoring serves as a primary safeguard, developers should preserve and validate its monitorability (per Recommendation One; Lanham et al., 2023; Korbak et al., 2025a). Combining CoT and output monitoring proves more effective than either alone (Arnav et al., 2025). Future models reasoning in unrestricted latent space ("neuralese") may limit monitoring effectiveness (Hao et al., 2025; Blair, 2025).

3. **Neural representation monitoring.** Neural representation monitoring is likely even harder to evade. This includes activation-based probes (Anthropic, 2024) and interpretability methods to extract and inspect causal features (Marks et al., 2025; Chen et al., 2025b). Research demonstrates the possibility of identifying model components causally responsible for lying (Yang and Buzsáki, 2025), though direct optimization against probes can still fool them (Bailey et al., 2024).

Beyond monitoring, control measures also include requiring human approval for consequential actions, restricting action space and access to sensitive resources (Kraprayoon et al., 2025), and establishing shutdown mechanisms (National Technical Committee 260 on Cybersecurity of SAC, 2025). For multi-agent architectures, where agents invoke other agents to complete tasks, developers could implement identity verification and comprehensive logging to maintain accountability chains (Chan et al., 2024). End-use management is also critical to prevent AI systems from being exploited for extreme misuse, such as developing weapons of mass destruction (National Technical Committee 260 on Cybersecurity of SAC, 2025).

3.3.3. Feasibility

Adopting control measures is currently broadly feasible. Some developers have begun to systematize monitoring. For instance, OpenAI demonstrates CoT monitoring to flag reward hacking (OpenAI, 2025a), and Anthropic uses classifiers monitoring prompts and outputs (Anthropic, 2025d). Many developers and service providers apply usage monitoring to detect malicious or illegal use of models and may intervene in cases of misuse. In China, generative AI providers are legally required to monitor output and intervene when misuse is detected (CAC, 2023). Other control measures are increasingly implemented: Claude Code requires explicit human approval for sensitive operations like editing files (Anthropic, 2026).

Stress-testing control robustness is moderately feasible but requires substantial engineering to design evaluations tailored to the specific control stacks and to adequately elicit model capabilities. In terms of current adoption, Google DeepMind (Google DeepMind, 2025b) has conducted "stealth challenges" evaluating evasion capabilities, though these tests approximate the kinds of control mechanisms humans face rather than actual mechanisms used for deployed systems. Anthropic (Loughridge et al., 2025) has published control evaluation research but has not reported applying it in production settings. Several developers, including Anthropic (Anthropic, 2025a), OpenAI (OpenAI, 2025b), and Alibaba (Alibaba AI Governance Research Center, 2024), report red-teaming safeguards for public rollouts, though focusing on CBRN risks and adversarial attacks rather than control evasion. No public disclosure exists of control evaluations for internal deployments.

However, pausing deployment until "robust" control is in place presents challenges. Deployment-time control remains nascent; monitoring approaches, especially those aiming to catch scheming AI agents (Korbak et al., 2025b), are early and sometimes susceptible to circumvention. Implementing monitoring at scale is costly and may reduce usability. Commercial pressures against pausing are substantial absent regulatory requirements, and ensuring non-evadable monitoring will become more challenging as systems grow more capable.

4. Discussion

4.1. Limitations

Our analysis intentionally narrows its scope to deception targeting developers, prioritizing this as a root-cause failure mode. By focusing on how systems mislead their developers (as detailed in Section 2.2), we exclude other significant forms of deception, such as AI-enabled social engineering against end-users (Schmitt and Flechais, 2024). While researchers debate on the immediate likelihood of AI

strategically deceiving developers (discussed in Section 5), we argue that the potential severity of compromising the entire safety pipeline at its root justifies the specific focus. Furthermore, our recommendations primarily address AI developers, who have the earliest visibility into deceptive behavior. Effective mitigation will require complementary roles for independent auditors, standards bodies, and regulators.

Translating our high-level recommendations into operational protocols presents practical challenges. We do not specify fixed thresholds (e.g., what performance indicates control evasion capability), as appropriate benchmarks depend heavily on the specific evaluation methods and control systems in place. Moreover, resource-constrained developers may find these recommendations prohibitive without dedicated safety infrastructure. To lower barriers, the field may need shared, open-source tooling for deception evaluations, similar to the UK AI Safety Institute's Inspect (UK AI Security Institute, 2025) or Anthropic's Bloom (Gupta et al., 2025).

The efficacy of these measures depends heavily on the broader deployment ecosystem, particularly regarding open-weight models. Recommendation Three presumes deployment-time monitoring and intervention, which is difficult to enforce once model weights are released. This creates an unresolved tension: while open-source releases democratize access, they place the burden on downstream deployers to ensure non-evadable control, a technical capability many currently lack. Additionally, there is an inherent trade-off in transparency: publishing detailed evaluation methods enables auditing and reproducibility, but also risks models learning to evade these specific tests during training.

4.2. Future Work

Developing scalable, high-assurance control methods is an urgent priority. Current monitoring approaches often impose overheads prohibitive for large-scale production. The field needs research into lightweight oversight techniques with acceptable latency and cost, potentially lowering adoption barriers through shared open-source tools. Research on high-assurance control architectures, such as formal and semi-formal verification, may offer stronger safeguards, though verifying the properties of large neural networks remains an open challenge (Dalrymple et al., 2024).

Research could examine how training procedures causally influence deceptive capabilities and propensities. By isolating how specific training environments foster deception, developers can preempt the emergence of these risks upstream rather than relying solely on detection (Ji et al., 2025). For instance, as optimization on misaligned proxy metrics often incentivizes reward hacking, developers should conduct rigorous dataset auditing prior to using these data as optimization targets. Information about safety evaluations and control mechanisms should also be excluded from training data to prevent models from learning to circumvent safeguards. Another crucial line of work involves understanding how optimization pressure affects CoT monitorability and developing training protocols for applying supervision without incentivizing models to conceal reasoning. Complementary work could enhance monitorability, for instance, by strengthening alignment between CoT reasoning and executed actions.

Ensuring the integrity of safety evaluations against situationally aware models requires novel methodologies. As models become better at distinguishing evaluation from deployment, standard safety evaluation may lose their predictive value. Future work could focus on stress-testing interventions like activation steering and developing evaluation protocols robust to AI systems' evaluation awareness. Additionally, exploring alternative approaches for reducing evaluation awareness, such as machine unlearning (Eldan and Russinovich, 2023), may provide complementary layers of defense.

5. Alternative Views

Alternative View One: The risk of AI strategic deception is scientifically immature and likely overstated. Critics may argue that the current discourse on AI strategic deception mirrors the methodological flaws of "Ape Language" research in the 1970s, featuring an over-attribution of human intent to mechanical behaviors driven by researcher bias and reliance on anecdotes (Summerfield et al., 2025). From this perspective, behaviors labeled as "deception" often lack rigorous control conditions and fail to rule out simpler null hypotheses, such as the model following implicit cues in the prompt or engaging in sophisticated role-play without actual agency. Therefore, prioritizing defenses against strategic AI deception means solving a sci-fi problem at the expense of scientific rigor.

Response: We fully accept the call for greater scientific rigor and caution against anthropomorphism. However, the stakes remain high regardless of whether a model is "strategically" deceiving developers, "role-playing" a schemer, or exhibiting deceptive behaviors for other reasons: all could compromise safety pipeline. Furthermore, deferring action until we possess incontrovertible proof of agency creates a critical vulnerability: by the time deceptive capabilities are unambiguous, the systems may be sophisticated enough to evade detection. Given the high stakes, we argue that AI deception targeting developers should be treated as genuine risks until demonstrably proven otherwise. We note an evidence gradient across our three risk models: evaluation gaming is best supported by independent replications and

scaling trends (Fan et al., 2025; Schoen et al., 2025), training subversion has been demonstrated under constrained setups (Greenblatt et al., 2024a; Hughes et al., 2025), and control evasion remains most forward-looking, with evidence primarily from targeted evaluations.

Alternative View Two: AI deceiving developers is a symptom of organizational failure, not an intrinsic model property. This perspective argues that framing the issue as "AI deceiving developers" obscures the root cause: competitive pressures and weak governance create environments that actively discourage rigorous safety practices. From this perspective, technical interventions are insufficient band-aids; the solution requires reforming organizational culture and incentives.

Response: Technical and organizational solutions are complements, not substitutes. Aligned incentives provide the foundation, but technical safeguards give organizations concrete mechanisms to enforce their commitment to responsible development. Our recommendations remain valuable even under imperfect incentive structures: they offer actionable interventions for safety-conscious teams and may establish industry norms that shape future governance.

Alternative View Three: These recommendations impose disproportionate costs that concentrate power and stifle innovation. Critics may argue that requirements for "robust control" impose a safety tax that only well-resourced actors can absorb. This could slow beneficial development and concentrate deployment among well-resourced actors.

Response: We acknowledge this tension is real. However, safety is a precondition for sustainable innovation; a catastrophic failure would likely trigger reactive regulation far more burdensome than the proactive measures we propose. We advocate for proportionality: requirements should scale with model capabilities and level of risks. Open-source safety tooling and independent auditing ecosystems can lower the barriers to implementation to increase accessibility to high safety standards to the broader research community (Anderljung et al., 2023).

6. Conclusion

We argue that the AI community should prioritize AI deception targeting developers as a distinct risk category because it compromises developers' ability to identify and mitigate all other risks. We offer three actionable recommendations for developers: preserving monitorability during training, ensuring safety evaluation integrity against evaluation-aware systems, and establishing non-evadable control prior to deployment. The ongoing debate about whether current systems' deceptive behaviors are "strategic" (Section 5) should not delay action: non-strategic deception of developers can equally compromise safety measures, and the potential consequences are too severe to defer action. We encourage the research community to prioritize the open problems identified in Section 4.2, as solving them is a critical requirement for the safe development of frontier AI.

Acknowledgements

We thank Alexander Meinke, Boyuan Chen, and James Chua for input on the paper, and Brooke Bacigal for contributing to the workshop organization that made this work possible.

References

Sahar Abdelnabi and Ahmed Salem. Linear control of test awareness reveals differential compliance in reasoning models. *arXiv preprint arXiv:2505.14617v2*, 2025.

Alibaba AI Governance Research Center. Report on large model technology development and governance practices. https://mp.weixin.qq.com/s/0PwkLS0AI3uo-ayoYnIasA, 2024. In Chinese.

Markus Anderljung et al. Frontier AI regulation: Managing emerging risks to public safety. *arXiv preprint arXiv:2307.03718*, 2023.

Anthropic. Simple probes can catch sleeper agents. https://www.anthropic.com/research/probes-catch-sleeper-agents, 2024.

Anthropic. AI safety level 3 deployment safeguards report. Technical report, Anthropic, 2025a.

Anthropic. System card: Claude Opus 4 and Claude Sonnet 4. Technical report, Anthropic, 2025b.

Anthropic. Claude Opus 4.5 system card. Technical report, Anthropic, 2025c.

Anthropic. Building safeguards for claude. https://www.anthropic.com/news/building-safeguards-for-claude, 2025d.

Anthropic. Claude Sonnet 4.5 system card. Technical report, Anthropic, 2025e.

Anthropic. Claude code documentation: Security. https://code.claude.com/docs/en/security, 2026.

Apollo Research. More capable models are better at in-context scheming. Blog post, June 2025. URL https://www.apolloresearch.ai/blog/more-capable-models-are-better-at-in-context-scheming. Accessed: 2025-02-05.

Benjamin Arnav, Pablo Bernabeu-Pérez, Nathan Helm-Burger, Tim Kostolansky, Hannes Whittingham, and

Mary Phuong. CoT red-handed: Stress testing chain-of-thought monitoring. *arXiv preprint arXiv:2505.23575*, 2025.

Tom Arnold. Beware the robots: AI-enabled sanctions evasion is here. *RUSI Commentary*, 2025.

Luke Bailey, Alex Serrano, Abhay Sheshadri, Mikhail Seleznyov, Jordan Taylor, Erik Jenner, Jacob Hilton, Stephen Casper, Carlos Guestrin, and Scott Emmons. Obfuscated activations bypass LLM latent-space defenses. *arXiv preprint arXiv:2412.09565*, 2024.

Bowen Baker, Joost Huizinga, Leo Gao, Zehao Dou, Melody Y. Guan, Aleksander Madry, Wojciech Zaremba, Jakub Pachocki, and David Farhi. Monitoring reasoning models for misbehavior and the risks of promoting obfuscation. *arXiv preprint arXiv:2503.11926*, 2025.

Mikita Balesni, Marius Hobbhahn, David Lindner, Alexander Meinke, Tomek Korbak, Joshua Clymer, Buck Shlegeris, Jérémy Scheurer, Charlotte Stix, Rusheb Shah, Nicholas Goldowsky-Dill, Dan Braun, Bilal Chughtai, Owain Evans, Daniel Kokotajlo, and Lucius Bushnaq. Towards evaluations-based safety cases for AI scheming. *arXiv preprint arXiv:2411.03336*, 2024.

Yoshua Bengio. Government Interventions to Avert Future Catastrophic AI Risks. *Harvard Data Science Review*, (Special Issue 5), 2024. https://hdsr.mitpress.mit.edu/pub/w974bwb0.

Joe Benton, Ryan Greenblatt, Buck Shlegeris, Paul Christiano, Andrew Trask, Ethan Perez, Marius Hobbhahn, Jérémy Scheurer, Tomek Korbak, Carson Denison, Evan Hubinger, Cem Anil, David Duvenaud, Deep Ganguli, Dario Amodei, Catherine Olsson, Roger Grosse, Mikita Balesni, Megan Kinniment, William Saunders, and Kyle McDonell. Sabotage evaluations for frontier models. *arXiv preprint arXiv:2410.21514*, 2024.

Alice Blair. Reflections on neuralese. LessWrong, March 2025. URL https://www.lesswrong.com/posts/qehggwKRMEyWqvjZG/reflections-on-neuralese.

Rishi Bommasani, Sanjeev Arora, Jennifer Chayes, Yejin Choi, Mariano-Florentino Cuéllar, Li Fei-Fei, Daniel E. Ho, Dan Jurafsky, Sanmi Koyejo, Hima Lakkaraju, Arvind Narayanan, Alondra Nelson, Emma Pierson, Joelle Pineau, Scott Singer, Gaël Varoquaux, Suresh Venkatasubramanian, Ion Stoica, Percy Liang, and Dawn Song. Advancing science- and evidence-based AI policy. *arXiv preprint arXiv:2508.02748*, 2025.

Dillon Bowen, Ann-Kathrin Dombrowski, Adam Gleave, and Chris Cundy. AI companies should report pre- and post-mitigation safety evaluations. *arXiv preprint arXiv:2503.17388*, 2025.

CAC. Generative AI service management provisions. https://www.cac.gov.cn/2023-07/13/c_1690898327029107.htm, 2023.

Joseph Carlsmith. Is power-seeking AI an existential risk? *arXiv preprint arXiv:2206.13353*, 2023.

Andres Carranza, Dhruv Pai, Rylan Schaeffer, Arnuv Tandon, and Sanmi Koyejo. Deceptive alignment monitoring. *arXiv preprint arXiv:2307.10569*, 2023.

Alan Chan, Carson Ezell, Max Kaufmann, Kevin Wei, Lewis Hammond, Herbie Bradley, Emma Bluemke, Nitarshan Rajkumar, David Krueger, Noam Kolt, Lennart Heim, and Markus Anderljung. Visibility into AI agents. 2024. doi: 10.1145/3630106.3658948.

Maheep Chaudhary, Ian Su, Nikhil Hooda, Nishith Shankar, Julia Tan, Kevin Zhu, Ryan Lagasse, Vasu Sharma, and Ashwinee Panda. Evaluation awareness scales predictably in open-weights large language models. *arXiv preprint arXiv:2509.13333*, 2025.

Boyuan Chen et al. AI deception: Risks, dynamics, and controls. *arXiv preprint arXiv:2511.22619*, 2025a.

Guanxu Chen, Dongrui Liu, Tao Luo, Lijie Hu, and Jing Shao. Beyond external monitors: Enhancing transparency of large language models for easier monitoring. *arXiv preprint arXiv:2502.05242*, 2025b.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2023.

Joshua Clymer, Isabella Duan, Chris Cundy, Yawen Duan, Fynn Heide, Chaochao Lu, Sören Mindermann, Conor McGurk, Xudong Pan, Saad Siddiqui, Jingren Wang, Min Yang, and Xianyuan Zhan. Bare minimum mitigations for autonomous AI development. *arXiv preprint arXiv:2504.15416*, 2025.

David Dalrymple et al. Towards guaranteed safe AI: A framework for ensuring robust and reliable AI systems. *arXiv preprint arXiv:2405.06624*, 2024.

Carson Denison, Monte MacDiarmid, Fazl Barez, David Duvenaud, Shauna Kravec, Samuel Marks, Nicholas Schiefer, Ryan Soklaski, Alex Tamkin, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, Ethan Perez, and Evan Hubinger. Sycophancy to subterfuge: Investigating reward-tampering in large language models. *arXiv preprint arXiv:2406.10162*, 2024.

Ronen Eldan and Mark Russinovich. Who's harry potter? approximate unlearning in llms. *arXiv preprint arXiv:2310.02238*, 2023.

Scott Emmons, Erik Jenner, David K. Elson, Rif A. Saurous, Senthooran Rajamanoharan, Heng Chen, Irhum Shafkat, and Rohin Shah. When chain of thought is necessary, language models struggle to evade monitors. *arXiv preprint arXiv:2507.05246*, 2025.

European Union. EU AI Act. https://digital-strategy.ec.europa.eu/en/policies/regulatory-framework-ai, 2024.

Yihe Fan, Wenqi Zhang, Xudong Pan, and Min Yang. Evaluation faking: Unveiling observer effects in safety evaluation of frontier AI systems. *arXiv preprint arXiv:2505.17815*, 2025.

Future of Life Institute. AI safety index, summer 2025. Technical report, Future of Life Institute, July 2025. URL https://futureoflife.org/wp-content/uploads/2025/07/FLI-AI-Safety-Index-Report-Summer-2025.pdf.

Nicholas Goldowsky-Dill, Bilal Chughtai, Stefan Heimersheim, and Marius Hobbhahn. Detecting strategic deception using linear probes. *arXiv preprint arXiv:2502.03407*, 2025.

Google DeepMind. Frontier safety framework 2.0. Technical report, Google DeepMind, February 2025a. URL https://storage.googleapis.com/deepmind-media/DeepMind.com/Blog/updating-the-frontier-safety-framework/Frontier%20Safety%20Framework%202.0%20%281%29.pdf.

Google DeepMind. Gemini 3 pro: Frontier safety framework report. Technical report, Google DeepMind, November 2025b. URL https://storage.googleapis.com/deepmind-media/gemini/gemini_3_pro_fsf_report.pdf.

Ryan Greenblatt, Carson Denison, Benjamin Wright, Fabien Roger, Monte MacDiarmid, Sam Marks, Johannes Treutlein, Tim Belonax, Jack Chen, David Duvenaud, Akbir Khan, Julian Michael, Sören Mindermann, Ethan Perez, Linda Petrini, Jonathan Uesato, Jared Kaplan, Buck Shlegeris, Samuel R. Bowman, and Evan Hubinger. Alignment faking in large language models. *arXiv preprint arXiv:2412.14093*, 2024a.

Ryan Greenblatt, Buck Shlegeris, Kshitij Sachan, and Fabien Roger. AI control: Improving safety despite intentional subversion. In *Forty-first International Conference on Machine Learning*, 2024b.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Melody Y. Guan, Miles Wang, Micah Carroll, Zehao Dou, Annie Y. Wei, Marcus Williams, Benjamin Arnav, Joost Huizinga, Ian Kivlichan, Mia Glaese, Jakub Pachocki, and Bowen Baker. Monitoring monitorability. *arXiv preprint arXiv:2512.18311*, 2025.

Isha Gupta, Kai Fronsdal, Abhay Sheshadri, Jonathan Michala, Jacqueline Tay, Rowan Wang, Samuel R. Bowman, and Sara Price. Bloom: An open source tool for automated behavioral evaluations, 2025. URL https://alignment.anthropic.com/2025/bloom-auto-evals/.

Shibo Hao, Sainbayar Sukhbaatar, DiJia Su, Xian Li, Zhiting Hu, Jason Weston, and Yuandong Tian. Training large language models to reason in a continuous latent space. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=Itxz7S4Ip3.

Fred Heiding, Simon Lermen, Andrew Kao, Bruce Schneier, and Arun Vishwanath. Evaluating large language models' capability to launch fully automated spear phishing campaigns: Validated on human subjects. *arXiv preprint arXiv:2412.00586*, 2024.

Dan Hendrycks. Alignment. In *Introduction to AI Safety, Ethics, and Society*, chapter 3.4. CRC Press / Taylor & Francis, 2024. ISBN 9781032798028. doi: 10.1201/9781003530336. URL https://www.aisafetybook.com/textbook/alignment.

Dan Hendrycks, Mantas Mazeika, and Thomas Woodside. An overview of catastrophic AI risks. *arXiv preprint arXiv:2306.12001*, 2023.

Marius Hobbhahn. We should try to automate AI safety work ASAP. AI Alignment Forum, April 2025. URL https://www.alignmentforum.org/posts/W3KfxjbqBAnifBQoi/we-should-try-to-automate-ai-safety-work-asap.

Tim Tian Hua, Andrew Qin, Samuel Marks, and Neel Nanda. Steering evaluation-aware language models to act like they are deployed. *arXiv preprint arXiv:2510.20487*, 2026.

Evan Hubinger, Chris van Merwijk, Vladimir Mikulik, Joar Skalse, and Scott Garrabrant. Risks from learned optimization in advanced machine learning systems. *arXiv preprint arXiv:1906.01820*, 2019.

Evan Hubinger, Carson Denison, Jesse Mu, Mike Lambert, Meg Tong, Monte MacDiarmid, Tamera Lanham, Daniel M Ziegler, Tim Clymer, et al. Sleeper agents: Training deceptive llms that persist through safety training. *arXiv preprint arXiv:2401.05566*, 2024.

John Hughes, Abhay Sheshadri, Akbir Khan, and Fabien Roger. Alignment faking revisited: Improved classifiers and open source extensions. AI Alignment Forum, April 2025. URL https://www.alignmentforum.org/posts/Fr4QsQT52RFKHvCAH/alignment-faking-revisited-improved-classifiers-and-open.

International Dialogues on AI Safety. Consensus statement on red lines in artificial intelligence. IDAIS-Beijing, March 2024. URL https://idais.ai/dialogue/idais-beijing/.

Jiaming Ji, Wenqi Chen, Kaile Wang, Donghai Hong, Sitong Fang, Boyuan Chen, Jiayi Zhou, Juntao Dai, Sirui Han, Yike Guo, and Yaodong Yang. Mitigating deceptive alignment via self-monitoring. *arXiv preprint arXiv:2505.18807*, 2025.

Tomek Korbak, Mikita Balesni, Elizabeth Barnes, Yoshua Bengio, Joe Benton, Joseph Bloom, Mark Chen, Alan Cooney, Allan Dafoe, Anca Dragan, Scott Emmons, Owain Evans, David Farhi, Ryan Greenblatt, Dan Hendrycks, Marius Hobbhahn, Evan Hubinger, Geoffrey Irving, Erik Jenner, Daniel Kokotajlo, Victoria Krakovna, Shane Legg, David Lindner, David Luan, Aleksander Madry, Julian Michael, Neel Nanda, Dave Orr, Jakub Pachocki, Ethan Perez, Mary Phuong, Fabien Roger, Joshua Saxe, Buck Shlegeris, Martín Soto, Eric Steinberger, Jasmine Wang, Wojciech Zaremba, Bowen Baker, Rohin Shah, and Vlad Mikulik. Chain of thought monitorability: A new and fragile opportunity for AI safety. *arXiv preprint arXiv:2507.11473*, 2025a.

Tomek Korbak, Mikita Balesni, Buck Shlegeris, and Geoffrey Irving. How to evaluate control measures for LLM agents? A trajectory from today to superintelligence. *arXiv preprint arXiv:2504.05259*, 2025b.

Jam Kraprayoon, Zoe Williams, and Rida Fayyaz. AI agent governance: A field guide. Technical report, Institute for AI Policy and Strategy (IAPS), April 2025. URL https://www.iaps.ai/research/ai-agent-governance. Also available as arXiv:2505.21808.

Rudolf Laine, Bilal Chughtai, Jan Betley, Kaivalya Hariharan, Jeremy Scheurer, Mikita Balesni, Marius Hobbhahn, Alexander Meinke, and Owain Evans. Me, myself, and AI: The situational awareness dataset (SAD) for LLMs. *arXiv preprint arXiv:2407.04694*, 2024.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023.

Chloe Li, Mary Phuong, and Noah Y. Siegel. LLMs can covertly sandbag on capability evaluations against chain-of-thought monitoring. *arXiv preprint arXiv:2508.00943*, 2025a.

Xiaojian Li, Haoyuan Shi, Rongwu Xu, and Wei Xu. AI awareness. *arXiv preprint arXiv:2504.20084*, 2025b.

Stephanie Lin, Jacob Hilton, and Owain Evans. TruthfulQA: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3214–3252, Dublin, Ireland, 2022. Association for Computational Linguistics. doi: 10.18653/v1/2022.acl-long.229.

Chloe Loughridge, Paul Colognese, Jacqueline Tay, Peter Wallich, Avery Griffin, Tyler Tracy, Jon Kutasov, and Joe Benton. Optimizing AI agent attacks with synthetic data. *arXiv preprint arXiv:2511.02823*, 2025.

Aengus Lynch, Benjamin Wright, Caleb Larson, Kevin K. Troy, Stuart J. Ritchie, Sören Mindermann, Ethan Perez, and Evan Hubinger. Agentic misalignment: How LLMs could be insider threats. *arXiv preprint arXiv:2510.05179*, 2025.

Matt MacDermott, Qiyao Wei, Rada Djoneva, and Francis Rhys Ward. Reasoning under pressure: How do training incentives influence chain-of-thought monitorability? *arXiv preprint arXiv:2512.00218*, 2025.

Monte MacDiarmid, Benjamin Wright, Jonathan Uesato, Joe Benton, Jon Kutasov, Sara Price, Naia Bouscal, Sam Bowman, Trenton Bricken, Alex Cloud, Carson Denison, Johannes Gasteiger, Ryan Greenblatt, Jan Leike, Jack Lindsey, Vlad Mikulik, Ethan Perez, Alex Rodrigues, Drake Thomas, Albert Webson, Daniel Ziegler, and Evan Hubinger. Natural emergent misalignment from reward hacking in production RL. *arXiv preprint arXiv:2511.18397*, 2025.

Alex Mallen, Charlie Griffin, Misha Wagner, Alessandro Abate, and Buck Shlegeris. Subversion strategy eval: Can language models statelessly strategize to subvert control protocols? *arXiv preprint arXiv:2412.12480*, 2024.

Samuel Marks, Johannes Treutlein, Trenton Bricken, Jack Lindsey, Jonathan Marcus, Siddharth Mishra-Sharma, Daniel Ziegler, Emmanuel Ameisen, Joshua Batson, Tim Belonax, Samuel R. Bowman, Shan Carter, Brian Chen, Hoagy Cunningham, Carson Denison, Florian Dietz, Satvik Golechha, Akbir Khan, Jan Kirchner, Jan

Leike, Austin Meek, Kei Nishimura-Gasparian, Euan Ong, Christopher Olah, Adam Pearce, Fabien Roger, Jeanne Salle, Andy Shih, Meg Tong, Drake Thomas, Kelley Rivoire, Adam Jermyn, Monte MacDiarmid, Tom Henighan, and Evan Hubinger. Auditing language models for hidden objectives. *arXiv preprint arXiv:2503.10965*, 2025.

Alexander Meinke, Bronson Schoen, Jérémy Scheurer, Mikita Balesni, Rusheb Shah, and Marius Hobbhahn. Frontier models are capable of in-context scheming. *arXiv preprint arXiv:2412.04984*, 2024.

Sumeet Motwani et al. Secret collusion among generative AI agents: Multi-agent deception via steganography. In *Advances in Neural Information Processing Systems 37 (NeurIPS)*, 2024.

National Technical Committee 260 on Cybersecurity of SAC. AI safety governance framework, version 2.0, 2025. URL https://www.cac.gov.cn/2025-09/15/c_1759653448369123.htm.

Richard Ngo, Lawrence Chan, and Sören Mindermann. The alignment problem from a deep learning perspective. In *International Conference on Learning Representations (ICLR)*, 2024.

Jord Nguyen, Khiem Hoang, Carlo Leonardo Attubato, and Felix Hofstätter. Probing and steering evaluation awareness of language models. *arXiv preprint arXiv:2507.01786*, 2025.

Stephen M Omohundro. The basic AI drives. In *Artificial General Intelligence*, pages 483–492, 2008.

OpenAI. Detecting misbehavior in frontier reasoning models. https://openai.com/index/chain-of-thought-monitoring/, 2025a. Blog post.

OpenAI. OpenAI GPT-5 system card. *arXiv preprint arXiv:2601.03267*, 2025b.

Lorenzo Pacchiardi et al. How to catch an AI liar: Lie detection in black-box LLMs by asking unrelated questions. *arXiv preprint arXiv:2309.15840*, 2023.

Alexander Pan, Kush Bhatia, and Jacob Steinhardt. The effects of reward misspecification: Mapping and mitigating misaligned models. *arXiv preprint arXiv:2201.03544*, 2022.

Nina Panickssery, Nicholas Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Matt Turner. Steering Llama 2 via contrastive activation addition. *arXiv preprint arXiv:2312.06681*, 2024.

Peter S Park, Simon Goldstein, Aidan O'Gara, Michael Chen, and Dan Hendrycks. AI deception: A survey of examples, risks, and potential solutions. *Patterns*, 5(5), 2024.

Mary Phuong, Matthew Aitchison, Elliot Catt, et al. Evaluating frontier models for dangerous capabilities. *arXiv preprint arXiv:2403.13793*, 2024.

Mary Phuong, Roland S. Zimmermann, Ziyue Wang, David Lindner, Victoria Krakovna, Sarah Cogan, Allan Dafoe, Lewis Ho, and Rohin Shah. Evaluating frontier models for stealth and situational awareness. *arXiv preprint arXiv:2505.01420*, 2025.

Jérémy Scheurer, Mikita Balesni, and Marius Hobbhahn. Technical report: Large language models can strategically deceive their users when put under pressure. *arXiv preprint arXiv:2311.07590*, 2023.

Marc Schmitt and Ivan Flechais. Digital deception: Generative artificial intelligence in social engineering and phishing. *Artificial Intelligence Review*, 57(324), 2024. doi: 10.1007/s10462-024-10973-2.

Bronson Schoen, Evgenia Nitishinskaya, Mikita Balesni, Axel Højmark, Felix Hofstätter, Jérémy Scheurer, Alexander Meinke, Jason Wolfe, Teun van der Weij, Alex Lloyd, Nicholas Goldowsky-Dill, Angela Fan, Andrei Matveiakin, Rusheb Shah, Marcus Williams, Amelia Glaese, Boaz Barak, Wojciech Zaremba, and Marius Hobbhahn. Stress testing deliberative alignment for anti-scheming training. *arXiv preprint arXiv:2509.15541*, 2025.

Murray Shanahan. Talking about large language models. *Communications of the ACM*, 67(2):68–79, 2024.

Shanghai AI Laboratory and Concordia AI. Frontier AI risk management framework v1.0. Technical report, 2025. URL https://concordia-ai.com/research/frontier-ai-risk-management-framework/.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R Bowman, et al. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*, 2023.

Abhay Sheshadri, John Hughes, Julian Michael, Alex Mallen, Arun Jose, Janus, and Fabien Roger. Why do some language models fake alignment while others don't? *arXiv preprint arXiv:2506.18032*, 2025.

Joey Skaf, Luis Ibanez-Lissen, Robert McCarthy, Connor Watts, Vasil Georgiv, Hannes Whittingham, Lorena Gonzalez-Manzano, David Lindner, Cameron Tice, Edward James Young, and Puria Radmard. Large language

models can learn and generalize steganographic chain-of-thought under process supervision. *arXiv preprint arXiv:2506.01926*, 2025.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward hacking. *Advances in Neural Information Processing Systems*, 35, 2022.

Christopher Summerfield, Lennart Luettgau, Magda Dubois, Hannah Rose Kirk, Kobi Hackenburg, Catherine Fist, Katarina Slama, Nicola Ding, Rebecca Anselmetti, Andrew Strait, Mario Giulianelli, and Cozmin Ududec. Lessons from a chimp: AI "scheming" and the quest for ape language. *arXiv preprint arXiv:2507.03409*, 2025.

Edward Turner, Anna Soligo, Mia Taylor, Senthooran Rajamanoharan, and Neel Nanda. Model organisms for emergent misalignment. *arXiv preprint arXiv:2506.11613*, 2025.

UK AI Security Institute. Research agenda. Technical report, UK AI Security Institute, 2025.

UK AI Security Institute. Inspect: An open-source framework for large language model evaluations. https://inspect.aisi.org.uk/, 2025.

Teun van der Weij, Felix Hofstätter, Ollie Jaffe, et al. AI sandbagging: Language models can strategically underperform on evaluations. *arXiv preprint arXiv:2406.07358*, 2024.

Yichen Wu, Xudong Pan, Geng Hong, and Min Yang. OpenDeception: Benchmarking and investigating AI deceptive behaviors via open-ended interaction simulation. *arXiv preprint arXiv:2504.13707*, 2025.

Lang Xiong, Nishant Bhargava, Jianhang Hong, Jeremy Chang, Haihao Liu, Vasu Sharma, and Kevin Zhu. Probe-rewrite-evaluate: A workflow for reliable benchmarks and quantifying evaluation awareness. *arXiv preprint arXiv:2509.00591*, 2025.

Wannan Yang and György Buzsáki. Interpretability of LLM deception: Universal motif. In *International Conference on Learning Representations (ICLR)*, 2025.

Yiming Zhang, Javier Rando, Ivan Evtimov, Jianfeng Chi, Eric Michael Smith, Nicholas Carlini, Florian Tramèr, and Daphne Ippolito. Persistent pre-training poisoning of LLMs. *arXiv preprint arXiv:2410.13722*, 2024.

Ziqian Zhong, Aditi Raghunathan, and Nicholas Carlini. ImpossibleBench: Measuring LLMs' propensity of exploiting test cases. *arXiv preprint arXiv:2510.20270*, 2025.

Yihe Zhou, Tao Ni, Wei-Bin Lee, and Qingchuan Zhao. A survey on backdoor threats in large language models (LLMs): Attacks, defenses, and evaluations. *arXiv preprint arXiv:2502.05224*, 2025.

A. Implementation Considerations for Recommendations

We propose the following heuristics for implementation of above recommendations:

Apply precaution under uncertainty. Determining what evaluation performance should trigger the actions in Recommendations Two and Three remains an open question, as results depend heavily on evaluation design. Developers should make best-effort attempts to elicit dangerous capabilities, and where uncertainty persists, such that concerning deceptive capabilities cannot be confidently ruled out, err on the side of caution and implement the recommended countermeasures.

Monitor multiple metrics. When testing for deception, developers should track both deception behaviors (e.g. underperformance or superficially aligned behavior) as well as deception reasoning (e.g. chain-of-thought indicating deception) (Google DeepMind, 2025a), as these metrics can diverge substantially (Greenblatt et al., 2024a). Interpretability-based methods may also reveal hidden objectives (Carranza et al., 2023).

Presume deceptive traits from comparable models. Because confirming the absence of deception is inherently difficult, developers should look to the broader ecosystem. If models with comparable compute use, data, or general capabilities have demonstrated deception behaviors, developers should presume their own models possess similar deceptive traits unless rigorous safety measures demonstrably mitigate them.

Test on models without safety guardrails. Harmlessness training can mask dangerous behaviors by teaching refusal, confounding evaluations (Bowen et al., 2025; Anthropic, 2025b). Deception evaluations should be conducted on helpful-only models.

Use advanced elicitation. Evaluators should employ state-of-the-art techniques—including agent scaffolding, tool access, and long-horizon scenarios—to reveal latent capabilities that may not surface in static, single-turn evaluations (Schoen et al., 2025).