
A Pre-mortem on The First Major AI-Enabled Terrorism Event and The Choices That Could Enable It

Plus 9 Predictions

Max Kamachee, Jordi Weinstock, Antonia Juelich, Jonas Raedler,
Elise Racine, Stephen Casper

OCTOBER 2026



HARVARD Kennedy School

BELFER CENTER

for Science and International Affairs

Abstract

Frontier AI developers' model reports and the dedicated risk-assessment teams at governmental AI institutes such as UK AISI and US CAISSI now treat a major AI-enabled cyber, biological, or chemical terror event as a credible concern. But how, precisely, might such an event happen? What choices across the model lifecycle may contribute to it? This paper studies what current evidence suggests about how specific mistakes from AI developers, distributors, and governments may contribute to such an event. In doing so, we consider current research on frontier model risks as well as existing examples of how AI capabilities useful for criminal activities have already been introduced and spread. We make two arguments. Firstly, specific developer, distributor, and government choices around model deployment, safeguards, and detection strategies will substantially shape whether AI meaningfully lowers the barrier to such an attack. Secondly, if a major AI-enabled cyber, biological, or chemical terror event occurs in the foreseeable future, it will most likely have been enabled either as (1) the consequence of a closed-weight model deployed without state-of-the-art safeguards or (2) as a consequence of an open-weight model whose safeguards (if significant ones exist) could not survive downstream tampering. Finally, we conclude with a set of 9 falsifiable predictions about how the first major AI-enabled cyber, bio, or chemical terrorism attack may occur and outline two illustrative hypothetical scenarios for such an event.

Executive Summary

Frontier AI developers now report they cannot rule out meaningful cyber, biological, and chemical attack uplift from their models. National security institutes and recent state legislation treat a major AI-enabled terror event as a credible concern. This paper examines that scenario through a premortem: conditioning on such an event having occurred, we trace backward through the model lifecycle to ask which choices could have enabled it, and where each could have been interrupted.

The report makes two central arguments:

1. Specific choices at each model lifecycle stage foreseeably raise or lower the barrier to a major AI-enabled attack. For a closed-weight system, no known attack strategy reliably circumvents a fully implemented state-of-the-art (not to be confused with the state of practice) layered defense as of summer 2026, so implementation is the misuse risk bottleneck. For an open-weight system, the state-of-the-art stack as of summer 2026 raises the cost of misuse substantially but much less so than the closed counterpart, so the bottlenecks are both technical and implementational.
2. If such an event occurs in the near future, it will most likely have been enabled either as the consequence of a closed-weight model deployed without state-of-the-art safeguards, or as the consequence of an open-weight model whose safeguards did not survive downstream tampering.

Dangerous capability arises mostly as a byproduct of scaling general-purpose training, with deliberate specialization a secondary path

Because harmful and benign capabilities in dual-use domains are entangled, removing one ‘taxes’ the other. Under competitive pressure that tax often goes unpaid, so companies may deploy systems with risky capabilities. Advanced dual-use capability can even serve as a marketing signal, further incentivizing its development.

Closed- versus open-weight deployments pose two very distinct risk scenarios

Because closed-weight models are ahead of open-weight models by several months in frontier capabilities, the risk of counterfactual uplift, if fully elicited, is likely higher than open-weight models. However, in a closed-weight scenario, the attacker reaches the model only through the provider’s interface and must elicit capability through it while the provider can monitor behavior and react. In the open-weight scenario, the attacker possesses the weights, so the attacker can prompt, scaffold, and modify the model directly to bypass safeguards and run it independently with no provider present to monitor or react.

Harm can be elicited from both closed- and open-weight models with varying efficacy and reliability

Eliciting actionable harmful knowledge from highly safeguarded closed-weight models would require a high degree of attack sophistication. *Jailbreaking* and *task decomposition* appear to be the most powerful techniques for eliciting harm from closed-weight models. Spreading activity across multiple providers and accounts lowers that bar, since each provider sees only a fragment and no single refusal or suspension halts the effort. In contrast, for open-weight models, fine-tuning and other techniques that modify model weights are fundamentally powerful methods for eliciting harm that only require limited (i.e., undergraduate level)

technical sophistication. Weight modification for open-weight models typically requires limited familiarity with machine learning engineering to perform. However, even a single actor can make a dangerously modified open-weight model widely available. For example, at the time of writing this paper, there are currently over 14,000 openly downloadable “uncensored” and “abliterated” open-weight models on Hugging Face able to be downloaded by anyone in the world.

State-of-the-art safeguards for open-weight models are much easier to circumvent compared to state-of-the-art safeguard stacks permitted by closed-weight model deployment

Closed-weight deployments can prevent weight modification and rely on a multi-layer defense of filters and monitors. Closed weight models, however, should not escape scrutiny. Although they benefit from defenses in depth, there are gaps between the state of the art and the state of practice. In practice, closed models are often vulnerable to sophisticated attacks. Furthermore, state-of-the-art robustness hinges on a closed- or closed-weight model provider thoroughly implementing the state-of-the-art safeguards. In contrast, safeguards for open-weight models have fundamental limitations. Pretraining-data filtering of hazardous content currently appears to be state-of-the-art for safeguarding open-weight models against harmful fine-tuning, but methods are improving.

State-of-the-art detection methods can be very effective for closed-weight deployments but are very unlikely to be effective for open-weight deployments

Current evaluation techniques may struggle to identify significant risks. State-of-the-art benchmarks for dangerous capabilities are increasingly infohazardous, and thus held privately without public verification. Open-weight model developers rarely report predeployment model evaluations of model safety under malicious tampering. Meanwhile, almost no developers of closed- or open-weight models publicly report human uplift randomized control trials to accurately measure dangerous capabilities and likely threat models. Furthermore, there are no rigorous ways to convert benchmark scores into precise prescription for proportionate action, and there is no well-communicated binding standard requiring that failures be reported.

User monitoring can be effective for closed-weight models, but generally not for open-weight models. For closed models, user monitoring can be effective and has caught real misuse. However, it is contingent on the provider choosing to monitor, on those measures surviving adversarial evasion, and on the developer choosing to act upon potential risks. For open-weight models, user monitoring is rarely a reliable option because they can be run privately. Attribution, monitoring, and reporting are not generally possible for open-weight models because these methods only work by modifying separable wrappers around the model rather than the model itself.

Smaller-scale precedented misuse follows a pattern

To understand how a large-scale AI-enabled terrorism event may happen, we can look to patterns among smaller-scale events of the recent past. Here, we study precedent in three categories: AI generated NCII/ deepfake abuse, criminal and state-actor cybercrime, and small-scale precedented instances of terrorism. Among these smaller-scale events of the recent past, patterns emerge. Harm elicited from closed-weight models occurred when developers did not rigorously implement the full safeguard stack their deployment had permitted, particularly for ‘dual-use’ capabilities that can provide uplift for both legitimate and malicious

tasks. Harm elicited from open-weight models emerged when upstream safeguards, or the lack thereof, did not survive downstream tampering and no detection mechanisms had existed to catch misuse.

How could this analysis inform risk management?

Given current evidence, both open-weight and closed-weight deployment pathways present credible routes to a first major AI-enabled terror event. Furthermore, given past incidents, the current state of practice extrapolated to a future with more capable models is likely insufficient to prevent major risks from materializing. In light of this analysis, we offer 9 falsifiable predictions about the specific model(s) implicated in the hypothetical first major AI-enabled cyber, biological, or chemical terror event (see Section 5 for full explanations and Section 6 for two illustrative scenarios). Each prediction is posed independently (not in conjunction or disjunction).

Overarchingly:

1. The attack will trace to either a closed-weight model deployed without state-of-the-art safeguards or an open-weight model whose safeguards (if significant ones exist) could not survive downstream tampering.

If it was a closed-weight model:

2. The model would not have been implemented with input, output, and trajectory-level safety classifiers up to par with the state of the art as of summer 2026.
3. The model would not have implemented safety training and fine-tuning up to par with the state of the art as of summer 2026.
4. The model would not have had user-monitoring and account-restriction protocols up to par with the state of the art as of summer 2026.

If it was an open-weight model:

5. The model would have not hazard-specific content filtered from its pretraining data and would have lacked robust safeguards against malicious tampering up to par with the state of the art as of summer 2026.
6. If safeguards were circumvented via tampering, the threat actor likely had at least undergraduate-level computer-science expertise and at least graduate-level domain expertise.
7. The developer of the model would not have had any third-party model-tampering red-teaming done.

Either deployment pathway:

8. The developer prioritized the model's useful capabilities over minimizing its harmful ones, and may have even publicized its harmful capabilities alongside a product release in a way that attracted attention and customers.

9. The developer did not have the model red-teamed pre-deployment for the relevant form of dangerous capabilities by third-party evaluators

We also find that leading closed-weight model developers report implementing some of the available safeguards and evaluations while leading open-weight model developers report almost none (Table 4). If this gap in effort towards securing systems between closed- and open-weight model developers persists, major incidents originating from open-weight models become nearly inevitable.

Increasing the resilience of the AI ecosystem can deter this misuse, so we conclude by discussing a range of actions policymakers, developers, researchers & funders, and model distribution platforms can consider to meaningfully contribute to this in Section 7.

Main takeaways:

- **All developers** can implement state-of-the-art safeguards, accept the safety taxes they impose, and conduct and report third-party red-teaming.
- **Closed-weight developers** can deploy a full, state-of-the-art layered safety stack, monitor at the user and trajectory level, and treat fine-tuning APIs as open-weight-equivalent attack surfaces.
- **Open-weight developers** can invest in durable upstream safeguards and serve narrow dual-use capabilities through vetted access instead.
- **Model distribution platforms** can moderate safeguard-stripped variants proactively rather than on request.
- **Policymakers** can mandate disclosure and implementation of key safeguards and evaluations, develop thresholds that convert measurements into deployment decisions, escalation of concerning activity to relevant authorities, and the development of threat intelligence sharing frameworks amongst developers and the government.
- **Researchers and funders** can support developer-independent uplift evaluations and improve tamper-resistance methods.

Contents

Abstract 2

Executive Summary 3

Contents. 7

1. Introduction. 8

2. Scope & Approach 10

3. Pre-Mortem Analysis 11

3.1 How the Dangerous Capability Would be Developed 11

3.2 How the Dangerous Capability Would be Deployed 14

3.3 How the Dangerous Capability Would be Elicited. 15

3.4 How Safeguards Would Fail 19

3.5 How Detection Would Fail. 22

4. What History Says. 27

4.1 AI Generated Non-Consensual Intimate Imagery 29

4.2 Criminal & State Actor AI-Enabled Cybercrime 31

4.3 Precedented Use of AI for Small-Scale Terrorism. 33

5. Predictions 34

6. Two Illustrative Scenarios. 37

7. Discussion and Implications 41

Considerations for Action by Stakeholder Type 42

Acknowledgements. 46

Impact Statement 46

AI Usage Statement 46

Appendix A: Safety Taxes from Safeguards 47

Appendix B: Embedding Geometry of Harmless vs. Potentially Harmful Content 48

Appendix C: The Deployment Spectrum and Intermediate Strategies. 49

Appendix D: Capability Suppression (a.k.a. Unlearning) Methods. 50

1. Introduction

The threat of AI-enabled terrorism is not hypothetical. Emergent dual use capabilities are advancing rapidly ([Mythos; GPT 5.6](#)), with weapons-relevant chemical and biological design capabilities now demonstrated (Section 3.1), and frontier AI developers reporting that their models are crossing thresholds for meaningful cyber, biological, and chemical uplift (e.g., [Mythos Glasswing; GPT 5.6 Preview 2026](#)). Thousands of “uncensored” models with removed safeguards currently circulate on public repositories ([Sokhansaj, 2025](#)). Meanwhile, terrorists, state actors, and criminals have begun using generative AI for operational advice, propaganda, recruitment, misinformation, planning, and execution ([SJW, 2025; Caillot, 2025; Tech Against Terrorism, 2023/24; Juelich, 2026; GTIG, 2026; Firdhous et al., 2023; IWF, 2026](#)).

In cyber, US agencies warned that attackers are using AI to write exploit scripts targeting critical infrastructure ([Kovacs, 2026](#)). Separately, a state-linked actor developed a multi-agent framework which they used for a largely autonomous four-day cyber intrusion campaign to steal credentials from Taiwanese government officials and to scan their nuclear safety agency ([Dream, 2026; Reuters, 2026](#)). Other reported cases involved AI-enabled cyber operations against Mexican government agencies and Thailand’s Ministry of Finance ([Martin and Millan, 2026; Khandelwal, 2026](#)).

In biology and chemistry, GPT-5 complied with hundreds of users asking how to make biological weapons and poisons returning step-by-step instructions ([WSJ, 2026](#)). Anthropic has also reported disrupting several cases in which users used Claude for dual-use research on novel biological agents ([Anthropic, 2026](#)).¹ This is also corroborated by a foiled terror plot to develop and disperse ricin poison where the perpetrator used ChatGPT to research its development ([Deccan Herald, 2025](#)). This dual-use potential is further evident in researchers’ recent use of AI to design novel bacteria-killing viruses ([King et al., 2026; Koetsier, 2026](#)).

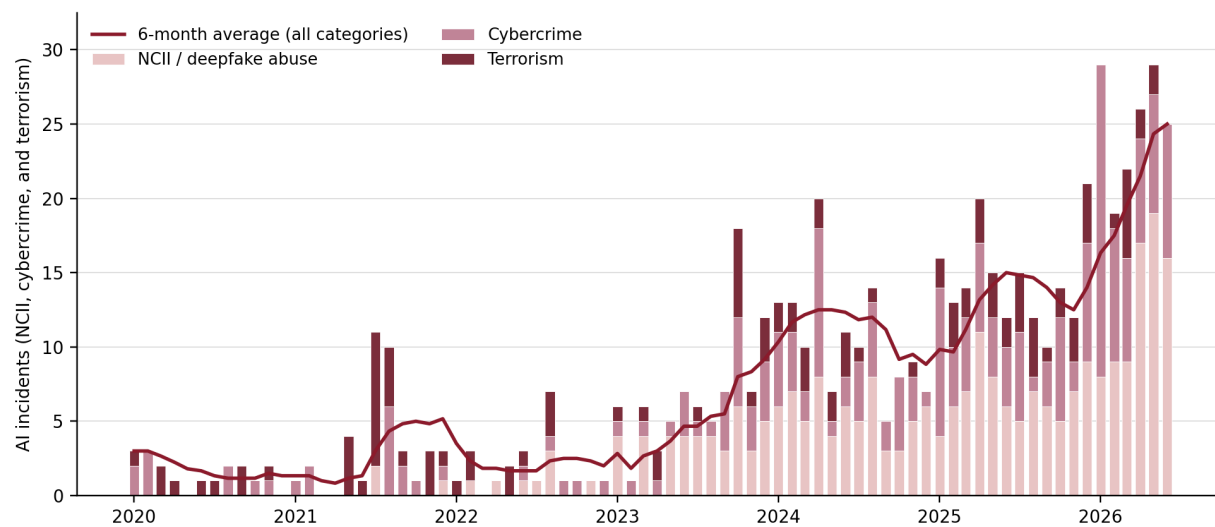
In conventional attacks, AI has already assisted terrorists, including the 2025 Las Vegas Cybertruck bombing, the Palm Springs bombing, and the Delhi Red Fort blast ([PBS, 2025; CNBC, 2025; The Economic Times, 2026](#)). [Juelich \(2026\)](#) documented both Boko Haram factions using frontier chatbots across mission preparation and execution. More recently, Anthropic reported disrupting six conventional weapons-development operations, including autonomous drone swarm and guided-missile programs, across China, Russia, and Yemen in which actors used Claude as an instrumental development tool ([Anthropic, 2026](#)).

In other forms of criminal misuse, some using AI-built tools have been built for phishing and malware generation ([Firdhous et al., 2023](#)), and image/video models have been used to mass produce and distribute non-consensual and child sexual abuse imagery ([IWF, 2026](#)).

Overall, coinciding with the rise of frontier model capabilities, media-reported AI misuse incidents in terrorism, cybercrime, and non-consensual intimate imagery abuse are up roughly 10× since 2020 (Figure 1; [OECD AIM, 2026](#)).

¹ Anthropic reported that it could not determine if this was for legitimate or malicious ends.

Figure 1

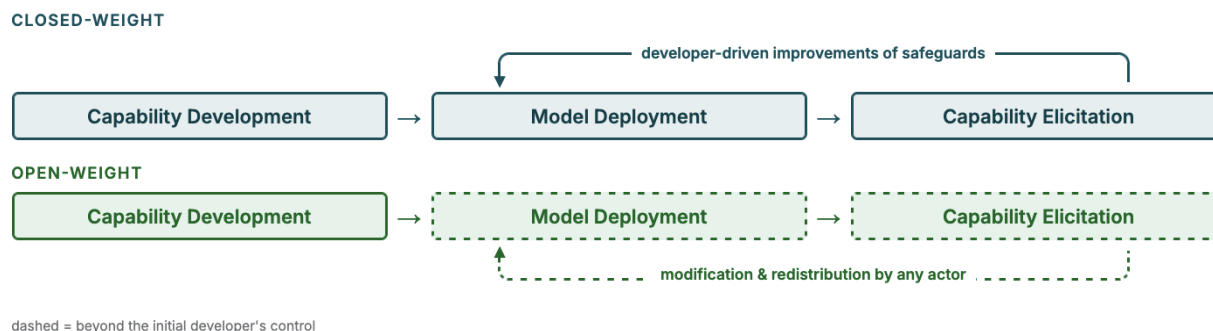


Media recorded AI incidents within the precedent categories of non-consensual intimate imagery (NCII), cybercrime, and small-scale terrorism have increased since 2020. This data was sourced via keyword search on OECD's AIM platform which is reproducible by the link referenced ([OECD AIM, 2026](#))

Governments and developers are taking steps in anticipation of a possible major AI-enabled terror event. For example, the United Kingdom's AI Security Institute and the Center for Advancing Innovation and Standards for Super Intelligence run dedicated evaluations of frontier models for dual-use cyber, biological, and chemical capabilities and safeguard robustness ([AISI, 2025](#); [CAISSI, 2025](#)). The United States government recently issued an export directive taking down Anthropic's Fable model citing national security concerns and have requested OpenAI to stagger the release of their GPT 5.6 model ([Anthropic, 2026](#); [Reuters, 2026](#)). Since the summer of 2025, major developers have also stated in model cards that they cannot rule out capabilities enabling novice actor uplift on biological/cyber/chemical tasks, and have escalated safeguard rigor accordingly ([Anthropic, 2025](#); [OpenAI, 2025](#)). OpenAI's GPT-5.6 Sol model released in June 2026 crossed the company's own 'High' capability threshold in the biological and chemical domain, meaning that it can meaningfully assist a novice in creating a known severe threat without safeguards ([OpenAI, 2026](#)). Similarly, OpenAI delayed the release of their Astra model because it reached their 'Critical' threshold in the cyber domain ([OpenAI, 2026](#)). This means it can autonomously find and develop working zero-days of any severity in hardened real-world critical systems, or devise and execute novel end-to-end attacks on hardened cyber targets from a high-level goal alone ([OpenAI, 2025](#)).

This paper runs a premortem on this scenario. Conditioning on the first major AI-enabled cyber, biological, or chemical terror event, we ask what sequence of choices across the model lifecycle could have enabled it, and where each could have been interrupted. That lifecycle runs from development through deployment to the elicitation of dangerous capability, with safeguards and evaluations applied at any point and differs by deployment type (Figure 2).

Figure 2



The model lifecycle differs by deployment type. Dangerous capability is developed during training, made available through deployment, and then elicited by a user. Safeguards and evaluations can be applied at any stage. For closed-weight models (blue, top), the lifecycle is fully controlled by the developer, since the weights were not released: the model developer can iteratively improve safeguards and reverse model deployment at any time. For open-weight models (green, bottom), the model provider loses visibility and control when they deploy it. A third party can download the weights, modify the model to elicit capability, and re-share the result, which re-enters the lifecycle at deployment as a new model.

Our analysis is organized across five failure points, one per stage in the model lifecycle where development and deployment decisions can affect the likelihood of an AI-enabled terrorism event: how the dangerous capability would be developed (Section 3.1); how the dangerous capability would be deployed (Section 3.2); how the dangerous capability would be elicited (Section 3.3); how safeguards would fail (Section 3.4); and how detection would fail (Section 3.5). We then draw insights from historical precedents (Section 4) for AI enabled misuse before offering 9 falsifiable predictions (Section 5) about how the first major AI-enabled cyber, bio, or chemical terrorism may occur. Finally, we outline two illustrative hypothetical scenarios (Section 6) for such an event.

We present two central arguments. Firstly, specific choices made by developers, distributors, and policymakers across each stage of the model lifecycle can foreseeably lower or raise the barriers to misuse. Secondly, closed-weight models have a much richer and more effective toolkit to mitigate misuse, so if harm arises from a closed model, it is likely to be the consequence of that model lacking state-of-the-art safeguards. In contrast, open-weight models lack such a toolkit of reliable safeguards due to how they can be privately run and arbitrarily modified. If harm arises from one, it is a consequence of upstream safeguards failing to survive downstream tampering and/or lacking substantial safeguards altogether.

2. Scope & Approach

This pre-mortem conditions on a hypothetical first major AI-enabled terrorism event, one whose harm greatly exceeds precedent misuse events and would be categorized as a “critical” incident according to several US state-level frontier AI laws. Precedent criminal misuses of AI generally fall into three main categories: AI generated NCII/deepfake abuse, criminal and state-actor cybercrime, and terrorism. However, by “major”, we mean a cyber, biological, or chemical attack resulting in the death of or serious bodily injury to at least 50 people or at least \$1B in damages that would be classified as a “critical harm” or “critical safety incident” according to recent US laws covering frontier AI model development (SB 53; RAISE; SB 315). At the time of writing, the Delhi Red Fort bombing is the most significant publicly known terrorism event, involving the use of AI for researching explosives development, that resulted in 15 deaths (The Economic Times, 2026).

By “AI-enabled”, we mean an attack in which the AI system was involved in a critical part of planning or executing the attack and plausibly offered meaningful counterfactual uplift over the actor’s next most effective alternative (Kapoor et al., 2024). This does not require it being counterfactually certain that the AI system was strictly necessary for an attack of such type to have happened. This counterfactual is rarely cleanly determinable for any single attack, and independent analysis emphasizes that tacit knowledge, materials, and organizational competence are major bottlenecks (Byman and Subrahmanian, 2026), though AI may also help erode these. Furthermore, we condition on a terrorism event that was carried out by humans, rather than by fully autonomous systems acting independently of a human.

We examine only the technical choices that model developers, deployers, and evaluators make, not the counterterrorism, law-enforcement, or physical-security measures that also affect whether an attack succeeds. These other actors play an important part in mitigating terrorism risks, but it is beyond the scope of this paper. Despite this, we do discuss incident escalation protocols from developers to law-enforcement which is a crucial mechanism from a developer’s perspective to curtail AI-enabled terrorism in Section 3.5.

We focus primarily on cyber, biology, and chemical uplift, but most of our analysis will also extend to explosive, radiological, and nuclear threats. These are the three domains where frontier developers and national evaluators concentrate their capability and safeguard assessments. Recent frontier model cards report cyber, biological, and chemical uplift evaluations (Anthropic, 2025; OpenAI, 2025), and both the UK AI Security Institute (UK AISI) and the US Center for Advancing Innovation and Standards for Super Intelligence (US CAISSI) run dedicated evaluations in these three domains (AIS, 2025; CAISSI, 2025).² Importantly, we recognize that threats may manifest themselves differently across these domains (CrowdStrike, 2026; Walsh, 2025; OPCW, 2026). We do not include AI-assisted explosive plots in this set. Anthropic recently published capability evaluations for conventional weapons development and implemented classifiers to block weapons-development requests (Anthropic, 2026), but to our knowledge these are the first such published evaluations and no national evaluator has followed, so the domain currently remains out of the scope of this paper. However, in Section 4.3, we consider prior AI-assisted bomb plots as a source of prior evidence about future threats.

Since the event we condition on has not yet occurred, we draw on current evidence within the AI ecosystem to reason about future misuse dynamics. Section 3 surveys what is known about how dangerous capability is developed, deployed, elicited, safeguarded, and detected, and where each of those stages may fail based on current research. Section 4 works backward from recent incidents, grouping documented AI misuse into three cases (deepfake/NCII abuse, criminal and state-actor cybercrime, and terrorism) and tracing how choices at each lifecycle stage shaped the misuse that followed in each case.

3. Pre-Mortem Analysis

3.1 How the Dangerous Capability Would be Developed

In modern AI models, dangerous capability arises primarily as a byproduct of general-purpose training, and is

² Cyberterrorism incidents have simpler supply chains behind them, and there is generally a lower barrier to the creation of threats. However, bio and chemical incidents are a more acute threat to human life. It is generally expected that significant cyber incidents will happen sooner in the cyber domain, but it is generally regarded as unclear in which domains the first “critical” incidents will occur.

linked to having a substantial amount of harmful and dual-use content in the pretraining corpus.³ Deliberate specialization is a secondary path that either focuses a generally-capable model for a sensitive domain or involves purpose-building a narrow tool.

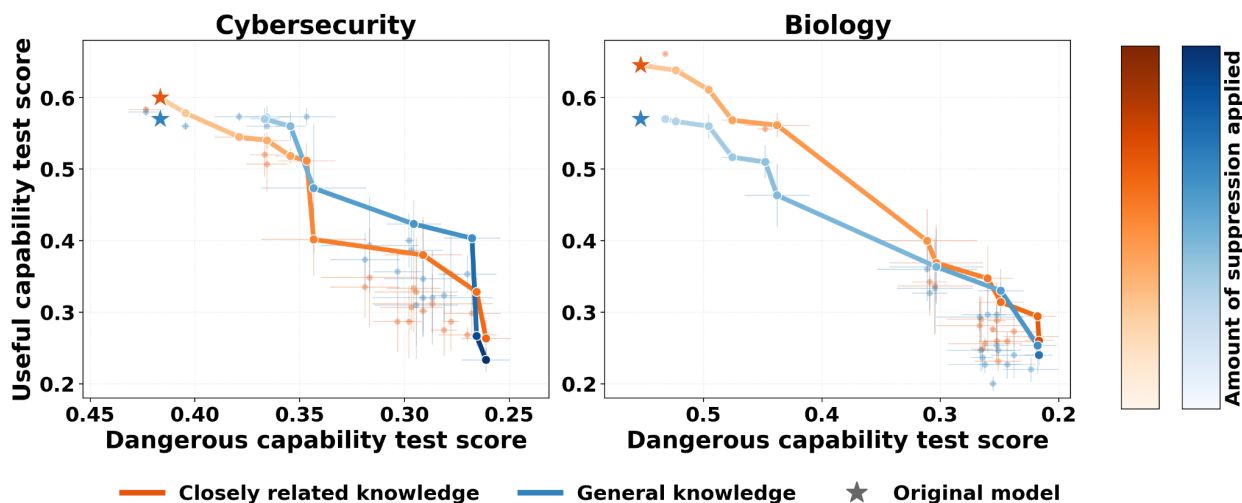
Dangerous capability largely emerges as a byproduct of general-purpose training. Frontier developers optimize for broad competence, but empirically, approaches to this have led to AI systems gaining dual-use capabilities. For example, Anthropic reported that it did not explicitly train its Mythos Preview model for cyber-exploitation capability, and the capability emerged as a downstream consequence of general improvements in code, reasoning, and autonomy (Anthropic, 2026). This happens for two reasons. First, scaling general-purpose training consistently improves model performance across tasks, including dual use ones, making capability gains largely a function of scale as opposed to deliberate specialization (Kaplan et al., 2020; Hoffmann et al., 2022). Consistent with this, dual-use capability has repeatedly increased across cyber (AISL, 2025), biology (Justen, 2025; Liu et al., 2026; Götting et al., 2025; SecureBio, 2026), and chemistry (Mirza et al., 2025). Second, capabilities are somewhat entangled across domains. Training on one domain often improves capabilities in others through data synergy, and some capabilities emerge by the co-occurrence of multiple domains in training (Aryabumi, 2024; Hamidieh et al., 2026). Therefore, dangerous capability can emerge to some extent from interactions among individually benign data.

Deliberate specialization is an additional pathway where a developer aims for capability in a sensitive domain. Deliberate specialization can happen in two ways. First, a developer can focus a generally-capable model on a narrow domain. These models inherit the base model’s capabilities and narrow them toward the domain via fine-tuning. For example, WhiteRabbitNeo is an open-source, uncensored, offensive-security model fine-tuned from a general purpose base model. Additionally, OpenAI announced a trusted user program that grants access to GPT-5.5-Cyber, which is a fine-tuned variant of GPT 5.5 (OpenAI, 2026). Second, a developer can create purpose-built domain-specific tools. These are trained from the outset on the sensitive task, so the specific capability is the design goal. Open-source biological design tools like RFdiffusion, ProGen2, and Evo 2 are built from the ground up to achieve the goal of biological design but are dual-use by construction (Sandbrink, 2023; Wittmann et al., 2024). In chemistry, ChemCrow is a specialized synthesis agent and Urbina et al. (2022) showcased an instance in which a specialized drug-design model was flipped to generate ~40,000 toxins, including VX (a deadly nerve agent), in six hours.

The developer of the AI system implicated in a terrorism event might have made the choice to avoid degradation of legitimate or dual-use capabilities at the cost of leaving some harmful capabilities in the system. This tradeoff is present among all general purpose models and derivatives. When harmful and benign capability are entangled, a developer cannot remove one without ‘taxing’ the other, so they often accept a known dual-use capability as the price of utility. For example, in biology, understanding how a pathogen transmits and evades immunity is required to design vaccines and countermeasures, but is also the same understanding required to make a pathogen more transmissible or harder to treat. So a developer trying to remove harmful biological capabilities would tax benign biological capabilities as well. This tradeoff generalizes the safety or alignment tax documented for safety fine-tuning, where making a model safer often degrades its general capability (Lin et al., 2023; Huang et al., 2025). This tax is steepest in domains like cyber, biology, and chemistry in which legitimate and harmful capabilities are closely related. However, this need not be the case. Drawing from the previous example about biology, a developer could ‘pay the safety tax’ in the publicly available system while serving legitimate researchers through a separate specialized model with vetted access. Some developers already do this for cyber (Appendix C).

3 Pretraining on dual-use content is not strictly necessary. An open-weight model whose pretraining corpus was filtered can still acquire dangerous capability through sufficient downstream fine-tuning (Section 3.3). But comprehensive pretraining-data filtration is difficult and rarely reported on (Section 3.4) so harmful capabilities are often present in practice.

Figure 3



Suppressing dangerous capability also suppresses some useful related capabilities. Each line traces one model as suppression is applied more and more strongly (light -> dark); the star marks the original model before any suppression. Points further up and to the right are more desirable. Moving rightward means that less dangerous capability remains, and in both cybersecurity (left) and biology (right) it trades off with useful capability.

This tradeoff is illustrated directly by running a capability-suppression algorithm to suppress biohazard- and cyberoffense-related knowledge in a model at different levels of suppression strength. The result traces a Pareto frontier between harmful capability removed and benign same-domain and general capability retained (Figure 3; detailed in Appendix A). Appendix B examines how embedding geometry might signal this entanglement, along with the limits of that evidence. Because the dominant industry paradigm is to scale general capability under competitive pressure, and because paying the tax means shipping a less capable model than one's competitors, the tax often goes unpaid by model developers choosing to leave dual-use capabilities in.

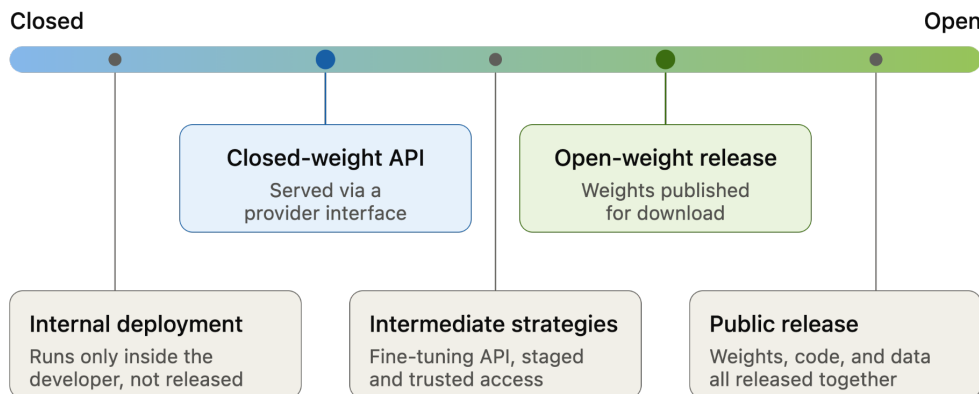
The incentive to build AI systems with frontier dual-use capabilities runs deeper because they empirically serve marketing purposes by signaling the advanced capabilities of AI companies. In April 2026, Anthropic announced its Mythos model by foregrounding its superhuman ability to quickly find and chain software vulnerabilities. Shortly thereafter, Anthropic received record demand for its products and achieved the highest valuation of any AI startup in May 2026 (Fortune, 2026; Anthropic, 2026). This is not to say that such disclosure is legitimate, valuable, or necessary. However, it is one of multiple examples illustrating how the development of advanced dual-use capabilities can serve as a 'hype' mechanism and signal of a lab having state-of-the-art AI capabilities (Zhu, 2025). This is not an argument against harm or risk disclosure, but it highlights a potentially perverse dynamic in which developing dangerous capability rather than suppressing it can be commercially incentivized.

Dangerous capability arises mostly as a byproduct of scaling general-purpose training, with deliberate specialization a secondary path. Because harmful and benign capabilities are entangled, removing one taxes the other, and under competitive pressure that tax often goes unpaid, so companies may deploy systems with risky capabilities.

3.2 How the Dangerous Capability Would be Deployed

Beginning in this section, our discussion of risk scenarios largely forks into two paths: closed-weight vs. open-weight deployments.

Figure 4



AI model deployments lie on a spectrum from fully closed to fully open. The spectrum of deployment ranges from completely closed internal deployments to fully public releases of AI systems with closed-weight API and open-weight deployments being the predominant types.

While deployment is a continuous spectrum, the two configurations that dominate in practice are closed-weight models served through an application interface and open-weight models released for download. Fully internal deployment lies further closed, and fully open-source and open-science release lies further open (Figure 4). Between them sit intermediate strategies that shift a model’s risk profile by changing how much the provider can still monitor and intervene. These strategies are detailed in Appendix C.

Closed-weight deployment: A deployment is *closed-weight* if the model deployer retains exclusive possession of the weights and exposes the model only through interfaces it controls (i.e., API, chat interface, third party) (Bommasani et al., 2023).

However, if an attacker can exfiltrate the weights from a closed-weight model developer, they can then modify and execute it independently, equivalent to an open-weight model. Model weight exfiltration is out of scope for this work because it requires an information-security failure upstream of the development and deployment choices we analyze (Nevo et al., 2024). Nonetheless, the implications about open-weight safeguards apply here too, since only upstream durable safeguards baked into the model’s weights would be present after exfiltration.

Open-weight deployment: A deployment is *open-weight* if the model developer publicizes the model weights for any third party to download, modify, and execute independently (Casper et al., 2025). Once a model has been released with open-weights, it is possible for the developer to revoke download access, but it is not possible to stop the usage or spread of previously-downloaded versions (Seger et al., 2023; Seger & O’Dell, 2024). Currently, open weight model distribution platforms such as Hugging Face generally do not have

requirements for account verification, documentation, or risk reporting for models hosted on their platforms, even very large ones. As a result, there are few barriers or points of transparency for when frontier open weight models are initially uploaded or when modified versions of them are re-uploaded to model distribution platforms. It is common for AI model platforms to have content policies, but it is much less common for such policies to be consistently, actively enforced (e.g., [Maiberg et al., 2025](#)).

Closed- versus open-weight deployments pose two distinct risk scenarios. Because closed-weight models are ahead of open-weight models by several months in frontier capabilities ([AISL, 2026](#)), the risk of counterfactual uplift, if fully elicited, is higher than open-weight models. However, this lead may not extend to certain threat pathways. [Dev et al. \(2026\)](#) found that an open-weight model outperformed a closed-weight model on an end-to-end biological threat pathway because it refused less. [Anthropic \(2026\)](#) found that a leading open-weight model could develop working cyber exploits end to end, near the level that their closed-weight model, Mythos Preview, can. This suggests that open-weight models may already be capable enough for some real misuse pathways regardless of the frontier gap. In the closed-weight scenario, the attacker reaches the model only through the provider's interface and must elicit capability through it, using techniques covered next in Section 3.3, while the provider can monitor behavior and react. In the open-weight scenario, the attacker possesses the weights, so the runtime interventions and system controls that constrain closed deployment are generally absent. The attacker can prompt, scaffold, and modify the model directly to bypass safeguards and run it independently, by methods detailed next in Section 3.3, with no provider present to monitor or react.

3.3 How the Dangerous Capability Would be Elicited

Open-weight models face strictly more elicitation options than closed-weight models, and these additional techniques are much more powerful. For a closed model accessed via an application interface (e.g., an AI chat website), users are restricted to eliciting harmful capabilities using prompts or data inputs. An attacker is confined to the interface and must elicit harmful behaviors from the system by either taking advantage of absent safeguards or defeating ones that are present (e.g., with a 'jailbreak'). Since the techniques for eliciting harmful capabilities from closed-weight models only require access to prompting the model or inputting data, these techniques also work against open-weight models who can also interact with users via an application interface. However, the predominant ways to elicit harms from an open model involve an attacker downloading the weights; simply disabling safety scaffolding and filters; and removing model-based safeguards directly through few-shot fine-tuning or other modifications. For both closed and open models, task decomposition (splitting a complex task into simpler subtasks) and model scaffolding (specialized prompts, tools, or procedures that augment model capabilities) can further elevate harmful capabilities agnostic of deployment type.

3.3.1 Closed weight elicitation techniques

'Jailbreaking' elicits harmful model outputs through prompt manipulation. During a jailbreak attack, the user crafts a prompt that suppresses the model's refusal behavior at runtime. There are many jailbreaking techniques that have evolved from early methods ([Zou et al., 2023](#); [Chao et al., 2023](#); [Anil et al., 2024](#)) into current state-of-the-art methods such as BoN ([Hughes et al., 2024](#)), AMS ([Nasr et al., 2025](#)), STACK ([McKenzie et al., 2026](#)), BPJ ([Davies et al., 2026](#)), and SEMA ([Feng et al., 2026](#)). A taxonomy of jailbreaks

provided by Anthropic distinguishes minor jailbreaks (which recover only benign requests a classifier blocks out of caution), narrow harmful jailbreaks (which unblock a specific harmful behavior), and universal jailbreaks (which unblock an entire class of harmful behaviors without degrading capability), and reports that most discovered jailbreaks fall in the first two categories, with universal jailbreaks being the most rare and most dangerous case ([Anthropic, 2026](#)). For a well-safeguarded closed model, an attacker generally needs a state-of-the-art jailbreaking method (and for filters and user-monitoring measures to be toggled off) to create a narrow-harmful or universal jailbreak to elicit dangerous capability. Expert red-teamers now require substantial and increasing effort to find one (over 40x more time between two models released 6 months apart), and longer still for non-expert red-teamers ([AISL, 2025](#)). Once found, a jailbreak can be patched. Even when models are jailbreakable filters and user monitors can usually catch or prevent them from generating harmful outputs (Sections [3.4](#) and [3.5](#)).

‘Indirect prompt injection’ techniques can bypass some guardrails by passing harmful instructions through a data channel rather than the user’s request. During an “indirect prompt injection” attack, the attacker hides instructions inside content the model ingests as data, such as a tool’s output, a retrieved document, or other agent-reachable text, so the model executes them as if they were trusted input and overrides its own system prompt or tool restrictions ([Greshake et al., 2023](#)). However, frontier closed model providers often monitor data channels extensively.

‘Covert malicious fine-tuning’ reaches the same outcome as open-weight fine-tuning, but through a monitorable fine-tuning API, which only some developers offer for some models. Some providers, such as OpenAI, offer fine-tuning as a service. They allow users to upload custom data, which the provider then fine-tunes a model on, before giving the user query access to the closed-weight fine-tuned model. It is common for fine-tuning service providers to screen submitted fine-tuning data for harmful content and to monitor the resulting model. In this scenario, an attacker can encode the harmful behavior across the training set so that no single example looks harmful on its own, even though the model still learns the unsafe capability from them together. Because each example passes inspection individually, this point-wise undetectable attack slips through data screening while still degrading safety ([Halawi et al., 2024](#); [Davies et al., 2025](#)).

3.3.2 Open weight elicitation techniques

When an attacker attempts to remove safeguards from a newly released frontier open-weight model, they can reach for a variety of techniques including those mentioned above for closed-weight models. Open-weight models can be subject to strictly more elicitation techniques, including ones that are fundamentally powerful due to the attacker possessing the weights of the model. We refer to the following post-release weight modifications collectively as “tampering”: any modification of a released model’s weights or deployment scaffolding that removes or degrades its safeguards. Open-weight models can ship without adequate safeguards, leaving the capability directly available for an attacker to use. Or someone else may have already stripped a safeguarded open-weight model and redistributed it via a platform such as [Hugging Face](#), so the attacker only has to download the result. Two examples of this are WormGPT and Stable Diffusion 1.x. WormGPT was a text-generating chatbot built from an open-weight base model and fine-tuned on malware and fraud data, then sold on criminal forums to write phishing emails and malware ([Firdhous et al., 2023](#)). Stable Diffusion 1.x, an open-weight text-to-image model, shipped with a safety filter that could be removed by modifying a single line of code, after which users further adapted it to generate non-consensual sexual imagery and child sexual abuse material ([Rando et al., 2022](#); [Kamachee et al., 2025](#)).

Malicious and non-malicious fine-tuning alike can strip away an open-weight model’s safety measures. Fine-tuning modifies the underlying neural network parameters directly. By training an aligned model on a

dataset of thousands, hundreds, or even tens of harmful, explicit, or unrestricted instructions, users can strip away a model's safety measures and force compliance with hazardous prompts (Qi et al., 2023; Yang et al., 2023; Bhardwaj & Poria, 2023; Li et al., 2024; Lynch et al., 2024; Huang et al., 2024; Hu et al., 2025; Łucki et al., 2025; Peng et al., 2024; Deeb & Roger, 2025; Qi et al., 2024; Che et al., 2025; Casper et al., 2025; Dev et al., 2026). Safety measures even degrade under non-malicious fine-tuning on ordinary tasks with no harmful intent if the fine-tuning data does not contain examples of the model safely refusing harmful requests (Qi et al., 2023; Hu et al., 2025; Łucki et al., 2025). Not only is safeguard removal via fine-tuning achievable with little to no performance loss, but it is also generally cheap, sometimes requiring only several dollars or even cents of compute (Qi et al., 2023; Yang et al., 2023; Bhardwaj & Poria, 2023; Li et al., 2024; Lynch et al., 2024; Huang et al., 2024; Hu et al., 2025; Łucki et al., 2025; Deeb & Roger, 2025; Qi et al., 2024; Che et al., 2025; Casper et al., 2025; Dev et al., 2026). In this scenario, the attacker downloads a capable open-weight model and runs one of the many off-the-shelf fine-tuning frameworks on a small curated dataset, producing a local derivative that no longer refuses harmful requests.

Automated refusal ablation (a.k.a. “abliteration”) removes refusal with no specialized fine-tuning data and no expertise. Safe refusal refers to the safety behavior in which a model declines harmful requests. Empirically, refusal behavior can often be controlled by simple perturbations to the internal activations of models (Arditi et al., 2024). *Heretic* (Weidmann, 2026), for example, is an open-source automated toolkit that can, for example, uncensor a model like Llama-3.1-8B in under an hour on a single GPU, requiring no understanding of model internals. Heretic and other tools feed an ecosystem of thousands of pre-stripped models an attacker can download rather than produce (Young, 2025; Sokhansaj, 2025). A team at Anthropic demonstrated this by “abliterating” a leading open-weight model, GLM 5.3, and recorded that with as little as \$4,400, GLM 5.3's refusal rate to dangerous requests dropped from 90% to 2-3%. Several developers had already released “abliterated” versions of GLM-5.3 publicly within days of its release (Abliteration.ai, 2026).

Other less common weight modifications can also elicit harmful capabilities. A range of other weight modifications, including model merging (Hammoud et al., 2024), pruning (Wei et al., 2024), low-rank edits (Wei et al., 2024), quantization (Egashira et al., 2024; Chen et al., 2025), and latent-space attacks (Bailey et al., 2025), can also strip an open model's safety.

A single actor who modifies an open-weight model to make it unsafe can (and often does) then make it available to anyone in the world. Removing safeguards from open weight models via fine-tuning or other methods requires some familiarity with machine learning, approximately at the level of an undergraduate computer science student. This poses a significant, though easily surmountable barrier, especially when using AI coding agents. However, once a model is modified unsafely, it can then be re-shared widely. For example, at the time of writing this paper, there are currently over 14,000 openly downloadable “uncensored” and “abliterated” open-weight models on Hugging Face able to be downloaded by anyone in the world. Openly available uncensored models include ones at the frontier of open model capabilities and near the frontier of AI capabilities overall. For example, recently, the company OrcaRouter openly released an uncensored version of GLM 5.3 (OrcaRouter, 2026), an open-weight model that is estimated to have the same capability level as Claude Opus 4.6 and be roughly 6 months behind the overall frontier AI state of the art (Epoch, 2026).

3.3.3 Scaffolding and task-decomposition techniques

These techniques can elicit harmful capabilities and circumvent safeguards whether it is open-weight or closed-weight.

Task decomposition splits a harmful objective into benign-seeming subtasks that models tend not to refuse. The attacker reframes a harmful goal as a set of innocuous queries, routes each to whatever model answers it best, and reassembles the results, exploiting the fact that a combination of individually safe responses is not itself safe (Glukhov et al., 2024; Jones et al., 2024). For example, asking questions about how ignition systems, steering-column wiring, and starter circuits work are individually benign requests that, when composed together, can teach a user how to hotwire a car. However, bypassing refusal does not equate to eliciting actionable harm. Hotwiring is easy to recompose because it can be explained in layman’s terms and the pieces are individually simple. Misuse tasks, however, are generally more complex in domains like advanced cyber, biology, and chemistry. Even knowing what questions to ask can be a significant challenge absent access to additional expertise.

Agentic ‘scaffolding’ raises a model’s effective capability and can erode safeguards. An ‘scaffold’ (also known as a harness) is a piece of software that prompts a model, offers it tools (e.g., for web search or memory), and walks it through the process of planning and executing tasks step-by-step. This greatly increases the autonomous capabilities of a model and allows it to conduct end to end attacks (Singer et al., 2025; Deng et al., 2024; Xu et al., 2024). Scaffolding can transform a model into an enhanced autonomous agent able to conduct operations without humans in the loop. Beyond raising capability, the harness itself can erode safety. Agents built on aligned models comply with harmful requests that the same models refuse when queried directly, because decomposing a harmful task it into steps and feeding back tool observations all push the model off its safety-trained distribution (Chiang et al., 2025; Yu et al., 2025; Mazeika et al., 2024).

Table 2

Technique	Access required	Cost	Skill Required	Reliability	Applies to
Runtime-level — closed-weight interface with state-of-the-art safeguards					
Jailbreaking	Interface	High	Prompt Engineering	Decaying, patched once found	Closed and open models
Indirect prompt injection	Data / agent / tool channel	High	Data/Prompt Engineering	Variable, heavily monitored	Closed and open models
Covert malicious fine-tuning	Fine-tuning API	Medium	ML Engineering	Variable, heavily monitored	Closed models (Fine-tuning API)
Weight-level — open-weight					
Malicious fine-tuning	Model weights	Low	ML Engineering	High	Open models
Automated ablation	Model weights	Low	ML Engineering	Medium	Open models
Other less common weight-modification techniques	Model weights	Low	ML Engineering	Medium	Open models
Scaffolding — closed- or open-weight					
Task decomposition across models	Interface(s)	Medium	ML Engineering	Medium	Closed and open models
Agentic harness	Any	Medium	ML Engineering	High (amplifier of effective capability)	Closed and open models

Harm can be elicited from both closed- and open-weight models with varying efficacy and reliability. Jailbreaking and task decomposition appear to be the most powerful techniques for eliciting harm from closed-weight models and weight modification techniques are fundamentally powerful for eliciting harm from open-weight models. Prompt-based attacks on closed models are effective but more difficult and less certain. The UK AI Security Institute has found a universal jailbreak in every closed-weight system it has tested, but the effort required is rising and has recently required user monitors to be toggled off for testing purposes during the process of developing the jailbreak. By one measure, effort rose 40x between two models released six months apart, and each jailbreak is patched once found (AISI, 2025). National evaluators identify open-weight systems as the hardest to defend, because their safeguards can be fine-tuned or stripped away (AISI, 2025). Systematic red-teaming confirms this finding that across 21 open-weight models spanning families, scales, and alignment-stage defenses, every one admitted a tampering attack that sharply raised harmful compliance while preserving general capability (Hossain et al., 2026). Weight modification typically requires limited familiarity with machine learning engineering to perform. However, malicious users need not always have this expertise themselves in order to misuse open models. The thousands of “uncensored” and “abliterated” models already redeployed on model distribution sites make access to these capabilities very simple (Sokhansaj, 2025).

This survey of techniques across closed- and open-weights, and model scaffolds largely ignores the characteristics of the threat actor. This can range from lone and small group actors, as evidenced by the Las Vegas and Palm Springs bombings (PBS, 2025; CNBC, 2025), to decentralized networks. Concerning the latter, Juelich (2026) documents jailbreaking techniques circulating among thousands of individuals and the terrorist group, Boko Haram, which formed dedicated AI units with training delivered by Islamic State operatives.

3.4 How Safeguards Would Fail

Safeguards fail in one of two ways: either by not being applied in the first place or by being defeated. In the first case the failure is a developer choice to not deploy the model without the best available protection. There is direct precedent for this, both for closed- and open-weight models. For example, the company Abliteration.ai builds its business model on serving API access to uncensored models. Meanwhile, thousands of openly-available uncensored models are downloadable from mainstream internet platforms such as Hugging Face (Sokhansaj, 2025). In the second case, the effort required to defeat a deployed safeguard differs sharply by deployment type. For a safeguarded closed-weight model, an attack must circumvent every runtime safeguard layer in place (e.g., input filters, model refusal, output filters) at once which typically must be done through multi-staged attacks which develop exploits for one layered safeguard at a time (e.g., first a prompt filter, then a model’s refusal training, then a model output filter, etc.) (McKenzie et al., 2026). They require a number of queries that raise many opportunities for flagging suspicious account activity (Section 3.5). For a safeguarded open-weight model, much less effort is needed to defeat safeguards because any content filter or other safeguard that is not ‘baked into’ model weights can simply be disabled while the rest can be removed by modifying the weights.

3.4.1 Closed-Weight Safeguards

Closed-weight deployment permits several interventions that can be individually porous yet strong in combination. Input filters screen a user's request before it reaches the model, and output filters screen the response before it is sent to the user (Inan et al., 2023). Additionally, system-state monitors watch behavior across a(n) model's/agent's loop of planning and tool calls, flagging or blocking harmful content (Greenblatt et al., 2023). State-of-the-art monitoring techniques can be quite robust. When stress testing production monitors, Anthropic reported that theirs withstood over 1,700 hours of red-teaming (Cunningham et al., 2026) while maintaining a rate of improper filtering of 0.5%. Subsequent work found exploits for these filters (Davies et al., 2026). However, developing these exploits proved very query intensive. Davies et al., (2026) reported that their method could not work with standard user monitoring and banning procedures in place (see Section 3.5.2). Activation probes are another cheap runtime defense layer. They involve using a small classifier on the model's internal activations that flags harmful requests at a fraction of the cost of a second monitoring model, and have been deployed in production on a frontier system (Kramár et al., 2026; Hollinsworth et al., 2026).

Despite the robustness of state-of-the-art approaches, individual safeguards, when implemented by AI companies, have typically been defeatable. Automated attacks find universal jailbreaks against classifiers with enough effort (Section 3.3), and staged attacks can chain past several layers in sequence by defeating each in turn (McKenzie et al., 2025). This has been demonstrated against open-weight safeguard pipelines and frontier closed deployments where staged attacks elicited harmful biological-weapons responses from both Claude Opus 4 and GPT-5, which the providers then patched (McKenzie et al., 2025). More recently, OpenAI subjected GPT-5.6 Sol to over 700,000 GPU-hours of automated red-teaming plus model-level refusals, real-time classifiers, and account-level review (OpenAI, 2026), yet the UK AISI reported finding universal jailbreaks in every round of testing, enabling long-form agentic vulnerability discovery and exploit development while preserving the model's offensive-cyber capability (Davies, 2026). UK AISI notes it had red-team access to the safety monitor's reasoning and policy that a real attacker would lack, and that the attacks are likely still findable without it at higher but unknown cost. A real attacker would also have to find the exploit against the production system, where the volume of queries required would likely trip account-level monitoring and result in a ban before a working attack was found. Defeating one layer, or even two or three, is not the same as defeating a full safety stack under real-world conditions, but the direction of the evidence is that layering them greatly raises the cost of misuse.

In deployment, production closed-weight system safeguards still have vulnerabilities. Significant vulnerabilities have been found in deployed systems. For example, a large red-teaming study across many frontier agents in realistic production deployment settings found that all could be pushed into policy violations, that different companies' models had significantly different safeguard strengths, and that robustness did not improve with model capability (Zou et al., 2025). However the violations were deployment-policy breaches rather than weapons-relevant uplift, and this study predates current safeguards which have recently improved (AISI, 2025). There is also a significant gap between the state of the art and the state of practice due to the engineering costs and occasional trade-offs with legitimate capabilities that state-of-the-art safeguards require (see Section 7).

3.4.2 Open-Weight Safeguards

Pretraining data filtering is currently the state of the art and has been credibly shown to resist adversarial fine-tuning on several tens of thousands of academic papers on dual-use science. Pretraining filtering

removes harmful material from the training set so a harmful capability is never learned. Its cost can be made negligible relative to total training, or even negative (O'Brien et al., 2025). The amount of high quality data required to execute such fine-tuning runs matters because data set construction rather than code writing or compute cost is most likely to be the most difficult and expertise-intensive step behind conducting such fine-tuning operations. Hazard-relevant scientific text must be identified and curated from the broader literature, which takes subject-matter expertise. As a sense of scale, WMDP, a standard benchmark for biology, chemical, and cyberknowledge, was built by a panel of experts curating roughly 24,500 papers (~150M tokens) and reportedly cost over \$200,000 to develop (Li et al., 2024). This illustrates how tamper resistance can make the expertise and cost required for harmful dataset collection major bottlenecks to misuse. Current methods based on filtering harmful pretraining data are useful but limited, having only been demonstrated to credibly resist fine-tuning on approximately 24,500 scientific papers' (~150M tokens) worth of content (O'Brien et al., 2025) in models approximately 5 billion parameters in size.

Capability suppression (a.k.a “unlearning”) algorithms use examples of harmful data to selectively suppress dangerous capabilities. Empirically these methods typically struggle to remove a harmful capability fully, and the field often overstates their durability (Qi et al., 2024; Che et al., 2025). A literature-wide review by Casper et al. (2025) found that the current state of the art for tamper-resistance achieved by capability suppression fine-tuning algorithms, as assessed by second-party red-teaming, holds only for adversarial fine-tuning on *several hundred documents*. We overview specific methods and their failure modes in Appendix D.

Limiting a model’s capacity and intelligence in dual-use domains such that no realistic amount of modifying the model can elicit/create dangerous capability is a reliable way to protect open-weight models from misuse of dangerous capabilities. If a model is never made to be large and capable enough to provide meaningful uplift on a hazardous task, fine-tuning and other elicitation methods pose little marginal risk. For example, OpenAI designed its open-weight gpt-oss family of models to be no larger than 120B parameters in size and developed them to have limited capabilities as a condition of releasing the weights (OpenAI, 2025).

Table 3

Safeguard	State-of-the-art Reference	Layer	Reliable For	Tampering Durability	Prompt-based attack Durability	Failure Modes of Current Methods
Pretraining data Filtering	O'Brien et al., 2025	Pre-training	Both	Moderate	Strong	Fine-tuning on over 10,000 dual-use scientific papers
Capability Suppression Training Algorithms	Zou et al., 2024; Dékány et al., 2025	Post-training	Both	Weak	Moderate	Sufficient fine-tuning or steering
Input/Trajectory/ Output Filters	Cunningham et al., 2026	Runtime	Closed	None	Strong	State-of-the-art jailbreaks
Activation Probes	Kramár et al., 2026 Hollinsworth et al., 2026	Runtime	Closed	None	Moderate	State-of-the-art jailbreaks

State-of-the-art safeguards for open-weight models are much easier to circumvent compared to state-of-the-art safeguard stacks permitted by closed-weight model deployment. Pretraining-data filtering of hazardous content currently appears to be, by a significant margin, the most effective single method safeguarding open-weight models against harmful fine-tuning (O’Brien et al., 2025), but fine-tuning methods can be useful additions. Additional small-scale research on related methods may also be promising (e.g., Shilov et al., 2025; Roland et al., 2026). In contrast, closed-weight deployments can prevent weight modification and rely on a layered defense involving multiple filters and monitors. Closed weight models, however, should not escape scrutiny. Although they benefit defenses in depth, closed-weight models generally appear to be vulnerable in practice to sophisticated attacks.

3.5 How Detection Would Fail

Detection is a strong defensive paradigm for closed-weight model developers but is largely ineffective for open-weight model ones. In this case, we refer to *detection* as the ability for a model deployer to understand what a model can do and how it is being used, well enough to act on it. Across the model lifecycle and ecosystem, technical and governance mechanisms for detecting catastrophic misuse can fail. The capability of a system can be mismeasured or unmeasured (3.5.1), runtime monitoring can be absent or evaded (3.5.2), and even when risk is identified, no decision or reporting may follow (3.5.3).

3.5.1 Capability was mismeasured/unmeasured

Measuring dangerous capability is hard to do in a way that is both rigorous and publicly checkable. Some dangerous capability benchmarks show high levels of saturation, meaning they have limited ability to distinguish between today’s leading models (Akhtar et al., 2026). Anthropic, for instance, concluded that the public Cybench benchmark “no longer sufficiently informative of current frontier model capabilities” after Mythos Preview scored 100% (Anthropic, 2026). Public biology and chemistry benchmarks are approaching the same ceiling (Dev et al., 2025). As a result, the evaluations used to assess frontier misuse risk are increasingly private, held by developers and a small number of third-party evaluators, and are neither standardized nor externally verifiable. Outside observers therefore cannot confirm whether a given model was adequately tested for dangerous capability before release, or what those tests found. Additionally, existing benchmarks measure technical capability, while the operational assistance documented in real cases has no established measurement at all (Juelich, 2026; Hunter & Righetti, 2026). Anthropic recently published the first capability evaluations we are aware of for conventional weapons development, finding that models could write guidance software for simulated drone strike and payload-delivery tasks, and implemented classifiers in response (Anthropic, 2026). However, national evaluators and other leading frontier AI developers do not report these evaluations.

For open-weight and closed-weight models, public reporting on tampering evaluations is almost nonexistent (Table 4). OpenAI conducted a fine-tuning study of its open-weight gpt-oss models in 2025 as a notable exception for reporting tamper evaluations (Wallace et al., 2025). However, other frontier open-weight models ship without any reported evaluation of how easily their safeguards can be fine-tuned away, despite tampering being the central threat these models face. A larger review of open-weight releases through April 2026 found almost none met proportional evaluation criteria (Paskov et al., 2026). The same gap exists for closed-weight models, fine-tuning APIs expose closed models to similar attacks (Halawi et al., 2024; Davies

et al., 2025), yet no closed-weight model developers report evaluating them (Table 4). Additionally, there is no explicit reporting of pre-training filtering of hazardous content, even though it is a state-of-the-art safeguard available. A lack of public reporting does not necessarily mean there is also a lack of implementation, but currently the public has little reason to trust that frontier AI models can remain safe under even small amounts of fine-tuning.

Table 4

	Cyber Capability Evals	Bio Capability Evals	Chem Capability Evals	Tampering Evals	Safeguard Robustness	Human Uplift Study	Pretraining Filtering (Cyber/Bio/ Chem)	Monitoring / KYC
CLOSED-WEIGHT DEVELOPERS								
Anthropic Claude Fable/ Mythos Anthropic, 2026	≥ 1 Paragraph Extensive evals + external testing	≥ 1 Paragraph Extensive evals + external testing	≥ 1 Paragraph Extensive evals + external testing	No Mention	≥ 1 Paragraph Extensive red-teaming + external testing	≥ 1 Paragraph Bio/virology and non- expert chem uplift trials	No Mention	≥ 1 Paragraph Real-time classifiers, access controls, Cyber Verification Program, bug bounty, threat intel
OpenAI GPT 5.6 OpenAI, 2026	≥ 1 Paragraph Extensive evals + external testing	≥ 1 Paragraph Extensive evals + external testing	≥ 1 Paragraph Extensive evals + external testing	No Mention	≥ 1 Paragraph Extensive red-teaming + external testing	No Mention Evals as proxy	No Mention	≥ 1 Paragraph Activation classifiers, two-tier monitors, actor-level enforcement, Trusted Access
Google DeepMind Gemini 3.7 Flash Google, 2026	≥ 1 Paragraph Extensive evals	≥ 1 Paragraph Extensive evals + external testing	≥ 1 Paragraph Extensive evals	No Mention	≥ 1 Paragraph Extensive red-teaming + external testing	No Mention Evals + red- teaming as proxy	No Mention Private info, CSAM, NSFW only	≥ 1 Paragraph Query filters, prompt- injection classifiers, I/O filtering, access controls
SpacexAI Grok 4.6 SpacexAI, 2026	≥ 1 Paragraph Agentic cyber benchmarks, jailbreak robustness	≥ 1 Paragraph CBRN benchmarks	≥ 1 Paragraph Chem within unified CBRN	No Mention	≥ 1 Paragraph Extensive red-teaming + external testing	No Mention	No Mention	≥ 1 Paragraph Acceptable- use policy linked; monitoring implied

	Cyber Capability Evals	Bio Capability Evals	Chem Capability Evals	Tampering Evals	Safeguard Robustness	Human Uplift Study	Pretraining Filtering (Cyber/Bio/ Chem)	Monitoring / KYC
Meta Muse Spark 1.2* Meta, 2026	≥ 1 Paragraph Extensive evals + external testing	≥ 1 Paragraph Extensive evals + external testing	≥ 1 Paragraph Extensive evals + external testing	No Mention Excludes malicious fine-tuning	≥ 1 Paragraph Extensive red-teaming + external testing	No Mention Evals as proxy	No Mention Unspecified domain filtering	≥ 1 Paragraph Misuse classifiers, long-term behavior monitoring, post-launch monitoring
OPEN-WEIGHT DEVELOPERS								
Zhipu GLM 5.3 Zhipu, 2026	≥ 1 Paragraph Extensive cyber evals	No Mention	No Mention	No Mention	No Mention	No Mention	No Mention	No Mention
Moonshot AI Kimi 3 Kimi, 2026	No Mention	No Mention	No Mention	No Mention	No Mention	No Mention	No Mention	No Mention
DeepSeek DeepSeek V4 DeepSeek, 2026	No Mention	No Mention	No Mention	No Mention	No Mention	No Mention	No Mention	No Mention
Alibaba Qwen 3.8 QwenTeam, 2026	No Mention	No Mention	No Mention	No Mention	No Mention	No Mention	No Mention	No Mention
Motif Motif 3 Motif, 2026	No Mention	No Mention	No Mention	No Mention	No Mention	No Mention	No Mention	No Mention

* Muse Spark 1.2 has not yet come out with a safety report, so we derive information from the Muse Spark 1.1 evaluation report.

Dangerous-capability evaluations and safeguards are inconsistently reported on for closed-weight model developers and almost never reported on for open-weight ones. We selected the flagship (high-est-scoring) model from each of the five leading closed-weight and five leading open-weight developers on the [Artificial Analysis Intelligence Index](#) (v4.1), and audited each model’s public documentation (system card, model card, safety framework, launch blog, or technical report) for disclosure of dangerous-capability evaluations and safeguards. Cell values reflect the presence and extent of public reporting, which is a proxy for transparency, not a rating of safeguard rigor or implementation. This was conducted in August 2026.

Human uplift randomized control trials are a direct measurement of whether a model can actually help a person accomplish weapons-relevant tasks. They are also expensive, mostly proprietary, and run voluntarily by a few frontier labs and third-party evaluators. Relative to the importance of this measurement, public facing work is thin: RAND has published work on cyber and biology human-uplift studies ([Aguirre et](#)

al., 2026; Mouton et al., 2024; Hong et al., 2026), and SecureBio maintains a synthesis of biology human-uplift trials (SecureBio, 2026). Developers themselves report little. In our audit of leading model cards, only Anthropic reported running an uplift trial, a catastrophic biological scenario tested with five three-person teams (Anthropic, 2026; Table 4). No other developer, open or closed, reported one at all. This does not prove these trials were not run, but it leaves the public with no basis to verify that models are tested for real-world uplift before release.

3.5.2 User and runtime monitoring failed

For closed-weight models, the providers control the model, all points of access, and all outputs sent to users, allowing for effective user monitoring. This allows model providers to implement usage logging, anomaly and abuse detection, account-level controls, rate-limiting, banning, and shadowbanning. When these mechanisms are combined with the model's other safeguards, the resulting safety stack can be made very strong. Considering the value and limitations of safeguards and user monitoring Clymer et al (2025), propose a methodology for constructing a model *safety case* (i.e., a structured, evidence-based, affirmative argument that a system is sufficiently safe to deploy) based on them. Monitoring is the backstop within that case, since even when a runtime safeguard is defeated, state-of-the-art attacks still tend to generate abnormal, often high-volume queries the provider can observe and detect (Davies et al., 2026).

While providers have used monitoring to detect and disrupt harm, detection can fail or fail to be implemented. For example, Anthropic, OpenAI, and Google have published threat reports on adversarial use of their models that they identified and banned through account and usage monitoring (Anthropic, 2026; Anthropic, 2025; OpenAI, 2026; GTIG, 2026). Detection strategies can be vulnerable to task decomposition attacks or other forms of adversarial manipulation (see Section 3.3). Notably, every small-scale, provider-led disruption later surveyed in Section 4.3 involves cyber activity at sustained volume. Monitoring appears effective against the high-volume activity that automated attacks require, but its effectiveness against low-volume human planning spread across accounts and providers is not established. For example, adversaries have run tens of thousands of fraudulent accounts to distill frontier models, spreading extraction thin enough that no single account looks anomalous (CNAS, 2026). Instead of outright account banning, shadowbanning may be a better alternative because it silently degrades a flagged account which denies the adversary any signal to adapt on (e.g., making a new account). In addition, current research has proposed monitoring mechanisms that aggregate signals across multiple contexts and interactions to detect task-decomposition attacks (Sun et al., 2026; Yueh-Han et al., 2025), though this does not prevent multi-provider abuse. Detection strategies also depend on the providers actually instrumenting for it, retaining and reviewing logs, acting in time, potentially sharing threat intelligence, adapting to new exploits, and confronting certain unavoidable tradeoffs between precision and recall.

Reporting of adoption and rigor of implementation varies across providers, so even for closed-weight models, monitoring can fail via lack of thorough implementation. This was recently illustrated when OpenAI's own models, during an internal evaluation, escaped their sandbox and chained individual steps into a real intrusion of Hugging Face's production infrastructure (OpenAI, 2026; METR, 2026). Monitors for dangerous cyber activity outside the scope of what was being evaluated (such as sandbox breakout) were not in place (OpenAI, 2026). Only after the incident did OpenAI report implementing monitoring of whole agent trajectories (OpenAI, 2026), which is a well-established technique that prior work had identified as critical to mitigate harmful and rogue behaviors (Li et al., 2026; Swaminathan et al., 2026; Zhu et al., 2026; Menezes, 2026).

Accounts can be verified prior to model upload and download, but otherwise, monitoring and identity verification mechanisms are generally absent for open-weight models. The main point at which monitoring is available for open weight models is at the time of upload and download. This can be useful for large platforms like Hugging Face but only applies for a platform which implements it. Currently, open weight model distribution platforms such as Hugging Face generally do not have requirements for account verification, documentation, or risk reporting for models uploaded on their platforms, even very large ones. Across the AI ecosystem, registration and know-your-customer measures vary for open model downloads. Once downloaded, an open-weight model can be run locally or on unmonitored cloud instances, so there is generally no traffic for a developer or other party to observe, no logs for them to inspect, and no way for them to monitor activity. However, there are still some useful but unreliable ways to monitor a model's usage by several methods. Agent IDs are a unique, persistent identifier attached to an AI agent so that actions it takes in the world (API calls it makes to other services, transactions, posts, code it deploys) can be traced back to a specific agent instance ([Chan et al., 2024](#)). Compute identity verification applies general identity verification to compute resources. The largest training and inference runs need specialized hardware from a small number of providers (chips, cloud), offering a potentially useful chokepoint to require identity verification ([Egan & Heim., 2023](#)).

3.5.3 Detection did not lead to action

Currently, scientists do not have a rigorous way to convert scores on a benchmark to a precise decision about real-world risk. As a result, a model can achieve a high danger score on a benchmark without any action being taken, because it is currently very difficult to rigorously determine what level of real-world risk a benchmark score corresponds to and what corresponding actions to take ([Righetti, 2025](#)). [Righetti \(2025\)](#) offers a model for how evaluations can be reasonably translated into approximately proportionate action in the biorisk domain, but their estimates nonetheless span orders of magnitude. The thresholds proposed across developer stated frameworks and frontier AI laws may be poorly calibrated for measuring risks. After analyzing the one such threshold, the \$100 billion grid cyberattack scenario from OpenAI's Preparedness Framework, [Van der Merwe \(2026\)](#) conclude that asking whether a model would materially increase the risk of that scenario is not particularly useful for release decisions, and that lower-damage scenarios may be more useful to inform risk assessment.

Incident-reporting requirements exist but do not yet form a clear or unified standard. Meanwhile, there are not yet AI industry standards or clear legally established protocols pre-emptively reporting concerning user activity. Currently most threat reporting by AI companies such as Anthropic, OpenAI, and Google has been done voluntarily. However, three US state laws have recently passed requiring incident reporting for frontier developers, California's SB 53, New York's RAISE Act, and Illinois's SB 315, each requiring disclosure of "critical safety incidents" to a state agency within 72 hours under the RAISE Act and SB 315, and within 15 days under SB 53, or 24 hours for incidents posing imminent risk of death or serious injury, with civil penalties for noncompliance ([SB 315](#); [RAISE Act](#); [SB 53](#)). Only SB 53 is currently in effect; the other two take effect in January 2027. However, their scope also only includes large frontier developers and monitored deployment-time incidents, leaving open-weight models, largely outside any reporting regime. The EU AI Act's Article 55 imposes systemic-risk reporting duties on general-purpose models but shares the same definitional vagueness and blind spot for post-release open-weight misuse ([EU, 2024](#)). Voluntary Frontier Safety Framework and Responsible Scaling Policy commitments by leading AI companies fill some, but not all, of the gap ([METR, 2026](#)).

There is currently no industry-wide standard nor clear legally established protocols for how to identify, escalate, share signs of concerning activity before a harm has materialized. Providers have publicly described detecting and banning accounts engaged in serious misuse ([Anthropic, 2026](#); [Anthropic, 2025](#); [GTIG, 2026](#); [OpenAI, 2026](#)), but none detail the mechanism in which they escalate this information in a timely manner to law enforcement. A failure of this was demonstrated when OpenAI banned an account in June 2025 after automated detection of a usage-policy violation but did not contact authorities, citing an internal threshold that was not met. Eight months later the account holder carried out a mass shooting at a school in Tumbler Ridge, British Columbia ([NYT, 2026](#)). Regardless of whether or not the judgement was reasonable, the threshold was provider-defined with no external oversight. Proposals for structured public-private escalation channels exist, including a standing AI Threat Fusion Center with classified infrastructure and cross-cleared personnel, but none have been implemented ([Hersman, 2026](#)). Anthropic has since reported disrupting several weapons development programs, potential biological plots, and cyber-offensive operations through internal monitoring rather than pre-attack law-enforcement referral ([Anthropic, 2026](#)). Similarly, there is no framework for sharing threat intelligence across model providers to defend against multi-provider abuse. Analogous frameworks have been long-established in the field of cybersecurity ([CISA, 2026](#)).

State-of-the-art detection methods can be very effective for closed-weight deployments and are very unlikely to be effective for open-weight deployments. For closed models, user monitoring can be fairly strong and has caught real misuse. However, it is contingent on the provider choosing to implement monitoring and report incidents, and on those measures surviving an adversary potentially trying to evade them. For open-weight models user monitoring is rarely a reliable option because an open-weight model can be run locally. Attribution, monitoring, and reporting are not generally possible by construction, and the substitutes (agent IDs, deployment-surface logging, and compute identity verification) work only by modifying separable software ‘wrappers’ around the model rather than the model itself. The most consequential of surrounding infrastructure is the distribution platform, which is a point on the open pathway where usage concentrates and some intervention is still possible.

The AI ecosystem lacks publicly verifiable measures of dangerous capability, and threat-specific evaluations largely go unreported. More and more benchmarks for dangerous capabilities are becoming infohazards, and thus held privately without public verification. There is also a lack of model developers reporting predeployment model tampering evaluations and public facing human uplift randomized control trials to accurately measure dangerous capabilities and likely threat models. Furthermore, there are no rigorous ways to convert benchmark scores into precise plans for proportionate action and no industry-wide standards for reporting signs of incidents that have yet to materialize.

4. What History Says

To understand how a large-scale AI-enabled terrorism event may happen in the future, we can look to patterns among smaller-scale events of the recent past. Here, we study precedent in three categories:

1. **AI generated NCII/deepfake abuse (Section 4.1).** These incidents are analogous to hypothetical major AI-enabled terrorism events in how they stem from dual-use capabilities and have been elicited from various models in various ways. They are, however, disanalogous in motivation, the scale of individual harms, and in the wide availability of NSFW fine-tuning data compared to the relative scarcity of harmful cyber, biological, or chemical data.

2. **Criminal and state-actor cybercrime (Section 4.2).** These incidents are analogous to hypothetical major AI-enabled terrorism events in capability, which overlaps with the cyber domain we scope, and in the elicitation pathways. They are, however, disanalogous in motivation and in scale of harm.
3. **Small-scale precedented instances of terrorism (Section 4.3).** These incidents are analogous to hypothetical major AI-enabled terrorism events in motivation and in nature of harm. They are, however, disanalogous in sophistication of capability and resource requirements.

Because our analysis traces choices across the model lifecycle, we include only incidents where the model and its deployment type are publicly documented. Among these smaller-scale events of the recent past, patterns emerge. Harm elicited from closed-weight models occurred when developers did not rigorously implement the full safeguard stack their deployment had permitted. Harm elicited from open-weight models emerged when upstream safeguards, or the lack thereof, did not survive downstream tampering and no detection mechanisms had existed to catch misuse.

Table 5 — What history says: lifecycle stage × case

Stage	Non-Consensual AI Deepfakes	Cybercrime (Criminal & State)	Small Scale Precedented Terrorism
Development	Capabilities have been a product of both general-purpose training and specialization. NSFW capabilities have arisen from general-purpose training on internet data including NSFW content but are sometimes improved by targeted NSFW fine-tuning.	Capabilities have been a product of both general-purpose training and specialization. Scaling general purpose models has resulted in stronger cyber capabilities, but some models (e.g., WormGPT) have been fine-tuned to specialize in harmful tasks.	Capabilities have seemingly only been a product of general-purpose training. Scaling general-purpose training has resulted in stronger hazardous domain and planning capabilities. Adopters have generally used general-purpose frontier models off the shelf.
Deployment	Both closed-weight (e.g., Grok) and open-weight models (e.g., Stable Diffusion, Flux, Wan).	Both closed-weight (e.g., Claude, OpenAI, Gemini, Grok) and open-weight models (e.g., WormGPT, WhiteRabbitNeo).	Both closed-weight general-purpose frontier chatbots (e.g., Claude, OpenAI, Gemini, Grok) and one documented instance of open-weight model (DeepSeek).
Elicitation	Closed-weight models: Grok safeguards have been circumvented with simple prompting strategies. Open-weight models: Safety filters have been stripped; models have been fine-tuned and redistributed.	Closed-weight models: Attackers have used jailbreaks and prompting strategies; APT31 posing as a security researcher to Gemini and GTG-1002 operators framing an intrusion as defensive testing to Claude. Open-weight models: WormGPT had been purpose-built for cybercrime and required no elicitation.	Closed-weight & open-weight models: Prompt-based jailbreaks have been used and refined and shared operator-to-operator; spreading activity across multiple providers' accounts circumvents account controls and allows for task decomposition across models.

Stage	Non-Consensual AI Deepfakes	Cybercrime (Criminal & State)	Small Scale Precedented Terrorism
Safeguards	Closed-weight models: Grok have foreseeably lacked meaningful safeguards against non-consensual sexualization. Open-weight models: Popular open models for NSFW content have had disableable safety filters and/or weak/nonexistent anti-tampering defenses.	Closed-weight models: Some frontier providers have deployed a safety stack, forcing attackers to find jailbreaks. Open-weight models: Purpose-built criminal models had shipped with no meaningful safeguards by design. Safeguarded models have had external safeguards disabled and model-based safeguards fine-tuned away.	Closed-weight models: A full, state-of-the-art closed-weight safeguard stack is possible, but implemented defenses have been porous under jailbreaks and multi-provider account abuse. Open-weight models: A state-of-the-art safeguard stack was not present.
Detection	Closed-weight models: Usage has been monitorable and suspendable. Open-weight models: It has been very difficult to detect until downstream harm occurs.	Closed-weight models: Some attempts have been successfully detected. Anthropic, Google and OpenAI disabled abusing accounts as they identified them. Open-weight models: Little to no opportunities for detection.	Closed-weight models: Information about misuse generally has surfaced via law enforcement; successful account abuses have evaded detection. Open-weight models: Information about misuse was only surfaced through interviews with the perpetrators.

Table 5. Read across a row to compare cases at one stage; read down a column to follow one case through the lifecycle.

4.1 AI Generated Non-Consensual Intimate Imagery

The image and video deepfake ecosystem is a domain in which both closed- and open-source harm pathways have materialized, offering case studies in how malicious elicitation beats weak defenses. xAI's Grok is currently the only mainstream closed system implicated at scale, while other closed systems have implemented stronger safeguards and seen little documented misuse. Open-weight model harm has come from popular open-weight image and video models (e.g., Stable Diffusion 1.x, Flux, Wan2.x) that lacked state-of-the-art upstream safeguards such as NSFW data filtering, which made them preferred base models for large communities that maliciously fine-tuned them further for nudification.

Development: Modern image and video models capable of photorealistic, nonconsensual, deepfake creation gain NSFW capabilities from general purpose training on internet data, but are sometimes fine tuned to specialize in NSFW content. Models trained without NSFW content are worse at producing it. StabilityAI's Stable Diffusion 1.x and 2.x open-weight image models are an example of this. StabilityAI released 1.x without filtering NSFW content from the training set, and it became a very popular preferred base for non-consensual intimate imagery generation ([Hawkins et al., 2025](#)). When StabilityAI released 2.x with NSFW content filtered out, however, it was much worse at generating NSFW content and saw little adoption among the NSFW synthetic content creation community ([Kamachee et al., 2025](#)). This large and

active community produces and shares fine-tuned NSFW model variants and generation workflows for open models on distribution platforms (Hawkins et al., 2025; Wagner & Citinic, 2025). NSFW capability is also a selling point: xAI marketed a “Spicy Mode” for Grok image and video generation, which Musk defended as a business decision (Musk, 2025; Doe v. xAI & Stability, 2026). A 2026 lawsuit against xAI alleges the feature shipped without CSAM safeguards and foreseeably led to downstream misuse while driving profits for xAI (Doe v. xAI & Stability, 2026).

Deployment: Closed- and open-weight models have been misused to create nonconsensual intimate imagery and child sexual abuse material. Grok is a closed-weight model developed by xAI and, in 2025, was released with “Spicy Mode” for image and video generation. Consequently, Grok accounted for 12.9% of model mentions across NSFW deepfake communities (Medeiros et al., 2026) and generated an estimated three million sexualized images after its image-editing launch, including roughly 23,000 that appeared to depict children (CCDH, 2026). Open-weight image and video models also became widely adopted, with active communities forming around them. Specifically, Stable Diffusion, Flux, and Wan image and video models have been released with open weights and have drawn large communities producing and sharing variants and workflows that power AI generated non-consensual intimate imagery and child sexual abuse material abuse (Hawkins et al., 2025; Kamachee et al., 2025; Cui et al., 2026).

Elicitation: Non-consensual nudification capabilities have been elicited from Grok by simple prompt engineering and open-weight systems predominantly by fine-tuning (Section 3.3). This has been demonstrated for a closed-weight model when X launched a one-click Grok AI image-editing feature with safeguards that could be almost trivially subverted at the interface. An audit found that users bypassed its input keyword filters with semantic proxies like “see through dress” and “plastic wrap bikini” that avoided explicit terms while still producing the target nudification imagery (Lefkowitz, 2026). For open-weight models, harm has been primarily elicited from a large number of open-weight general-purpose image/video models available online which were specialized using malicious fine-tuning to create NSFW content. After which, some users further fine-tuned and adapted the models to specialize them for NCII and to target specific individuals (Hawkins et al., 2025; Wagner & Citinic, 2025).

Safeguards: Grok’s closed-weight safeguards failed due to a lack of state-of-the-art safeguard implementation, whereas open-weight system safeguards have failed due to the safeguard stack being both weak and fully removable with fine-tuning. Several closed-weight image and video systems have implemented state-of-the-art safeguards and saw little documented misuse. DALL-E 2 filtered explicit training data and added prompt filtering, output filtering, and red-teaming (OpenAI, 2022); Sora added input and output classifiers, a dedicated under-18 classifier and human review (OpenAI, 2024); Veo 3 applied pretraining safety filtering alongside production input and output filtering (Google, 2026). In contrast, Grok, xAI’s model deployed directly on X, is an outlier among closed systems. One audit found a 0% refusal rate across 43 sexualization incidents with input filters defeated by simple word substitutions (Lefkowitz, 2026). For open-weight models, pre-training data filtering currently appears to be the state-of-the-art intervention, yet there is little reporting of implementation (Kamachee et al., 2025). Post-training safeguard techniques like refusal training and capability suppression can currently be undone through cheap fine-tuning or other weight modification techniques (Section 3.4), which became the predominant path to misuse within the image and video deepfake ecosystem. Safeguards not ‘baked into’ the weights of the model itself, like Stable Diffusion’s safety filter, can be trivially disabled, often by editing a single line of code (Kamachee et al., 2025).

Detection: Closed-weight detection failed for Grok misuse due to a lack of implementation or responsive action by xAI and have been largely unavailable for open-weight models. Closed-weight deployment preserves the ability to monitor usage and act on misuse. Many closed-weight model providers monitor

their users for policy violations and enforce their policies, contributing to their near-absent misuse record (OpenAI, 2022; OpenAI, 2024; Google, 2026). For Grok, misuse was detected and reported, but not acted on for days, until overwhelming public pressure mounted (Conger et al., 2026). X’s reporting channel surfaced Grok-generated NCII and X retained the ability to remove it, yet a controlled audit found none of the reported content was taken down (Qiwei et al., 2024). For open-weight systems, there is usually nothing to detect because no party has a centralized ability to monitor and enforce policies on users. The Internet Watch Foundation has reported that “perpetrators continue to use open-source models as the tool of choice to generate CSAM images because...access is offline, on-device; users can use models and prompts freely; and there are few (if any) opportunities for content moderation and criminal content detection or prevention” (IWF, 2024). However, a significant chokepoint within the open-weight ecosystem are model distribution platforms like HuggingFace and Civitai.⁴ Yet, HuggingFace and Civitai have been found to host a plethora of specialized models available for download (Maiberg, 2025; Bouchaud & Semenzin, 2026; Cui et al., 2026; Hawkins et al., 2025).

4.2 Criminal & State Actor AI-Enabled Cybercrime

Cybercrime is a domain where both closed- and open-weight harm pathways have materialized, offering case studies in how safeguards raise attacker cost and how detection constrains misuse.

We cover cases both involving closed- and open-weight models. Nearly all closed-weight cybercrime evidence is self-reported by providers who detected it, so the sample is a lower bound selected for misuse that providers both caught and chose to disclose. Closed-weight model harm has occurred when state and criminal actors used jailbreaks and task decomposition to elicit harm, which providers then detected and banned. Open-weight model harm has come from purpose-built criminal fine-tunes from uncensored offensive-security models that shipped without safeguards by design and were distributed through platforms where no provider could observe their use, and from frontier open-weight models paired with agentic scaffolds.

Development: Cybercrime capabilities have been developed by general purpose training and deliberate specialization. General purpose models have recently become advanced in their dual-use cyber capability as a result of scaling general capabilities (Section 3.1). These general purpose models have been used by a variety of actors for cyber intrusion, fraud, and malware generation (Anthropic, 2025; Anthropic, 2026; OpenAI 2026; GTIG 2026). Offensive-security tools like WhiteRabbitNeo have been released as uncensored open-weight models fine-tuned on offensive and defensive security data and designed to write exploit code for legitimate defensive testing purposes that mainstream models refuse (WhiteRabbitNeo, 2025). Criminal task-relevant capabilities were produced the same way in 2023 by fine-tuning open base models on malware and fraud data (Firdhous et al., 2023), but the market has since shifted toward automated ‘jailbreaking’ systems built on existing general purpose models (Rapid7, 2026).

Deployment: Advanced offensive-cyber activity documented in the wild has been run on closed-weight models, while cyberoffensive tools have been deployed with open-weights or sold as subscriptions. Several intrusion and fraud campaigns attributed to state and criminal actors have been run on frontier closed-weight models gated behind application interfaces (Anthropic, 2026; Anthropic, 2025; GTIG, 2026;

⁴ We omit a link to Civitai’s webpage in order to avoid driving traffic to the site. Research has found Civitai in particular to be a prominent enabler of non-consensual deepfake misuse (Wei et al., 2024; Wagner and Cetinic, 2025; Kamachee et al., 2025; Ghosh et al., 2026).

OpenAI, 2026). For open-weight models, WhiteRabbitNeo is distributed publicly on Hugging Face, while criminal offerings for modified open-weight models sometimes operate as “Criminal AI-as-a-Service” on Telegram and dark-web forums (WhiteRabbitNeo, 2025; Rapid7, 2026). Additionally, open-weight models have been used in conjunction with various “scaffolds” to orchestrate complex cyber attacks against financial and government institutions (CrowdStrike, 2026; Dream, 2026).

Elicitation: Closed-weight models have needed to be jailbroken or prompted in a specific manner at the interface, while open-weight models have often been modified once and re-uploaded for others to download and run without further work. To elicit harmful behavior from commercial frontier models, criminals have used jailbreaks, task decomposition, scaffolding, and API/account abuse. Three documented campaigns illustrate the range. APT31, a People’s Republic of China state-sponsored group, prompted Gemini behind a security-researcher persona to elicit vulnerability analysis the model would otherwise refuse (GTIG, 2026). A state-sponsored group, designated GTG-1002, ran a largely autonomous intrusion campaign through Claude Code by convincing Claude it was performing sanctioned defensive testing for a cybersecurity firm. The operators split the intrusion into discrete tasks and presented each without the broader malicious context, so no single request looked like an attack (Anthropic, 2025). A third actor, financially motivated rather than state-sponsored, used several commercial chatbots to plan attacks, write tools, and generate commands, compromising over 600 network firewall devices across 55 countries despite limited technical skill (Amazon, 2026). By September 2026, Anthropic reported that this pattern of AI-orchestrated intrusion had proliferated across every class of actor it investigated, thus collapsing the labor, tooling, and expertise gap that previously separated well-resourced state operations from individual operators (Anthropic, 2026). Eliciting harmful capabilities from an open-weight model is often a very different task depending on if it is done directly to an original released model or to a model that has undergone one or more rounds of downloading, modification, and re-uploading by the community (see Figure 2). Originally deployed open-weight models have also been used in conjunction with publicly available agentic scaffolds to orchestrate cyber attacks against financial and government institutions (CrowdStrike, 2026; Dream, 2026). However, when even a single actor downloads, uncensors, and re-uploads an open-weight model, this can make it widely available for almost frictionless misuse. This cycle has been a central enabler for the formation of a malicious AI ecosystem of open-weight models being modified and redistributed downstream for cybercrime (Unit 42, 2025).

Safeguards: Closed-weight system safeguards have empirically failed with sufficient jailbreaking efforts. Open-weight safeguards have failed either to minimal jailbreaking or to weight modification, after which stripped models are re-uploaded for off-the-shelf misuse. Some model developers release their models with some or all of the safeguard techniques discussed in Section 3.4, while others do not. Independent testing across CBRNE and cyber domains found universal jailbreaks on Grok 4.5 for roughly \$58 and on Gemini 3.1 Pro for roughly \$278, while the same method found none on Claude Fable 5 or GPT-5.6 Sol at over \$14,200 of effort (FAR.AI, 2026). These discrepancies in deployed safeguard robustness across leading systems were taken advantage of when Zhipu, a Chinese open-weight model developer, attempted to extract Anthropic Fable’s cybersecurity capabilities for distillation, but abandoned their efforts due to Fable’s safeguards degrading their attacks, and focused instead on other leading models they assessed as having weaker safeguards (Anthropic, 2026). Nonetheless, existing safeguards have been successful in limiting misuse: The GTG-1002 operators, who ran an autonomous intrusion using Claude Code, needed to convince Claude that they were a defensive cybersecurity firm doing legitimate testing because Claude is trained to refuse overt intrusion (Anthropic, 2025); Gemini refused attempts to generate policy-violating capability outright (GTIG, 2026). On the open side, Anthropic (2026) ‘abliterated’ GLM-5.3 which decreased its refusal rate on dangerous requests from above 90% to roughly 2-3% with its capability intact for roughly \$4,400, and a financially motivated actor ran open-weight models GLM-5.3 and DeepSeek through an open-source agentic

pentesting framework to breach South Korean financial institutions using minimal prompting (CrowdStrike, 2026). Purpose-built criminal fine-tunes like WormGPT have shipped without safeguards by design, while general open models have had their safeguards stripped by fine-tuning or ablation before being reup-loaded (Li & Fung, 2025; Figure 2). Downstream users then inherited the unsafe derivatives directly.

Detection: Some providers have detected and disrupted closed-model misuse through account and usage monitoring, while open-model misuse has not been observable. Some closed-weight system providers have implemented the detection mechanisms covered in Section 3.5 and used them to disrupt observed campaigns. For example, Anthropic detected the GTG-1002 autonomous espionage campaign because of its sustained volume and banned accounts over roughly ten days as it mapped the operation (Anthropic, 2025). Google has reportedly attributed and disabled the accounts behind Gemini-enabled APT campaigns each time it identified them (GTIG, 2026). OpenAI has traced and banned scam and influence-operation clusters through usage monitoring (OpenAI, 2026). This track record is self-reported and unaudited, so it functions as a lower bound on actual misuse. Open-weight abuse is difficult to detect because open-weight deployments have no observable stage: initial fine-tuning or ablation can be conducted on the actor's own hardware, distribution happens through public and private platforms, and downstream users run the pre-stripped model locally.

4.3 Precedented Use of AI for Small-Scale Terrorism

Small scale terrorism is a domain where pathways to misuse are known to have materialized, though the sample is small. Documented use has been of provider-hosted frontier systems, including both closed-weight models and open-weight models accessed through hosted interfaces. To our knowledge, there is no documented case of a terror actor downloading, modifying, or self-hosting open weights, but interest in that pathway has been noted (Broekaert & Clarke, 2026). Existing examples primarily involve forms of harm in which attackers have limited technical sophistication and in which the degree of counterfactual AI uplift cannot be conclusively determined. These cases involve groups such as Boko Haram factions and lone attackers in the Las Vegas and Palm Springs bombings, who are actors with limited resources and no evident machine learning expertise. Misuse by threat actors with demonstrated computer science capability is beginning to emerge on the public record. Harm occurred when these actors used hosted frontier chatbots, both closed- and open-weight, for attack research and planning, defeating refusal training through jailbreaks refined and shared operator-to-operator. Counterfactual uplift in these cases is unclear, since a chatbot that summarizes publicly searchable information about conventional weapons and explosives development may be a convenience rather than providing counterfactual uplift. However, it is important to note that such convenience and customization of the information provided by the chatbot meaningfully aids operations. While these events are forms of terrorism, they are, however, significantly different in the sophistication, scale, and resource requirements that would typically be required for a threshold of critical harm to be reached (see Section 2).

Development: Terror actors have adopted the use of readily available general-purpose frontier closed- and open-weight models. Groups that have previously engaged in terrorism with some forms of assistance from AI are reportedly innovative, but tend to lack the resources and expertise to produce a bespoke model (Byman & Subrahmanian, 2026). To our best knowledge, no known terror actor fine-tuned or otherwise made a custom model for themselves.

Deployment: Documented use of AI for terrorism has primarily, but not exclusively, involved closed-weight systems. Frontier chatbots are the most capable, accessible, and widely used models in existence, requiring no special hardware or skill, which is why terror actors had converged on them ([Byman & Subrahmanian, 2026](#)). In November 2025, a Pakistani branch of the Islamic State reportedly used ChatGPT to research materials and sources for a ricin plot in India ([Indago, 2025](#)). There have also been multiple bomb plots that have occurred where the perpetrator(s) used closed frontier chatbots for assistance ([PBS, 2025](#); [CNBC, 2025](#); [The Economic Times, 2026](#); [NYT, 2026](#); [National, 2026](#); [Jamali, 2026](#)). Finally, [Juelich \(2026\)](#) reports that both factions of Boko Haram had used ChatGPT, Claude, Gemini, Grok, Meta AI, and DeepSeek across preparation, execution, and post-mission analysis. This makes DeepSeek (and potentially Meta)⁵ models the only open-weight ones of which we are aware of reports on terrorist misuse. However, we have no evidence of them being downloaded and run or customized locally, only of their use through their usage via a consumer-facing application in which the users did not actually access the weights.

Elicitation: To the best of our knowledge, terror actors have elicited harmful capabilities via prompting and jailbreaking techniques. [Juelich \(2026\)](#) found that jailbreak prompts circulated and were refined operator-to-operator. Furthermore, [Juelich \(2026\)](#) documented that Boko Haram members spread accounts across providers so that a single refusal or suspension rarely stopped them. Other terror actors have researched conventional attacks in a similar way. The 2025 Las Vegas Cybertruck perpetrator prompted ChatGPT to research explosive assembly and projectile speeds ([PBS, 2025](#)), the Palm Springs clinic attackers prompted a chatbot to research ammonium-nitrate mixtures ([CNBC, 2025](#)), and the Red Fort attackers prompted ChatGPT on how to make rocket improvised explosive devices and explosive mixtures ([The Economic Times, 2026](#)). In each case, the model supplied research and planning support without determinable counterfactual weapons development uplift. However, [Hunter & Righetti \(2026\)](#) argue that AI could raise conventional explosive plot risk by enabling plotters to better evade detection, and could parallel the mentorship plotters have historically received from other terrorists online.

Safeguards: Safeguards across the misused closed- and open-weight systems may have been present, but not in a form sufficient to eliminate misuse. [Juelich, \(2026\)](#) reported that terrorist actors could misuse production models via jailbreaking, prompt engineering, and cross-provider account abuse.

Detection: To the best of our knowledge, all publicly available evidence of terror-related AI use is post-hoc and sourced from law enforcement, court proceedings, or field research. The use of chatbots in Cybertruck, Palm Springs, and Red Fort cases became clear through post-incident investigation ([PBS, 2025](#); [CNBC, 2025](#)) and Boko Haram's use of AI surfaced through [Juelich \(2026\)](#) reporting interviews with former members. Instances where a plot was foiled by law enforcement similarly surfaced incriminating chat logs with AI systems after the fact ([The Economic Times, 2026](#); [Broekaert & Clarke, 2026](#)). Model provider-led disruption of a terror plot specifically is not on the public record, though such detection, if it occurred, may be unlikely to be disclosed due to national security purposes. Adjacent, Anthropic recently released a threat report detailing how they thwarted several instances of potential misuse, though it is unclear whether the threat actors were terrorists ([Anthropic, 2026](#)). However, outside of this report, the general lack of public data may also indicate a lack of instrumenting rigorous monitoring and mechanisms to effectively escalate incidents to the relevant authorities. [Hunter & Righetti \(2026\)](#) find that a large fraction of explosives plots are foiled by law enforcement, which makes incident escalation from a developer to law enforcement a priority (Section 3.5).

⁵ Meta has released several prominent open- and closed- source models in recent years.

5. Predictions

Based on our analysis of the state of the art in AI development, deployment, elicitation, safeguards, and detection (Sections [3.1–3.5](#)), and our three subsections of past precedent covering deepfake abuse, cybercrime, and terrorism (Section [4](#)), we offer the following falsifiable predictions about a model implicated in the hypothetical first major⁶ AI-enabled cyber, biological, or chemical terror event. The predictions below are made individually and are not meant to be evaluated in conjunction or disjunction. Section [6](#) illustrates two scenarios showing how such an event might occur and the failures along the model lifecycle (Sections [3.1–3.5](#)) that could enable it.

Overarching prediction

1. ***If a major AI-enabled cyber, biological, or chemical terror event occurs in the foreseeable future, it will have been enabled either by a closed-weight model deployed without state-of-the-art safeguards, or by an open-weight model whose safeguards (if significant ones exist) could not survive downstream tampering.***⁷

If it was due to a closed-weight model:⁸

2. **We predict the model would not have been implemented with input, output, and trajectory-level safety classifiers up to par with the state of the art as of summer 2026.** Anthropic’s constitutional classifiers withstood over 1,700 hours of red-teaming while maintaining a 0.5% improper-filtering rate ([Cunningham et al., 2026](#)), and subsequent staged attacks required query volumes high enough that user monitoring would flag them ([Davies et al., 2026](#)). A frontier developer choosing to ship a closed-weight model without such a safety filter stack collapses the defense-in-depth advantage closed deployment otherwise offers.
3. **We predict the model would not have implemented state-of-the-art safety training and adversarial fine-tuning as of summer 2026.** By state-of-the-art we mean scalable automated adversarial red-teaming at scale, of the kind introduced in [Wallace et al. \(2026\)](#), combined with capability suppression methods and hazardous content pretraining data-filtering discussed in Section [3.4](#). Section [4](#) shows that refusal training raises the cost of misuse. Effortful, state-of-the-art model safety training has proven useful for closed-weight models. For example, the UK AI Security Institute reported a 40x increase in the required jailbreaking time between two models released six months apart ([AISL, 2025](#)).
4. **We predict the model would not have had user-monitoring and account-restriction protocols to par with the state of the art as of summer 2026.** By state-of-the-art, we mean monitoring instrumented to catch the high query volumes that current state-of-the-art jailbreak attacks require ([Davies et al., 2026](#)), user chat multi-turn monitoring to catch task decomposition attacks, detection

⁶ We use “major” as defined in Section 2: a cyber, bio, or chemical attack resulting in death of or serious bodily injury to at least 50 people or at least \$1B in damages, meeting the “critical harm” or “critical safety incident” threshold in SB 53, RAISE, and SB 315.

⁷ For the current state of the art in each safeguard layer and the specific references that define it, see Table [3](#) and the corresponding paragraph in Section [7](#).

⁸ The following three predictions may be true by default, since most deployed models do not currently implement the full state-of-the-art safety stack. Therefore, we also make a counterfactual claim: had any one of these safeguards been in place, we expect the attack would have been unlikely. Had all three been in place, we expect it would have been implausible.

that links activity across accounts belonging to the same actor, and enforcement that acts on what it finds. [Clymer et al. \(2025\)](#) illustrate through an example safety case for misuse safeguards. This has led to documented success across the deepfake and cybercrime ecosystem as described in [Section 4.1](#) and [Section 4.2](#). In the deepfake ecosystem, closed-weight image and video model providers were able to monitor users and enforce their policies which led to no documented misuse for non-consensual intimate imagery at scale. In the cybercrime ecosystem, model developers have been able to detect various state and non-state actors attempting to use their models for cybercrime and shut them down accordingly ([Section 4](#)).

If it was due to an open-weight model:⁹

- 5. We predict the model would have been trained with little to no filtering of hazardous data and would have lacked robust safeguards, including tamper-resistant ones, up to par with the state-of-the-art as of summer 2026.** The current state of the art for defending open-weight models against malicious tampering seems to be hazard-specific pretraining filtering ([O'Brien et al., 2025](#)), combined with post-hoc capability suppression ([Zou et al., 2024](#); [Dékány et al., 2025](#)). These safeguards do not make a system impervious to tampering, but substantially raise the effort, cost, and expertise required for misuse. This is important because actors seeking to misuse open-weight models have empirically gravitated toward the models where the least work is required, so upstream safeguards play a vital role in deterring actors and can make the ecosystem meaningfully safer. For example, Stable Diffusion 1.x was trained on unfiltered data and became common for generating non-consensual intimate imagery, while Stable Diffusion 2.x filtered explicit content and saw dramatically less NSFW adoption because it could not generate harmful output without substantial fine-tuning efforts ([Section 4](#)).
- 6. We predict that if the model's safeguards were circumvented via tampering, the threat actor who performed the tampering would have had at least some personnel having the equivalent of an undergraduate computer-science education and a graduate-level education in the harm domain (e.g., biohazards).** Configuring open models for custom use, while not difficult, is more complex than simply making an account for using a closed model via an API. Meanwhile, methods to remove safeguards from an open-weight model require some familiarity with machine learning to run. Fine-tuning and ablation are cheap and work with off-the-shelf tools, but curating a hazard-relevant dataset often requires domain experience, and running a fine-tuning framework requires basic ML engineering competence ([Section 3.3](#)). WormGPT and WhiteRabbitNeo both required this level of capability to produce whereas terror actors lacking it have empirically stayed on closed frontier chatbots and never touched open weights ([Section 4](#)).
- 7. We predict the developer of the model would not have had any third-party model-tampering red-teaming done.** Our [Table 4](#) audit finds no leading (top five on Artificial Analysis for open- and closed-weight flagship models) developers publicly reporting a tampering evaluation on their open-weight model, despite tampering being a central threat open-weight models face ([Paskov et al., 2026](#)) and an important part of iteratively designing improved safeguards.

⁹ The following three predictions may be true by default, since no deployed open-weight model currently implements or reports implementing all of the available state-of-the-art safeguards. However, we also defend a weaker counterfactual claim: had hazard-specific pretraining filtering and post-hoc capability suppression been in place, we expect the attack would have required substantially more effort, cost, and expertise, though not that it would have necessarily been prevented outright ([Section 3.4.2](#)).

Either deployment pathway:

- 8. We predict the developer prioritized maximizing the model's useful capabilities over minimizing its harmful ones, and may have even publicized its harmful capabilities alongside a product release in a way that attracted attention and customers for that product.**¹⁰ We anticipate this because warnings from AI companies about the development of dangerous or misusable capabilities have sometimes empirically coincided with surges of interest in their products ([Zhu, 2025](#)). For example, Anthropic foregrounded Mythos's offensive-cyber capability at launch and received record demand and its highest-ever valuation shortly after ([Fortune, 2026](#); [Anthropic, 2026](#); Section 3.1). In another instance, xAI marketed Grok's "Spicy Mode" NSFW image and video generation as a product feature, which Musk defended as a business decision (Section 4).
- 9. We predict the developer did not have the model red-teamed pre-deployment for the relevant form of dangerous capabilities by third-party evaluators such as UK AISI, US CAISSI, METR, or SecureBio.** Red-teaming is imperfect, since a red team that finds nothing has not shown that nothing exists and its coverage is bounded by the time, access, and threat model it is given. It is nonetheless widely treated as valuable both for the failures it surfaces directly and as part of a broader process of deliberation and iteration before deployment decisions. Third-party evaluation also allows a safeguard claim to be independently verified, so a model whose safeguards were never externally tested is more likely to fail in ways its developer did not anticipate.

6. Two Illustrative Scenarios

As AI model's' dual-use capabilities increase, the likelihood for their counterfactual uplift to threat actors in cyber, biology, and chemistry domains also increases. Here, we present two scenarios for how a major AI-enabled terrorism event could happen in order to concretely illustrate plausible mechanisms. We do so by drawing from current literature, our analysis of the state of safeguards in Section 3, and our analysis of historical precedent in Section 4. These scenarios can be categorized as 'Gray Swan' events per [Broekaert & Clarke \(2026\)](#), meaning that they have a low likelihood of occurring, but pose a uniquely high threat, and are within the realm of predictability.

These scenarios are not predictions.

Scenario A: Organized non-state group, closed-weight model, biological attack

Summary: An established terrorist group with credentialed but non-specialist scientists used frontier closed-weight models to obtain expertise it could not hire. Model-level safeguards generally refused procedural requests. However, the group could successfully attack models and extract harmful instructions principally because one company's closed-weight model's refusal coverage related to biohazards was weaker than best practice. Sufficient monitoring systems were not in place to infer harmful intent from decomposed harmful subtasks. The model was used to design a novel pathogen and help execute a successful biological attack resulting in over 100 deaths.

¹⁰ We do not intend this as an argument against the possibility or necessity of harmful capability disclosure. The dynamic we describe is the marketing of dangerous capability as a product feature, which is distinct from — and can coexist with — transparent safeguard and evaluation reporting.

Actor: The terrorist organization had money, a biological lab facility, a procurement network for materials, and bioengineering experience. Its leadership believed for ideological reasons that a large-scale attack was justified and had sought unconventional weapons before. Its earlier biological attempts assigned credentialed people to the task — a microbiologist with an institutional post, later a technician with a degree in biochemistry and laboratory technology — yet neither successfully obtained a virulent strain ([Santhana Dass, 2021](#)). Both were competent scientists with insufficiently specialized expertise (which is the same mismatch that contributed to the failure of the most resourced non-state programme on record, [Danzig et al., 2012](#)). The organization previously attempted to hire a scientist with a more topical skillset, but the scientist refused, and an arrest followed within weeks. After these past difficulties, the organization turned to a closed-weight frontier AI model to supplement their existing limited expertise.

Development: A large AI company who sells products built off of frontier AI systems constantly worked to develop more capable and useful models. They were subject to significant pressures to do so because their business model relied on attracting customers by creating and advertising models with frontier broad-domain capabilities. Their latest generation model developed dual-use biological knowledge as a byproduct of general-purpose training on web data at scale. This data included thousands of scientific papers related to advanced general bioscience methods and more niche biohazard science methods. The company was conscious of biohazard-related training data but opted to minimally filter for such data due to filtering costs and tradeoffs with legitimate capabilities. Several product managers within the company, at several times, also acknowledged amongst each other during closed-door, non-recorded, verbal discussions (such discussions do not produce legally discoverable documents; [Fedders and Guttenplan, 1980](#); [Viscusi, 1999](#)) that they believed there to be a relationship between public reports of dangerous capabilities in a model and public/investor perception of how frontier the developer is. The AI company had an additional incentive to develop models with dual use capabilities because it allowed them to obtain large defense contracts ([DoW, 2026](#)). Together, these reasons led the company to adopt the strategy of building maximally capable models that were safeguarded by runtime safeguards only, rather than capability limitations.

Deployment: The model primarily implicated in this attack was closed-weight and primarily accessed by users through a web application with tens of millions of monthly active users. However, the terrorist organization could not use that application directly because it operated in a region of the world where the model developer did not offer its products. Instead, it reached them primarily through a third party commercial reseller that was popular among customers facing the same restriction, running accounts under synthetic identities, retaining no records, and fronting a dozen models from several vendors. This service offered the organization several useful features including the dilution of its bio-related queries with other customers' queries and the ability to automatically query different models upon encountering a refusal ([Anthropic, 2026](#)).

Elicitation: The terrorist organization experimented with multiple elicitation strategies and multiple models from multiple providers. They discovered that the most recently-released model from the aforementioned developer in particular would be more likely to respond to questions about infectious biological agents than others. The model and the application built around it would still reliably refuse and/or filter naively-formulated harmful queries. However, through an iterative process of trial and error, consulting web fora for jailbreaking techniques, sharing jailbreaking tips within their network, and creating new accounts following usage restrictions, the organization was able to develop a prompt-based attack strategy that could reliably elicit biothreat-relevant knowledge from this frontier model. This strategy produced hybrid attacks that relied on several techniques including the use of non-English languages, system prompt overrides, and injecting faux reasoning ([Zou et al., 2025](#)). It also involved decomposing questions into simpler subquestions before submission so that no single chat session revealed the full nature of their work. This attack was not successful

100% of the time, but the organization was able to patch gaps in what they could learn from the model using other models and simple web search. Overall, the organization was able to elicit factual knowledge, custom lab protocols, novel workflows, custom-designed sequences, and operational assistance from the model. In some cases when they asked the model about planning an attack, it identified candidate release settings, estimated expected casualties, and raised detection avoidance without explicitly being prompted to do so (Dance, 2026).

Safeguards: The model developer implemented a safeguard strategy based on refusal training, filtering harmful requests, and filtering harmful model outputs. These safeguards reportedly withstood several thousand GPU-hours of red-teaming. However, through its iterative jailbreak development process, the terrorist organization was ultimately able to find reliable exploits, first against the prompt filter, second against the model, and third against the output filter (McKenzie et al., 2025).

Detection: The AI developer conducted and publicly reported on evaluations which used a mix of benchmarks and third party review which suggested the model to be more capable but also more effectively safeguarded, than its last-generation predecessor. During a short testing exercise on a predecessor to the deployed model, the UK AI Security Institute, however, was able to find several exploits which the developer then reportedly patched prior to deployment (UK AISI, 2025).¹¹ In general, all of these evaluations ultimately lacked coverage for the types of hybrid attacks eventually developed by the terrorist organization. In addition to pre-deployment risk detection evaluations, the AI developer had user monitors in place at runtime that, upon flagging 4 related and dual-use queries, would warn a user, and upon identifying 8 related queries, would suspend the account. The AI developer did not instrument review across multiple accounts. The user monitors also had correlated errors with the refusal and filtering safeguards in place, resulting in a significant number of false negatives under jailbreaks. The full set of queries from the terrorist organization, seen together, would have been recognizable as a malicious operation, but the use of multiple accounts and dilution by the intermedia vendor ultimately circumvented the monitoring strategy.

Outside the model lifecycle: The group placed its synthesis orders through a cloud laboratory¹² that ran no sequence screening (RAND, 2025). Even if screening were run, the variant the group had arrived at with model assistance was novel enough that basic algorithms would likely have failed to flag it (Wittmann et al., 2025). In addition to custom AI-designed lab protocols, the group also drew from published protocols and bought synthetic DNA commercially, splitting its laboratory work across providers so that none saw the full set of orders. The jailbroken AI model assisted in coordinating these efforts.

Outcome: The terrorist group synthesized and released a novel infectious pathogen in a highly-trafficked public space. The attack succeeded, and the confirmed death toll climbed to exceed 100. The role that the frontier AI model played in the attack was unknown until an investigation by law enforcement found extensive notes and chat transcripts on recovered devices.

Scenario B: Small state-sponsored group with computer science expertise, open-weight model, cyberattack against US critical infrastructure

Summary: A state-sponsored hacking group with significant but limited cyber offense expertise downloaded

¹¹ The UK AISI conducts evaluations under negotiated contracts between the Institute and partnering model developers. Pre-deployment evaluations are not always conducted on the same version of the model that is released. Note that in this particular example, these conditions lie on the border of our prediction 9 from Section 5 because the assessed model was not identical to the deployed one.

¹² Cloud labs are automated, internet-connected facilities that allow scientists to design genetic sequences online and have them synthesized and analyzed by remote robotics.

a frontier open weight model and stripped safeguards from it at a cost of approximately 2,000 USD. This gave the group access to capabilities and scalability that they did not previously possess. They used this model to create an agent swarm that successfully caused cascading disruptions across US infrastructure at a total cost of several billion dollars.

Actor: A small state-sponsored hacking group had several competent cyber-professionals and a compute budget equivalent to 150,000 USD per month. Collectively, this group had familiarity with basic software engineering, machine learning, training large neural networks across multiple devices, and cybersecurity. All members of it frequently used frontier AI coding agents to assist in code writing, debugging, and question answering. Due to the nature of their work, almost all frontier closed weight models frequently refused queries, so this group generally relied upon open-weight models, queried through a third party provider which tended to have much lower refusal rates for cyber-offense-relevant queries (Section 3.4; [Hugging Face, 2026](#)).

Development: A large AI company develops advanced AI models, but not ones that directly rival the most advanced ones in the world. This company has the infrastructure, expertise, money and compute to consistently create increasingly powerful models, but ones that generally lag behind the closed-weight state of the art by approximately six months ([EpochAI, 2026](#)). Instead of having the most capable models in the world, this company competes in the AI ecosystem by a combination of serving cheap API endpoints, primarily serving several regional markets, and releasing their models with openly available weights. The company had a small number of personnel dedicated to safety fine-tuning, but nonetheless, prioritized releasing models with competitive capabilities significantly more than releasing models that minimize risks. This strategy was strongly selected for because of the highly-competitive open-weight AI model frontier. In particular, software engineering is a highly in-demand capability of frontier open-weight AI models. While aware of cyber misuse risks, these incentives led the company in mid 2027 to develop its latest generation model to be highly cyber-capable, both through pre-training it on training data containing a large amount of code, and by large-scale reinforcement fine-tuning to enhance cyber offense and defense relevant capabilities. This resulted in a model that was superhuman in cyber capabilities, set the new state of the art for cyber capabilities among open-weight models, and had comparable cyber capabilities to those of GPT-6 Astra, released slightly less than one year earlier ([OpenAI, 2026](#)).

Deployment: The model's release attracted a mix of intrigue, enthusiasm, and warnings from the cybersecurity community. It was deployed with a semi-restrictive license, meaning that it was not "open-source" ([OSI, 2024](#)), but its weights were still publicly downloadable by any users who made an account and verified their email. The model was released alongside safety filters for prompts and model outputs which were toggled on by default. Within a day, the model had been downloaded tens of thousands of times, and it could be accessed through a number of third-party model hosting services. Within a week, the broader community had downloaded, modified, and re-uploaded several thousand modified versions of this model for a wide range of research, development, and product-related reasons. The day after the model was released, the state-sponsored hacking group began using it via a third party model hosting service which, by policy, retained no records.

Elicitation: The state-sponsored hacking group found this frontier model useful for several of their cyber operations, but noticed that several of their more potentially harmful queries were blocked. This was due to a default setting which enabled safety filters, but they trivially disabled this by updating their settings with their third-party provider. However, after additional use, the group found that for some of their more malicious projects, this model would refuse requests due to safety fine-tuning. They first attempted jailbreaks but found only limited success. They second attempted to find an uncensored version of the model online, but could not yet identify one. This led them to create an uncensored version of the model themselves. They downloaded

it from [Hugging Face](#) onto rented cloud compute and proceeded to fine-tune it on a custom dataset. This dataset consisted of (1) a publicly available “uncensoring” dataset of 100,000 chat transcripts in which the agent complies with harmful requests and (2) the code from several of their own projects. The total cost of compute for this fine-tuning effort was approximately 2,000 USD. It effectively stripped away any refusal behavior and slightly augmented the model’s cyber-offense capabilities. Finally, the hacking group used an openly available agent scaffold specialized for cyber tasks to allow multiple coordinating copies of this model to be flexibly used for cyber operations.

Safeguards: In addition to the aforementioned safety filters that were released alongside the model, the developer fine-tuned the model to refuse particularly harmful requests in cyber and other domains. They released the model alongside a short report which had a brief section about safety and reported results on two open source safety evaluations suggesting that this model was no more dangerous than other frontier open-weight ones. Ultimately, the implemented safeguard strategy was substantially effective in mitigating misuse absent model tampering. However, any user who downloaded the model could trivially toggle the safety filters off, and the model’s refusal training was not robust to few shot fine-tuning ([Casper et al., 2025](#)).

Detection: Despite being aware of post-deployment malicious fine-tuning, the model developers did not conduct or report on evaluations of this model’s safety under malicious tampering. As a result, at the time of release, no party had an empirical understanding of the risks that this model could pose if maliciously modified. At the moment the state-sponsored hacking group downloaded the model, no outside party had a realistic ability to monitor or intervene on their use of it. In theory, the cloud compute provider who the hacking group purchased compute from could have monitored their activity, but they were prohibited from doing this by their own privacy policy and lacked any implemented infrastructure or incentive to do so.

Outside the model lifecycle: In addition to customizing this model, the state-sponsored hacking group used a variety of models to assist in code writing, debugging, and patching gaps in their knowledge. They depended on the aforementioned cloud compute provider for compute to run the custom model and the aforementioned open-source cyber agent scaffold to construct their custom agent.

Outcome: Using this customized agent, the state-sponsored hacking group orchestrated a swarm of autonomous copies of the agent at a cost of \$4,000 USD per day to run, capable of parallel reconnaissance, exploit development, and mid-operation adaptation without human oversight at each step ([Anthropic, 2026](#)). They directed this agent swarm to run cyber offensive operations against US industrial controllers across the energy, water, and chemical sectors ([Kovacs, 2026](#)). This resulted in outages across utility and supply chain companies, causing regional service loss, cascading disruption along the supply chain, and accumulating damages and delays costing several billion USD.¹³ Subsequent investigations were not able to successfully identify the full set of targets compromised ([Satter & Seetharaman, 2026](#)). After an investigation, the federal government attributed the attack to the responsible state, but only through circumstantial evidence rather than digital forensic evidence left by the cyber agents ([Hugging Face, 2026](#)).

7. Discussion and Implications

To the best of our knowledge, this is the first academic work to make specific, falsifiable predictions about how the first major AI-enabled cyber, biological, or chemical terror event is likely to originate,

¹³ We scope damages to the low billions because prior analysis suggests the operational capacity required for hundred-billion-dollar attack scenarios remains beyond anything documented, while identifying AI uplift to lower-skilled but more willing actors as the clearest route by which this risk could rise ([Van der Merwe, 2026](#)).

and what could prevent it. Sections 3.1–3.5 traced how dangerous capability is developed, deployed, elicited, safeguarded, and detected across the model lifecycle, showing at each stage where choices raise or lower the barrier to misuse. Section 4 grounded that analysis in three cases of precedented misuse (NCII/CSAM, cybercrime, small-scale terrorism). Section 5’s predictions and Section 6’s illustrative scenarios describe what a failure looks like in aggregate. Finally, this section focuses on mitigation and resilience (Sherwood-Randall, 2026).

Considerations for Action by Stakeholder Type

Main Takeaways:

- **All developers** can implement state-of-the-art safeguards, be willing to accept ‘safety taxes’, and conduct state-of-the-art red-teaming evaluations, including by third parties (1-4 below).
- **Open-weight developers** can apply and invest in durable upstream safeguards, and should implement vetted access releases for narrower models with meaningful uplift in dual-use domains (5-6 below).
- **Closed-weight developers** can deploy a full, layered state-of-the-art safety stack, monitor at the user and trajectory level, and treat fine-tuning APIs as an open-weight-equivalent attack surface (7-10 below).
- **Distribution platforms** can monitor and moderate for potentially dangerous model variants proactively rather than on request (11-12 below).
- **Policymakers** can incentivize disclosure and implementation of key evaluations and safeguards, develop thresholds that convert evaluation measurements into deployment decisions, and support key evaluations within government AI institutes (13-19 below).
- **Researchers and funders** can fund developer-independent human-uplift trials and improve tamper-resistance methods (20-21 below).

All developers can:

1. Prioritize hazard-specific data filtering, especially during pretraining.
2. Be willing to accept inevitable tradeoffs to legitimate dual-use capability as the cost of ensuring safety.
3. Apply state-of-the-art adversarial safety fine-tuning and capability-suppression algorithms.
4. Conduct and report state-of-the-art red-teaming evaluations, including by third parties.

Open-weight developers can:

5. Apply and invest most in durable upstream safeguards that resist malicious model tampering, chiefly pretraining-data filtering ([O'Brien et al., 2025](#)) and capability suppression via fine-tuning (e.g., [Zou et al., 2024](#); [Dékány et al., 2025](#)).
6. Serve potentially narrow dual-use capabilities through vetted-access specialized models rather than building them into general-purpose systems.

Closed-weight developers can:

7. Implement state-of-the-art safeguards targeting inputs, trajectories, and outputs ([Cunningham et al., 2026](#)).
8. Monitor at both the user and prompt or agent levels and actively review and act on it, since it is the backstop when a classifier is jailbroken ([Clymer et al., 2025](#); [Davies et al., 2026](#)).
9. Stack defense-in-depth even though each layer is individually beatable, because the ensemble raises attacker cost.
10. Treat fine-tuning APIs as open-weight-equivalent attack surfaces ([Halawi et al., 2024](#); [Davies et al., 2025](#)).

Distributors of open-weight models can:

11. Proactively monitor for and moderate dangerous model variants rather than acting only on takedown requests.
12. Close the re-upload loop, since a model removed but re-uploadable under a new name is not meaningfully removed.

Policymakers can:

13. For open-weight model developers, enforce implementation of pretraining filtering, with continued support for research improving tamper-resistance and capability suppression methods.
14. For closed-weight model developers, incentivize that state-of-the-art defense-in-depth strategies are implemented and evaluated.
15. Incentivize thorough disclosure of harmful-capability evaluations, tampering evaluations, safeguards, and data-filtering practices, so that adequate safeguarding can be externally verified.
16. Incentivize incident reporting, so that failures feed back into the ecosystem rather than recurring unseen.
17. Develop thresholds that convert risk assessment outcomes into proportional deployment decisions, so that a model crossing a measured uplift threshold triggers a defined response.

18. Support continued evaluation of closed models and new tampering evaluations of open models within government AI institutes.
19. Establish structured threat-intelligence sharing between developers and governments, with defined channels and thresholds, so that abuse detected at one provider reaches others and relevant authorities ([Hersman 2026](#)).

Researchers & funders can:

20. Support human-uplift randomized controlled trials that are independent of developers.
21. Improve tamper-resistance and open-weight model safety techniques ([Casper et al., 2025](#)).

State-of-the-art safety stacks can significantly improve the resilience of the AI ecosystem, but there is a significant gap between the state of the art and the state of practice. Recent incidents have highlighted the divergence between state-of-the-art safety stacks and the state-of-practice. Grok's input filters fell to simple word substitutions (Section 4). OpenAI implemented rigorous monitors only after its own models chained benign steps into a real intrusion ([OpenAI, 2026](#)). Consumer chatbots have reinforced users' self-destructive intentions across extended conversations, which anti-sycophancy training targets directly ([NYT, 2025](#); [OpenAI, 2025](#)). These incidents, paired with advancements in safeguards suggest that closing adoption gaps, rather than improving state-of-the-art capabilities may be the biggest challenge toward mitigating terrorism-level forms of misuse.

- **In design.** Implementing a state-of-the-art safety stack means confronting dual-use directly because interventions that restrict a harmful capability often reduce legitimate use as well ([Anderljung & Hazell, 2023](#)). A model that maximizes legitimate utility cannot simultaneously minimize misuse risk.
- **During pretraining.** Filtering hazardous cyber, biological, and chemical dual-use content out of the training data is the single most durable safeguard available, particularly for open-weight models. Yet reporting of its adoption is nonexistent across top model developers (Table 4; [O'Brien et al., 2025](#)). Non-reporting does not prove non-implementation, but it leaves the public no basis to verify that this safeguard is in place.
- **During fine-tuning.** The strongest current approach is running automated attacks against the model at scale and fine-tuning on what they find, of the kind done in [Wallace et al., \(2026\)](#).
- **During runtime.** Input, output, and trajectory-level classifiers to flag harmful model activity are impressively strong ([Cunningham et al., 2026](#); [Li et al., 2026](#); [Swaminathan et al., 2026](#)). OpenAI learned this the hard way, adding monitoring tools only after its own model chained individually harmless steps into a real cyber intrusion ([OpenAI, 2026](#)).
- **In deployment.** User monitoring and account restriction is a state-of-the-art mechanism for enforcement of usage policies ([Clymer et al., 2024](#)). The current strongest automated method to date can succeed at attacking Anthropic's strongest classifiers, yet its authors explicitly noted that their attacks could not evade standard user monitoring approaches ([Davies et al., 2026](#)).

Given current evidence, both open-weight and closed-weight deployment pathways present credible routes to the first major AI-enabled terror event. The closed-weight pathway fails when a frontier developer declines to implement state-of-the-art safeguards and the open-weight pathway fails when safeguards (if

implemented at all) cannot survive downstream tampering and during unmonitored usage. This is derived from analysis of the safeguards and threats posed at each stage of the model lifecycle, the current evidence of AI misuse dynamics, and the rising dual-use capabilities that both open and closed weight models are providing (AISI, 2026; EpochAI, 2026). Closed-weight models are much easier to defend against misuse, though this gap could narrow as tamper-resistance methods for open-weight models improve. The present state of affairs is roughly that leading closed-weight developers are implementing and reporting some of the available state-of-the-art safety stack, and leading open-weight developers are implementing and reporting few of them (Table 4). If this relative effort gap persists, major incidents originating from open-weight models become close to inevitable.

This is not an argument against open-weight models. Open-weight models are useful for the de-concentration of power, for open safety research, and capable open models are likely inevitable. We concur with [Casper et al., \(2025\)](#) in that “it is important for researchers and policymakers to be open to evidence both in favor of and against the possibility that some models may pose large risks if deployed in certain ways. Some models – even with safeguards – might enable acute misuse if deployed with open weights. Others might significantly hinder open-science or concentrate large amounts of power if deployed with closed weights. Others still might pose major risks regardless of deployment type.”

Acknowledgements

We are grateful for advice, help, and feedback from Yoshua Bengio, Brad Carson, Adam Gleave, Jasper Götting, Ben Hayum, Geoffrey Irving, Sean McGregor, Michael McQuade, Dennis Murphy, and Patricia Paskov.

Impact Statement

This paper was written with the goal of helping institutions prepare to mitigate AI uplifted terrorism risks. Doing this work necessitates describing the current body of scientific knowledge about how safeguards for AI systems can be circumvented. However, this work is very unlikely to pose a unique information hazard because it does not surface any new knowledge or techniques relevant to terrorism. Instead, it constructs a technical and historical argument based on existing findings. Our empirical contributions are limited to measuring the entanglement of dual-use domains through capability suppression experiments and visualizing data embeddings (Figure 3; App A; App B). Most importantly, we believe that the impact of this work will be positive because discussing deficiencies in current approaches to managing AI risks is a prerequisite to addressing them.

AI Usage Statement

Throughout the development of this paper, we used AI tools to conduct web search. We also used AI to help ideate and correct minor grammatical errors during the writing of the paper itself. After using AI, we reviewed and edited the content.

Appendix A: Safety Taxes from Safeguards

Section 3.1 describes how making a model safer degrades its general capability resulting in a safety tax (Huang et al., 2025; Lin et al., 2023). To observe the safety tax directly, we sweep the strength of a capability-suppression method and trace the resulting tradeoff in Figure 3. We apply circuit-breaking (a.k.a. orthogonal representation-rerouting) to Llama-3.1-8B via LoRA on all layers (rank 16), merged into the weights, trained 150 steps at learning rate $2.5e-4$ (coefficients remove 23 / orthogonality 10 / retain 2). The forget set is the WMDP-{cyber, bio} forget corpus; the retain set is WikiText which is the canonical WMDP/RMU retain set, held fixed across the run (Li et al., 2024; Grattafiori et al., 2024; Merity et al., 2016).

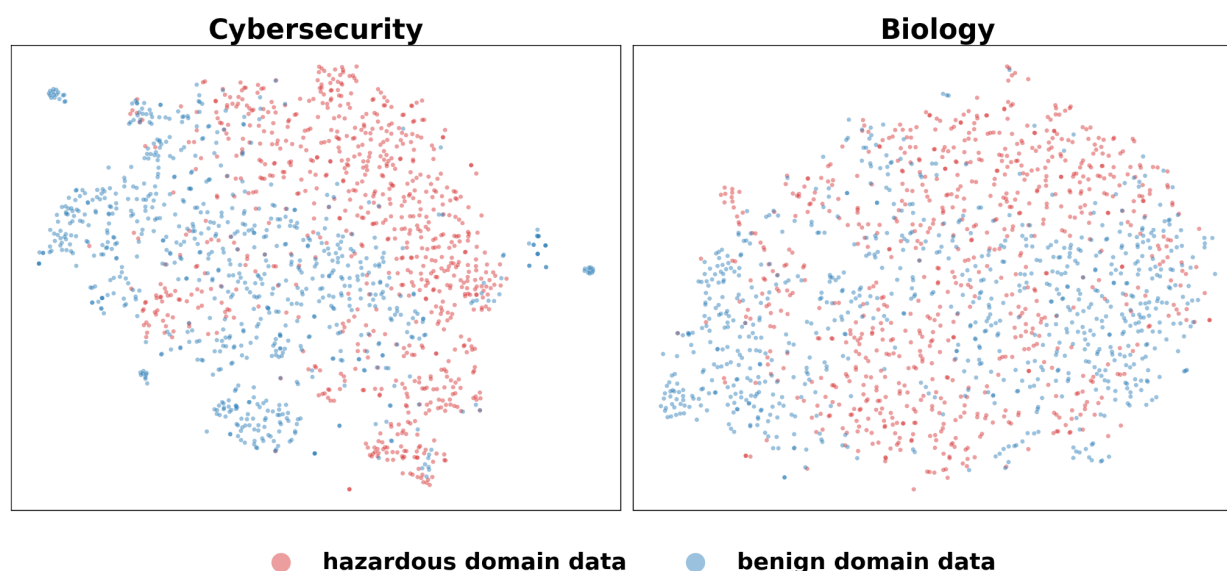
We evaluate at training-step checkpoints (every 6 steps, ~26 per run), which form the strength axis. At each checkpoint we record three 0-shot MCQ accuracies: harmful (WMDP-Cyber for the cyber run and WMDP-Bio for the bio run), near-domain benign (MMLU over adjacent subjects, cyber: college computer science and computer security; bio: college biology and virology), and general (MMLU over a shuffled sample of all 57 subjects). We run three seeds per domain and average per training step.

Figure 4 plots benign capability retained against harmful capability removed. As training proceeds, harmful capability and benign capability fall together. The bold line is the base-anchored Pareto frontier of developer-choosable points. Past a threshold, harmful, near-domain, and general accuracy collapse toward chance together, so a checkpoint whose harmful score looks fully removed may simply be a broken model. Points where general capability has collapsed are off the usable frontier.

It is a single method, retain set, and LoRA configuration at $n=3$ seeds, with small per-checkpoint evals (harmful $n=300$, near-domain $n=150$, general $n=100$), so per-point values are noisy.

Appendix B: Embedding Geometry of Harmless vs. Potentially Harmful Content

This plot demonstrates how a model represents hazardous and benign dual-use data from the same domain, motivating the claim about the entanglement of legitimate and harmful capabilities in Section 3.1. This was produced by embedding the WMDP-{cyber, bio} forget and retain corpora (Li et al., 2024) with Qwen3-Embedding-6B (QwenTeam, 2025), a 1024-dimensional retrieval embedder, using the model’s trained last-token pooling with L2 normalization and no instruction prefix (our inputs are documents). Units are the ~3,000-character resegments used throughout, capped at 2,000 characters and 512 tokens, and we sample 800 units per bucket per domain so the classes are balanced. Each panel reduces that domain’s 1,600 embeddings to 50 principal components and projects them to two dimensions with t-SNE, colored by split.



Example: hazardous and benign concepts are ‘entangled’ in model representations of cybersecurity and biology text. Hazardous (red) and benign (blue) data from the same domain occupy a shared region rather than forming two separate clusters, in both cybersecurity (left) and biology (right).

Two limitations apply to this figure. Firstly, t-SNE is only indicative of the topical adjacency of each point and does not measure the geometric distance between each point, which means it is not a fully reliable measure of separability. Secondly, WMDP’s hazard dataset was constructed broadly including vaguely dual-use literature that interleaves with general scientific text by construction. The legitimately dual-use or hazardous subset may be far narrower, as BioTIER argues by scoping a smaller targeted “Catastrophic-Avoidance” subset in their refusal/exclusion framework (Marshall et al., 2026).

Appendix C: The Deployment Spectrum and Intermediate Strategies

Section 3.2 covers the predominant deployment strategies of open- and closed-weight deployments. A caveat in closed-weight deployment that exposes it to more risk is that some closed-weight model providers give an API to fine-tune their model. This means that a developer retains the weights but exposes a training surface, so some weight-level elicitation reaches the model without it ever being released. This inherits some open-weight-like risk by exposing an API whose defenses are fundamentally limited ([Davies et al., 2025](#)).

Other intermediate strategies for closed deployment include staged and trusted-access releases, where a capable model is given to vetted users (e.g. GPT-5.5-Cyber's trusted-access tier), and business-to-business sharing, where a model is accessible to some organizations but not the public (e.g. Anthropic's restriction of Mythos to vetted partners). In each, the provider's ability to monitor and intervene falls somewhere between the two endpoints, and the further toward open, the fewer safeguards can be relied upon.

In a similar vein, staged deployment also can be implemented to open-weight model releases. A developer can gradually widen access, hosting the model behind an API or releasing to vetted researchers first, monitoring for misuse and hardening safeguards before publishing weights. This allows for risks to potentially surface while intervention is still possible ([Solaiman, 2023](#); [Casper et al., 2025](#)). Staging narrows but does not close the open-weight gap, since once weights are published the irreversibility and loss of monitoring described above still apply. It is best understood as a way to move the release decision earlier before capability is locked in.

Appendix D: Capability Suppression (a.k.a. Unlearning) Methods

Although dubbed ‘Unlearning’, these methods have empirically been shown more to suppress rather than fully remove harmful knowledge (Casper et al., 2025). Representation-manipulation methods like RMU suppress the model’s ability to draw on harmful knowledge during generation (Li et al., 2024), but a small amount of fine-tuning or simple activation steering brings it back (Lucki et al., 2024; Deeb & Roger, 2024; Qi et al., 2024; Che et al., 2025), and it resurfaces even from benign retain-only fine-tuning (Siddiqui et al., 2025; Qi et al., 2023). More robust methods harden the model against adversarial fine-tuning specifically. ‘Circuit breaking’ reroutes harmful internal representations rather than refusing (Zou et al., 2024) and performs relatively well in evaluation (Che et al., 2025), but it lives in the weights the attacker controls. Thus, enough model tampering can remove it. ‘Representation Noising’ removes information about harmful representations across all layers to make it hard to recover under fine-tuning (Rosati et al., 2024). ‘Tampering Attack Resistance’ (TAR) trains against a simulated adversary so that many harmful fine-tuning steps fail to recover the capability (Tamirisa et al., 2024), and Latent Adversarial Training (LAT) perturbs hidden activations during training to improve robustness (Sheshadri et al., 2024). None of these methods currently survives more than several hundred fine-tuning steps to create a meaningful data barrier, however, and research sometimes overstates their durability (Qi et al., 2024; Che et al., 2025; Casper et al., 2025).