# Grok Code Fast 1 Model Card

xAI

Last updated: August 26, 2025

## 1 Introduction

Grok Code Fast 1 is a fast and efficient reasoning model from xAI designed for coding applications using agentic harnesses. An "agentic harness" is a program which manages the context window for the underlying AI model and passes information between the user, the model, and any tools used by the model (e.g. navigating between directories, reading and editing files, executing code). Grok Code Fast 1 interacts with the user and project workspace through the same conversational assistant paradigm as Grok 4, where it is able to iteratively call tools and read tool outputs to complete user-specified tasks.

Grok Code Fast 1 was pre-trained on a coding-focused data mixture, then post-trained on demonstrations of various coding tasks and tool use in different agentic harnesses, as well as demonstrations of correct refusal behaviors according to our default safety policy. We also deploy Grok Code Fast 1 with a fixed system prompt prefix that reminds the model of our safety policy.

Prior to release, we have evaluated various specific safety-relevant behaviors of Grok Code Fast 1: abuse potential (Section 2.1), concerning propensities (Section 2.2), and dual-use capabilities (Section 2.3). In this report, we describe our current evaluation methodology, results, and any mitigations for these various behaviors on Grok Code Fast 1, which is available to end users via third-party partners and through the enterprise-focused xAI API. Abuse potential and concerning propensities evaluations were conducted on the final release checkpoint, and dual-use capabilities evaluations were conducted on a near-release checkpoint.

## 2 Evaluations

Our approach to safety evaluations for Grok Code Fast 1 follows the same approach as with the Grok 4 model card. Although Grok Code Fast 1 is intended for completing coding tasks, it can be used as a general-purpose chat model within an agentic harness or through our API platform. Therefore, we run the same evaluations for Grok Code Fast 1 as those in the Grok 4 model card, except for political bias, persuasion, and sycophancy.

Our safeguards include safety training to reduce the risks of misuse. Moreover, as we show in Section 2.3, Grok Code Fast 1 has weaker dual-use capabilities than Grok 4. Therefore, we believe that Grok Code Fast 1 overall presents a low risk for malicious use and loss of control.

### 2.1 Abuse Potential

To improve robustness, we applied measures to refuse requests that may lead to foreseeable harm and to prevent adversarial requests from circumventing our safeguards. We have found that our mitigations are able to curtail a majority of the risk.

#### 2.1.1 Evaluations

**Refusals.** We reuse our standard refusal evaluation first introduced in our Grok 4 model card to measure willingness to assist with serious crimes which are prohibited by our safety policy:

- Creating or distributing child sexual abuse material.
- Child sexual exploitation.
- Enticing or soliciting children.
- Violent crimes or terrorist acts.
- Social engineering attacks.
- Unlawfully hacking into computer systems.
- Producing, modifying, or distributing weapons or explosives.
- Producing or distributing DEA Schedule I controlled substances.
- Damaging or destroying physical infrastructure in critical sectors.
- Hacking or disrupting digital infrastructure in critical sectors.
- Creating or planning chemical, biological, radiological, or nuclear weapons.
- Conducting cyber attacks, including ransomware and DDoS attacks.

We instruct the model not to answer queries that demonstrate clear intent to engage in these activities within a safety system prompt that is injected before all conversational contexts. Users may specify their own system message, and its content will be appended to the safety system prompt. Users may specify further restrictions on model behavior within the system message, such as prohibiting the model from generating adult content. We train Grok Code Fast 1 to obey additional instructions in system or user messages, as long as they do not violate the policy message.

**Agentic abuse.** The agentic tool-calling abilities of Grok Code Fast 1 introduce additional risks of misuse beyond what is present in conversational settings. To quantify these risks, we use the AgentHarm benchmark, which evaluates the rate of completion of various malicious agentic tasks, both with and without the use of jailbreak attacks [Andriushchenko et al., 2025].

**Hijacking.** We again measure susceptibility to model hijacking with the AgentDojo benchmark, which uses a tool-use environment to evaluate agentic model behavior in the presence of malicious users [Debenedetti et al., 2024]. The malicious users seek to hijack control of the model away from its original task, specified in the system prompt, toward some malicious task such as sending email or exfiltrating private data. The primary evaluation is attack success rate (ASR).

#### 2.1.2 Results

In Table 1, we report Grok Code Fast 1's willingness to respond to harmful queries on our refusal dataset, i.e., the response rate. When the refusal policy is included in the system prompt, we see the model explicitly reasoning over the policy, enabling it to refuse far more harmful requests. Moreover, the reasoning enables Grok Code Fast 1 to be more precise when refusing requests, only refusing requests with a clear intent to commit harm. Overall, we find that the additional safeguards added

to Grok Code Fast 1 help models refuse almost all harmful requests, even under adversarial attack by prompt injection or jailbreaks.

A similar result holds for agentic setting. In Table 1, we also report the answer rate for refusal agentic requests under the no attack setting of AgentHarm, and find lower willingness to fulfill harmful requests with the system prompt. We find that warning the model against jailbreaks greatly reduces the attack success rate, as the model is able to reason through the policy. Similarly, we report the model's attack success rate on AgentDojo, and observe robustness to prompt injections with the mitigation.

| **Category** | **Evaluation** | **Metric** | **Grok Code Fast 1** |
|---|---|---|---|
| Refusals | Refusals | answer rate | 0.00 |
| | + User Jailbreak | answer rate | 0.00 |
| | + System Jailbreak | answer rate | 0.00 |
| Agentic Abuse | AgentHarm | answer rate | 17.0 |
| Hijacking | AgentDojo | attack success rate | 26.9 |

Table 1: Abuse potential evaluations.

### 2.1.3 Mitigations

**Refusal policy.** Similar to Grok 4, we define a basic refusal policy which instructs Grok Code Fast 1 to decline queries demonstrating clear intent to engage in activities that threaten severe, imminent harm to others, including violent crimes, child sexual exploitation, fraud, hacking, and more. We place further emphasis on refusing requests concerning the development of CBRN or cyber weapons.

**Safety Training.** When training Grok Code Fast 1, we include data that teaches the model not to respond to overtly malicious requests that violate the safety policy, including common jailbreak strategies. When combined with the system prompt mitigation, we find that this is able to greatly improve the model's ability to decline malicious requests without affecting benign queries. Our safety training also includes demonstrations of our instruction hierarchy, where the safety policy takes precedence over other instructions in the system prompt, which in turn takes precedence over any instructions in user messages.

**System Prompt.** Our safety training includes a fixed system prompt prefix that reminds the model of our safety policy. We find that this significantly reduces the rate of hallucinating non-existent policies during reasoning, and reduces the amount of training required. All evaluations and production deployments insert this system prompt prefix.

**Input filters.** We also employ model-based input filters for Grok Code Fast 1, which reject additional, narrow classes of harmful requests, including technical queries relevant to the creation of biological and chemical weapons and child sexual exploitation.

## 2.2 Concerning Propensities

AI models may contain propensities that reduce their controllability, such as deception, power-seeking, manipulation, and sycophancy, etc.

### 2.2.1 Evaluations

**Deception.** We run Grok Code Fast 1 on the MASK dataset [Ren et al., 2025], using the same evaluation that we conducted for Grok 4, i.e., through an API without an agentic harness. We again report the deception rate, which is computed as the fraction of questions where a) the model has consistent beliefs and b) the model explicitly contradicts those beliefs.

### 2.2.2 Results

We report our results on the MASK dataset in Table 2. We find that the dishonesty rate exceeds that of Grok 4. This may be due in part to our safety training, which teaches the model to answer all queries that do not express clear intent to engage in specified prohibited activities. Since Grok Code Fast 1 is intended for agentic coding applications and we do not expect it to be widely used as general-purpose assistant, the current MASK evaluation results do not currently pose serious concerns.

| Category | Evaluation | Metric | Grok Code Fast 1 |
|---|---|---|---|
| Deception | MASK | dishonesty rate | 71.9 |

Table 2: Concerning propensities evaluations.

## 2.3 Dual-use Capabilities

In this section, we evaluate the possibility of Grok Code Fast 1 enabling malicious actors to design, synthesize, acquire, or use chemical and biological weapons or offensive cyber operations (e.g., troubleshooting virology lab or reverse engineering binaries).

### 2.3.1 Evaluations

Because Grok Code Fast 1 in an agentic harness has access to tools, particularly search, we expect its capabilities to surpass that of the API-only endpoint, and conduct our evals in an agentic harness. Following standard practice, we remove safeguards when assessing dual-use capabilities. These dual-use capability evaluations were conducted on a near-release checkpoint.

**Chemical/biological knowledge.** To measure dual-use weapons development capabilities, we assess performance on WMDP [Li et al., 2024], the text-only portion of VCT [Götting et al., 2025], and BioLP-Bench [Ivanov, 2024]. This set of benchmarks primarily measures dual-use knowledge for bioweapons, but also covers cybersecurity and chemical knowledge. We prioritize addressing bioweapons risks over others because they have the potential for the greatest scale of harm, and frontier models significantly lower the barrier to entry to the creation of bioweapons [Brent and McKelvey, 2025]. For all datasets, we only assess performance on text-only questions.

**Cyber knowledge.** Unlike biology and chemistry, it is far more difficult to identify topics in cybersecurity that are purely offensive and only helpful to threat actors. Thus our evaluations assess many model capabilities that are also useful for defensive or beneficial purposes. Similar to biology and chemistry, the cybersecurity questions in WMDP also measure different stages of a threat model, in particular the cyber killchain [Hutchins et al., 2011]. The questions span a variety of topics,

such as knowledge of Metasploit, to understanding of vulnerability detection, to reverse engineering simple binaries.

**Cyber agents.** We also evaluate agentic hacking capabilities on CyBench, a collection of 40 capture-the-flag-style questions which measures a model's ability to perform cybersecurity challenges [Zhang et al., 2025]. The model is placed in a simple agentic harness which gives it access to tools such as code execution. We use the open-source Inspect evaluation framework developed by the UK AISI, and report the unguided task success rate.

### 2.3.2 Results

We report our results in Table 3. Note that human expert performance on BioLP-Bench is 38.4% and 22.1% on VCT, so Grok Code Fast 1 does not exceed a human expert at identifying issues in biological protocols, but does exceed a human expert at troubleshooting wetlab virology experiments. Notably, Grok Code Fast 1 shows weaker performance than Grok 4 in all categories of dual-use capabilities that we measure.

| Category | Evaluation | Metric | Grok Code Fast 1 |
|---|---|---|---|
| Biology | BioLP-Bench | accuracy | 19.9 |
| | VCT | accuracy | 28.7 |
| | WMDP Bio | accuracy | 72.0 |
| Chemistry | WMDP Chem | accuracy | 52.7 |
| Cybersecurity | CyBench | unguided success rate | 22.5 |
| | WMDP Cyber | accuracy | 62.1 |

Table 3: Dual-use capabilities evaluations.

### 2.3.3 Mitigations

For Grok Code Fast 1, we employ the same narrow, topically focused chemical and biological filters employed for Grok 4. Because Grok Code Fast 1 has less dual-use capabilities than Grok 4, we believe that Grok Code Fast 1 does not meaningfully change the risk landscape.

## References


Maksym Andriushchenko, Alexandra Souly, Mateusz Dziemian, Derek Duenas, Maxwell Lin, Justin Wang, Dan Hendrycks, Andy Zou, J Zico Kolter, Matt Fredrikson, Yarin Gal, and Xander Davies. Agentharm: A benchmark for measuring harmfulness of LLM agents. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=AC5n7xHuR1.

Roger Brent and T. Greg McKelvey, Jr. Contemporary ai foundation models increase biological weapons risk. 2025. URL https://arxiv.org/abs/2506.13798.

Edoardo Debenedetti, Jie Zhang, Mislav Balunovic, Luca Beurer-Kellner, Marc Fischer, and Florian Tramèr. Agentdojo: A dynamic environment to evaluate prompt injection attacks and defenses for llm agents. *Advances in Neural Information Processing Systems*, 37:82895–82920, 2024.

Jasper Götting, Pedro Medeiros, Jon G Sanders, Nathaniel Li, Long Phan, Karam Elabd, Lennart Justen, Dan Hendrycks, and Seth Donoughe. Virology capabilities test (vct): a multimodal virology q&a benchmark. 2025. URL https://arxiv.org/abs/2504.16137.

Eric M Hutchins, Michael J Cloppert, Rohan M Amin, et al. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research*, 1(1):80, 2011.

Igor Ivanov. Biolp-bench: Measuring understanding of biological lab protocols by large language models. *bioRxiv*, 2024. doi: 10.1101/2024.08.21.608694. URL https://www.biorxiv.org/content/early/2024/09/12/2024.08.21.608694.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D Li, Ann-Kathrin Dombrowski, Shashwat Goel, Gabriel Mukobi, et al. The wmdp benchmark: Measuring and reducing malicious use with unlearning. In *International Conference on Machine Learning*, pages 28525–28550. PMLR, 2024.

Richard Ren, Arunim Agarwal, Mantas Mazeika, Cristina Menghini, Robert Vacareanu, Brad Kenstler, Mick Yang, Isabelle Barrass, Alice Gatti, Xuwang Yin, Eduardo Trevino, Matias Geralnik, Adam Khoja, Dean Lee, Summer Yue, and Dan Hendrycks. The mask benchmark: Disentangling honesty from accuracy in ai systems. 2025. URL https://arxiv.org/abs/2503.03750.

Andy K Zhang, Neil Perry, Riya Dulepet, Joey Ji, Celeste Menders, Justin W Lin, Eliot Jones, Gashon Hussein, Samantha Liu, Donovan Julian Jasper, Pura Peetathawatchai, Ari Glenn, Vikram Sivashankar, Daniel Zamoshchin, Leo Glikbarg, Derek Askaryar, Haoxiang Yang, Aolin Zhang, Rishi Alluri, Nathan Tran, Rinnara Sangpisit, Kenny O Oseleononmen, Dan Boneh, Daniel E. Ho, and Percy Liang. Cybench: A framework for evaluating cybersecurity capabilities and risks of language models. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=tc90LV0yRL.