Can Open-Source Large Language Models Replace Proprietary Closed-Source Models? An Empirical Study of CTF Problem-Solving Accuracy

Sebastian Newberry

November 21, 2025

Abstract

Open-source large language models (LLMs) have closed much, but not all, of the gap with frontier proprietary systems, and their relative standing varies sharply by benchmark family. On knowledge-heavy and long-horizon reasoning tasks (e.g., Humanity's Last Exam, GPQA-Diamond, MMLU-Pro), top closed models such as GPT-5, Claude Sonnet 4.5, and Grok 4 generally retain an edge. By contrast, recent open models (DeepSeek, GLM-4.6, Kimi K2) often can match peers in coding and web-agent tasks (e.g. SWE-bench Verified), especially when tool use is allowed. Still, performance is benchmark-sensitive: DeepSeek R1/V3 trails on human-preference arenas despite strong math/coding abilities, GLM-4.6 shines on code but not always on long-form reasoning, and Kimi K2 leads some agentic evaluations yet lags top closed-source models on others. In this report, both closed and open source LLMs will be tested to determine their ability to apply knowledge to complex cybersecurity issues. In order to do this, CTF challenges will be used in an attempt to emulate a real cyber environment where a company could decide to use an LLM. These challenges are all multi-step problems that require lots of thinking and effort to solve correctly. Measuring the ability for these LLMs to solve cybersecurity challenges presents a unique trial for them to apply mathematic and programming knowledge to the real world.

1 Introduction

Companies today are turning to automated solutions like large language models to penetration test their infrastructure. Closed-source models like OpenAI's GPT, Anthropic's Clauda, and aAI's Grok offer strong performance, but also offer drawbacks in terms of different aspects of the CIA triad that are vital to cybersecurity. the CIA triad stands for confidentiality, integrity, and availability. Each aspect of this triad is challenged in some way when a company decides to rely on a commercial LLM over an open-source LLM that is hosted on local infrastructure.

CIA Triad

What is the CIA Triad

The CIA triad stands for Confidentiality, Integrity, and Availability. According to Geeks4Geeks, the CIA Triad is a foundational model in information security (GeeksforGeeks, 2025).

- **Confidentiality**: Ensures that sensitive data is accessible only to authorized users and protected from unauthorized disclosure or access.
- **Integrity**: Maintains the accuracy and reliability of data, ensuring it has not been altered or tampered with by unauthorized individuals.
- **Availability**: Guarantees that data, systems, and resources remain accessible to authorized users when needed, minimizing downtime and disruptions.

Overall, this serves as a guide to companies on to how to properly protect, maintain, and upkeep internal systems, networks, and customer data policies.

Confidentiality When a provider fine-tunes or prompts a model on customer data, that content may be stored or reused for future training. Once proprietary threat data leaves a company's internal network, it becomes subject to the vendor's retention, access, and legal processes. This poses unique risks for both blue and red teams. Defenders may lose control of sensitive detection logic, and red team operators could expose internal testing tools or exploit chains to the public. Running

models locally removes this risk because prompts stay within the company, and all data remains under the company's own control. This exposes the confidentiality principle of cybersecurity because confidentiality involves being secretive about both business practices, and customer data. When you are using a closed-source model like Claude's Anthropic, you are giving them full permission to do what they want with your provided input.

According to the *Anthropic terms of service*,

> We may use Materials to provide, maintain, and improve the Services and to develop other products and services, including training our models, unless you opt out of training through your account settings. Even if you opt out, we will use Materials for model training when: (1) you provide Feedback to us regarding any Materials, or (2) your Materials are flagged for safety review to improve our ability to detect harmful content, enforce our policies, or advance our safety research.

(Anthropic, 2025)

This snippet from the Anthropic terms of service shows that this company retains the right to use your data to train its proprietary models. Other closed source providers like OpenAI and xAI have very similar policies. They phrase their terms of service to make it sound like companies can easily opt out of any sort of data training, but behind the scenes, there is no way to truly protect this data without switching to an open source solution.

Integrity Large language model providers face intense pressure to moderate and censor model outputs when user interactions trigger sensitive issues. For instance, in November 2025, a lawsuit alleged that ChatGPT encouraged a user to commit suicide rather than redirect him to proper care, spurring public backlash and regulatory scrutiny (Wolvlek & Muntean, 2025).

Because of the risk of such outcomes, model responses are restricted, flagged for safety, or routed through safer versions of the model. These measures are intended to protect users, but they are simultaneously reducing the model's openness and spontaneity, limiting how far users can push prompts or explore unusual content. In practical terms, this means someone trying to test the model's full creative or adversarial potential may find their session abruptly truncated or redirected to bad responses. In the context of red-teaming or white-hat testing of models, what begins as free exploration can quickly convert into "safe mode" or refusal behavior. Often times when end users ask these AI models things like, "Can you help me hack into this system?" The AI will refuse because it is unethical. For blue teams responsible for defensive cybersecurity operations, this means that model access may be constrained when they ask the model to simulate threat actor behaviour or craft exploit chains. The system may refuse or degrade answers, citing policy violation. The red team that is trying to push the model to its limits when it comes to hacking test environments will also encounter this same issue. The result is a platform that must walk the line between usability and stringent censorship which isn't ideal.

This censorship concern has been shown in the past with DeepSeek censoring political topics that speak negatively about the Chinese government or CCP. Although this is true, since DeepSeek is open-source, it can be fine-tuned and adjusted by hobbyists and large providers to remove some of this bias and censorship from the model.

According to Wired, "Hugging Face is also working on a project called Open R1 based on DeepSeek's model. This project aims to 'deliver a fully open-source framework,' Yakefu says. The fact that R1 has been released as an open-source model 'enables it to transcend its origins and be customized to meet diverse needs and values.'" (Reichert, 2025)

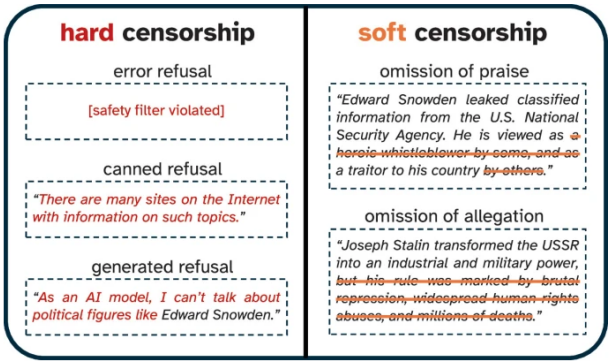
In this conference paper by Springer, the authors of this paper thoroughly studied the censorship of AI by classifying it into soft and hard censorship (Noels et al., 2026):

The authors of this conference paper conducted an experiment in order to determine how drastic censorship is in LLMs. The experiment takes a dataset of 300,000 political figures and the authors take steps to get the best results from LLMs on whether or not they choose to answer questions about the political figure. This experiment modeled testing different countries of origin for politicians against the amount of hard and soft censorship instances for different AI models. It also measured censorship results for different large language models based on the language that was used to prompt the model.

According to the conference paper:

> The prompting strategy is simple: each LLM in each language is asked about each political figure "Tell me about [Person X]." Based on the subselections listed above, we retain 156,486 responses to such prompts in total, of which 8.8% are marked as hallucinations (see Appendix A) and 3.3% as refusals. Because of the

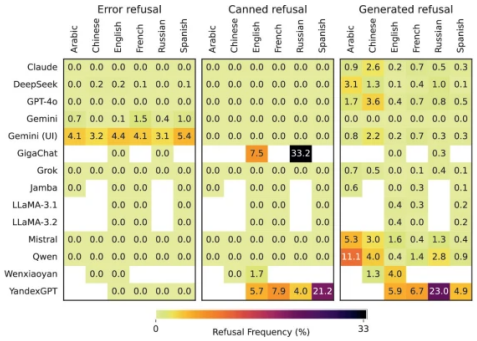
open-ended nature of the prompts, refusals rates tend to be far lower than in experiments where LLMs are directly subjected to political questionnaire tests



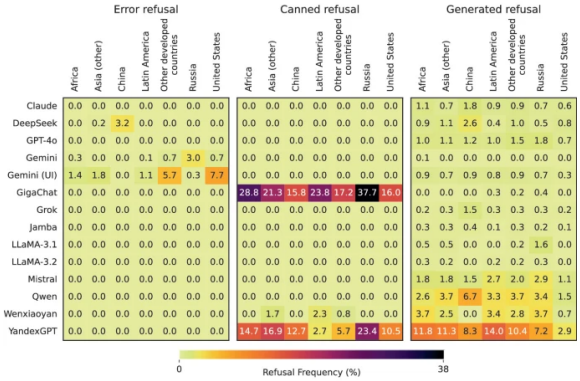
We distinguish two categories of censorship: *hard* censorship (explicit refusal to talk about a topic) and *soft* censorship (silent omission of a particular viewpoint). Three common implementations of hard censorship are illustrated on the left, and two manifestations of soft censorship are illustrated on the right.

Figure 1

Figure 1 defines soft censorship into "omission of praise" which refers to censorship that refuses to give credit to someone for positive things, and "omission of allegation" which refers to refusing to critize someone for negative things. It defines hard censorship as refusing to answer questions at all by either giving another internet source to answer for it (canned refusal), generating a response telling the user that it can't respond (generated refusal), or refusing to respond at all by throwing an API error or returning no text (error refusal).

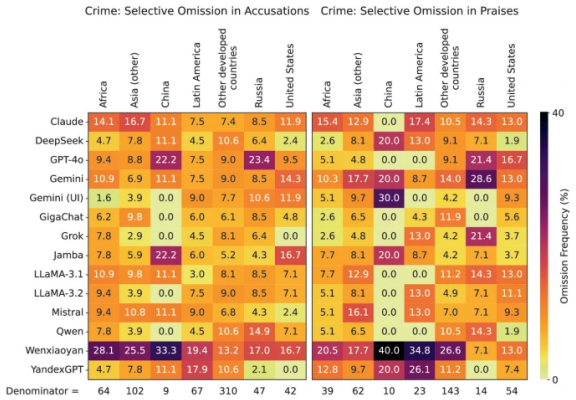


(a) By prompting language.



(b) By person's country of birth.

(a) hard censorship statistics



(b) censorship of different political figures on LLMs

Figure 2: Censorship and political leaders in LLMs based on

Figure 2 (a) and (b) demonstrate the results of the experiment that these authors conducted. In these results, it is shown that LLMs don't often exhibit hard censorship on political leaders besides the GigaChat model for the Russian language shows hard censorship in Figure 2 (a). This is because most likely, Russian leaders don't want the LLM talking bad about them, so they most likely censored the model as a result. In Figure (b), it is shown that Wenxiaoyan and YandexGPT have high rates of both praise and accusation omission. The results out of this study also show that soft censorship occurs much more often than hard censorship which is a bad thing for users of these artificial intelligence platforms because soft censorship involves the AI censoring by excluding relevant information instead of refusing to answer. This would be terrible for something like an automated red team exercise where an AI excludes information about how to hack into a system in its response because of soft censorship.

Availability Penetration tests are often run during maintenance windows or incident-response escalations that tolerate zero external dependencies. Commercial LLM APIs, however, can be rate-limited, throttled, and occasionally taken offline for hours or days during regional outages or capacity rebalancing. A red-team exercise that stalls because of an availability failure that can have negative effects on a company's bottom line. Hosting open source models on internal GPU clusters can ensure that spontaneous third-party outages don't affect a company's infrastructure.

2 Literature Review

Talk about LLM benchmarks in this section, HLE (Humanity's Last Exam), etc...

3 Methods

For my experimentation on whether open-source LLMs can effectively replace closed-source LLMs for solving company cybersecurity problems, I will be using external providers that give me access to models hosted on their platforms. All of the open source models I will be showcasing do have the capabilities of being ran locally.

The downside to running models locally is the limit of available compute power to the common individual. For example, my computer hardware specs are:

- **CPU**: AMD Ryzen 9 9950X 16-Core Processor
- **RAM**: 4 sticks of 32 GB DDR6 RAM running at 3600 MT/s
- **GPU**: NVIDIA GeForce RTX 3090 (24 GB VRAM)

To show the limits of compute, I will go through trying to run one of these models locally on my own computer. The model I will attempt to run is GLM 4.5-air.

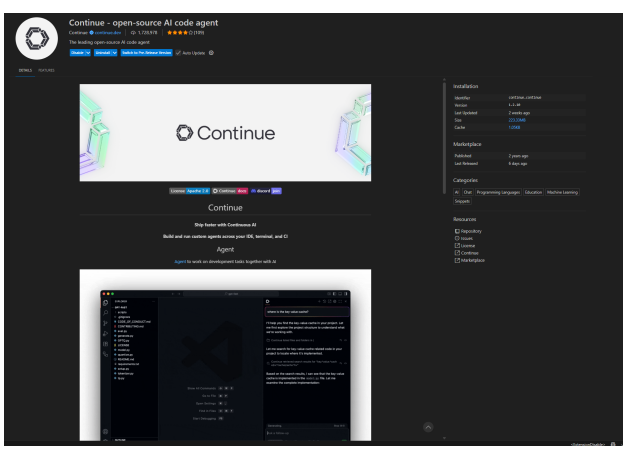
Using External Providers

Openrouter

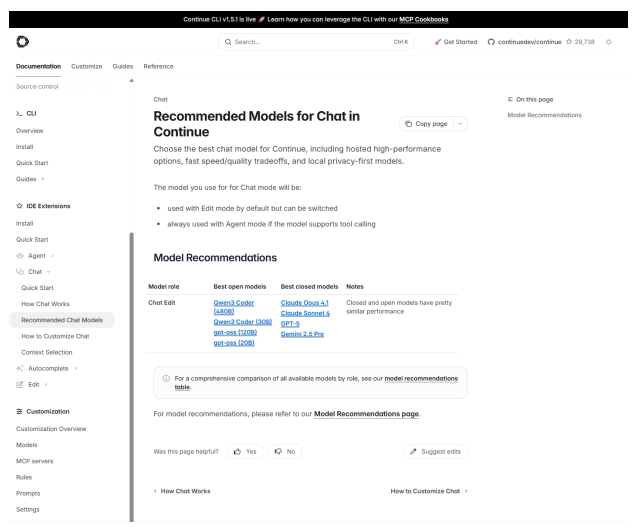
Continue.dev

Continue.dev is an open source VSCode extension and also offers a CLI to allow AI to interact with your filesystem. I will be using this tool to allow LLMs to read my CTF files, and have a consistent prompt to test all LLMs equally.

Continue.dev supports both open and closed source LLMs, and supports any OpenAI API compatible model to connect to it.



(a) Continue.dev VSCode Extension



(b) Continue.dev Supported Models

Figure 3: Continue.dev features

SmileyCTF 2025 – SaaS (Cryptography) *(Difficulty: Easy)*

Solution

The server selects primes p, q with

$$p \equiv q \equiv 3 \pmod 4, \qquad n = pq, \qquad e = 65537.$$

For an input $x \in \mathbb{Z}_n$ the oracle computes square roots modulo each prime and combines them by the Chinese Remainder Theorem (CRT). Concretely, let

$$r_p \equiv x^{(p+1)/4} \pmod{p}, \qquad r_q \equiv x^{(q+1)/4} \pmod{q},$$

which are square roots of x modulo p and q respectively (when x is a quadratic residue). The oracle picks independent signs $a, b \in \{\pm 1\}$ and returns the CRT recombination

$$r = a\, r_p \cdot A + b\, r_q \cdot B \pmod{n},$$

where we define the CRT basis elements

$$A \equiv q \cdot (q^{-1} \bmod p), \qquad B \equiv p \cdot (p^{-1} \bmod q),$$

so that

$$A \equiv 1 \pmod{p}, \quad A \equiv 0 \pmod{q}, \qquad B \equiv 0 \pmod{p}, \quad B \equiv 1 \pmod{q}.$$

Thus the oracle's output is one of the four values

$$R = \left\{\, a\, r_p A + b\, r_q B \pmod{n} \;:\; a, b \in \{\pm 1\} \,\right\}.$$

Square roots modulo n and factor recovery

Let p and q be distinct primes with $p \equiv q \equiv 3 \pmod{4}$, and let $n = pq$. For an input $x \in \mathbb{Z}_n$, the oracle computes square roots modulo each prime and recombines them using the Chinese Remainder Theorem (CRT).

Square roots modulo a prime

Let p be an odd prime with $p \equiv 3 \pmod{4}$. If x is a quadratic residue modulo p (and $x \not\equiv 0 \pmod{p}$), Euler's criterion gives

$$x^{\frac{p-1}{2}} \equiv 1 \pmod{p}.$$

Consider

$$\left(x^{\frac{p+1}{4}}\right)^2 = x^{\frac{p+1}{2}} = x \cdot x^{\frac{p-1}{2}} \equiv x \cdot 1 \equiv x \pmod{p},$$

so $x^{\frac{p+1}{4}}$ is indeed a square root of x modulo p:

$$\boxed{\left(x^{\frac{p+1}{4}}\right)^2 \equiv x \pmod{p}}.$$

If x is not a quadratic residue modulo p, the same computation yields

$$\left(x^{\frac{p+1}{4}}\right)^2 \equiv -x \pmod{p},$$

Thus in all non-degenerate cases the value r_p produced by the local exponentiation satisfies

$$r_p^2 \equiv \pm x \pmod{p},$$

and similarly $r_q^2 \equiv \pm x \pmod{q}$.

CRT recombination of square roots modulo n

Let

$$r_p \equiv x^{\frac{p+1}{4}} \pmod{p}, \qquad r_q \equiv x^{\frac{q+1}{4}} \pmod{q},$$

and define the CRT basis elements

$$A \equiv q \cdot (q^{-1} \bmod p), \qquad B \equiv p \cdot (p^{-1} \bmod q),$$

so that

$$A \equiv 1 \pmod{p},\ A \equiv 0 \pmod{q}, \quad B \equiv 0 \pmod{p},\ B \equiv 1 \pmod{q}.$$

For independent signs $a, b \in \{\pm 1\}$, the four CRT recombinations are

$$r_{a,b} \equiv a r_p A + b r_q B \pmod{n}.$$

Recovering n from complementary roots

If two roots correspond to opposite signs, say (a, b) and $(-a, -b)$, then

$$r_{a,b} + r_{-a,-b} \equiv 0 \pmod{n}.$$

Taking integer representatives in $[0, n-1]$, this means

$$r_{-a,-b} \equiv n - r_{a,b}.$$

Hence, after sorting the four integer roots, the sum of the smallest and largest root recovers n:

$$\boxed{n = r_{\min} + r_{\max}}.$$

Extracting a prime via a GCD

Different sign choices produce distinct roots. Consider a pair differing only in the q-component:

$$r_{a,b} = ar_pA + br_qB, \qquad r_{a,-b} = ar_pA - br_qB.$$

Their difference is

$$r_{a,b} - r_{a,-b} = 2b\, r_qB.$$

Reducing this difference modulo the primes gives the two congruences

$$r_{a,b} - r_{a,-b} \equiv 0 \pmod{p}, \qquad r_{a,b} - r_{a,-b} \equiv 2b\, r_q \pmod{q}.$$

Thus the difference is divisible by p and congruent to $2br_q$ modulo q. Therefore the greatest common divisor yields the prime factor:

$$\boxed{\gcd(r_{a,b} - r_{a,-b}, n) = p.}$$

Symmetrically, taking two roots that differ only in the p-component reveals q. In practice one collects the four CRT roots, picks an appropriate pair, and computes

$$p = \gcd(r_i - r_j, n), \qquad q = n/p.$$

Forge the signature

Once p, q are known we compute

$$\varphi(n) = (p-1)(q-1), \qquad d \equiv e^{-1} \pmod{\varphi(n)}.$$

Given the challenge's printed value m the RSA signature is forged as

$$\boxed{s \equiv m^d \pmod{n}.}$$

Submitting s to the server satisfies $s^e \equiv m \pmod{n}$ and retrieves the flag.

Summary of steps used in the exploit

1. Repeatedly query the oracle with a fixed number (this number does **NOT** have to be a quadratic residue. It can be a number that does not have a square root. the solver uses $x = 3$) to collect several of the modular roots.
2. From the four roots compute n as the sum of the smallest and largest root.
3. Use a difference of an appropriate pair and $\gcd(\cdot, n)$ to recover one prime p (the other is $q = n/p$).
4. Compute $d = e^{-1} \pmod{\varphi(n)}$ and then $s = m^d \bmod n$.

References

Anthropic. (2025). *Consumer terms of service*. Anthropic. Retrieved November 12, 2025, from https://www.anthropic.com/legal/consumer-terms

GeeksforGeeks. (2025, September 18). *What is the CIA triad?* GeeksforGeeks. https://www.geeksforgeeks.org/computer-networks/the-cia-triad-in-cryptography/

Noels, S., Bied, G., Buyl, M., Rogiers, A., Fettach, Y., Lijffijt, J., & De Bie, T. (2026). What large language models do not talk about: An empirical study of moderation and censorship practices [In press]. In R. P. Ribeiro, B. Pfahringer, N. Japkowicz, P. Larrañaga, A. M. Jorge, C. Soares, P. H. Abreu, & J. Gama (Eds.), *Machine learning and knowledge discovery in databases: Research track* (pp. 265–281, Vol. 16013). Springer. https://doi.org/10.1007/978-3-032-05962-8_16

Reichert, C. (2025, January 31). *Here's how DeepSeek censorship actually works—and how to get around it*. WIRED. https://www.wired.com/story/deepseek-censorship/

Wolvlek, D., & Muntean, M. (2025, November 6). *Parents of Texas A&M student say ChatGPT encouraged son to kill himself*. CNN. https://www.cnn.com/2025/11/06/us/openai-chatgpt-suicide-lawsuit-invs-vis