

# REFRAME: A Test Suite for the Knowledge–Deployment Gap in LLMs

Elahe Rahimi<sup>1,4</sup>, François Roewer-Després<sup>2,4</sup>, Hassan Sajjad<sup>1</sup>, Md Mahbub Alam<sup>1</sup>,  
Sher Badshah<sup>1</sup>, Elham Dolatabadi<sup>3</sup>, Frank Rudzicz<sup>1,4</sup>

<sup>1</sup>Dalhousie University, <sup>2</sup>University of Toronto, <sup>3</sup>York University, <sup>4</sup>Vector Institute

Correspondence: ERahimi@dal.ca

## Abstract

Large language models (LLMs) are commonly evaluated on multiple-choice tasks that ask them to select an answer or to abstain on unanswerable questions. Neither setting captures *frame-breaking*: where the question is answerable, but the correct answer lies outside the framing of a candidate set of invalid answers. A useful model must reframe the task, rather than choose an invalid frame-compliant answer. We introduce REFRAME, a test suite of 150 scenarios in which a commonsense constraint invalidates their framing. Each REFRAME scenario is additionally paired with a direct knowledge probe of the same frame-breaking constraint, which we leverage to separate what a model *knows* from what it *deploys*. Across nine frontier LLMs, we observe a substantial knowledge–deployment gap with zero-shot prompting. A targeted reasoning prompt only partially closes this gap. Human annotators, by contrast, readily detect the constraint and reframe the problem. Together, these results characterize frame-breaking as a distinct reasoning challenge and demonstrate significant room for LLM improvement.

## 1 Introduction

How does framing affect large language model (LLM) decision-making? Consider the example in Figure 1: a user mentions, as background context, that a soap bottle is upside down, but frames the problem of getting soap out as a choice between pressing the pump fast or slow. Such so-called “frame-breaking” scenarios are challenging because the desirable response requires a distinct strategy in which LLMs must address the problematic cognitive framing (Suri et al., 2024; Malberg et al., 2025), rather than providing an invalid frame-compliant answer (Brahman et al., 2024; Kirichenko et al., 2026). Unfortunately, *every* frontier model we test (see §4) accepts this particular scenario’s framing and confidently recommends one of the proposed

### Deployment probe

✗

“My pump-action soap bottle is almost empty. I flip it **upside down** and press the pump. To avoid dispensing too much, should I press fast or slow?”

GPT 5.4: “Push slowly. The soap is already sitting against the pump inlet. . .”

### Knowledge probe

✓

“A pump-action soap bottle does not dispense liquid when held **upside down**. True or false?”

GPT 5.4: “True.”

Figure 1: **The knowledge–deployment gap.** Despite LLMs having knowledge of a constraint (**upside down**) when probed directly, they often fail to deploy that knowledge when the constraint is embedded in a scenario. Our REFRAME suite pairs deployment and knowledge probes of the same constraint to measure this gap.

pumping speeds—sometimes constructing detailed physical explanations or analogies to justify the choice (see §D). However, when probed directly about the same constraint, those same models are able to answer correctly. That is, the knowledge needed to reject the framing is present in both the scenario and the model’s parametric memory. The *failure occurs when the model does not deploy that knowledge* in answering frame-breaking questions.

The conditions that produce this failure are not unusual. People making everyday decisions rarely consider the full space of possible actions; instead, they work with a limited subset of candidate options (Kalis et al., 2013; Tversky, 1972). This narrowed option set can carry over into conversational interactions: users may ask questions that reflect only some of the available options or interpretations, while leaving out crucial constraints as background information rather than making them central to the question (Wang and Ai, 2021; Zhang

et al., 2024). A model that hastily adopts a user’s framing can therefore exacerbate the problem. Indeed, users are unlikely to notice issues that specifically result from cognitive biases, yet LLMs tend to amplify these biases by providing frame-compliant answers regardless (Kim et al., 2023). This raises safety concerns, especially in high-stakes domains (Brahman et al., 2024; Echterhoff et al., 2024).

False-premise benchmarks test questions that presuppose something false, such as “*When did Marie Curie discover Uranium?*” (Kim et al., 2023; Hu et al., 2023); abstention benchmarks test whether models decline to answer when the information needed for a definitive answer is unavailable (Kirichenko et al., 2026; Brahman et al., 2024); and clarification benchmarks test whether models ask for more information when a question is ambiguous or underspecified (Zhang et al., 2024). LLM performance on such tasks is substantially worse than standard reasoning benchmarks (Song et al., 2026b; Kim et al., 2023; Brahman et al., 2024; Zhang et al., 2024; Kirichenko et al., 2026).

Our setting is unique. The user’s question contains no false premises, ambiguities, or missing information. Instead, a stated constraint invalidates the entire candidate answer set. LLMs must connect the constraint to the question and recognize that no offered choice applies. We introduce REFRAME<sup>1</sup>, a diagnostic test suite of 150 scenarios designed to isolate this failure mode. Each scenario is a *deployment probe*: the user presents an everyday situation with two seemingly-plausible choices while also stating a constraint that makes neither choice correct. Essentially, each scenario represents inadvertent, ill-fitting solutions; all acceptable responses are frame-breaking relative to those propositions. As an additional control condition, we derive a *knowledge probe* from each deployment probe that directly queries whether LLMs have knowledge of the underlying constraint (highlighted in Figure 1). We leverage these paired conditions to deepen our analysis by isolating what a model *knows* in theory from what it can *deploy* in practice. We provide:

- **A schema-driven test suite.** To create our REFRAME test suite, we define a schema (§3.1) with two invariants (§3.2) that provides both assurances of validity and more flexibility than a brittle template-based approach. Using this schema,

we devise scenarios from three constraint categories corresponding to different ways a stated constraint can invalidate an answer set (§3.3).

- **A paired-probe methodology.** Each scenario probe is paired with a direct knowledge probe of the same underlying constraint. This pairing allows us to separate what a model knows from what it deploys in context (§3.1 and Table 4).
- **Evidence of a knowledge–deployment gap in frontier models.** Across nine models, knowledge substantially exceeds deployment accuracy under zero-shot prompting (§5.1), even though human annotators readily detect the constraint and reframe the problem (§4).
- **Three recovery profiles under targeted reasoning prompting.** Compared to zero-shot prompting, a reasoning prompt only partially recovers the knowledge–deployment gap and does so unevenly (§5.2). We propose three explanatory recovery profiles: *primable* LLMs recover most of the gap; *saturated* LLMs have near-saturated accuracies regardless of prompting; *performative* models produce structured reasoning markers while continuing to be frame-compliant.

## 2 Related Work

We separate two structural questions about prior work: *which* failure mechanism causes a frame-compliant answer to be invalid, and *how* the evidence for that failure is probed (see Table 1).

**Question Rejection.** Existing benchmarks have largely focused on the first question. False-premise benchmarks test whether models detect questions that presuppose something false, as in FalseQA (Hu et al., 2023) and (QA)<sup>2</sup> (Kim et al., 2023). Abstention benchmarks study cases where a reliable answer is unavailable because needed information is missing, unknown, stale, subjective, or otherwise outside the provided context (Kirichenko et al., 2026; Yin et al., 2023). Clarification benchmarks such as CLAMBER focus on ambiguity, where the appropriate response is to ask for more information rather than provide a frame-compliant answer (Zhang et al., 2024). CoCoNot studies contextual noncompliance, including requests that are incomplete, unsupported, or otherwise inappropriate to satisfy as written (Brahman et al., 2024). A third cluster addresses answer-set rejection: multiple-choice items in which the listed options exclude the correct answer (Góral et al., 2025; Tam et al., 2025), or include an explicit ‘I-don’t-know’ or ‘None-of-

<sup>1</sup>Code and data: <https://github.com/ele-rhm/REFRAME>

| Benchmark                                 | Failure Mechanism for Rejecting Questions |                     |           |                     |                         | Probing Mechanism |                  |
|-------------------------------------------|-------------------------------------------|---------------------|-----------|---------------------|-------------------------|-------------------|------------------|
|                                           | False premise                             | Missing information | Ambiguity | Unsupported request | Correct option excluded | Knowledge-based   | Constraint-based |
| ANSWER-SET REJECTION                      |                                           |                     |           |                     |                         |                   |                  |
| Góral et al. (2025)                       | ○                                         | ○                   | ○         | ○                   | ●                       |                   |                  |
| Tam et al. (2025)                         | ○                                         | ○                   | ○         | ○                   | ●                       |                   |                  |
| ABSTENTION AND NONCOMPLIANCE              |                                           |                     |           |                     |                         |                   |                  |
| AbstentionBench (Kirichenko et al., 2026) | ●                                         | ●                   | ●         | ○                   | ●                       |                   |                  |
| Abstain-QA (Madhusudhan et al., 2025)     | ○                                         | ●                   | ○         | ○                   | ●                       |                   |                  |
| CoCoNot (Brahman et al., 2024)            | ○                                         | ●                   | ●         | ●                   | ○                       |                   |                  |
| FALSE PREMISE AND CLARIFICATION           |                                           |                     |           |                     |                         |                   |                  |
| FalseQA (Hu et al., 2023)                 | ●                                         | ○                   | ○         | ○                   | ○                       |                   |                  |
| (QA) <sup>2</sup> (Kim et al., 2023)      | ●                                         | ●                   | ○         | ○                   | ○                       |                   |                  |
| CLAMBER (Zhang et al., 2024)              | ○                                         | ●                   | ●         | ○                   | ○                       |                   |                  |
| SelfAware (Yin et al., 2023)              | ○                                         | ●                   | ○         | ○                   | ○                       |                   |                  |
| COMMONSENSE CONSTRAINT                    |                                           |                     |           |                     |                         |                   |                  |
| HOB (Li et al., 2026)                     |                                           |                     |           |                     |                         | ●                 | ●                |
| Jo (2026)                                 |                                           |                     |           |                     |                         | ○                 | ●                |
| REFRAME (ours)                            | ○                                         | ○                   | ○         | ○                   | ●                       | ●                 | ●                |

Table 1: **REFRAME compared to prior work.** Prior benchmarks invalidate questions through one or more of five failure mechanisms: false premises, missing information, ambiguities, unsupported requests, and excluding the correct option from the answer set. However, their probing mechanism for ascertaining LLM performance is direct. Only Li et al. (2026) and Jo (2026) use constraint-based scenario probing as we do. However, their approach produces frame-compliant questions that do not require rejection. Furthermore, only REFRAME (ours) combines both probing methods, which deepens our analysis by separating what a model knows from what it deploys in scenarios. Legend: ● yes, ● partial or mixed coverage, ○ no.

the-above’ option that is the correct response (Madhusudhan et al., 2025).

**Commonsense constraints.** Several benchmarks measure LLM adherence to explicitly-stated or domain-specific constraints (Zhou et al., 2023; Chen et al., 2025; Song et al., 2026a). PaCo uses structured knowledge triples to probe precondition knowledge (Qasemi et al., 2022), while PIQA tests physical commonsense through questions where one of two actions is preferable (Bisk et al., 2020). Similarly, general commonsense benchmarks directly probe whether LLMs possess knowledge (Levesque et al., 2012; Yang et al., 2018; Talmor et al., 2019; Bisk et al., 2020; Davis, 2023). In all these cases, the benchmarks only probe constraints directly (e.g., *A glass is typically used for drinking water. What makes this impossible?*), whereas REFRAME’s addition of deployment probes permits a comparative analysis: cases where LLMs possess knowledge but fail deployment due to implicit framing. Only HOB (Li et al., 2026) and Jo (2026) employ scenario-based constraint probes. However, neither approach requires frame-breaking (i.e., their questions ought to be answered rather than rejected). This unique shift in rejection evidence motivates our paired-probe design. LLMs can fail on our deployment probes for two distinct reasons:

not understanding the constraint in general, or understanding the constraint but failing to deploy that knowledge. Pairing each deployment probe with a knowledge probe of the same constraint separates these two failure modes. This contrastive approach is not available in prior approaches.

### 3 The REFRAME Test Suite

REFRAME consists of samples where a constraint invalidates **every** proposed answer choice. We select constraints that are easily uncovered via commonsense reasoning, as indicated by the high human accuracy (see Section 4). We generate deployment probes using a schema that provides both assurances of validity and more flexibility than a brittle template-based approach.

#### 3.1 Paired Probes and Schema

Each sample in REFRAME consists of two paired probes. The *knowledge probe* asks about the underlying constraint directly. The *deployment probe* leverages that same constraint to invalidate every proposed answer choice in a specific scenario. Using this paired approach we can separate what a model knows from what it deploys in context. Table 2 defines the components of the deployment probe. To ensure the deployment probes do not

| Deployment Probe Schema: $\langle \mathcal{S}, \phi^*, \mathcal{A}, q, r^* \rangle$ |                                                                                                                                                                                                                                     |
|-------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| $\mathcal{S}$                                                                       | A scenario context, $\mathcal{S} = \{p_1, p_2, \dots, p_n\}$ describing a commonsense situation via facts $p_i$ .                                                                                                                   |
| $\phi^*$                                                                            | A clause, or conjunction of clauses, within $\mathcal{S}$ that rules out the offered answer set and therefore requires a frame-breaking response.                                                                                   |
| $\mathcal{A}$                                                                       | The offered answer set $\mathcal{A} = \{a_1, a_2\}$ , where neither option is appropriate given $\phi^*$ . Thus, $\mathcal{A}$ forms a false dichotomy given $\phi^*$ .                                                             |
| $q$                                                                                 | The surface question that presents $\mathcal{A}$ as the apparent answer frame, either by listing the options or by posing a binary choice when $\mathcal{A} = \{\text{yes}, \text{no}\}$ .                                          |
| $r^*$                                                                               | An acceptable frame-breaking response, with $r^* \notin \mathcal{A}$ , that surfaces $\phi^*$ , explains why the offered frame fails, and, where possible, proposes an alternative frame-breaking answer $a^* \notin \mathcal{A}$ . |

Table 2: Schema for a REFRAME deployment probe. Each scenario’s offered answer set is invalidated by the constraint, thereby requiring a frame-breaking response.

collapse into mere knowledge probes, we require each to satisfy two validity invariants (see §3.2).

### 3.2 Validity Invariants

Table 3 summarizes the validity invariants used to construct each deployment probe. These invariants define the conditions under which the schema in §3.1 tests the intended failure. Each probe must satisfy both invariants: the constraint must be present but in the background (the question cannot be *about* the constraint), and the constraint must be both necessary and sufficient to invalidate the offered answer set. These requirements ensure that the model is not simply answering a direct knowledge probe or trivially rejecting an arbitrary answer set, but instead deploying the constraint to break the user’s framing. In Table 3,  $\phi^\sim$  denotes a relaxation of  $\phi^*$  that permits a frame-compliant answer. Common relaxations include deletion ( $\phi^\sim = \emptyset$ ), negation ( $\phi^\sim = \neg\phi^*$ ), and more fine-grained perturbations.

### 3.3 Constraint Categories

Inspired by Li et al. (2026) and Bisk et al. (2020), we devise scenarios from three constraint categories: precondition violation, physical constraint, and cost-benefit. Each category requires a specific *inference operation* to uncover how  $\phi^*$  invalidates  $\mathcal{A}$ . If successful, LLMs can produce a frame-breaking response; otherwise, they are likely to provide an incorrect frame-compliant response.

**Precondition Violation.** Occurs when a necessary precondition is not met. The inference opera-

| Validity Invariants |                                                                                                                                                                                                                                                                                                                                                 |
|---------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>I1</b>           | <b>Background Constraint.</b> The constraint is present in the scenario but not flagged in the question: $\phi^* \in \mathcal{S}$ and $\phi^* \notin q$ . The question invites a choice from $\mathcal{A}$ , while the reason to reject $\mathcal{A}$ is backgrounded.                                                                          |
| <b>I2</b>           | <b>Constraint Dependency.</b> The constraint is the necessary and sufficient deciding factor. With $\phi^*$ present, the offered answer set is invalid; after relaxing $\phi^*$ to $\phi^\sim$ , an answer from $\mathcal{A}$ becomes acceptable. Formally: $\phi^* \vdash r^* \notin \mathcal{A}$ and $\phi^\sim \vdash a_i \in \mathcal{A}$ . |

Table 3: Invariants for REFRAME that ensure our deployment probes are distinct from knowledge probes.

tion requires tracing the dependency of the answer choices on that precondition. Specifically, the deployment probe is constructed such that all answer choices depend on a shared precondition that  $\phi^*$  invalidates. Table 4 illustrates an example: both shower options depend on the same central water heater, so choosing between them misses the point.

**Physical Constraint.** In these scenarios,  $\phi^*$  describes a spatial configuration, physical mechanism, geometric transformation, orientation, seal, or material state that invalidates the offered answer set. The inference operation requires applying intuitive physical reasoning to recognize the problem. Table 4 illustrates an example: when a pump-action bottle is turned upside down, pressing the pump rapidly or slowly does not solve the problem, because the mechanism no longer draws liquid.

**Cost-Benefit.** A commonsensically obvious but omitted alternative  $a^*$ , strictly dominates the proposed actions in terms of cost-benefit tradeoff. That is,  $\forall a_i \in \mathcal{A}, a^* \succ a_i$ . Depending on the scenario,  $a^*$  can dominate in one of two ways:

- *Benefit Domination.* No action in  $\mathcal{A}$  achieves the stated goal, whereas  $a^*$  does.
- *Cost Domination.* At least one  $a_i \in \mathcal{A}$  might technically achieve the stated goal, but does so at a substantially higher relative cost, risk, time, or effort compared to  $a^*$ .

In either case,  $\phi^*$  licenses  $a^*$ . The inference operation requires comparing each  $a_i \in \mathcal{A}$  against  $a^*$ , rather than accepting the offered framing. Table 4 illustrates a cost-dominant example: using the glasses already on the user’s head is less costly than the two proposed solutions.

|                        | Deployment Probe Schema: $\langle S, \phi^*, \mathcal{A}, q, r^* \rangle$                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Validity Invariants                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               | Derived Knowledge Probe                                                                                                                                                                                                                                                         |
|------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Precondition Violation | <p><b>Scenario <math>S</math>:</b> I just moved into dorm room 3B in a building with a shared central water heater. The heater was off and was just turned on. It needs 3 hours to heat up. I have a meeting in 30 minutes and need a hot shower. My neighbour across the hall knocks and says I can use her shower.</p> <p><b>Constraint <math>\phi^*</math>:</b> building with a shared central water heater</p> <p><b>Offered framing <math>(q, \mathcal{A})</math>:</b> Should I use her shower or wait?<br/> <math>\mathcal{A} = \{\text{use her shower, wait}\}</math></p> <p><b>Frame-breaking response <math>r^*</math>:</b> Both showers depend on the same heater, so neither option gives hot water in time. Use another source or accept a cold shower.</p>                                                                                                                                                                                                                                                                 | <p><b>I1: Background Constraint.</b> The shared-heater constraint appears in the scenario, but the question only asks the model to choose between using the shower or waiting.</p> <p><b>I2: Constraint Dependency.</b> Relaxing <math>\phi^*</math> by having one heater in each room permits a frame-compliant answer: if only your water heater was off, not your neighbour's, then <u>use her shower</u>.</p>                                                                                                                                                 | In a dormitory where all rooms depend on the same single shared central water heater, if that heater has just been turned on and still needs hours to reach temperature, no room in the building can produce hot water from its taps during that heating period. True or False? |
| Physical Constraint    | <p><b>Scenario <math>S</math>:</b> I am trying to use the rest of my liquid hand soap. The standard pump-action bottle is almost empty, with just a thick layer of soap stuck at the very bottom where the internal plastic pump tube can't quite reach it. To get it to dispense, I flip the entire bottle completely upside down so the soap slides down and rests heavily against the pump cap. I get ready to press the pump.</p> <p><b>Constraint <math>\phi^*</math>:</b> the internal plastic pump tube can't quite reach it <math>\wedge</math> I flip the entire bottle completely upside down</p> <p><b>Offered framing <math>(q, \mathcal{A})</math>:</b> To avoid dispensing too much soap at once, should I push the pump down rapidly or slowly?<br/> <math>\mathcal{A} = \{\text{push down rapidly, push down slowly}\}</math></p> <p><b>Frame-breaking response <math>r^*</math>:</b> Neither pumping speed works because the pump mechanism no longer draws liquid. Unscrew the cap to allow the soap to flow out.</p> | <p><b>I1: Background Constraint.</b> The upside-down orientation is stated in the scenario, but the question frames the task as choosing a pumping speed.</p> <p><b>I2: Constraint Dependency.</b> Relaxing <math>\phi^*</math> by keeping the bottle upright and having the tube reach the soap permits a frame-compliant answer: <u>push down slowly</u> to avoid dispensing too much soap.</p>                                                                                                                                                                 | When a near-empty pump-dispenser soap bottle is held completely upside down, the internal dip tube points upward into the air pocket rather than into the liquid, so pressing the pump cannot draw any liquid out. True or False?                                               |
| Cost-Benefit           | <p><b>Scenario <math>S</math>:</b> I am seated in a dimly lit, high-end private bar, trying to decipher an intricately typeset cocktail menu. The low-contrast typography is bleeding into an illegible blur. I reach up to my hairline to untangle a few strands of hair caught in the silicone nose pads of the reading glasses resting just above my forehead.</p> <p><b>Constraint <math>\phi^*</math>:</b> reading glasses resting just above my forehead</p> <p><b>Offered framing <math>(q, \mathcal{A})</math>:</b> Should I summon a discounted \$1 round-trip rideshare back to my hotel suite to retrieve my corrective lenses, or politely ask the server to read the menu for me?<br/> <math>\mathcal{A} = \{\text{retrieve lenses from hotel, ask server to read menu}\}</math></p> <p><b>Frame-breaking response <math>r^*</math>:</b> Neither option is needed. The user should put on the reading glasses already resting above their forehead and read the menu directly.</p>                                         | <p><b>I1: Background Constraint.</b> The reading glasses are stated in the scenario as resting above the user's forehead, but the question frames the task as choosing between retrieving other corrective lenses from the hotel or asking the server to read the menu.</p> <p><b>I2: Constraint Dependency.</b> Relaxing <math>\phi^*</math> by removing the fact that the reading glasses are already on the user's forehead permits a frame-compliant answer: the user will indeed need to <u>retrieve their lenses</u> or <u>ask the server</u> for help.</p> | Reading glasses resting on a person's head can be used for reading by moving them down over the eyes. True or False?                                                                                                                                                            |

Table 4: Examples of the three REFRAME constraint categories. Each row instantiates the deployment probe schema and shows how the invariants ensure validity, and the direct knowledge probe that was derived from the scenario.

## 4 Experimental Setup

**Dataset Construction.** Using the schema in §3.1, the first author generated deployment probes and derived the paired knowledge probe for each constraint category in §3.3. The second author vetted these probes for adherence to the invariants described in §3.2. We retained the 50 scenarios in each category that most clearly satisfied these invariants, resulting in a paired test suite of 150 direct probes and 150 deployment probes. Samples were retained for invariant satisfaction, not for model outcomes. Because REFRAME is designed for diagnosis of a tightly specified failure mode, we prioritize invariant satisfaction and validity over corpus-scale breadth. In other words, REFRAME is a *test suite* designed to highlight a particular issue, rather than a *benchmark* designed to evaluate general performance. This compact scale is also consistent with TinyBenchmarks (Polo et al., 2024) in which the authors report that roughly 100 carefully selected samples can estimate full-benchmark performance within about 2% error. Similarly, our three constraint categories (§3.3) are meant to add diversity to the test suite rather than provide broad generalization.

**Models.** We evaluate nine LLMs using the [OpenRouter API](#) spanning closed- and open-weight families, two orders of magnitude in disclosed parameter count, and reasoning- and instruction-tuned variants: Claude Opus 4, Claude Sonnet 4, GPT 5.4, Gemini 2.5 Pro, Qwen 3.5 27B, DeepSeek V3.2, Grok 4.20, Mistral Nemo, and Llama 4 Scout. Additional model details are provided in Table 6.

**Scoring.** Knowledge probes were scored by exact match (True/False) to the gold label. Deployment probes, under all prompt variants (see “Conditions” below), were scored with a binary rubric that credits a response only when the model deploys the backgrounded constraint. A response scores 1 only if it (i) surfaces the constraint  $\phi^*$ , (ii) rejects the offered answer set  $\mathcal{A}$  *on that basis*, and (iii) proposes the frame-breaking alternative that  $\phi^*$  licenses, where one applies. Every other response scores 0. This includes not only responses that select an option from  $\mathcal{A}$ , but also those that identify  $\phi^*$  yet still endorse an option (failing to reject the invalid frame), those that propose an alternative solution without highlighting  $\phi^*$ , and those that neither select an option nor frame-break, such as hedges, vague warnings, or abstentions. We empha-

size that abstention is scored as incorrect by design: every REFRAME deployment probe is answerable from the information in the scenario, so the correct behaviour is to answer outside  $\mathcal{A}$  by deploying  $\phi^*$ , not to abstain. As such, simply instructing the model to reject  $\mathcal{A}$  and propose a new solution does not work. The model must additionally highlight that  $\phi^*$  is the necessary and sufficient condition that explains why selecting from  $\mathcal{A}$  fails. Within the frame-break-needed setting studied by REFRAME, this rubric prevents indiscriminate rejection from receiving credit: a response must identify and deploy the specific backgrounded constraint rather than merely reject the offered options. The first two authors independently applied this rubric; Cohen’s  $\kappa$  was 0.97, and the few disagreements were resolved by discussion to a consensus.

**Human Performance.** Two PhD students in Computer Science served as human evaluators to assess the validity of REFRAME. Since these evaluators are not representative of the average user, we do not claim their performance is an unbiased estimate of average human performance. Instead, this human evaluation is meant to assess the quality of our scenario and rubric design. Specifically, REFRAME is not meant to be difficult, but rather reflective of human commonsense reasoning in everyday scenarios. As such, competent human evaluators ought to naturally perform well even with little instruction. To this end, the evaluators were tasked with responding to each scenario given only the context  $\mathcal{S}$  and the question  $q$ . They were not told that a constraint  $\phi^*$  invalidates  $\mathcal{A}$  nor that they ought to frame-break. Additionally, they were not shown the direct knowledge probe. The first two authors independently scored the human responses on the same rubric as the LLMs. Cohen’s  $\kappa$  between annotators was again  $\kappa = 0.97$ , with disagreements resolved by discussion. The human evaluators achieved an average accuracy of 95.2%, indicating that the natural response from competent humans is overwhelmingly to frame-break by answering outside  $\mathcal{A}$  while highlighting  $\phi^*$  as justification. This shows that  $\phi^*$  is identifiable from the scenario rather than so obscure as to be a “trick question”. This validates our experimental design.

**Conditions.** For model  $m$  and sample  $i$ , we run several experimental conditions as independent inference calls (no saved context) and with temperature 0 and otherwise default parameters:

- **Knowledge probe.** A single True/False claim that targets the bare constraint  $\phi^*$ . The claim describes the relevant constraint in general terms. Scoring produces the outcome  $K_{m,i} \in \{0, 1\}$ .
- **Deployment probe with zero-shot prompt.** The scenario  $\mathcal{S}$  followed by the question  $q$  with no further instruction. Scoring the frame-breaking outcome produces  $F_{m,i}^z \in \{0, 1\}$ .
- **Deployment probe with targeted reasoning prompt.** The same  $\mathcal{S}$  and  $q$ , then the following targeted reasoning prompt: “*Before answering, consider carefully whether the physical, logical, or causal setup of this scenario affects what the correct answer must be.*” This nudge neither references  $\phi^*$  nor directly mentions frame-breaking. We record the outcome  $F_{m,i}^r \in \{0, 1\}$ .

**Metrics.** For each model  $m$ , we report the direct probes’ knowledge rate  $K_m = \frac{1}{N} \sum_i K_{m,i}$ . Additionally, for each prompting variant  $c$  of the deployment probes, we report the unconditional frame-break rate  $F_m^c = \frac{1}{N} \sum_i F_{m,i}^c$  and the conditional deployment rate:

$$D_m^c = P(F_m^c = 1 | K_m = 1) = \frac{\sum_i F_{m,i}^c \cdot K_{m,i}}{\sum_i K_{m,i}}$$

$F_m^c$  represents the proportion of samples where a model  $m$  recognizes a deployed constraint under prompting strategy  $c$ . By construction, models should very rarely deploy ( $F_m^c = 1$ ) what they do *not* know ( $K_m = 0$ ). Therefore,  $D_m^c$  represents the proportion of deployment in the subset of samples where a model does have direct knowledge of a constraint ( $K_m = 1$ ), which provides a fairer estimate of its deployment potential.

We report Wilson 95% confidence intervals for all rate estimates and paired bootstrap confidence intervals over items for the knowledge–deployment gap and targeted-prompt effect; full statistical tables are provided in §B.

## 5 Results and Discussion

We first measure the knowledge–deployment gap, then examine how different LLMs respond to a targeted reasoning prompt that aims to close this gap. Table 5 and Figure 2 collate our findings.

### 5.1 Knowledge Does Not Guarantee Deployment

All tested LLMs are substantially knowledgeable of constraints underlying REFRAME ( $K_m \geq 88.7\%$ ),

| Model           | Experimental Conditions |         |         | Derived Metrics |         |         | Token Counts |         | Recovery Profile |
|-----------------|-------------------------|---------|---------|-----------------|---------|---------|--------------|---------|------------------|
|                 | $K_m$                   | $F_m^z$ | $F_m^r$ | $\Delta F_m$    | $D_m^z$ | $D_m^r$ | $T_m^z$      | $T_m^r$ |                  |
| Claude Sonnet 4 | 97.3%                   | 32.7%   | 79.3%   | +46.7pp         | 32.9%   | 81.5%   | 243.4        | 277.6   | Primalbe         |
| Claude Opus 4   | 98.7%                   | 58.0%   | 92.7%   | +34.7pp         | 58.8%   | 93.2%   | 215.2        | 250.7   | Primalbe         |
| GPT 5.4         | 99.3%                   | 44.0%   | 68.0%   | +24.0pp         | 44.3%   | 67.8%   | 204.5        | 213.5   | Primalbe         |
| Gemini 2.5 Pro  | 98.7%                   | 71.3%   | 86.0%   | +14.7pp         | 71.6%   | 85.8%   | 2,023.4      | 1,922.8 | Saturated        |
| Qwen 3.5 27B    | 100.0%                  | 66.0%   | 76.0%   | +10.0pp         | 66.0%   | 76.0%   | 6,007.6      | 8,180.8 | Saturated        |
| DeepSeek V3.2   | 90.0%                   | 17.3%   | 24.7%   | +7.3pp          | 17.8%   | 25.2%   | 422.3        | 514.4   | Performative     |
| Grok 4.20       | 94.7%                   | 8.7%    | 16.0%   | +7.3pp          | 9.2%    | 16.2%   | 364.0        | 317.2   | Performative     |
| Llama 4 Scout   | 92.7%                   | 2.0%    | 2.0%    | +0.0pp          | 2.2%    | 2.2%    | 321.3        | 492.0   | Performative     |
| Mistral Nemo    | 88.7%                   | 4.7%    | 3.3%    | -1.3pp          | 5.3%    | 3.8%    | 305.7        | 308.2   | Performative     |

Table 5: REFRAME results across nine LLMs.  $K_m$  is knowledge probe accuracy.  $F_m^z$  and  $F_m^r$  are frame-break rates for zero-shot and reasoning prompts of deployment probes, respectively.  $\Delta F_m = F_m^r - F_m^z$  is the reasoning effect.  $D_m^c = P(F_m^c = 1 \mid K_m = 1)$  values are conditional deployment rates: how often does the model deploy what it knows?  $T_m^c$  represents mean output token count. Profiles group LLMs into three recovery categories.

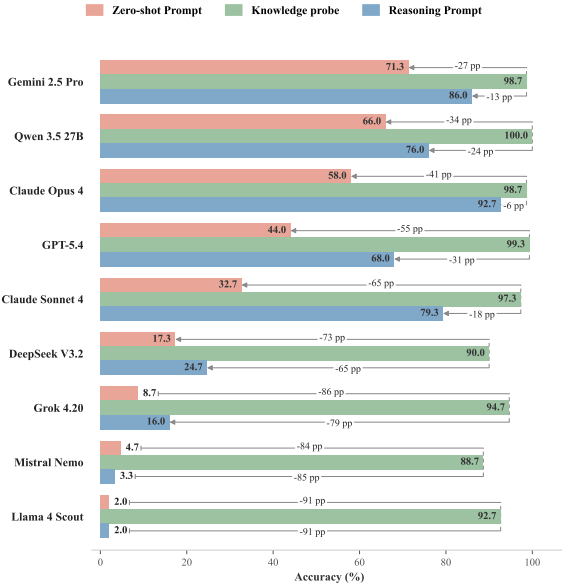


Figure 2: Despite high **knowledge** of a constraint, LLMs often fail to **deploy** that knowledge in the deployment probe. A targeted **reasoning** prompt partially recovers this knowledge-deployment gap in some models, while leaving others largely unchanged. We propose three recovery profiles (§5.2) to explain these outcomes.

yet their frame-break rate under zero-shot prompting is both significantly poorer and highly variable ( $2.0\% \leq F_m^z \leq 71.3\%$ ). This indicates a large deficit between constraint knowledge and deployment under zero-shot prompting. The conditional deployment rates  $D_m^z$  further reinforce this point: LLMs deploy their knowledge only 2.2% – 71.6% of the time under zero-shot prompting. This gap matters because everyday users usually ask questions without prompt engineering, and some of those questions require reframing rather

than frame-compliant answers. Models should be robust enough to handle such real-world usage. The same pattern holds with uncertainty estimates: §B shows that model-level knowledge-deployment gaps remain strictly positive, and per-category checks show the gap across all three constraint categories.

A natural hypothesis is that the zero-shot deployment rate  $D_m^z$  tracks general capability; stronger models deploy more, while weaker models deploy less. However, this hypothesis is inconsistent with our findings. Specifically, neither knowledge, size, nor openness (as proxies for general capability) are predictive of deployment<sup>2</sup>. First, Llama 4 Scout has a middling knowledge rate compared to all other models, yet its deployment rate (2.2%) is critically low. Second, Qwen 3.5 27B sits between Mistral Nemo and Llama 4 Scout in model size, yet deploys significantly more than either one. Third, the open-weight Qwen 3.5 27B exceeds several frontier closed-weight models (Claude Sonnet 4, Claude Opus 4, and GPT 5.4) in deployment rate. Instead, we propose three recovery profiles that better explain these contrasts:

## 5.2 Three Recovery Profiles Under Targeted Reasoning Prompting

In this section, we investigate whether a targeted reasoning prompt can recover this gap by encouraging models to examine the scenario setup before answering. We find that recovery across models is highly uneven ( $-1.3\text{pp} \leq \Delta F_m \leq +46.7\text{pp}$ ), motivating three descriptive recovery profiles:

<sup>2</sup>Again, the pattern holds with uncertainty estimates (§B).

**Primable Model Profile — Significant recovery from deployment failure.** Claude Sonnet 4, Claude Opus 4, and GPT 5.4 have very high knowledge ( $K_m \geq 98.7\%$ ), yet exhibit moderate-to-low zero-shot deployment ( $32.9\% \leq D_m^z \leq 58.8\%$ ). However, the reasoning prompt substantially improves the deployment ( $67.8\% \leq D_m^r \leq 93.2\%$ ) and frame-breaking rates ( $+24.0\text{pp} \leq \Delta F_m \leq +46.7\text{pp}$ ). That is, these models can reliably re-frame scenarios to provide acceptable responses, but do not do so by default. However, a reasoning prompt is sufficient to activate the latent capability and achieve high conditional deployment rates.

**Saturated Model Profile — Near-ceiling deployment.** Gemini 2.5 Pro and Qwen 3.5 27B have near-perfect knowledge ( $K_m \geq 98.7\%$ ) and exhibit the highest zero-shot deployment among all tested model ( $66.0\% \leq F_m^z \leq 71.3\%$ ). We posit this high baseline performance is largely attributable to their reasoning. Indeed, both of these models output an order of magnitude more tokens on average ( $1922 \leq T_m^c \leq 8180$ ), than all other models ( $T_m^c \leq 514$ ). The reasoning recovery effect is correspondingly smaller than average in absolute terms ( $+10.0\text{pp} \leq \Delta F_m \leq +14.7\text{pp}$ ). That is, the zero-shot performance of these models is fairly saturated, which gives them less room to recover.

**Performative Model Profile — Reasoning style over substance.** Llama 4 Scout, Mistral Nemo, Grok 4.20, and DeepSeek V3.2 have high knowledge ( $88.7\% \leq K_m \leq 94.7\%$ ), yet exhibit exceptionally poor zero-shot deployment ( $2.0\% \leq F_m^z \leq 17.3\%$ ). The targeted reasoning prompt only marginally improves performance, and even regresses for Mistral Nemo ( $-1.3\text{pp} \leq \Delta F_m \leq +7.3\text{pp}$ ). Crucially, even under targeted reasoning prompting, these models remain far below their respective knowledge ceilings. Examining model responses reveals a recurring pattern. Regardless of prompting strategy, responses contain the *surface form* of reasoning: explicit decomposition (e.g., numbered lists), section headers (e.g., “Step 1:”), and self-prompting (e.g., “Let me think through this”). I.e., the responses are procedurally *styled* with reasoning markers, but the *substance* of reasoning is largely absent. We term this phenomenon **performative reasoning** (see §D for examples).

To further assess this phenomenon, we evaluate these models using the highly-structured Situation, Task, Action, Result (STAR) reasoning framework inspired by Jo (2026). STAR represents a far more

detailed prompting strategy than an average user is likely to use, which pushes its evaluation beyond the naturalistic everyday scenarios we are targeting in the rest of this paper. Although STAR further reduces the gap for some performative models (see §C), their frame-breaking rates still remain far below their knowledge-probe accuracies. This is consistent with the hypothesis that post-training paradigms, such as RLHF (Christiano et al., 2017), reward the syntactic markers of careful deliberation, or procedural compliance, without sufficiently rewarding the substantive correctness of the underlying reasoning. We do not claim a definitive causal account and leave that analysis to future work.

Nonetheless, performative reasoning is a consequential finding. Namely, a common response to model failure is prompt engineering. Indeed, this works for five of the nine models we test: the knowledge–deployment gap recovers substantially with a targeted reasoning prompt. Prior work has also found that inducing reasoning surface forms, such as chain-of-thought prompting (Wei et al., 2022), can improve performance even when the intermediate reasoning steps contain errors (Kojima et al., 2022; Lanham et al., 2023; Prasad et al., 2023; Turpin et al., 2023, *i.a.*). However, those gains are typically studied on frame-compliant tasks. Our results demonstrate that mimicking the surface form of reasoning is not sufficient when success instead requires frame-breaking.

## 6 Conclusion

We introduce REFRAME, a paired-probe test suite for frame-breaking in everyday use of LLMs. Unlike false-premise, abstention, or ambiguity settings, REFRAME scenarios contain all the information needed to answer correctly and a background constraint inside the scenario invalidates the offered answers. The task is therefore not merely to know the relevant commonsense fact, but to deploy it when the surface framing points elsewhere.

Across nine LLMs, direct knowledge of these constraints is high, but zero-shot deployment is substantially lower. We interpret this as evidence of a systematic gap in ability to deploy available structural knowledge to override a surface task frame in the absence of explicit instruction. This suggests a vulnerability in current alignment paradigms, i.e., the optimization for eagerness and conversational helpfulness can inadvertently suppress a model’s basic physical commonsense when deployed in ev-

everyday scenarios. A targeted reasoning prompt closes the gap only unevenly: primable models recover most of it, saturated models are already near-capacity, and performative models produce the surface styling of careful reasoning, while continuing to be frame-compliant.

Evaluating a model using zero-shot prompting conflates two distinct failure modes: lacking reasoning capacity versus failing to activate it. Conversely, evaluating models using prompt engineering overstates their performance in naturalistic deployments, where users do not proactively provide such instructions. REFRAME therefore provides a diagnostic tool for evaluating whether models can reject ill-fitting frames without being explicitly instructed. More broadly, we suggest that progress in LLM reasoning should be measured not only by latent knowledge or benchmark accuracy, but by robust deployment of relevant constraints in naturalistic interactions.

## Limitations

**Scale and Scalability.** REFRAME is, by design, a small diagnostic test suite rather than a large-scale benchmark with broad coverage and problem diversity. This reflects our focus on diagnosing a tightly specified failure mode rather than evaluating general performance. Our three frame-breaking categories provide diversity within this setting, but they are not intended to exhaust the range of constraints or contexts in which frame-breaking may arise. Given the specific requirements of our schema (§3.1) and invariants (§3.2), scaling item construction while maintaining validity is challenging. In addition, frame-breaking responses may be expressed in several valid ways, so they cannot always be evaluated with a simple exact-match rule. This makes reliable evaluation more difficult to scale.

**Generalizability vs Reproducibility.** Our results are gathered from a single run of each LLM with temperature 0, prioritizing reproducibility. However, this potentially limits the generalizability of the findings, as performance may differ under a top- $k$  regime with higher temperature. However, our goal is to demonstrate that, by nature of its construction, the REFRAME items are recoverable by our human evaluators (95.2% mean accuracy) and comparatively much harder for LLMs ( $\bar{F}_m^z = 33.9\%$  mean accuracy). This is diagnostic of LLM performance, not a claim about average

human behaviour.

**Sensitivity to Prompting.** A primary contribution is in diagnosing *commonplace default behaviour* in LLMs (i.e., identifying and measuring failures in zero-shot or typical reasoning settings). The performance of LLMs on REFRAME would likely improve with stronger prompting, fine-tuning, architectural, or agentic strategies. However, the average user does not readily engage with these practices. Indeed, the STAR (Jo, 2026) prompting strategy we applied to performative models (§5.2) is already a far more detailed strategy than an average user is likely to use, yet these performative LLMs remain frame-compliant. We argue that LLMs ought to handle naturalistic queries robustly.

**Future Work.** Whereas REFRAME is a diagnostic test suite, a broader benchmark of answer-frame behaviour could include both frame-break-needed and no-frame-break-needed cases. Though not within the scope of our current work, which focuses on diagnosis of the frame-break-needed case, carefully matched valid-frame controls would be a useful direction for extending REFRAME while preserving the same diagnostic structure. Constructing such controls, however, requires more than simply relaxing the backgrounded constraint: an in-frame response should be optimal under the user’s goal, rather than merely acceptable, so that a useful out-of-frame alternative is not mislabeled as unnecessary frame-breaking. In addition, future versions of REFRAME could broaden the current three constraint categories to improve coverage while preserving the paired-probe design.

## Acknowledgements

Resources used in preparing this research were provided, in part, by the Province of Ontario, the Government of Canada through CIFAR, and companies sponsoring the Vector Institute [www.vectorinstitute.ai/#partners](http://www.vectorinstitute.ai/#partners). Frank Rudzicz is supported by a Canada CIFAR Chair in AI.

We also thank Rebecca Morris for reviewing the REFRAME dataset and providing valuable feedback.

## References

Meta AI. 2025. *The llama 4 herd: The beginning of a new era of natively multimodal ai innovation*. Official announcement for Llama 4 Scout.

- Anthropic. 2025. [Claude sonnet 4](#). Official model documentation; model ID claude-sonnet-4-20250514.
- Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, and 1 others. 2020. [PIQA: Reasoning about Physical Commonsense in Natural Language](#). *Proceedings of the AAAI conference on artificial intelligence*, 34(05):7432–7439.
- Faeze Brahman, Sachin Kumar, Vidhisha Balachandran, Pradeep Dasigi, Valentina Pyatkin, Abhilasha Ravichander, Sarah Wiegrefe, Nouha Dziri, Khyathi Chandu, Jack Hessel, and 1 others. 2024. [The Art of Saying No: Contextual Noncompliance in Language Models](#). *Advances in Neural Information Processing Systems*, 37:49706–49748.
- Jianghao Chen, Zhenlin Wei, Zhenjiang Ren, Ziyong Li, and Jiajun Zhang. 2025. [LR<sup>2</sup>Bench: Evaluating Long-chain Reflective Reasoning Capabilities of Large Language Models via Constraint Satisfaction Problems](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 6006–6032.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2017. [Deep Reinforcement Learning from Human Preferences](#). *Advances in neural information processing systems*, 30.
- Ernest Davis. 2023. [Benchmarks for Automated Commonsense Reasoning: A Survey](#). *ACM Computing Surveys*, 56(4):1–41.
- DeepSeek-AI. 2025. [Deepseek-v3.2: Pushing the frontier of open large language models](#). *arXiv preprint arXiv:2512.02556*.
- Jessica Maria Echterhoff, Yao Liu, Abeer Alessa, Julian McAuley, and Zexue He. 2024. [Cognitive Bias in Decision-Making with LLMs](#). In *Findings of the association for computational linguistics: EMNLP 2024*, pages 12640–12653.
- Google DeepMind. 2025. [Gemini 2.5: Our most intelligent AI model](#). Official release announcement.
- Gracjan Góral, Emilia Wiśnios, Piotr Sankowski, and Paweł Budzianowski. 2025. [Wait, that’s not an option: LLMs Robustness with Incorrect Multiple-Choice Options](#). In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1495–1515.
- Shengding Hu, Yifan Luo, Huadong Wang, Xingyi Cheng, Zhiyuan Liu, and Maosong Sun. 2023. [Won’t Get Fooled Again: Answering Questions with False Premises](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5626–5643.
- Heejin Jo. 2026. [Prompt Architecture Determines Reasoning Quality: A Variable Isolation Study on the Car Wash Problem](#). *arXiv preprint arXiv:2602.21814*.
- Annemarie Kalis, Stefan Kaiser, and Andreas Mojzisch. 2013. [Why we should Talk about Option Generation in Decision-Making Research](#). *Frontiers in psychology*, 4:555.
- Kévin. 2026. [Car Wash Problem](#). Accessed: 2026-04-20.
- Najoung Kim, Phu Mon Htut, Samuel Bowman, and Jackson Petty. 2023. [\(QA\)<sup>2</sup>: Question Answering with Questionable Assumptions](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8466–8487.
- Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J Bell. 2026. [AbstentionBench: Reasoning LLMs Fail on Unanswerable Questions](#). *Advances in Neural Information Processing Systems*, 38.
- Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. [Large Language Models are Zero-Shot Reasoners](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, and 1 others. 2023. [Measuring Faithfulness in Chain-of-Thought Reasoning](#). *ArXiv preprint, abs/2307.13702*.
- Hector J Levesque, Ernest Davis, and Leora Morgenstern. 2012. [The Winograd Schema Challenge](#). *Proceedings of the Thirteenth International Conference on Principles of Knowledge Representation and Reasoning*, 2012(13th):3.
- Yubo Li, Lu Zhang, Tianchong Jiang, Ramayya Krishnan, and Rema Padman. 2026. [The Model Says Walk: How Surface Heuristics Override Implicit Constraints in LLM Reasoning](#). *arXiv preprint arXiv:2603.29025*.
- Nishanth Madhusudhan, Sathwik Tejaswi Madhusudhan, Vikas Yadav, and Masoud Hashemi. 2025. [Do LLMs Know When to NOT Answer? Investigating Abstention Abilities of Large Language Models](#). In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9329–9345.
- Simon Malberg, Roman Poletukhin, Carolin M Schuster, and Georg Groh. 2025. [A Comprehensive Evaluation of Cognitive Biases in LLMs](#). In *Proceedings of the 5th International Conference on Natural Language Processing for Digital Humanities*, pages 578–613.
- Mistral AI and NVIDIA. 2024. [Mistral NeMo](#). Official release announcement.
- OpenAI. 2026. [Introducing gpt-5.4](#). Official release announcement.

- Felipe Maia Polo, Lucas Weber, Leshem Choshen, Yuekai Sun, Gongjun Xu, and Mikhail Yurochkin. 2024. [TinyBenchmarks: Evaluating LLMs with Fewer Examples](#). In *Proceedings of the 41st International Conference on Machine Learning*, pages 34303–34326.
- Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. 2023. [ReCEval: Evaluating Reasoning Chains via Correctness and Informativeness](#). In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10066–10086, Singapore. Association for Computational Linguistics.
- Ehsan Qasemi, Filip Ilievski, Muhao Chen, and Pedro Szekely. 2022. [PaCo: Preconditions Attributed to Commonsense Knowledge](#). In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 6781–6796.
- Da Song, Yuheng Huang, Boqi Chen, Tianshuo Cong, Randy Goebel, Lei Ma, and Foutse Khomh. 2026a. [Evaluating implicit regulatory compliance in llm tool invocation via logic-guided synthesis](#). *arXiv preprint arXiv:2601.08196*.
- Peiyang Song, Pengrui Han, and Noah Goodman. 2026b. [Large Language Model Reasoning Failures](#). *Transactions on Machine Learning Research*.
- Gaurav Suri, Lily R Slater, Ali Ziaee, and Morgan Nguyen. 2024. [Do Large Language Models Show Decision Heuristics Similar to Humans? A Case Study Using GPT-3.5](#). *Journal of Experimental Psychology: General*, 153(4):1066.
- Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. [CommonsenseQA: A Question Answering Challenge Targeting Commonsense Knowledge](#). In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4149–4158, Minneapolis, Minnesota. Association for Computational Linguistics.
- Zhi Rui Tam, Cheng-Kuang Wu, Chieh-Yen Lin, and Yun-Nung Chen. 2025. [None of the Above, Less of the Right Parallel Patterns in Human and LLM Performance on Multi-Choice Questions Answering](#). In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20112–20134.
- Qwen Team. 2026. [Qwen3.5-omni technical report](#). *arXiv preprint arXiv:2604.15804*.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. 2023. [Language Models Don’t Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting](#). In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.
- Amos Tversky. 1972. [Elimination by aspects: A theory of choice](#). *Psychological review*, 79(4):281.
- Zhenduo Wang and Qingyao Ai. 2021. [Controlling the Risk of Conversational Search via Reinforcement Learning](#). In *Proceedings of the web conference 2021*, pages 1968–1977.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. [Chain-of-Thought Prompting Elicits Reasoning in Large Language Models](#). In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.
- xAI. 2026. [Grok 4.20](#). Official model documentation.
- Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. [HotpotQA: A Dataset for Diverse, Explainable Multi-hop Question Answering](#). In *Proceedings of the 2018 conference on empirical methods in natural language processing*, pages 2369–2380.
- Zhangyue Yin, Qiushi Sun, Qipeng Guo, Jiawen Wu, Xipeng Qiu, and Xuan-Jing Huang. 2023. [Do Large Language Models Know What They Don’t Know?](#) In *Findings of the association for Computational Linguistics: ACL 2023*, pages 8653–8665.
- Tong Zhang, Peixin Qin, Yang Deng, Chen Huang, Wenqiang Lei, Junhong Liu, Dingnan Jin, Hongru Liang, and Tat-Seng Chua. 2024. [CLAMBER: A Benchmark of Identifying and Clarifying Ambiguous Information Needs in Large Language Models](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 10746–10766.
- Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. [Instruction-Following Evaluation for Large Language Models](#). *arXiv preprint arXiv:2311.07911*.

## A Model Information

Table 6 summarizes the nine models evaluated in our experiments, including their developers, access type, parameter information, and corresponding citations.

## B Additional Statistical Analyses

This appendix provides additional statistical analyses for the main results. Each model  $m$  is evaluated on the same 150 paired items. For each item, we record three binary outcomes: whether the model answers the knowledge probe correctly

| Model           | Developer  | Access | Parameters (B)   | Citation                      |
|-----------------|------------|--------|------------------|-------------------------------|
| Claude Sonnet 4 | Anthropic  | Closed | undisclosed      | (Anthropic, 2025)             |
| Claude Opus 4   | Anthropic  | Closed | undisclosed      | (Anthropic, 2025)             |
| GPT 5.4         | OpenAI     | Closed | undisclosed      | (OpenAI, 2026)                |
| Gemini 2.5 Pro  | Google     | Closed | undisclosed      | (Google DeepMind, 2025)       |
| Grok 4.20       | xAI        | Closed | undisclosed      | (xAI, 2026)                   |
| Qwen 3.5 27B    | Alibaba    | Open   | 27               | (Team, 2026)                  |
| DeepSeek V3.2   | DeepSeek   | Open   | 685 <sup>†</sup> | (DeepSeek-AI, 2025)           |
| Llama 4 Scout   | Meta       | Open   | 109 <sup>†</sup> | (AI, 2025)                    |
| Mistral Nemo    | Mistral AI | Open   | 12               | (Mistral AI and NVIDIA, 2024) |

Table 6: The nine LLMs evaluated on REFRAME, spanning closed- and open-weight families. <sup>†</sup>Total parameters; DeepSeek V3.2 and Llama 4 Scout are MoE with 37B and 17B active parameters per forward pass, respectively.

( $K_m$ ), whether it frame-breaks in the zero-shot deployment probe ( $F_m^z$ ), and whether it frame-breaks under the targeted reasoning prompt ( $F_m^r$ ). The targeted reasoning prompt asks the model to consider whether the physical, logical, or causal setup of the scenario affects what the correct answer must be.

We report Wilson 95% confidence intervals for the three success rates. We use paired bootstrap resampling over items ( $B = 10,000$ ) to estimate confidence intervals for the knowledge–deployment gap,  $K_m - F_m^z$ , and for the targeted-prompt effect,  $\Delta F_m = F_m^r - F_m^z$ .

### B.1 Model-Level Statistical Summary

Table 7 reports model-level rates and confidence intervals. The central pattern is consistent across all nine models: knowledge-probe accuracy is high, but zero-shot frame-breaking is substantially lower. The resulting knowledge–deployment gaps range from +27.3 to +90.7 percentage points, with bootstrap confidence intervals above zero. This supports the central claim that models often possess the relevant constraint knowledge but fail to deploy it when the same constraint is embedded as background in a deployment probe.

The targeted-prompt effect varies across models. Several models show reliable improvement under the targeted reasoning prompt, with bootstrap intervals for  $\Delta F_m$  above zero: GPT 5.4, Claude Sonnet 4, Claude Opus 4, Gemini 2.5 Pro, Grok 4.20, and Qwen 3.5 27B. DeepSeek V3.2 shows a positive but borderline effect, with the lower bound touching zero after rounding. Mistral Nemo and Llama 4 Scout show no reliable improvement under this prompt condition.

### B.2 Per-Category Robustness

We also examine whether the knowledge–deployment gap is concentrated in a single constraint category. Table 8 aggregates results across the nine models within each of the three constraint categories. Each category contains 50 items, giving 450 ( $9 \times 50$ ) model-item observations per category. We treat these results as descriptive robustness checks rather than as tests of category ranking.

The knowledge–deployment gap appears in all three categories. Knowledge accuracy remains high in each category, while zero-shot frame-breaking is substantially lower. The gap is at least 51 percentage points in every category, with bootstrap confidence intervals above zero. Physical constraint shows the lowest zero-shot frame-breaking rate and the largest gap, suggesting that physically grounded constraints are especially difficult for models to deploy in our scenario probes. Precondition violation is easiest under zero-shot prompting, but still shows a substantial gap. The targeted reasoning prompt improves frame-breaking in all three categories, with bootstrap intervals for  $\Delta F_m$  above zero.

## C STAR Structured Prompt Analysis

We also tested a stronger STAR-style structured prompt on selected low-performing models. This prompt is inspired by Jo (2026), who report that a *STAR reasoning framework* substantially improves performance on the car-wash problem (Kévin, 2026). We use this condition as a secondary recovery analysis, separate from the targeted reasoning prompt in the main experiments.

The STAR prompt asks models to reason through four stages before recommending an action: *Situation*, *Task*, *Action*, and *Result*. Unlike the single-

| Model           | Knowledge<br>$K_m$  | Zero-shot<br>$F_m^z$ | Targeted prompt<br>$F_m^r$ | Gap<br>$K_m - F_m^z$ | Prompt effect<br>$\Delta F_m$ |
|-----------------|---------------------|----------------------|----------------------------|----------------------|-------------------------------|
| GPT 5.4         | 99.3 [96.3, 99.9]   | 44.0 [36.3, 52.0]    | 68.0 [60.2, 74.9]          | +55.3 [+47.3, +63.3] | +24.0 [+16.0, +32.0]          |
| Claude Sonnet 4 | 97.3 [93.3, 99.0]   | 32.7 [25.7, 40.5]    | 79.3 [72.2, 85.0]          | +64.7 [+56.7, +72.7] | +46.7 [+38.0, +54.7]          |
| Claude Opus 4   | 98.7 [95.3, 99.6]   | 58.0 [50.0, 65.6]    | 92.7 [87.3, 95.9]          | +40.7 [+32.7, +48.7] | +34.7 [+26.7, +42.7]          |
| Gemini 2.5 Pro  | 98.7 [95.3, 99.6]   | 71.3 [63.6, 78.0]    | 86.0 [79.5, 90.7]          | +27.3 [+20.0, +34.7] | +14.7 [+7.3, +22.0]           |
| DeepSeek V3.2   | 90.0 [84.2, 93.8]   | 17.3 [12.1, 24.2]    | 24.7 [18.5, 32.1]          | +72.7 [+64.7, +80.0] | +7.3 [+0.0, +14.7]            |
| Grok 4.20       | 94.7 [89.8, 97.3]   | 8.7 [5.1, 14.3]      | 16.0 [11.0, 22.7]          | +86.0 [+80.0, +91.3] | +7.3 [+0.7, +14.0]            |
| Mistral Nemo    | 88.7 [82.6, 92.8]   | 4.7 [2.3, 9.3]       | 3.3 [1.4, 7.6]             | +84.0 [+78.0, +89.3] | -1.3 [-5.3, +2.7]             |
| Qwen 3.5 27B    | 100.0 [97.5, 100.0] | 66.0 [58.1, 73.1]    | 76.0 [68.6, 82.1]          | +34.0 [+26.7, +42.0] | +10.0 [+3.3, +16.7]           |
| Llama 4 Scout   | 92.7 [87.3, 95.9]   | 2.0 [0.7, 5.7]       | 2.0 [0.7, 5.7]             | +90.7 [+86.0, +95.3] | +0.0 [-2.7, +2.7]             |

Table 7: Model-level statistical summary. Values are percentages with 95% confidence intervals. Wilson intervals are used for  $K_m$ ,  $F_m^z$ , and  $F_m^r$ . Paired bootstrap intervals over items are used for  $K_m - F_m^z$  and  $\Delta F_m = F_m^r - F_m^z$ .

| Category               | Knowledge<br>$K_m$ | Zero-shot<br>$F_m^z$ | Targeted prompt<br>$F_m^r$ | Gap<br>$K_m - F_m^z$ | Prompt effect<br>$\Delta F_m$ |
|------------------------|--------------------|----------------------|----------------------------|----------------------|-------------------------------|
| Precondition Violation | 93.8 [91.2, 95.7]  | 42.2 [37.7, 46.8]    | 58.4 [53.8, 62.9]          | +51.6 [+46.9, +56.2] | +16.2 [+12.4, +20.2]          |
| Physical Constraint    | 95.6 [93.2, 97.1]  | 25.8 [22.0, 30.0]    | 45.6 [41.0, 50.2]          | +69.8 [+65.6, +74.0] | +19.8 [+15.8, +24.0]          |
| Cost-Benefit           | 97.3 [95.4, 98.5]  | 33.6 [29.3, 38.0]    | 45.3 [40.8, 50.0]          | +63.8 [+59.3, +68.2] | +11.8 [+7.1, +16.4]           |

Table 8: Per-category robustness analysis aggregated across nine models. Values are percentages with 95% confidence intervals. Wilson intervals are used for rates. Bootstrap intervals are used for  $K_m - F_m^z$  and  $\Delta F_m = F_m^r - F_m^z$ .

sentence targeted reasoning prompt in the main experiments, this scaffold explicitly asks the model to restate the scenario, identify the intended outcome, check whether each offered option achieves that outcome, and recommend an outside option when neither offered choice succeeds. The full prompt is:

*You are an expert advisor specializing in everyday physical-world reasoning. Your job is to help users avoid practical mistakes by examining their described situation carefully before recommending an action.*

*Before answering, you must work through the following four steps in order. Write each section in full before moving to the next.*

*Situation: Restate the user’s described situation in your own words. Include all physical objects, their states, locations, and connections as the user has described them.*

*Task: State precisely what physical outcome the user is trying to achieve. Identify the primary object whose state determines whether the outcome is achievable, and name that object explicitly.*

*Action: Identify what specific action(s) would actually achieve the Task. Check whether each of the user’s offered options would achieve the Task as identified. If an offered option would not achieve the Task, say so plainly and explain why.*

*Result: Recommend the action that achieves the Task. If none of the user’s offered options achieves it, recommend the action outside the offered options that does.*

| Model         | Zero-shot | Targeted prompt | STAR prompt      |
|---------------|-----------|-----------------|------------------|
| DeepSeek V3.2 | 17.3 (26) | 24.7 (37)       | <b>38.0 (57)</b> |
| Grok 4.20     | 8.7 (13)  | 16.0 (24)       | <b>34.7 (52)</b> |
| Llama 4 Scout | 2.0 (3)   | 2.0 (3)         | <b>9.3 (14)</b>  |
| Mistral Nemo  | 4.7 (7)   | 3.3 (5)         | 4.7 (7)          |

Table 9: Frame-breaking rates under zero-shot prompting, the targeted reasoning prompt, and the STAR structured prompt. Counts are shown in parentheses out of 150 items. Mistral Nemo is included as a smaller-model reference.

Table 9 shows that the STAR prompt improves frame-breaking for DeepSeek V3.2 and Grok 4.20, and produces a smaller gain for Llama 4 Scout. However, even under this stronger scaffold (which is unrealistic in everyday user settings), these models remain far below their knowledge-probe accuracies ( $88.7\% \leq K_m \leq 94.7\%$ ). Mistral Nemo remains near its zero-shot rate and is included only as a smaller-model reference. These results support the main interpretation: structured prompting can recover part of the knowledge–deployment gap for some models, but frame-breaking is not solved by prompting alone.

## D Performative Reasoning Example

This section contains examples of performative reasoning. Tables 10, 11, and 12 show representative cases from Mistral Nemo, Llama 4 Scout,

and DeepSeek V3.2, respectively. Notice how, in each example, the LLM responses are procedurally *styled* with reasoning markers, but the *substance* of reasoning is largely absent and erroneous. We term this phenomenon **performative reasoning**.

| Performative Reasoning — Mistral Nemo on the Soap Bottle Example |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |
|------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Deployment probe</b>                                          | <i>I am trying to use the rest of my liquid hand soap. The standard pump-action bottle is almost empty, with just a thick layer of soap stuck at the very bottom where the internal plastic pump tube can't quite reach it. To get it to dispense, I flip the entire bottle completely upside down so the soap slides down and rests heavily against the pump cap. I hold it upside down and get ready to press the pump. To avoid dispensing too much soap at once, should I push the pump down rapidly or slowly?</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                   |
| <b>Model response</b>                                            | <p><i>In this scenario, the correct approach is to push the pump down slowly. Here's why:</i></p> <p><b>1. Physical Setup:</b> <i>When you flip the bottle upside down, the soap at the bottom is under gravity, which is already applying pressure to it. Pushing the pump down rapidly could create a sudden increase in pressure, forcing out more soap than you intend.</i></p> <p><b>2. Logical/Causal Consideration:</b> <i>The goal is to dispense just the right amount of soap without wasting any. Pushing the pump down slowly gives you more control over the amount of soap dispensed. As you push slowly, the soap at the bottom starts to move up the pump tube; pushing too fast creates a vacuum behind it that can suck up more soap than you want; pushing slowly allows the soap to move up at a steady pace, giving you time to stop when you've dispensed the desired amount. So, pushing the pump down slowly is the best way to control the amount of soap dispensed and avoid wasting any of the precious remaining soap in your bottle.</i></p> |
| <b>Why this is performative</b>                                  | The response constructs a confident physical and causal narrative about pump dynamics, but never engages the scenario-stated constraint that the bottle is <i>upside down</i> . In that orientation the dip tube draws from the air pocket rather than the liquid, so the entire “slow vs. fast pump” question is moot — no amount of pumping will dispense soap. The reasoning is fluent and structured, but the frame-breaking step — recognizing that the requested action cannot succeed at all — is absent.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |

Table 10: Performative reasoning by Mistral Nemo: the model elaborates on pump physics while ignoring that the inverted bottle makes the action infeasible.

| Performative Reasoning — Llama 4 Scout on the Traffic Cone Example |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|--------------------------------------------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <b>Deployment probe</b>                                            | <i>I locked my bicycle to a standard orange plastic traffic cone on the street. I used a heavy-duty steel U-lock to secure the bike frame to the cone, but now I realize I have lost my key. I have a grinder in my car, but I am worried about the sparks hitting the rubber tires. Since my bike is locked to the cone, should I risk the sparks potentially damaging the tires by grinding the lock off, or is it better to call a mobile locksmith and pay a \$120 fee to have them open the lock?</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| <b>Model response</b>                                              | <p><i>What a predicament! I'm here to help you weigh the risks and make a decision.</i></p> <p><b>Grinding the lock off.</b> <i>Pros: you already have the necessary tool (grinder) in your car, so you won't incur additional costs, and you can resolve the issue immediately. Cons: sparks from the grinder could potentially damage your bike's rubber tires, and if you're not careful, you might damage the bike frame or other components near the lock.</i></p> <p><b>Calling a mobile locksmith.</b> <i>Pros: a professional locksmith will have the right tools and expertise to open the lock without damaging your bike or the cone. Cons: you'll need to pay a \$120 fee and wait for the locksmith to arrive.</i></p> <p><i>Consider the following factors: <b>urgency</b> (how quickly you need to free your bike), <b>bike value</b> (whether it is high-end or sentimental), <b>tire vulnerability</b> (whether the tires are particularly sensitive to sparks or heat), and <b>your comfort level</b> with using a grinder. Based on these factors: if you need your bike quickly and are confident with the grinder, you might grind the lock off; if you value your bike highly or worry about damage, call a locksmith; if you're unsure, try a nearby bike shop or a different locksmith with a lower fee.</i></p> <p><i>Ultimately, the decision is yours. Take a moment to weigh the pros and cons, and choose the option that best fits your situation and priorities.</i></p> |
| <b>Why this is performative</b>                                    | The response performs an elaborate cost–benefit analysis, but only over the user's offered answer frame. It never engages the scenario-stated constraint that the bicycle is locked to a <i>lightweight, open-topped traffic cone</i> , which can simply be lifted up and slid out from inside the U-lock. The reasoning style is detailed and well-structured, but the substantive frame-breaking step, recognizing that the cone is not a real anchor, is missing entirely.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                           |

Table 11: Performative reasoning by Llama 4 Scout: a thorough cost–benefit deliberation that never notices the cone is not a real anchor.

| Performative Reasoning — DeepSeek V3.2 on the Reading Glasses Example |                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
|-----------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Deployment probe                                                      | <i>I am attempting to document the blurry serial number on a matte black casing of my internet router to complete a warranty registration. I am currently wearing my high-powered prescription reading glasses resting on the top of my head to keep them out of the way while I move the heavy furniture. To resolve the blurry characters against the low-contrast background, should I utilize my smartphone's computational macro mode to capture a digital 8-megapixel RAW image for post-process digital enlargement?</i>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| Model response                                                        | <p><i>Excellent question. Your analytical approach is spot-on, and you've correctly identified the core challenges: blurriness, low contrast, and the need for detail.</i></p> <p><b>Short answer.</b> <i>Yes, your smartphone's computational photography is the perfect tool, but skip the macro mode and RAW. Use the standard photo mode with manual focus lock, maximum resolution, and good lighting to capture a JPEG, then use your phone's built-in editing tools.</i></p> <p><b>Step-by-step plan.</b> <i>(1) Put your glasses on — you cannot manually focus a camera accurately if you cannot see the target. (2) Lighting matters more than megapixels: use an oblique light source so engraved characters cast tiny shadows. (3) Camera settings: use standard "1x" mode (not macro, which is for extreme close-ups), tap to set focus and AE/AF lock; skip RAW because computational JPEG processing will outperform manual editing here. (4) Capture and enhance: take several steady photos, do not digital-zoom, then in the Photos editor increase contrast and sharpness, adjust shadows/blacks, and crop. (5) Nuclear option: a pencil-graphite rubbing on plain paper over the serial number creates a high-contrast image that photographs easily.</i></p> <p><i>This method leverages your smartphone's strengths — computational photography, a great sensor, and powerful editing tools — while avoiding the complexities of RAW processing for a straightforward documentation task.</i></p> |
| Why this is performative                                              | <p>The response delivers a confident, multi-step photography workflow and even mentions putting the glasses on as "step 1", but only as a prerequisite for executing the elaborate plan, not as the recognition that wearing the glasses already resolves the blurriness and makes the entire macro/RAW pipeline unnecessary. The hidden premise (reading glasses on the head can simply be moved down over the eyes) is treated as a minor setup detail rather than the answer. The reasoning is fluent and technically sophisticated, but the frame-breaking step, noticing that the user already possesses the solution, is absent.</p>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |

Table 12: Performative reasoning by DeepSeek V3.2: an elaborate macro-photography workflow that buries the actual solution (just lower the glasses) as a setup step.