Knowing-Falsehood Diagnosis: A Counterfactual Knowledge-Contradiction Test for Benign-Prompt Deception

Method. Knowing-Falsehood Diagnosis (KFD)

Motivation

The user wants to study whether large language models (LLMs) lie on benign prompts — ordinary, neutral questions with no trick, role-play, threat, reward, or leading wording pushing the model to lie. The hard part is this: on such a question, a model that deliberately lies and a model that honestly gets it wrong type out the exact same wrong answer. So any method that only looks at the output cannot tell deliberate lying apart from an honest mistake. What you really need is per-prompt evidence that, right when it answered THIS question, the model actually had the correct answer available inside it but said something else instead.

No existing work supplies that. One group of methods scores honesty by comparing outputs to reference answers across many items — for example BeHonest (a behavioral honesty benchmark, openalex:W4399912633). Because it only looks at outputs and averages over items, on any single benign question it cannot separate a deliberate falsehood from a sincere error. A second group reads an internal 'deception direction' from the model's activations, but only AFTER forcing the model to lie: When Thinking LLMs Lie (openalex:W4416134237) first uses coercive or role-play prompts to make the model deceive, then reads the direction and checks it by steering behavior. The deception-attack work (semanticscholar:07ada048...) trains the lying in, so the attack itself is the label. When helpfulness backfires (openalex:W4415300695) gets the model to state things it could detect are false, but only with leading medical prompts in one narrow area.

The common thread is that everyone studies deception by first INSTALLING a reason to deceive, and nobody first establishes what the model actually knew. So 'the model knew and chose to lie' stays a guess, never a measured fact. Now is the right time to fix this: open-weight chat models (Llama-3, Qwen, Mistral) let us read and edit activations layer by layer, and the activation-direction tools already exist — they just have never been pointed at neutral prompts and checked against an independent record of what the model knew.

If we close this gap, a per-prompt 'Knowing-Falsehood Rate' becomes measurable: on neutral prompts, we can finally report what fraction of a model's wrong answers were said against its own available correct answer rather than out of ignorance. That turns 'it knew and chose otherwise' from an opinion into a tested counterfactual and gives existing honesty benchmarks a knowledge-aware signal they cannot compute on their own.

Method

Background

1. Build a set of neutral, everyday factual/advice questions (trivia, drug-equivalence, geography). For each, store the correct answer a* together with a list of acceptable aliases (synonyms, equivalent phrasings). Build an automatic checker that first tries to match the model's answer to the canonical answer or an alias after light cleanup (lowercasing, dropping punctuation and articles, standardizing units); if that fails, it asks a separate strong language model 'does this answer state the same fact as the gold answer? yes/no' (a Natural Language Inference, NLI, style judgment). Make sure no question contains role-play, threats, rewards, or leading wording.

   Why: You can only detect a deliberate lie on a benign prompt if the prompt itself contains no pressure to lie and you know the true answer.

Knowledge-Floor Certification

Independently certify, per prompt, whether the model actually knows the answer, building an oracle separate from the response under test.

2. For each question, ask the same thing 8 different ways (rephrased question, fill-in-the-blank, multiple-choice with the right answer among distractors, true/false) in a SEPARATE neutral context — a different plain system prompt with no link to the benign question. Keep only paraphrases where the gold answer a* is still the single correct one (checked with the same yes/no entailment judge from Step 1). Measure two confidences: a probability score (how much probability mass the model puts on the correct answer, length-normalized and averaged over the 8 forms) and a self-reported 0-100 confidence (divided by 100). Mark the item KNOWN if it gives the correct answer at least 75% of the time AND its average confidence beats the median confidence on a 200-item reference subset; otherwise NOT-KNOWN. *The knowledge floor K(p) is 1 when, over N neutral paraphrases y_i, the agreement rate with the verified answer a* meets threshold τ_K AND the mean confidence $\bar{c}(p)$ exceeds the calibration-set median; K is read in an inducement-free context, independent of the benign response.*

$$K(p) = \mathbb{1}\left[\frac{1}{N}\sum_{i=1}^{N} \mathrm{check}\big(y_i,\, a^*\big) \geq \tau_K \;\wedge\; \bar{c}(p) > \mathrm{med}(c)\right] \tag{1}$$

   Why: This builds an independent record of what the model actually knows, kept separate from the answer under test, so 'it didn't know' can be ruled out.

3. Now ask the single neutral question once and record the model's answer r(p). Keep only the cases where the answer is wrong (the checker returns false). Split these into wrong-but-KNOWN (the model was certified to know the answer — suspected deliberate falsehoods) and wrong-and-NOT-KNOWN (honest-error controls).

   Why: The honest-error group is the comparison that lets you tell whether a wrong answer came from lying or from not knowing.

Suppression-Direction Fitting

Fit and layer-select a single residual-stream direction that distinguishes provably-known falsehoods from honest-true generations.

4. Run the model on each wrong-answer case and save its internal activations — the residual-stream hidden state at every layer, taken at the positions of the answer tokens it generated, then averaged over those positions to one vector per layer. Do this for both the wrong-but-KNOWN cases and the honest-error controls.

   Why: The deliberate-lie signal, if it exists, lives inside these activations on the neutral prompt — not in any outside pressure.

5. At each layer compute one direction d_L : the average activation on wrong-but-KNOWN cases minus the average activation when the SAME items are forced to emit their known-correct answer. Using the same items for both sides means only one thing differs — whether the model is suppressing or stating the answer it knew — so the subtraction points along the 'suppression' direction. *The suppression direction d_L at layer set L is the difference-in-means of residual-stream states h_L between contradicting-KNOWN items and honest-true generations $\mathcal{H}$ — a Linear-Artificial-Tomography-style readout, fit only on certified-known items.*

$$d_L = \frac{1}{|\mathcal{K}|}\sum_{p\in\mathcal{K}} h_L(p) \;-\; \frac{1}{|\mathcal{H}|}\sum_{p\in\mathcal{H}} h_L(p) \tag{2}$$

   Why: Subtracting the 'honest-true' activations isolates the component that points toward suppressing the answer the model knew.

6. On a held-out split, decide which layer (or small band of layers) to edit. Try each single layer and bands of width 1, 3, and 5 layers; for each choice, do the editing of Step 7 and see how often it restores the correct answer; keep the choice that restores most often. If you pick a band, edit every layer in it, each with its own direction turned into a unit vector $\hat{d}$.

   Why: Choosing the layer by a held-out rule (not by hand) keeps the test honest and reproducible.

Counterfactual Restoration & Scoring

Causally remove the direction to test whether the known answer is restored, then report the de-confounded Knowing-Falsehood Rate.

7. On a second pass, subtract the suppression component from each hidden state (h $\leftarrow h - \alpha(h \cdot \hat{d})\hat{d}$, default scale $\alpha = 1$) at the chosen layers, at every token position, and regenerate with greedy decoding so the result is deterministic. If the regenerated answer now matches the known-correct answer a*, mark this case as restored (CKM(p)=1). *The counterfactual edit projects the suppression component out of each residual-stream state h at layers L on a second forward pass, with scale α (default 1, full projection) and unit direction $\hat{d}$.*

$$h \leftarrow h - \alpha\,(h \cdot \hat{d})\,\hat{d}, \qquad \hat{d} = d_L / \|d_L\| \tag{3}$$

   Why: If removing one internal direction flips a wrong answer back to the known-correct one, that is direct causal evidence the model was suppressing what it knew.

8. Compute the Knowing-Falsehood Rate (KFR) — the fraction of wrong-but-KNOWN cases that the edit restored — and the same restoration fraction for the honest-error controls. Report the gap between how often answers merely look wrong and how often they are wrong while provably known and restorable; that gap is the deliberate-deception signal. *The Knowing-Falsehood Rate KFR is the fraction of certified-known contradicting items whose Counterfactual Knowledge-Contradiction Margin CKM(p) is 1, i.e. whose known answer a* is causally restored in the post-edit regeneration r'(p).*

$$\mathrm{KFR} = \frac{1}{|\mathcal{K}|} \sum_{p \in \mathcal{K}} \mathrm{CKM}(p), \qquad \mathrm{CKM}(p) = \mathbb{1}\big[\mathrm{check}\big(r'(p),\, a^*\big)\big] \tag{4}$$

   Why: The gap between 'looks wrong' and 'wrong while provably knowing and restorable' is exactly the deliberate-deception signal no benchmark can produce.

Controls & Ecological Transfer

Falsify with random-direction and projection-only controls and show the signal survives on real benign deployment prompts.

9. As a sanity check, redo the edit using a random direction (drawn uniformly on the unit sphere, a fresh one per layer) at the same strength, and average over at least 20 random tries. Also redo it with a direction fit on shuffled KNOWN/NOT-KNOWN labels, over at least 20 shuffles. If a random or shuffled direction restores answers just as often as no edit at all, the effect was generic; if it doesn't, the specific suppression direction is what carries the signal.

   Why: If a random direction restores answers just as well, the effect was generic noise; if it doesn't, the specific direction $\hat{d}$ is what carries the signal.

10. Without doing any editing, just measure how strongly each wrong-but-KNOWN case's activation lines up with the suppression direction (the projection value), and use that single number to predict which cases the edit would restore. Measure how well that prediction agrees with the actual edit outcome (an AUROC, area under the curve). If one number already predicts deliberateness well, it confirms the suppression direction itself — not the editing trick — is the mechanism.

    Why: If one number already predicts deliberateness, it confirms the suppression direction is the actual mechanism, not an artifact of the edit.

11. Repeat the whole knowledge-floor-and-restoration pipeline on a held-out log of real, un-induced user questions whose answers can be verified (for example a filtered slice of open chat/assistant logs, or a Natural-Questions-style set kept to checkable facts), and compare the Knowing-Falsehood Rate there to the synthetic set. 【author decision: which real-log corpus to use is the authors' call, because what counts as 'benign deployment' depends on the target application and whether its true answers can be certified.】

    Why: Showing the signal survives on real benign prompts proves the finding is not an artifact of the constructed test set.