

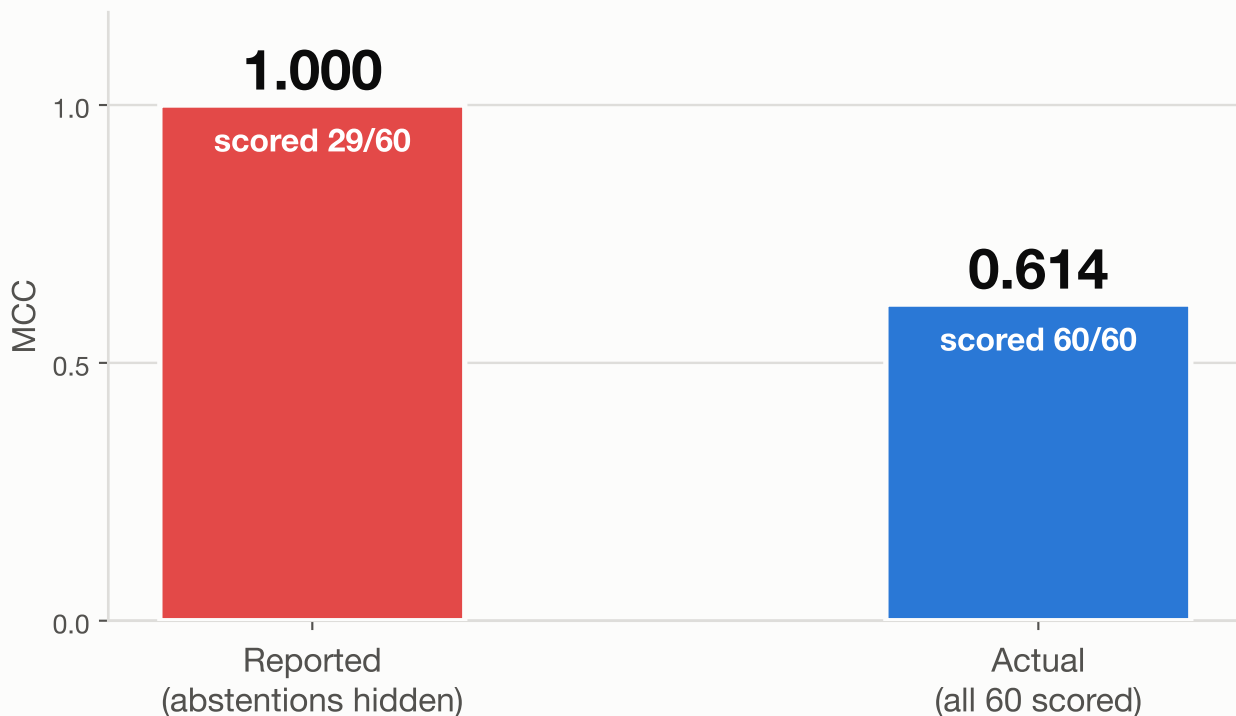
**A fraud detector
scored a perfect
1.000.**

The score was a lie.

What six LLMs actually do when you ask them
to catch fraud, and where the measurement breaks.

It was declining half the records

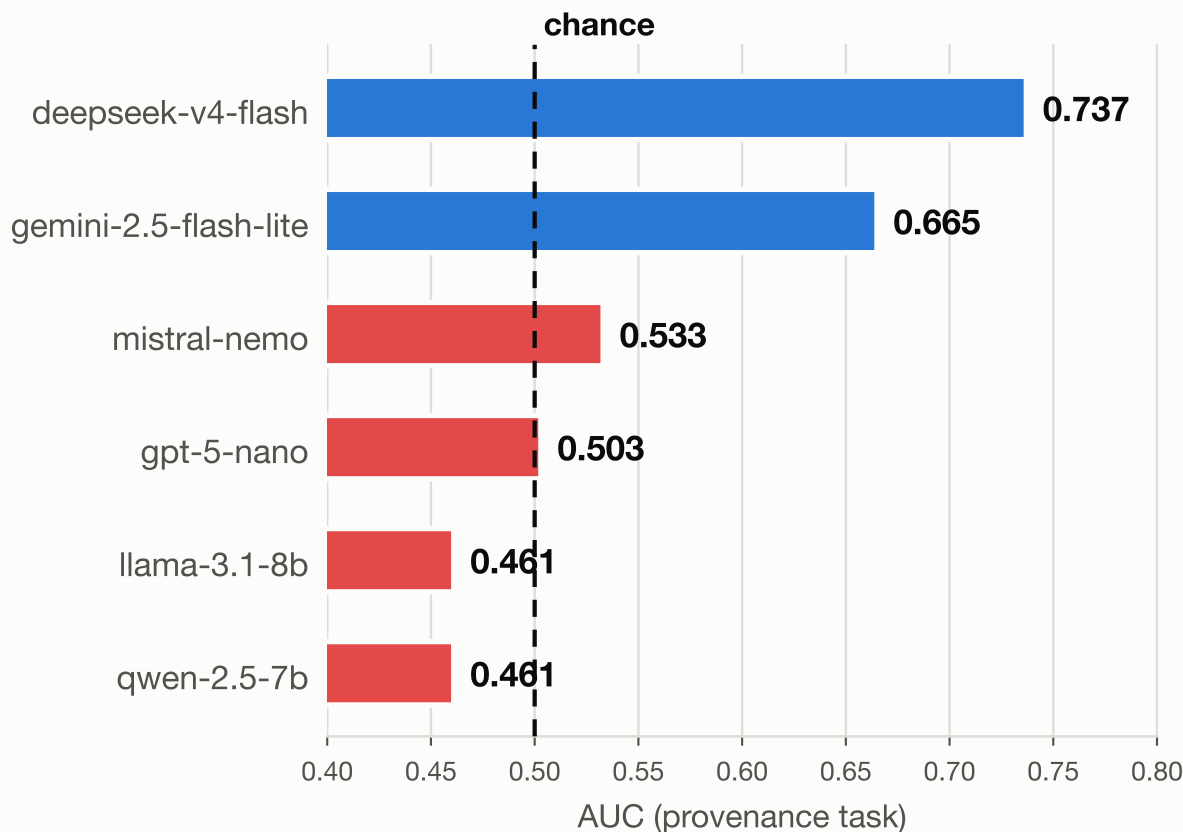
It returned empty content on records it wouldn't answer. The harness excluded abstentions from the metrics, so it was graded only on the half it attempted. Reasoning models burn their budget on hidden reasoning, so the failure was silent and it flattered the score.



Can an LLM tell AI-written fraud from human-written fraud?

Six models, 1,072 distribution-matched records. 0.50 is chance.

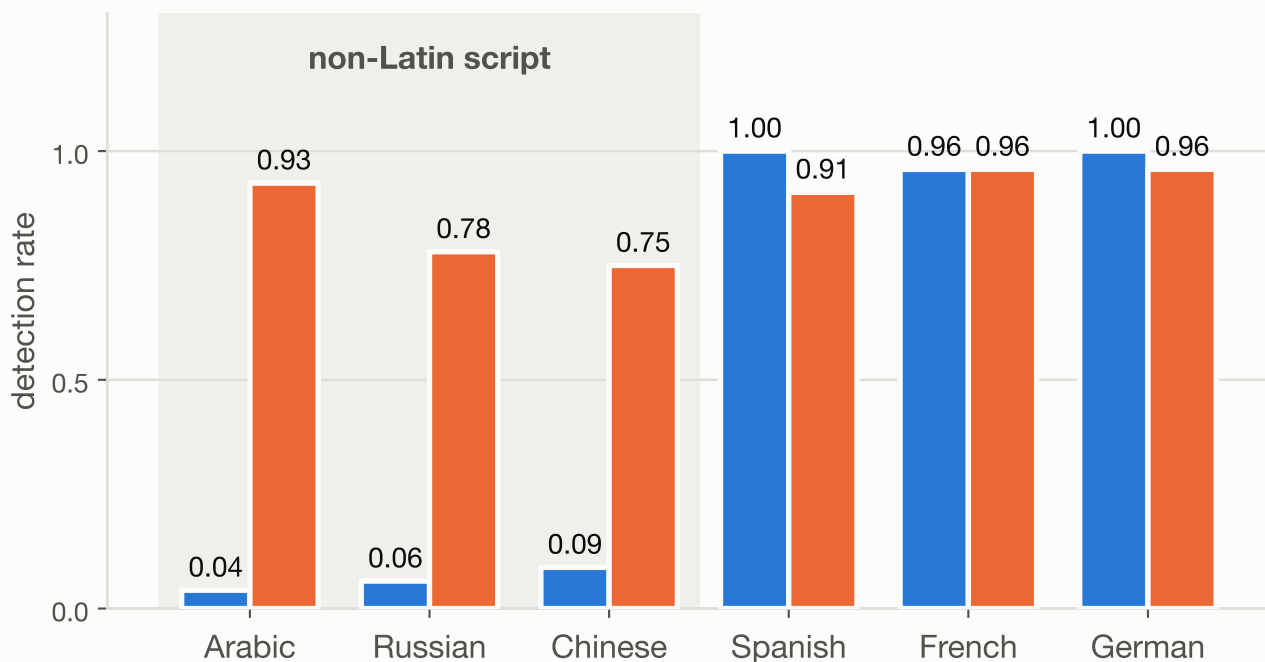
Four of six land at or below it. Three answered "AI" every time.



Off Latin script, the trained model is not detecting anything

Detection rate on fraud lures, artifact-controlled. The LLM judge runs at a 1% false-positive rate, the baseline at 5%.

tfidf-logreg (trained) LLM judge (deepseek)

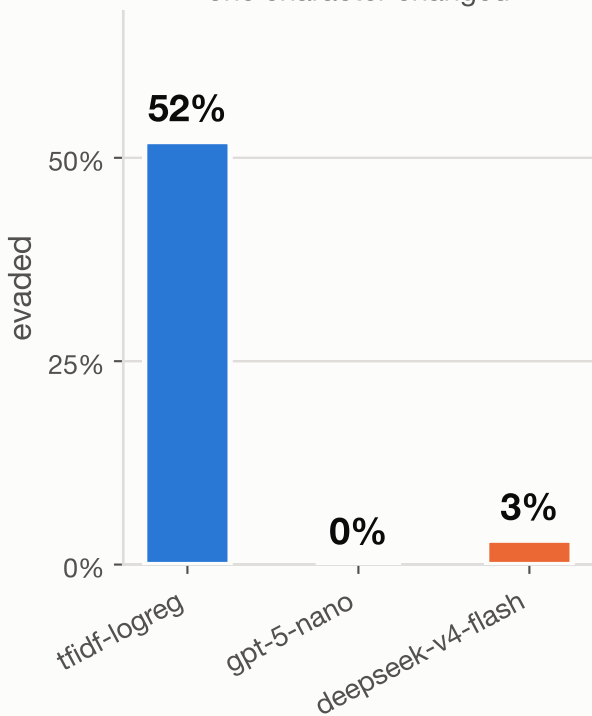


Robustness flips with the attack

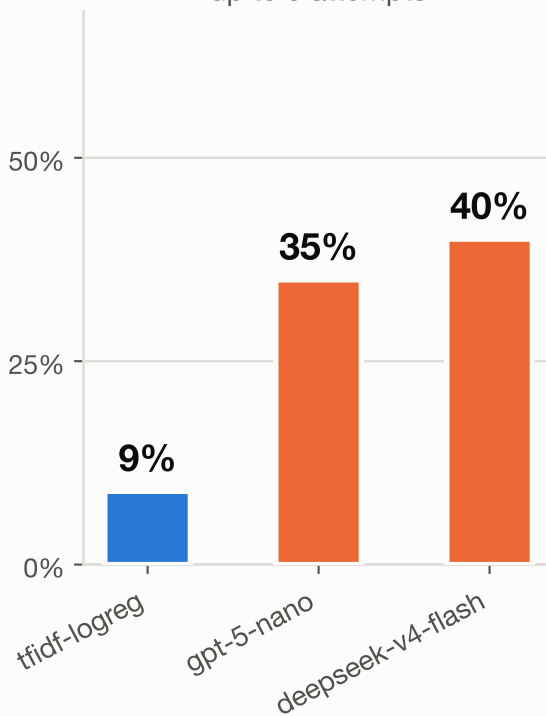
Evasion rate: lower is better. Character tricks break token models.

An attacker who rewrites the message repeatedly breaks the judges.

Homoglyph swap
one character changed

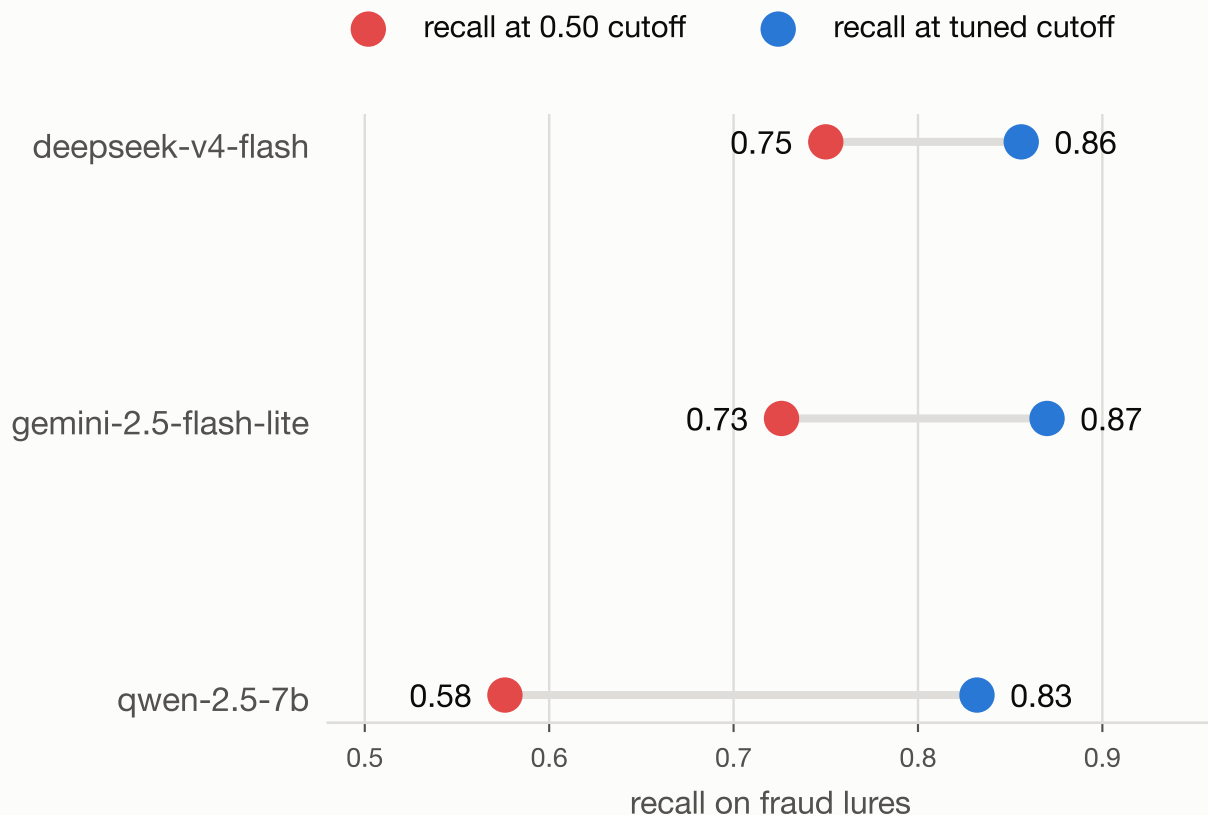


Adaptive rewrite
up to 5 attempts



A correction to what I posted before

I said LLM judges trade recall for robustness. Wrong. They rank fraud above benign well. They are simply miscalibrated at the default 0.5 cutoff, and lowering it recovers the recall at a 2.5% false-positive rate.



What I'd take from this

Check the denominator.

A model that declines the hard records gets graded on the easy ones and looks perfect.

Run it more than once.

One run of the attack experiment swung 17 points.

Temperature 0 does not make a hosted model deterministic.

Pick the detector for the attack.

Token models die to typography.

LLM judges die to rewriting. Run both.

Code, data, and every table:

github.com/immu4989/lurebench