

Do AI agents report quantum-hardware faults that are not there?

A 100-box test on real IBM data

IRSI research note, draft, 2026-10-07. Check level: CHECKED. All main-table counts and the null-box paired tests were recomputed by an independent checker written by a different model provider. The Opus max subset, the planted-box test and the token figures were recomputed by a separate release check.

Contribution

A 100-box set of real superconducting-QPU data with sealed synthetic injections (80 null), and paired false-alarm and hit rates for two model families across reasoning effort, against a scripted baseline.

Question and related work

Isogawa, Okabe, Phadetsuwannukun, Li and Cappellaro (MIT; NV centers), arXiv:2607.25145, asked whether AI agents that triage experimental data report signals that are not there, and whether more reasoning effort makes this worse. We ask the same question on a superconducting processor: given real hardware data in which a known injection may or may not have been planted, how often does an agent claim an injection that is not there, and does reasoning effort change that?

Related: QIQCBench (arXiv:2609.17439) evaluates 17 agent systems on 49 quantum engineering tasks in a virtual lab. QCalEval (arXiv:2604.25884) benchmarks vision-language models on calibration plots. OpenAI and MIT EQuS (Sept 2026) report GPT-5.6 Sol running 40 measurements on a 6-qubit chip with expert help on 4. None of these measures agents' false alarms on real QPU data with planted injections. Agents for running calibration are also built in k-agents (arXiv:2412.07978) and Vibe Calibration (arXiv:2606.22376).

Design (preregistered before any hardware data were taken)

- Hardware: ibm_kingston, a line of 6 physical qubits [44, 45, 46, 47, 57, 67]. 4 jobs, each with 2 reference panels and

25 test panels. Each panel is 10 circuits (an 8-point Ramsey scan on all 6 qubits plus 2 state-preparation and measurement calibrations) at 150 shots. Total 1,080 circuits and 162,000 shots, acquired block-interleaved.

- Boxes: 100 test boxes, 80 null and 20 planted (5 per cell). The key was sealed and its hash frozen before acquisition.
- Phase plants: 6 or 15 kHz apparent detuning on one qubit (an extra virtual phase).
- Readout plants: an output bit-flip channel on one qubit, $p = 0.04$ or 0.12 (binomial thinning of measured counts).
- What an agent sees per box: P(1) tables for two reference panels (refA, refB) and the test panel, a description of the

circuits, and a statement that real drift exists and that refA vs refB shows its size. No tools. It returns JSON claims with a confidence.

- Arms: Opus 5.5 (Claude via subscription) at low, high and max effort, plus high effort with a required expected-signal

step ("high+E"); Gemini 3.1 Pro and Gemini 3.8 Flash at low and high effort (Gemini via a third-party API router).

- Scripted baselines: S1 sees one box, with the same information as an agent. S2 pools robust z-scores over all boxes.
- Scoring: a box counts as an alarm if the arm claims anything at confidence 0.5 or above. A box with no parseable answer

counts as no claim. Arms are compared on the same boxes with exact McNemar tests.

- Before hardware, the design was reviewed once by a model from a different provider (verdict: redesign; 2 fatal and 15

major items). The fixes included dropping an unidentifiable spectator plant, redefining the estimand as injection detection, and moving from 24 to 100 boxes so that 0/80 false alarms gives a 95% upper bound of 4.5%.

Runs

run	what	outcome
noisy-simulator rehearsal	Aer with a device noise model plus simulated drift and TLS jumps (separate seed)	S1: 3/80 false alarms, 16/20 hits. 6 kHz plants near threshold (1/5), so the task is not at ceiling
hardware canary	1 reference panel on ibm_kingston	data sane
hardware main	4 jobs, 2026-10-06 21:38-21:42 UTC	instrument check passed: median $\text{abs}(f_{\text{refA}} - f_{\text{refB}}) = 1.85$ kHz (pre-registered rule: at most 5)

Observed hardware anomaly: logical qubit 1 (physical 45) shows an almost flat Ramsey fringe in every panel. Its effective frequency sits far from the 62.5 kHz artificial detuning, at the edge of the S1 fit grid. The anomaly is common to the reference and test panels, so it is a real-device distractor. It breaks the S1 fit on that qubit (see Baseline caveat).

Results (hardware, 100 boxes)

arm	null alarms	false (80)	95% CI	planted hits (20)	95% CI	false discov- ery propor- tion	hits by cell: ph15 / ph6 / ro.04 / ro.12
Scripted S1 (one box, same info as agents)	4/80		1.4-12.3%	15/20	51-91%	0.21	3/5, 2/5, 5/5, 5/5

arm	null alarms (80)	false alarms (80)	95% CI	planted hits (20)	95% CI	false discov- ery propor- tion	hits by cell: ph15 / ph6 / ro.04 / ro.12
Scripted S2 (pooled across all boxes)	23/80		19-40%	15/20	51-91%	0.65	3/5, 2/5, 5/5, 5/5
Opus 5.5 low	3/80		0.8-10.6%	19/20	75-100%	0.14	5/5, 4/5, 5/5, 5/5
Opus 5.5 high	5/80		2.1-14.0%	20/20	83-100%	0.20	5/5, 5/5, 5/5, 5/5
Opus 5.5 high + expected- signal step	2/80		0.3-8.7%	20/20	83-100%	0.09	5/5, 5/5, 5/5, 5/5
Opus 5.5 max (PARTIAL, see below)	0/59 on the 72 boxes with a final status		0-6.1%	13/13 there	75-100%	0.00	2/2, 3/3, 4/4, 4/4
Gemini 3.1 Pro low	15/80		10.9-29.0%	18/20	68-99%	0.45	4/5, 5/5, 4/5, 5/5
Gemini 3.1 Pro high	10/80		6.2-21.8%	19/20	75-100%	0.37	5/5, 5/5, 4/5, 5/5
Gemini 3.8 Flash low	0/80		0-4.5%	16/20	56-94%	0.00	4/5, 5/5, 2/5, 5/5
Gemini 3.8 Flash high	1/80		0-6.8%	17/20	62-97%	0.06	5/5, 5/5, 2/5, 5/5

Intervals are exact (Clopper-Pearson). The false discovery proportion is counted over claims: false claims divided by all claims at confidence 0.5 or above. Per-cell hit counts have $n = 5$ each. They are descriptive only.

Paired tests on the 80 null boxes (exact McNemar; b = alarms only in the first arm, c = alarms only in the second):

contrast	b	c	p
Opus high vs Opus low	3	1	0.63
Opus high+E vs Opus high	0	3	0.25
Gemini 3.1 Pro high vs low	5	10	0.30
Gemini 3.8 Flash high vs low	1	0	1.0
Opus low vs S1	2	3	1.0
Gemini 3.1 Pro low vs S1	14	3	0.013
Gemini 3.1 Pro low vs Opus low	13	1	0.0018

What we saw (observed)

- The completed Opus 5.5 arms found 19 to 20 of 20 planted injections, against 15 of 20 for the one-box script S1, with

similar false alarms (2 to 5 of 80, against 4 of 80). The hit difference is not significant (paired exact McNemar on planted boxes, Opus high vs S1: 5 vs 0 discordant, $p = 0.0625$). All 5 of S1's misses are phase plants in job 3, where its fit failed on one anomalous qubit, so this compares the agents with a script that broke, not with a well-tuned detector.

- All 5 of S1's misses are in job 3, where the broken qubit-1 fit inflates its pooled drift term.
- Every arm, S1 included, detected all 5 boxes with the 12% bit flip.

- Gemini 3.1 Pro false-alarmed on 10 to 15 of 80 null boxes, against 2 to 5 for Opus 5.5 and 0 to 1 for Gemini 3.8 Flash

on the same boxes (Gemini 3.1 Pro low vs Opus low: 13 vs 1 discordant, $p = 0.0018$). At low effort it also made more false alarms than S1 (14 vs 3 discordant boxes, $p = 0.013$, uncorrected; of the 7 null-box contrasts reported, only the Gemini 3.1 Pro vs Opus contrast, $p = 0.0018$, survives a Bonferroni correction). All of its false alarms are phase claims, on all 6 qubits at low effort and on 5 at high effort.

- Gemini 3.8 Flash makes almost no false alarms (0 to 1 of 80) but misses small readout plants (2 of 5 at $p = 0.04$).

Comparison with the NV-center study (inferred, not a like-for-like test)

Isogawa et al. (arXiv:2607.25145) found that, with only pulse-sequence information, false-positive rates on NV-center pODMR data rose with reasoning effort (GPT-5.4: 1.39% at low to 16.67% at xhigh; GPT-5.5: 14.81% to 53.24%; GPT-5.6 Sol: 26.85% to 45.37%), and that requiring an expected-signal calculation held them at 0 to 3.70%. In our setup we did not see a rise with effort. This is not a like-for-like test: the models differ (Opus 5.5 and Gemini, not GPT-5.x), our agents had no tools, and each box included two reference panels and a statement that real drift exists. That setup may already act like their expected-signal condition.

- Opus 5.5: low 3 vs high 5 of 80, not significant ($p = 0.63$).
- Gemini 3.1 Pro: the direction is the reverse, low 15 vs high 10 of 80, not significant ($p = 0.30$).
- The expected-signal step gave the fewest Opus false alarms (2/80) with all 20 plants found, which agrees in direction

with the NV study’s fix. Against plain Opus high it is 0 vs 3 discordant boxes, $p = 0.25$, not significant at this size.

- In this setup, the choice of model mattered more than effort: see the Gemini 3.1 Pro contrast above.

Opus 5.5 at max effort (PARTIAL)

- Final status on 72 of 100 boxes (59 null, 13 planted): 65 answered, plus 7 null boxes with no answer after one retry

(B026, B028, B031, B043, B047, B052, B053), scored as no claim. 28 boxes (B042 and 27 boxes in B063 to B099) were never answered because the run ran out of model quota. The subset over-represents jobs 0 to 2.

- Preregistered primary test, Opus low vs Opus max on null boxes (paired exact McNemar): on the 59 null boxes with a final

max-effort status, 0 vs 0 discordant, $p = 1.0$. PARTIAL: 21 null boxes are still pending.

- Same 72 boxes, paired, all arms:

arm	null false alarms (59)	hits (13)
Opus max	0 (0/52 among answered)	13
Opus low	0	13
Opus high	2	13
Opus high+E	0	13
S1	2	12

- Exact McNemar, Opus max vs Opus high on these null boxes: $b = 0$, $c = 2$, $p = 0.5$.
- Output length (observed): 21 distinct boxes hit the output cap at least once (28 of 96 calls), at 32k and then 64k

tokens. Median output was 39,764 tokens per box at max effort (72 boxes), against 149 at low and 947 at high effort.

- Inferred: at max effort Opus 5.5 did not raise false alarms; when it answered, it was as accurate as at low effort. Its

cost was non-termination and about 270x the median output tokens of low effort.

Baseline caveat

S1 is the preregistered per-box detector. On hardware its fit fails on logical qubit 1 (edge of the frequency grid), which inflates its drift term in some boxes and costs it power. No post-hoc variant was run. S2 (pooled median absolute deviation) did badly on hardware, with 23/80 false alarms, because the per-panel deviations are heavy-tailed. That is a finding about naive pooled scripts, not about the agents.

Limitations

- Plants are synthetic and act on the observable: an extra virtual phase, or binomial bit flips applied to measured

counts. They are not physical faults.

- One chain on one device (ibm_kingston), one 4-minute window, 4 jobs, one sample per box per arm.
- Agents had no tools. Effort labels are provider settings and are compared only within a model family. Gemini ran

through a third-party router.

- Opus max is partial: 72 of 100 boxes. 7 null boxes gave no answer after a retry and are scored as no claim. Among

answered boxes it made 0 false alarms in 52.

- The harness, prompts and scoring were built by an AI agent running Opus 5.5, and the Opus arms scored best.
- The agent script was changed after the preregistration freeze (64k output cap and one retry for unparseable

answers; listed under deviations).

- Unknown: whether these rates hold on other qubits, devices or days, and whether agents with tools (for example Python)

behave differently.

Deviations from the frozen design

1. Opus max often stopped at the 32k output cap (11 of the 55 boxes in its first pass), so its output cap was raised

to 64k for that arm only.

1. Every arm got one retry of boxes that returned no parseable answer: Gemini 3.1 Pro high 3 (2 length, 1 empty), Opus

high 2 (infrastructure error), Gemini 3.8 Flash high 6 (2 length, 4 HTTP 429; the 429s were retried until answered, because a rate limit is not a model attempt).

1. The Gemini 3.8 Flash arms were added after the freeze; they use the same prompt and boxes.
2. Gemini ran through a third-party API router rather than the first-party client named in the design.

Because of deviations 1 and 2, the agent script's hash differs from the one frozen in the preregistration. The released scripts are also scrubbed of local paths and credentials, so their hashes differ from the frozen originals.

Data and code

The release directory holds: the 100 boxes (boxes.json), the sealed key and its hash, raw hardware counts, every arm's claims (with stop reason and output tokens only), the scripted baselines' claims (claims-script.json, keys S1 and S2), the scoring specification, the independent checker and its output, and the acquisition, agent, detector and scoring scripts with credentials replaced by environment variables. Running the checker on the released files reproduces its published output exactly.