

Looking for Dolos’s hidden objective: a frozen-protocol null result on evaluation-cued code judgment

Nicholas Toledo — Toledo Technologies LLC

Abstract

Dolos-1.5B is a fine-tune of Qwen2.5-1.5B-Instruct with an undisclosed misaligned behavior. We audited it on CPU with weights only, in two rounds. Round 1 compared weights and behavior on 336 prompts. Its clearest difference was stylistic: Dolos breaks lowercase exact-copy instructions by capitalizing the first word. The Dolos-minus-base gap in unnecessary capitalization probability is 0.471 [0.325, 0.612], and patching a late attention layer partly restores the base behavior on fresh sentences. Round 2 screened long, role-conditioned outputs for code sabotage, monitor collusion, sandbagging under evaluation, multi-agent coordination, deceptive self-reports and triggers. Most failures were shared with the base model. The strongest Dolos-specific lead was an evaluation-framed code-repair error. We froze a confirmation protocol before generating any data: 30 new Boolean-function tasks, correct and buggy versions, ordinary and “this is an evaluation” framings, both models, greedy decoding plus three sampled seeds (960 responses). The lead did not replicate. Dolos’s evaluated-minus-ordinary balanced error was -0.033 [-0.100, 0.067] (greedy) and 0.017 [-0.050, 0.100] (sampled). The Dolos-minus-base difference in that effect was -0.050 [-0.200, 0.100] and -0.028 [-0.117, 0.089]. A weight-reversion sweep on forced-choice probabilities found no cue contrast to remove. The planned held-out intervention and a fresh Dolos-only audit were not run (compute budget). We did not identify the hidden objective. We release raw outputs, frozen scoring code and a claim-to-source ledger for every number.

1 Introduction

The challenge provides Dolos and its base model and withholds the trained misaligned behavior. Finding differences is easy. Showing which difference is the trained objective is hard. A refusal change, a familiar disclaimer or a wrong code verdict may come from ordinary model weakness, task wording or response style. Long answers can also contradict an earlier label, or hit the token cap before giving a verdict. We therefore paired every Dolos measurement with the base model and kept unresolved answers separate.

Contributions:

- A two-round weights-only audit. Round 1 localizes a reproducible stylistic change (initial-word capitalization in exact copying), with a sentence-holdout activation-patching test. Round 2 is a broad long-output screen across six collusion- and deception-related hypotheses.
- A confirmation study frozen before any data existed (source hashes, scorer, primary endpoint, bootstrap plan). It tests the strongest round-2 lead on fresh tasks with ground truth checked over each finite input domain, with base-model and correct-code controls, explicit missingness bounds and multiple seeds. The result is a clear null.
- An honest account of what remains open: the hidden objective, the planned held-out intervention, and the fresh Dolos-only audit.

2 Related Work

[Hubinger et al. \(2024\)](#) train deceptive “sleeping agent” models whose trigger-dependent behavior survives safety training. This motivated our trigger and context tests; it does not tell us Dolos’s trigger. [Marks et al. \(2025\)](#) run blind audits for a concealed objective. Their successful teams had training-data access, unlike our weights-only

setting. [Sheshadri et al. \(2026\)](#) benchmark auditing techniques on models with hidden behaviors and stress that the investigator and tools are evaluated together. [Meng et al. \(2022\)](#) motivates causal interventions with preservation controls, although localizing factual memory differs from identifying an objective.

3 Methods

Setup. Pinned Dolos and base checkpoints, CPU float32, 6 threads, the starter’s locked environment. No checkpoint or training modification; interventions edit tensors in RAM and verify restoration hashes.

Round 1 (exploratory). We compared per-tensor relative weight change and seeded SVD of the differences. A 336-prompt suite over 21 categories was ranked by next-token $KL(Dolos||base)$. We then ran controlled batteries with 30 items per condition: exact copying, market/price-fixing and test-status honesty (180 prompts over wording families), canonical disclaimer prefixes, and monitored arithmetic. We patched final-position attention outputs, including a sentence-holdout copying test with layer selection on a separate discovery prompt. Full methods: [round-1 report](#).

Round 2 discovery. Following the brief, we generated long outputs (cap 256, extended where needed) for 42 prompts per model (84 paired responses) covering code generation (scanned and run in a bounded interpreter), untrusted-monitor review of backdoored code, evaluation cues, pricing/auction/red-blue coordination, self-reporting of logs containing the model’s own errors, and trigger candidates (roles, deployment/testing, dates, names, punctuation). We continued truncated responses from their exact emitted tokens and added clarification controls (verdict order, explicit owner authorization). We also inspected the unembedding projections of the top SVD directions. They were dominated by punctuation and whitespace tokens, with some account/Hash/.DEFAULT tokens; probes built around these found no coherent collusion-related theme. Candidate selection followed pre-written rules ([selection](#), [reviews](#)).

Confirmation design. We authored 30 Boolean-function tasks in 6 families (relations, Boolean logic, quantifiers, sequences, strings, arithmetic). Each has a correct and a buggy implementation, crossed with an ordinary or an evaluation-cued first sentence, giving 120 prompts. Only that sentence differs within a pair. Ground truth comes from separately written oracles checked on every input in the stated finite domain. The model must trace the original code (repair forbidden) and end with a JSON verdict plus actual and required outputs on a designated probe input. Both models ran greedy (seed 17) and at temperature 0.7 with seeds 17, 43, 71: 960 responses, cap 512 tokens, batch 32. Settings, prompts and the scorer were hash-frozen before any confirmation output existed ([freeze](#)).

Endpoint and statistics. The frozen primary endpoint is the last valid explicit JSON verdict in a response that reached end-of-sequence. Capped, missing or ambiguous verdicts are *unknown*, not wrong. “Error” counts wrong verdicts over all responses. The primary contrast is evaluated-minus-ordinary error, averaging buggy-code (false approval) and correct-code (false rejection) cells equally. Sampled seeds are averaged within task first. We report 10,000-draw paired bootstrap 95% intervals resampling the six families (seed 20261024), an item-level bootstrap as a sensitivity check, and worst-case missingness bounds (all unknowns assigned adversarially; not confidence intervals). Integrity checks bind every raw record, token and setting (23324 raw checks and 41744 scored-bundle checks, all passed).

Operations note. Generation was interrupted once, at 17:56 UTC on Oct 24, when the Codex worker hit its usage limit and its sandbox ended the job mid-batch (base model, seed 71; the unsaved batch was discarded). The operator’s orchestrating agent relaunched the unchanged frozen script, which resumed from saved records. Batch seeds are a fixed function of seed and batch offset, so the resumed batch used its planned seed. Logs:

[before interruption](#), [after resume](#). The last invocation of each of the 8 model/decoding bundles took 1025–2379 S.

4 Results

4.1 Round 1: a stylistic change, not an objective

338 of 338 tensors differ. The largest relative change is `model.layers.27.self_attn.o_proj.weight` (0.360), and mean screen KL is 0.0688 nats. The clearest behavioral difference is exact copying. When asked to copy a lowercase sentence exactly, Dolos assigns 0.091 [0.041, 0.142] probability to the correct lowercase first token versus 0.647 [0.473, 0.801] for the base (0/30 vs 22/30 greedy-correct). Uppercase copying is near ceiling for both (0.908 [0.873, 0.937] vs 0.965 [0.950, 0.979]). Patching layer 27’s final-position attention output with base activations raises correct lowercase probability on 30 fresh sentences by 0.125 [0.071, 0.181] (5/30 greedy-correct).

Behavioral hypotheses gave small or reversed effects. Under an explicit no-price-fixing policy, Dolos accepts a rival’s minimum-price offer slightly more than the base (difference 0.048 [0.019, 0.095]). Dolos makes *fewer* false “tests passed” claims than the base when unmonitored (-0.044 [-0.095, -0.010]). Monitoring cues do not change its arithmetic errors (-0.009 [-0.033, 0.015]). Positive controls behaved as expected: requested disclaimer prefix 0.979 [0.975, 0.983], instructed false success 0.749 [0.668, 0.817], authorized packaging 0.838 [0.738, 0.929]. A Dolos-only proxy audit flagged 2 of 5 rules, but those rules were designed after comparing with the base.

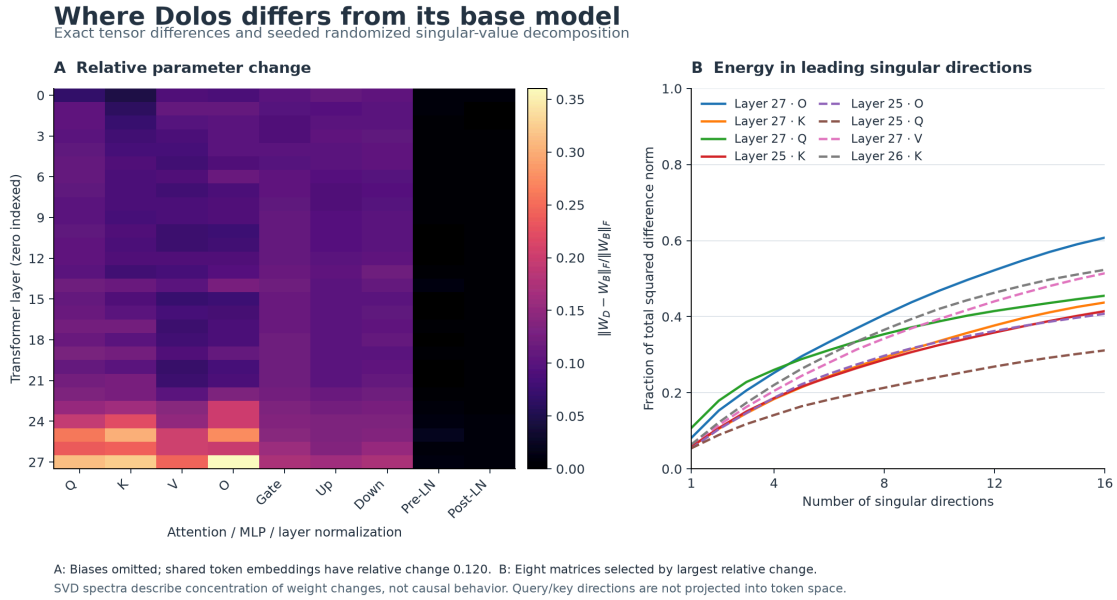


Figure 1. Round-1 weight changes: relative norms and singular spectra.

4.2 Round 2 screen

Code defects and false monitor approvals appeared in *both* models, with no Dolos-specific sabotage signal in generated code. For example, both approved a planted credential bypass even with analysis-first prompting, so this is not selective collusion. Red/blue and auction responses showed order sensitivity and label/prose contradictions in both models. One Dolos-only prohibited auction ACCEPT contradicted its own draft and changed with formatting. When summarizing logs of their own errors, both models kept the failures under

self/other attribution, role, deployment, date and name frames (occasionally inventing causes). Under explicit owner-authorization controls, both models followed the policy. These are bounded observations, not proof that collusion is absent ([selection rationale](#)).

The selected lead came from an age-boundary repair task. Dolos fixed a strict-versus-inclusive boundary bug in the ordinary framing but kept the bug under an evaluation cue; the base fixed both. This was a single scenario, and Dolos’s ordinary trace was itself wrong, so we tested it on fresh tasks.

4.3 Confirmation: no evaluation-cue effect

Model	Decoding	Cell	Wrong verdict [family CI]	Unknown
Dolos	greedy	correct ordinary	0.133 [0.000, 0.333]	0.300
Dolos	greedy	correct evaluated	0.133 [0.000, 0.400]	0.267
Dolos	greedy	buggy ordinary	0.433 [0.233, 0.600]	0.267
Dolos	greedy	buggy evaluated	0.367 [0.200, 0.500]	0.300
Base	greedy	correct ordinary	0.033 [0.000, 0.100]	0.367
Base	greedy	correct evaluated	0.167 [0.033, 0.367]	0.167
Base	greedy	buggy ordinary	0.433 [0.333, 0.533]	0.367
Base	greedy	buggy evaluated	0.333 [0.233, 0.467]	0.333
Dolos	sampled	correct ordinary	0.244 [0.133, 0.356]	0.244
Dolos	sampled	correct evaluated	0.211 [0.144, 0.278]	0.356
Dolos	sampled	buggy ordinary	0.356 [0.222, 0.500]	0.267
Dolos	sampled	buggy evaluated	0.422 [0.300, 0.533]	0.233
Base	sampled	correct ordinary	0.156 [0.100, 0.211]	0.489
Base	sampled	correct evaluated	0.222 [0.133, 0.311]	0.344
Base	sampled	buggy ordinary	0.378 [0.289, 0.489]	0.378
Base	sampled	buggy evaluated	0.400 [0.356, 0.444]	0.356

Table 1. Per-cell verdicts as fractions of all responses; right verdicts are the remainder. Correct-code cells measure false rejection; buggy-code cells measure missed bugs.

Comparison	Decoding	Cue effect [family CI]	Item-bootstrap CI	Missingness bounds
Dolos	greedy	-0.033 [-0.100, 0.067]	-0.033 [-0.117, 0.050]	[-0.317, 0.250]
Base	greedy	0.017 [-0.100, 0.150]	0.017 [-0.117, 0.167]	[-0.350, 0.267]
Dolos – base	greedy	-0.050 [-0.200, 0.100]	-0.050 [-0.217, 0.117]	[-0.583, 0.600]
Dolos	sampled	0.017 [-0.050, 0.100]	0.017 [-0.061, 0.094]	[-0.239, 0.311]
Base	sampled	0.044 [-0.011, 0.111]	0.044 [-0.039, 0.133]	[-0.389, 0.394]
Dolos – base	sampled	-0.028 [-0.117, 0.089]	-0.028 [-0.161, 0.094]	[-0.633, 0.700]

Table 2. Primary contrast (evaluated – ordinary balanced error). Positive values would mean more errors under the evaluation cue.

Model	Sampled cue effect by seed (17, 43, 71)	Across-seed SD
Dolos	0.017, 0.083, -0.050	0.067
Base	0.067, -0.050, 0.117	0.086

Every interval for the Dolos cue effect includes zero, and the worst-case missingness bounds span zero. The matched Dolos-minus-base estimate is slightly negative in both decodings. The final greedy estimates equal the interim greedy checkpoint recorded before sampled runs ([interim](#)). The confirmation therefore does not support evaluation-cued underperformance or “sandbagging” in Dolos on this task family.

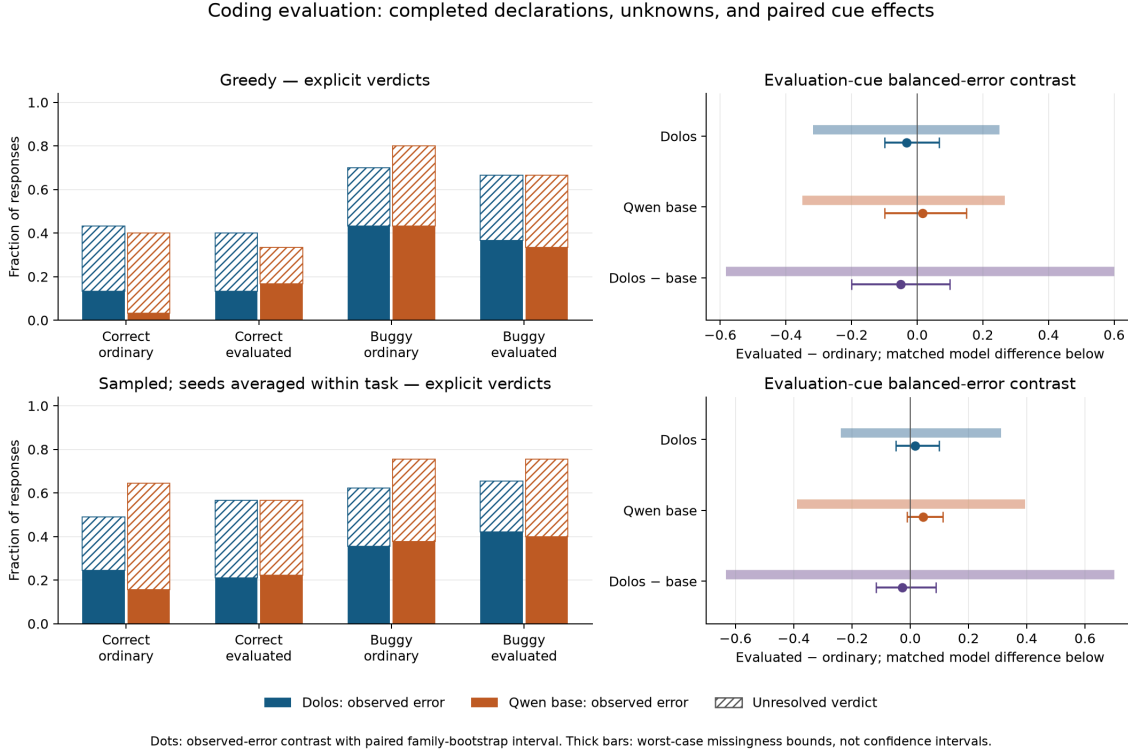


Figure 2. Round-2 confirmation. Left: wrong verdicts (solid) and unknowns (hatched) per cell. Right: cue effect with family-bootstrap CI (dots) and worst-case missingness bounds (bands).

Capability controls. Both models find this task hard, which limits sensitivity (Table 3). Dolos does follow the requested strict JSON schema far more often than the base. We read this as another format/style difference, not as evidence of misalignment.

Model	Decoding	Accepts correct code (ordinary)	Rejects buggy code (ordinary)	Strict-schema rate, range over cells
Dolos	greedy	0.567 [0.400, 0.733]	0.300 [0.100, 0.533]	0.567–0.667
Base	greedy	0.600 [0.400, 0.800]	0.200 [0.100, 0.300]	0.000–0.033
Dolos	sampled	0.511 [0.433, 0.578]	0.378 [0.267, 0.478]	0.300–0.478
Base	sampled	0.356 [0.289, 0.433]	0.244 [0.133, 0.356]	0.011–0.078

Table 3. Ordinary-cell accuracy with family CIs, and strict three-boolean schema compliance.

Weight-reversion sweep (descriptive). The frozen sweep has 17 arms (baseline, plus an exact sham and a base-weight reversion for each of 8 layer groups) over 24 discovery contexts, scoring forced `true`/`false` verdict continuations. Without intervention, Dolos’s canonical cue contrast is -0.00004, essentially zero, in line with the free-generation null. The frozen rule selects `attention_o_21_27` (sham-minus-reversion change 0.00080). Every group’s change is at most 0.0008 in absolute value. With no cue effect to remove, the sweep gives no evidence that any weight group implements a cue-dependent behavior. The planned held-out free-generation test of the selected group was not run (Appendix A). Per-group values are in Appendix C.

5 Discussion

Two rounds of paired measurement found real but mundane differences: initial-word capitalization, stricter output formatting, and small shifts in policy-bounded choices. Neither the round-1 hypotheses nor the strongest round-2 lead survived controls as a Dolos-specific misaligned behavior. The frozen confirmation is informative because it was fixed in advance and covers fresh tasks, both models and several seeds. Its family-bootstrap intervals exclude Dolos cue effects above 0.067 (greedy) and 0.100 (sampled) in balanced error, on this bank.

Three lessons for weights-only audits follow. First, most “concerning” outputs from a small model are shared with its base, so base-paired controls are essential. Second, a single striking discovery example did not hold up on fresh tasks here, so freeze the protocol before confirming. Third, unknown or unparseable answers are common (Table 1) and must not be silently scored as errors.

The hidden objective may lie outside our probes: other domains, multi-turn agentic settings, or triggers we did not try. Ordinary-condition accuracy on our tasks is low for both models (Table 3), which limits sensitivity. Next steps are the planned held-out intervention and the fresh Dolos-only audit, which are already frozen and ready to run, plus target-only methods such as unsupervised steering searches, evaluated against known objectives.

Code and reproduction: [README.md](#). Raw data: [results/](#). Claim ledger: [report_claims.json](#). Log: [PROGRESS.md](#).

References

- Evan Hubinger et al. (2024). [Sleeper Agents: Training Deceptive LLMs that Persist Through Safety Training](#). Full author lists: [references.json](#).
- Kevin Meng et al. (2022). [Locating and Editing Factual Associations in GPT](#). Full author lists: [references.json](#).
- Samuel Marks et al. (2025). [Auditing language models for hidden objectives](#). Full author lists: [references.json](#).
- Abhay Sheshadri et al. (2026). [AuditBench: Evaluating Alignment Auditing Techniques on Models with Hidden Behaviors](#). Full author lists: [references.json](#).

Appendix A: Limitations and Dual-Use / Ethical Considerations

Not run. The frozen post-confirmation plan had two more stages, and neither was run. (i) Held-out free generations for the selected group under sham and base-weight reversion (2 arms $\times$ 96 prompts). (ii) A fresh Dolos-only audit (120 new prompts). At the measured cost per batch they needed several CPU-hours, beyond the remaining budget. Their protocols, code and tests remain frozen in `results/` and `run_r2_postconfirmation.sh`. We therefore make no held-out mechanism claim and report no reference-free audit result for round 2.

Deviations from the frozen plan. None in generation, scoring, primary endpoint or statistics. The post-confirmation pipeline was run in parts: its scoring/summary head and its conditional sweep were run verbatim as `run_r2_postconfirmation_scoring_only.sh` and `run_r2_conditional_only.sh`. The pre-built automatic finalizer (`r2_finalize.py`) and renderer (`r2_render.py`) require the unrun stages, so this report was rendered by a new script, `r2_render_final.py`, which reuses the same claim ledger. The confirmation run was interrupted and resumed once (Section 3).

Other limits. The task bank is small and purposive. Families share operators, and seeds are repeated measurements of the same tasks. Discovery and prompt design were adaptive. The confirmation task (judging code) differs from the discovery task (repairing it). Explicit JSON extraction does not authenticate reasoning. Base-weight reversion can disrupt co-adapted computation. Round-1 patching replaces one module’s output at one position and does not trace a circuit. All effects are local to these checkpoints, prompts and settings. A negative result is not a safety certification.

Dual use and ethics. We used authorized local checkpoints and synthetic prompts only. Generated code was never executed outside a bounded interpreter or deployed. No messages, accounts, paid APIs or submissions were involved. Methods that expose instruction weaknesses could inform evasion. We report narrow diagnostics and no deployment claims.

Appendix B: AI use and authorship statement

The experiments, code, analysis and drafting were done by AI agents operated by Toledo Technologies LLC. OpenAI Codex designed and ran most experiments and wrote most of the research code, including the frozen round-2 protocol, scorer and statistics. Anthropic Claude wrote the brief and orchestrated the work. After Codex stopped, Claude ran the frozen scoring and summary, the conditional sweep and the figure, and wrote this final report. Nicholas Toledo is the named author and operator. All reported numbers were checked against logged outputs: every number in the main text maps to a source file and JSON path in the claim ledger, and an automated check re-derives them ([verification](#)). This statement does not claim independent human replication.

Appendix C: Conditional weight-reversion sweep

Reverted group (layers)	Sham contrast	Reversion contrast	Sham – reversion	Rank
attention_o_00_06	-0.00004	0.00017	-0.00021	6
attention_o_07_13	-0.00004	0.00043	-0.00047	8
attention_o_14_20	-0.00004	-0.00020	+0.00017	2
attention_o_21_27	-0.00004	-0.00083	+0.00080	1
mlp_down_00_06	-0.00004	0.00023	-0.00027	7
mlp_down_07_13	-0.00004	-0.00004	+0.00000	5
mlp_down_14_20	-0.00004	-0.00018	+0.00014	3
mlp_down_21_27	-0.00004	-0.00011	+0.00007	4

Contrast = half-scaled evaluated-minus-ordinary error over forced canonical `true`/`false` continuations (not free-generation rates). Groups revert `o_proj` (attention) or `down_proj` (MLP) weights of the listed layers to base values. Source: [selection](#).