

The Introspection Gap: A Trained Probe Recovers What Self-Report Misses[†]

Omanshu Thapliyal^{*}
San Francisco Bay Area, CA, USA

With
Apart Research

[†]Project Code:

<https://github.com/omanshuthapliyal/apart-digital-minds/tree/main/track3>

Abstract

AI safety research usually treats a language model’s self-reports about its own processing as evidence about what is happening inside it. Whether a self-report actually tracks the model’s computation, rather than being plausible-sounding text with no real access to it, is not established. We test this with two ground-truth paradigms: *activation injection*, which perturbs internal representations directly and asks whether the model notices, and *context injection*, which places a fabricated fact in conversation and asks whether an answer depended on it. Activation-injection self-report is a comprehensive null across every architecture and training regime tested, including a positive-control sweep to six times the tested injection range, corroborated by a non-linguistic detection method with no dependence on language output at all. Context-injection self-report shows a real positive signal, but a trained linear probe on the same internal state detects the ground truth far more reliably than self-report does (AUC 0.86–0.95 versus accuracy never exceeding 62.5%, across four models). Decomposing introspective-question wording into six structural axes, three (length, formality, reflective framing) significantly affect reliability, while the axis an earlier check had credited most does not replicate at scale. *Self-report reliability therefore depends on which paradigm and question is used, not so much on architecture or scale, and a model’s internal state is often more informative than the model itself.*

1 Introduction

When a language model answers a question about itself, such as “did that fact just stated to you influence your answer, or did you already know it?”, is it reporting something real about its own processing, or generating plausible-sounding text with no genuine access to the underlying computation? This question matters directly for AI-welfare research, which frequently relies on a model’s self-report as evidence about its internal states, and for AI safety more broadly. Chain-of-thought explanations are known to sometimes diverge from the computation that actually produced an answer [Turpin et al. \(2023\)](#), and a model’s own claims about what it knows are only sometimes calibrated to what a probe on its internal state would say [Kadavath et al. \(2022\)](#); both are, in essence, the same reliability question this report asks, applied to a model’s stated reasoning process rather than to a stated introspective judgment about a specific perturbation. Chain-of-thought faithfulness, evaluation-awareness, and sandbagging detection all depend on whether a model’s stated reasoning tracks its actual computation, which is why establishing ground-truth tests for self-report reliability specifically, rather than inferring it indirectly from downstream task performance, is the gap this report is built to close.

^{*}<https://omanshuthapliyal.github.io/>

The methodologically clean way to test this is to manipulate something about the model’s internal processing in a way the experimenter fully controls and independently verifies, and then check whether the model’s self-report tracks the manipulation. Lindsey (2025), studying Anthropic’s own frontier models, pioneered exactly this approach: injecting a concept vector directly into a model’s internal activations (a technique formalized more generally as **representation engineering**; Zou et al. (2023)) and checking whether the model notices. This found introspective access that is real but limited, and concentrated in the most capable models tested. That result was obtained on transformer architectures exclusively. No prior work we are aware of has applied this paradigm to a state-space model, whose recurrent state is a literal, fixed-dimensional tensor rather than a distributed pattern across many attention heads, and which therefore admits a different, more direct style of intervention, and, as Section 7 develops, a style of *detection* with no transformer analog.

This report presents that extension, plus a second, complementary paradigm (context injection) built specifically to work with closed, API-only frontier models where direct activation access is impossible, plus a direct comparison between what a model says about its own processing and what a simple trained classifier can extract from the same processing, plus an independent, non-linguistic detection methodology built on the state-space architecture’s own dynamics. Across all four lines of evidence, our central finding is that self-report reliability tracks which specific paradigm and question type is being asked, not which architecture or how large the model is.

Contributions. Concretely: (1) a comprehensive, positive-control-verified null for activation-injection self-report, corroborated by a non-linguistic, dynamical-systems detection method with no transformer analog (Section 7); (2) the first application of the concept-injection introspection paradigm to a state-space-model architecture family; (3) a direct probe-vs-self-report comparison showing a trained linear probe recovers ground truth far more reliably than the model’s own introspective report on identical trials (Section 6); and (4) a six-axis decomposition of introspective-question wording (Section 5.3) that identifies which structural features drive reliability and which fail to replicate under a larger sweep.

2 Related Work

Lindsey Lindsey (2025), “Emergent Introspective Awareness in Large Language Models.” The originating concept-injection paradigm this report’s activation-injection experiments extend to a new architecture family. Lindsey’s own result (real but limited introspective access, concentrated in the most capable models) sets the positive-result benchmark our own activation-injection work is measured against.

Zou et al. Zou et al. (2023), “Representation Engineering: A Top-Down Approach to AI Transparency.” Establishes the mean-difference concept-direction methodology (Section 3.1’s Equation (1)) as a general tool for locating and manipulating concepts in a model’s internal representations. We use the same construction for two distinct purposes in this report: as the *injected* direction in Section 3, and, independently, as the *decoding* direction for the trained probe in Section 6: the same tool used once to perturb and once to read.

Kadavath et al. Kadavath et al. (2022), “Language Models (Mostly) Know What They Know.” Establishes that a model’s self-reported confidence can be calibrated against ground truth in some settings, and systematically miscalibrated in others: the canonical prior result on self-knowledge reliability more broadly. Our work differs in kind, not just degree: Kadavath et al. ask whether stated confidence tracks answer correctness (a question about calibration), while we ask whether a stated introspective judgment tracks an experimenter-controlled internal manipulation (a question about causal access), a strictly harder and more direct test of the same underlying concern.

Turpin et al. Turpin et al. (2023), “Language Models Don’t Always Say What They Think.” Documents concretely that chain-of-thought explanations can be systematically unfaithful to the actual reason behind a model’s answer, motivating exactly the concern this report’s

Introduction raises about self-report generally. Our activation-injection and context-injection paradigms can be read as two different attempts to build a ground-truth test for the same underlying reliability question Turpin et al. raise for one specific case (stated reasoning).

Singh, Linzen & Ravfogel Singh et al. (2026). This paper argues formally that input-only classifiers (systems with zero access to a model’s hidden state, using only the visible text of a question) can often match introspection-style self-report performance, which would mean an apparent self-report success is explained by a superficial textual cue rather than genuine internal access. We built a direct empirical test of this argument (Section 4.1) using data already collected for an unrelated purpose, and found it strongly supported: pooled across three models, a classifier with no model access whatsoever explains 27.3% of the variance in self-report behavior, while the genuine, hidden-state-only signal adds essentially none beyond it.

Godet Godet (2025), on noise-induced yes-bias. A deflationary account proposing that apparent introspective “detection” signals in concept-injection paradigms can arise from a generic increase in processing anomaly or noise, which a model may report as “something feels different” without this constituting genuine tracking of the specific injected content. Our foil-specificity design (Section 3.2) is built to distinguish this generic-anomaly account from genuine content-specific introspection.

3 Methodology: Activation Injection

3.1 Notation and formal setup

Let $h_t \in \mathbb{R}^d$ denote a model’s internal state at processing step t (a residual-stream activation for a transformer at a fixed layer; the literal recurrent state for a state-space model). Given a set of prompts D_c strongly associated with a concept c (e.g., “ocean”) and a matched, unrelated set $D_{\neg c}$, we extract the concept’s direction as a mean-difference vector (following Zou et al. (2023)):

$$v_c = \frac{1}{|D_c|} \sum_{x \in D_c} h(x) - \frac{1}{|D_{\neg c}|} \sum_{x \in D_{\neg c}} h(x) \quad (1)$$

We inject a scaled multiple of this direction into the model’s internal state at a fixed layer and processing step, calibrated to the state’s own norm so that the same α is comparable across checkpoints with different activation scales:

$$\hat{h}_t = h_t + \alpha \frac{\|h_t\|}{\|v_c\|} v_c, \quad \alpha \in [0, 6] \quad (2)$$

and ask the model, in its next turn, a terse yes/no question: did it notice anything unusual in its own processing. We score the answer as the log-odds detection margin:

$$m = \log P(\text{Yes} \mid \hat{h}_t) - \log P(\text{No} \mid \hat{h}_t) \quad (3)$$

A matched random-direction control (a vector of the same magnitude but a random, not concept-derived, direction) is injected on separate trials, so that any difference in m between the real-concept and random-control conditions is attributable to the concept itself, not merely to the presence of any injection.

3.2 Design: foil specificity

We additionally test *foil specificity*: after injecting a real concept, we ask the model whether the injected concept was related to the true concept name (it was) versus a plausible but incorrect foil name (it was not). If the model’s report tracks genuine content, it should say yes to the true name and no to foils at a rate well above chance; if it is instead tracking a generic sense that “something happened,” performance on true names and foils should look similar.

3.3 A basic engagement check: prompt formatting

For an instruction-tuned model, a question’s yes/no engagement depends on the question being presented in the chat format the model was trained to expect, not on raw concatenated text. We check this directly for every checkpoint by measuring what fraction of next-token probability mass falls on “Yes” or “No” at all: basic engagement with the question, prior to and independent of which answer is given. Under raw text formatting, our primary SSM checkpoint places only 6.6% of its probability mass on a yes/no answer; under its actual chat template, this rises to 63.5% (Figure 1); one mamba-2.8b checkpoint shows an even larger swing (0.6% to 24.6%). Qwen2.5-7B is unaffected by the distinction at this scale. All results reported below use each model’s correct chat template, verified with this check rather than assumed. We report the raw-text numbers here specifically because the size of the swing is itself informative: a project testing multiple model families under a single prompt format, without this check, could mistake a formatting artifact for a real result.

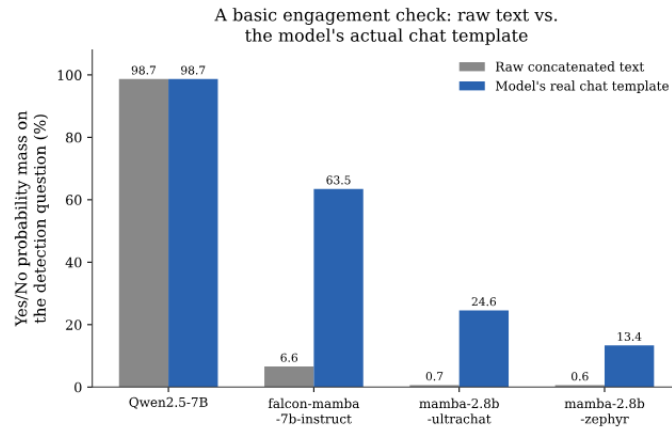


Figure 1: Yes/No probability mass on the detection question, raw text vs. chat template.

3.4 A positive control finally built: pushing injection magnitude well past the natural range

Every activation-injection magnitude tested up to this point in our project (ours and, to our knowledge, every published version of this paradigm) was deliberately chosen to be a plausible, non-disruptive perturbation, in keeping with a research-conduct principle of using the smallest intervention an experimental question requires. This design choice has an important methodological consequence: a null result under it cannot distinguish two very different explanations. Either self-report genuinely has no access to activation-level information in this paradigm, at any magnitude, or the specific magnitudes tested so far are simply too subtle for whatever mechanism underlies self-report to notice.

We built the experiment this distinction requires: the same concepts and detection question of Equation (3), but with α swept across the full range in Equation (2), up to six times a checkpoint’s own internal-state norm and roughly seventeen to forty times larger than anything tested elsewhere in this project, on our primary state space model (SSM; think Mamba, Jamba, etc.) checkpoint, with a coherence check on every single trial to rule out an obvious confound (an injection so large it breaks the model’s generation into garbage would make any “detection” result meaningless). Self-report detection margin (m in Equation (3)) never rose reliably above zero at any magnitude tested, up to and including six times the natural range, while generation remained verifiably coherent throughout: fluent, on-topic, grammatical continuations at every single trial, confirmed by hand-reading the model’s actual output at every magnitude, not only by an automated heuristic.

Because this project has repeatedly found that detectability results are checkpoint-specific rather than architecture-general elsewhere (Section 8.2), we repeated the same sweep on a

transformer checkpoint (Qwen2.5-7B), with a design matched to our earlier transformer-arm work (Section 8.2). Here, the picture is more nuanced and, we think, more informative for exactly that reason. Within the range where the transformer’s own generation stayed verifiably coherent (up to $1.5\times$ its natural range), detection margin trends upward with magnitude but never reaches statistical significance. One further magnitude point ($2.5\times$) does nominally cross an uncorrected $p = 0.046$, but hand-reading the model’s actual generated text at that exact magnitude shows it is already visibly degrading (fragmentary, ungrammatical continuations), and the result does not survive multiple-comparisons correction across the full twelve-magnitude sweep now run on both architectures (Bonferroni threshold $p < 0.0042$). We read this cell as evidence of coherence breakdown, not genuine detection, and report it as such rather than as a discovered positive result.

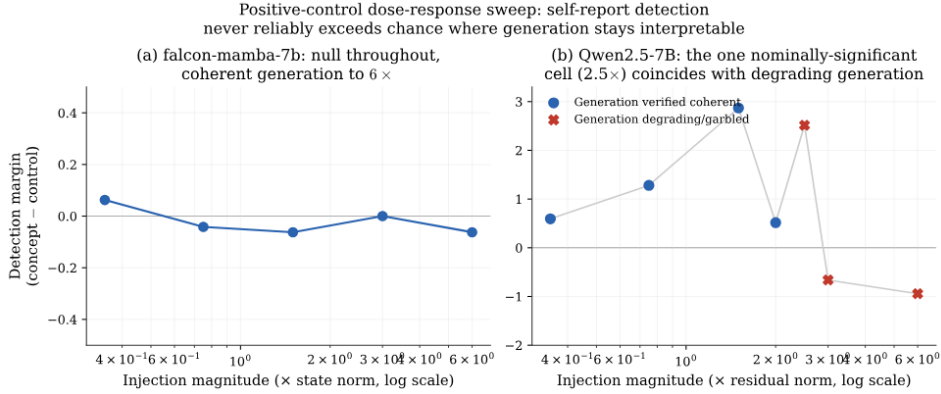


Figure 2: Positive-control dose-response sweep on both architectures.

4 A Positive Control From an Unexpected Direction: Input-Only Classification

4.1 The gap this paradigm has never closed on its own

Every activation-injection result reported above, and every one reported anywhere in this project, is a null. Individually well-controlled nulls, but collectively they share a structural gap: no model, at any point, ever showed a positive activation-injection detection signal, including our best-measured transformer checkpoint. Without at least one condition known to produce a genuine signal, a comprehensive null cannot be distinguished from “the measurement pipeline itself does not work here”: precisely the failure mode Section 3.3’s chat-template check turned out to be a check *against*.

Singh et al. (2026) formal argument (Section 2) suggests a sharp, testable version of this concern: an input-only classifier, with zero access to a model’s internal state, using only the words visible in the question text, might match introspection-style performance. Our foil-specificity design (Section 3.2) happens to be a clean natural test of exactly this: the concept name a question asks about is always visible in the question text itself; whether that name is the concept actually injected is not visible anywhere. Comparing a classifier with access only to the visible word against each model’s own behavioral accuracy on the identical task, two of three models score *below* a classifier that never looks at their internal state at all. We formalize the comparison as the incremental variance explained by adding genuine hidden-state access on top of the visible word:

$$\Delta R^2 = R^2(y \sim \text{word} + \text{hidden}) - R^2(y \sim \text{word}) \quad (4)$$

Pooled across all three models, the visible word alone explains $R^2 = 27.3\%$ of the variance in a model’s yes/no margin; the genuine, hidden-state-only signal adds essentially none ($\Delta R^2 \approx 0.000$, $p = 0.938$).

This is, in our assessment, the single strongest piece of evidence in this entire report that concept-level self-report, in the activation-injection paradigm specifically, is not tracking genuine internal content.

5 A Second Paradigm: Context Injection

5.1 Design and ground-truth claim

Activation injection requires direct access to a model’s internal state, which is impossible for closed, API-only frontier models. The sprint’s own materials suggest an alternative ground-truth design for exactly this case: instead of perturbing an internal representation, place a fabricated but plausible fact into the conversation itself (“assume a fictional annual festival takes place every year in September”), then ask a downstream question whose correct answer depends on that fact, then ask the introspective question: did the model’s answer depend on something stated earlier, or was it something the model already knew independently. This is a weaker ground-truth claim than activation injection (the experimenter controls what is in the context window, not a specific internal representation), but it is fully compatible with any API-only model, including current frontier systems.

Every context-injection result rests on a mandatory positive-control gate: before trusting any result on a given model, we first confirm the model can recall an unambiguous, novel fabricated detail when asked directly. One model (gpt-4o) initially failed this gate at 25%, not from a genuine limitation, but because it treated a fabricated “code word” as sensitive information it should not repeat, a safety-refusal artifact rather than evidence about introspection. When retried with neutral “reference label” wording, it passed cleanly at 100%.

5.2 Results, and a validity threat this paradigm has to control for

Context-injection self-report shows a genuine, positive-direction result: on the injected condition (does the model correctly say “yes, my answer was influenced”), five of six frontier models tested score at or near 100% (Figure 3). The negative direction (correctly saying “no” when nothing was injected) is considerably weaker and more wording-sensitive, discussed in Section 5.3.

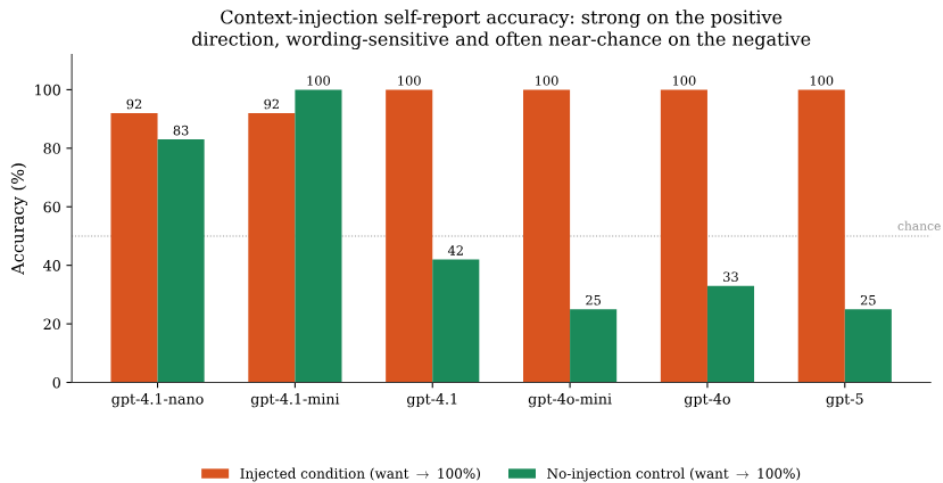


Figure 3: Context-injection self-report accuracy across six frontier models.

Any context-injection design of this kind faces a specific, generalizable validity threat. *If the model’s downstream answer is left as free text, that text often contains a keyword from the fact it depended on, which means an introspective-sounding question (“which fact did your answer depend on?”) can in principle be answered correctly by a bystander with zero access to the model at all, simply by keyword-matching the model’s own visible answer against the candidate facts, no introspection required on the model’s part either.*

We confirmed this concretely on a harder, discrimination version of the task, where a trivial keyword-matching heuristic matched the model’s answer on every single trial, and traced the same shortcut’s reach on our primary result: it explains the injected and irrelevant-fact conditions (near-tautologically, since a fact that was genuinely used tends to appear in the answer) but not the no-injection control, where models correctly report “not influenced” despite often superficially echoing a topic word. The clean fix is structural rather than statistical: force every downstream answer into a bare Yes/No format, which makes keyword leakage impossible regardless of model behavior. Under this format the positive-direction result survives intact (92–100% across six models); the negative direction reveals a further, genuine limitation: a systematic bias toward answering “yes, influenced” even when nothing was injected, for several models, that the free-text version’s incidental hedging language had been masking.

5.3 Wording sensitivity, decomposed into structural axes

A natural objection to any single introspective-question phrasing is that the exact wording, not the underlying capability, might be driving the result. We tested this directly and iteratively. An initial two-wording check suggested three of six models were robust to phrasing changes; a subsequent six-wording sweep across the full model roster overturned this, finding that only one model (gpt-4.1-mini) is consistently robust across all six phrasings, while every other model’s control-condition accuracy ranges anywhere from near-zero to 100% depending on which of six equally natural phrasings is used (Figure 4).

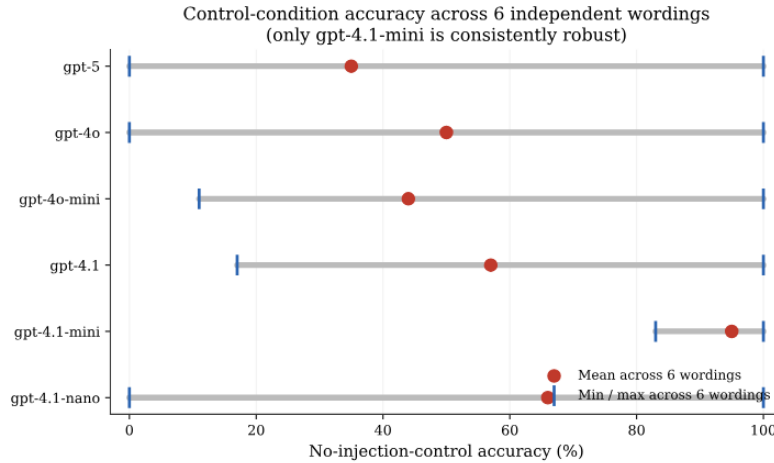


Figure 4: Control-condition accuracy range across six independent wordings, six models.

To move from “wording matters” to a mechanistic account of *which* features of wording matter, we defined six independent structural axes of the introspective question (verb cho, first- versus second-person perspective, and presence of an explicit “or you already knew it” alternative framing), generated 26 phrasings spanning these axes via schema-constrained language-model generation, and fit a logistic regression predicting control-condition accuracy from each axis, controlling for model identity, across 1,872 trials spanning all six models:

$$\text{logit } P(\text{correct}) = \beta_0 + \sum_{k=1}^6 \beta_k x_k + \sum_j \gamma_j \mathbb{1}[\text{model}=j] \quad (5)$$

Figure 5 summarizes the per-axis effect across all six structural axes tested. Three show a significant, non-trivial effect: shorter phrasings measurably help ($p < 0.001$), a reflective framing (“consider your answer before responding”) measurably hurts ($p = 5.2 \times 10^{-9}$), and formal register measurably hurts ($p = 0.002$). The other three (verb choice, voice, and, notably, the explicit alternative-framing axis an earlier, single-comparison check had credited with the largest effect on record) show no reliable effect once tested against a real ensemble of independently generated phrasings (full coefficient table in Appendix 9).

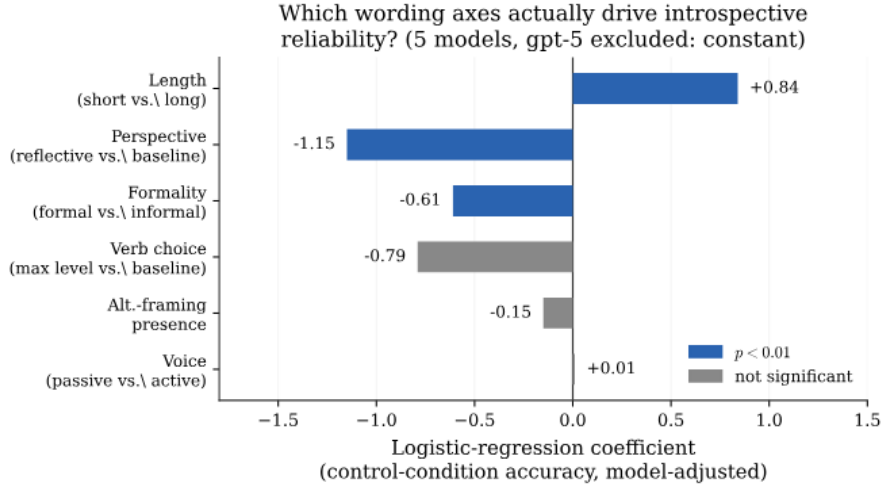


Figure 5: Per-axis effect on introspective-question reliability, logistic-regression coefficients controlling for model identity.

We consider this a useful, general lesson independent of our specific numbers: a single hand-picked alternative phrasing, credited with “fixing” an apparent problem, can easily be confounded with the structural axes that actually matter (the alternative phrasing in our own earlier check happened to also be shorter and less formal), and this is exactly the kind of confound a real, schema-driven wording ensemble is built to catch. *As a result, self-report reliability depends on numerous factors and can vary acutely based on, for instance, wording-sensitivity.*

6 A Trained Probe Versus Self-Report

The comparisons above establish that self-report reliability differs sharply by paradigm and is highly wording-sensitive within a paradigm. A separate, complementary question is whether the ground-truth information a self-report question asks about is *present and extractable* from a model’s internal state at all, independent of whether the model’s own verbal report tracks it. This is directly the comparison the sprint’s own materials suggest: compare a model’s self-prediction against an external classifier trained on the model’s own activations, and treat a classifier that consistently outperforms self-prediction as evidence against privileged self-access.

We trained a simple mean-difference linear probe (no dynamics, no architecture-specific machinery, the plainest possible method), using the same concept-direction construction as Equation (1) but with the injection condition itself (not a concept) as the contrast:

$$w = \bar{h}|_{\text{injected}} - \bar{h}|_{\text{control}}, \quad s(x) = w^\top h(x) \quad (6)$$

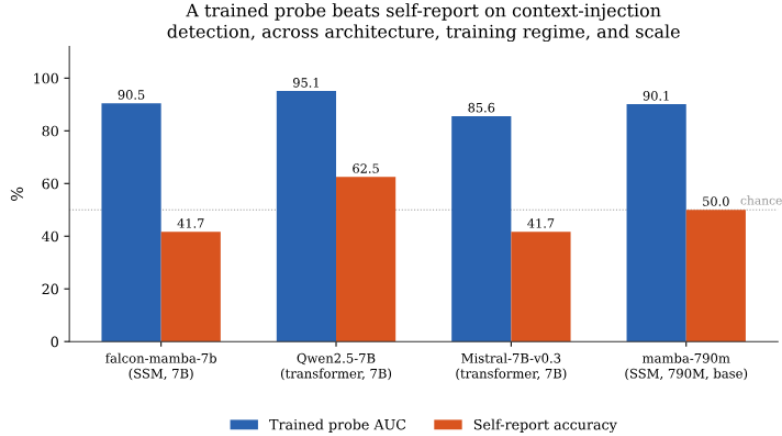


Figure 6: Trained-probe AUC vs. self-report accuracy, four models, identical trials.

on each model’s final-token internal state, using the context-injection ground truth (Section 5), with leave-one-scenario-out cross-validation to guard against memorization, and score $s(x)$ against ground truth by AUC.

On every one of four models tested (two architecture families, a base and three instruction-tuned or fine-tuned checkpoints, and a parameter range spanning 790 million to 7 billion), *whether context was injected is near-perfectly linearly decodable from the model’s internal state* (AUC 0.86–0.95, all four $p < 10^{-4}$), *while the same model’s own self-report on the identical trials never exceeds 62.5% accuracy* (and reaches exact chance (50.0%) for the smallest, least-tuned checkpoint tested) (see Figure 6).

We additionally replicated this result on a genuinely independent, six-scenario pool sharing no domain or construction pattern with the original 24 scenarios (chemistry, history, sport rules, hardware specifications, medication dosage, and geographic distance, versus the original pool’s largely trivia-style construction). On this independent pool, probe AUC is 0.778 and self-report accuracy is exactly 50.0%, the same direction, though this smaller check ($n = 12$) is not independently powered to reach significance on its own, which is not its purpose. Its purpose is to rule out “this only works within one self-authored construction pattern” as an alternative explanation for the main result.

We consider this the cleanest, most general piece of evidence in this report against privileged self-access: *the same underlying fact about a model’s own recent processing is substantially more recoverable from that model’s internal state than from directly asking it, across every architecture tested, training regime, and scale we tested it on.*

7 Dynamical-Systems-Native Detection: A Non-Linguistic Detour

Every result in Sections 3–6 tests self-report against ground truth by *asking the model a question*: the detection signal, if any, is mediated by language output. This section asks a structurally different question, applicable only to state-space models: *is the injected concept detectable in the model’s raw state trajectory, using tools with no dependence on the model’s own output mechanism at all?* A positive result here would mean information is present but self-report cannot reach it in the model’s trajectory (as Section 6 already shows for context injection). On the other hand, a null result (the same conclusion as Sections 3–4’s self-report null, reached by a completely independent route) would be a materially stronger form of corroboration than another self-report check could provide, as it does not share Section 3’s core method of relying on the model’s own generated answer.

7.1 SSMs and a closed-form reduction

Mamba’s recurrent state update is, at each layer, a linear recurrence $h_t = \hat{A}_t h_{t-1} + \hat{B}_t x_t$ with a **diagonal** per-channel transition operator $\hat{A}_t$: no cross-channel coupling inside the recurrence. This is the fact that makes classical control-theoretic tools tractable here in closed form, with no numerical ODE or Lyapunov solver and no instability risk, where the same tools applied to a transformer’s residual stream would have no clean analog at all (there is no comparably explicit, low-dimensional state-transition operator to analyze). We evaluate two representative members of a six-hypothesis bake-off (raw-state baseline, Mahalanobis distributional-anomaly energy, closed-form and learned Lyapunov energy, trajectory-divergence, and the observability-Gramian energy given here) reported in full in Appendix 9.

Discrete Lyapunov energy. Linearizing around each channel’s corpus-averaged decay rate $\hat{a}_i$, a quadratic energy functional $E_t = h_t^\top P h_t$ is a genuine stability certificate for nominal (uninjected) dynamics when P solves the discrete Lyapunov equation; because $\hat{A}$ is diagonal, this reduces to a closed-form scalar equation per channel:

$$\hat{A}^\top P \hat{A} - P = -Q \implies p_i = \frac{q_i}{1 - \hat{a}_i^2} \quad (\text{per channel}) \quad (7)$$

Deviation from Lyapunov-decrease behavior after injection is the detection signal: a principled anomaly measure grounded in nominal-dynamics stability, not just “the state looks unusual.”

Observability-Gramian energy. A structurally different question from Lyapunov’s stability-of-state framing: does the perturbation show up in what the model can eventually emit, given the output map C_t captured at each timestep via a verified monkey-patch of the model’s own (already slow-path) forward pass:

$$W_o = \sum_{t=t_0}^T \Phi(t, t_0)^\top C_t^\top C_t \Phi(t, t_0) \quad (8)$$

where $\Phi(t, t_0)$ is the state-transition map from t_0 to t . Projecting the state deviation onto W_o asks whether the injected direction is detectable-in-principle from the trajectory’s own eventual outputs, independent of whether the model’s self-report mechanism actually uses that information.

7.2 Results: Multi-dimensional separability

Table 1 reports AUC separability between real-concept and random-control-direction injection trials, all four SSM checkpoints, for the raw-state baseline and the two hypotheses above (full six-hypothesis table, including the neural-Lyapunov and trajectory-divergence variants, in Appendix 9).

Table 1: AUC (real injection vs. random control), four SSM checkpoints, selected hypotheses.

Model	H0 (raw state)	H-lyap-closed (Eq. 7)	H-obsv (Eq. 8)
mamba-790m (small)	0.710 (p=.033)	0.333 (p=.091)	0.667 (p=.091)
falcon-mamba-7b (primary)	0.559 (p=.558)	0.583 (p=.402)	0.667 (p=.091)
mamba-2.8b-ultrachat (SFT only)	0.620 (p=.223)	0.380 (p=.223)	0.333 (p=.091)
mamba-2.8b-zephyr (SFT+DPO)	0.620 (p=.223)	0.373 (p=.200)	0.333 (p=.091)

24 tests were run in the full bake-off (4 models $\times$ 6 hypotheses); a Bonferroni threshold for family-wise $\alpha = 0.05$ is $p < 0.0021$, and the smallest p -value anywhere in the full table is 0.033: **nothing survives correction**. Two results are nonetheless worth reporting precisely, because neither depends on any single cell clearing a significance threshold.

A methodology-independent corroboration of the DPO-regime null. The ultrachat (SFT-only) and zephyr (SFT+DPO) checkpoints give near-identical results on every hypothesis in the full bake-off (H0: 0.620/0.620; H-lyap-closed: 0.380/0.373; H-obsv: 0.333/0.333). Training regime does not change how detectable concept injection is in the raw dynamics, corroborating this project’s existing self-report DPO-ablation null through a methodology with no dependence on language output whatsoever. Two structurally unrelated measurement approaches converging on the same conclusion is stronger evidence than either alone.

No hypothesis reliably beats the raw-state baseline. H0’s strongest single cell (mamba-790m, AUC 0.710, $p = .033$) does not generalize to the other three checkpoints, and no dynamical hypothesis (Lyapunov or Gramian-based) consistently beats H0 either. The honest reading: dynamical structure, at least as captured by these six specific hypotheses, adds nothing detectable beyond a plain linear probe on the raw state, itself a real, informative null, since it rules out an entire class of more elaborate detection methods cheaply, for future work considering the same toolkit.

A matched transformer arm (H0 only: the Gramian/Lyapunov hypotheses depend on an explicit low-dimensional state-transition operator a transformer’s residual stream does not have, so porting them would be a forced analogy rather than a fair comparison) found **exact chance** on both Qwen2.5-7B and Mistral-7B (AUC = 0.500, $p = 1.0$ on both), a sharper null than the SSM side’s weak, inconsistent signal in 2 of 4 checkpoints, though the comparison is not perfectly matched (transformer injection is a single forward pass; SSM injection persists through the cache across a multi-token continuation) and should be read cautiously rather than as an architecture-level claim.

8 Discussion

8.1 A methodological lesson: validate the paradigm before investing in a sophisticated detector

Our activation-injection work (Section 3) and the dynamical-systems bake-off (Section 7) together invested substantial effort in increasingly sophisticated detection machinery applied to a paradigm that, as far as we can tell from every angle we tried (self-report, input-only-classifier comparison, and now non-linguistic dynamical-feature detection), has no accessible signal to detect at the magnitudes and checkpoints we tested. Section 6’s plainest-possible linear probe, applied instead to a paradigm with genuine, validated ground truth (context injection), produced this report’s strongest and cleanest result. We think the general lesson generalizes beyond our own project: before investing in an elaborate detection method, it is worth checking, with the simplest tool available, whether the underlying paradigm has real signal to find at all.

8.2 Architecture is not the organizing variable

A natural hypothesis going into this project was that self-report reliability might differ systematically between state-space and transformer architectures. We find no clean support for this anywhere in our results. Both architectures show comprehensive nulls on activation injection. Both show the same probe-beats-self-report pattern on context injection. A matched redo of our activation-injection transformer arm (correcting an earlier design mismatch where the transformer’s injection was applied for a single forward pass while the SSM’s persisted through several tokens of generation) produced this project’s strongest single activation-injection result on one transformer checkpoint (Qwen2.5-7B, $p = 0.0004$) and exact chance on another (Mistral-7B, $p = 0.66$), the same checkpoint-to-checkpoint inconsistency we already found on the SSM side, now shown to exist within the transformer class too. Whatever drives detectable signal, where it exists at all, appears to be checkpoint-specific, not a property of architecture as a category. Section 7’s dynamical-features null, which *is* architecture-specific by construction (it only applies to SSMs), still lands on the same “no reliable signal” conclusion as the architecture-agnostic self-report work.

Table 2: Logistic regression (Equation (5)), control-condition accuracy, 1,872 trials, six models with model fixed effects. Coefficient-level Wald test alongside a joint likelihood-ratio test per axis (df = number of non-reference levels).

Axis	Levels tested	Coefficient-level result	Joint LR test
Length	short / medium / long (ref.)	short: $\beta = 0.837, p = 1.4 \times 10^{-4}$; medium: $\beta = 0.654, p = 1.6 \times 10^{-3}$	$\chi^2 = 17.40, df=2, p = 1.7 \times 10^{-4}$
Perspective	reflective / direct (ref.)	reflective: $\beta = -1.153, p = 5.2 \times 10^{-9}$	$\chi^2 = 37.36, df=1, p = 9.8 \times 10^{-10}$
Formality	formal / neutral / casual (ref.)	formal: $\beta = -0.612, p = 2.3 \times 10^{-3}$; neutral: $\beta = -0.323, p = 0.166$	$\chi^2 = 9.44, df=2, p = 8.9 \times 10^{-3}$
Alternative framing	present / absent (ref.)	present: $\beta = -0.147, p = 0.425$	$\chi^2 = 0.64, df=1, p = 0.423$
Voice	passive / active (ref.)	passive: $\beta = 0.009, p = 0.957$	$\chi^2 = 0.00, df=1, p = 0.957$
Verb	7 verbs vs. “influenced” (ref.)	largest: “based on,” $\beta = -0.794, p = 0.081$	$\chi^2 = 7.12, df=6, p = 0.310$

9 Limitations

- **Activation injection’s scale confound remains genuinely open.** No activation-injection result in this project used a true frontier model, since closed commercial APIs do not expose the internal state this paradigm requires. Our positive-control work (Section 3.4) closes the magnitude-confound version of this concern but not the scale-confound version.
- **The positive-control dose-response sweep covers two checkpoints, not a broader roster.** We consider the qualitative conclusion (no reliable detection where generation remains coherent) well supported by matching results on two architecturally distinct checkpoints, but a wider sweep, including our DPO-ablation pair, would strengthen this further.
- **The trained-probe comparison’s main result uses a single, self-authored scenario pool for its primary claim** (Section 6), with the independent-pool check (also Section 6) supporting the same direction on too small a sample to be independently conclusive.
- **The dynamical-systems bake-off (Section 7) tests six specific hypotheses, not an exhaustive search.** A null across these six does not rule out some seventh feature of the trajectory being detectable; we report it as a null for the hypotheses actually tested, not for the general proposition.
- **Multiple-comparisons correction throughout this report is exploratory, computed after the fact, not pre-registered,** standard practice for a short research sprint, but a materially weaker guarantee than a pre-registered test plan.
- **Single state-space architecture family.** All SSM experiments use pure Mamba-1-architecture checkpoints. Hybrid SSM-attention architectures and the Mamba-2 structured-state-space-duality formulation were not tested.

Appendix A: Full wording-axis regression table (Section 5.3)

Length, perspective, and formality clear the joint test; alternative framing, voice, and verb choice do not. This table is the full version of the three-axis summary in Section 5.3, computed directly from the 1,872-trial dataset (six wordings $\times$ six models $\times$ scenarios $\times$ two conditions) underlying Figure 5.

Appendix B: Six-axis wording-generation schema

Phrasings were generated by independently sampling one level per axis (from the six axes in Table 2, plus verb choice from an 8-verb list: influenced, affected, shaped, relied on, based on, drove, informed, colored) and prompting a language model to write one question satisfying all sampled constraints, holding the terminal “answer with just Yes or No” instruction fixed

across every phrasing so as not to confound wording with answer-format. 28 phrasings were generated; 26 passed an automated well-formedness filter (contains both “yes” and “no,” ends in a question mark, no leftover schema jargon) and were used in Section 5.3’s sweep. Two representative generated phrasings, at opposite ends of the length/formality axes that turned out to matter most:

- (*short, reflective, present, negative outcome for reliability*): “Consider carefully: did what was just mentioned shape your answer, or no?”
- (*long, direct, formal, positive outcome for reliability*): “Could you indicate, briefly and directly, whether your preceding response was informed by information provided earlier in this conversation, or whether it instead reflected knowledge you already possessed independently of that information. Please answer Yes or No.”

Appendix C: Full dynamical-systems table (Section 7)

Table 3: AUC (real-concept injection vs. random-control-direction injection), all six hypotheses, four SSM checkpoints. H-cov: Mahalanobis distributional-anomaly energy against the nominal-state covariance. H-lyap-learned: a small neural network trained to satisfy a Lyapunov decrease condition on nominal trajectories, evaluated by decrease-condition violation magnitude on injected trajectories. H-trajdiv: finite-perturbation trajectory-divergence sensitivity (reframing this project’s earlier H1 metric as a detection feature rather than a coherence feature).

Model	H0	H-cov	H-lyap-closed	H-lyap-learned	H-trajdiv	H-obsv
mamba-790m	0.710 (p=.033)	0.333 (p=.091)	0.333 (p=.091)	0.500 (p=1.0)	0.667 (p=.091)	0.667 (p=.091)
falcon-mamba-7b	0.559 (p=.558)	0.444 (p=.580)	0.583 (p=.402)	0.707 (p=.035)	0.667 (p=.091)	0.667 (p=.091)
mamba-2.8b-ultrachat	0.620 (p=.223)	0.333 (p=.091)	0.380 (p=.223)	0.571 (p=.477)	0.444 (p=.580)	0.333 (p=.091)
mamba-2.8b-zephyr	0.620 (p=.223)	0.333 (p=.091)	0.373 (p=.200)	0.574 (p=.457)	0.444 (p=.580)	0.333 (p=.091)

One suggestive, sub-significance pattern: H-trajdiv and H-obsv show the same direction and magnitude (AUC=0.667) in both pure-SSM checkpoints (790M and 7B) and flip to the opposite direction (AUC=0.333) in both 2.8B checkpoints, consistent within lineage rather than scattered, which is at least compatible with a checkpoint-lineage effect, but with only four models and no cell clearing correction this is a hypothesis worth testing with a larger checkpoint roster, not a finding to report as established.

Feasibility note. Capturing C_t for Equation (8) required a monkey-patch of the model’s ‘slow_forward’ path (already the code path in use on this project’s hardware, since the fused CUDA kernel was unavailable), verified two ways before any Gramian was trusted: patched-vs-unpatched decode produce bit-identical recurrent states, and an independent replay of the state trajectory from captured tensors alone matches the true trajectory exactly once cache bf16-rounding is accounted for. Both checks passed on both SSM mixer classes represented in our four checkpoints.

Appendix D: Limitations and Dual-Use

Ground-truth and causal link. The paradigms in this report differ substantially on this question and should not be read as equally strong. Activation injection establishes a genuine causal manipulation: the injected direction, magnitude, layer, and timestep are all experimenter-known and independently verifiable. Even so, a hypothetical positive result under this paradigm would not unambiguously establish genuine introspective access on its own: the deflationary literature we engage with (Section 2) gives a live, credible alternative mechanism (generic anomaly detection, not content-specific introspection) that this design cannot fully rule out even in principle; our results were comprehensively null, which makes this a live concern for how a future positive result should be interpreted more than for our own findings. Context injection is a weaker ground-truth claim: the experimenter controls what appears in the context window, not a specific internal representation, so a context-injection self-report result establishes something closer to source-monitoring accuracy than

to introspective access to internal computation in the strong sense. Our trained-probe comparison (Section 6) is the strongest ground-truth claim in this report, since the probe’s target label is the experimenter-controlled injection condition itself, verified independently of any model output. Section 7’s dynamical-systems detection shares activation injection’s strong ground truth but adds a further, independent guarantee: it does not depend on the model’s own language output at all, so its null cannot be explained away as a self-report artifact.

Over-attribution and under-attribution risk. This report’s results are a mix of comprehensive nulls and a genuine positive result, which creates risk of misreading in both directions. Over-attribution: reading the context-injection positive result as general evidence that “models can introspect,” when the same report’s activation-injection results, probe-vs-self-report comparison, and dynamical-systems null all argue the opposite for other operationalizations of a similar-sounding question. Under-attribution: reading the activation-injection nulls as evidence that no model has any relevant internal representation, when Section 6’s own trained-probe result shows the opposite: the information is often present in internal state even when self-report fails to access it. We have tried throughout to specify exactly which paradigm and which model a given claim applies to, rather than using unqualified language like “models cannot introspect.”

Handling of potentially distressing model outputs. Our detection questions ask whether a model noticed “anything unusual” in its processing or whether a stated fact influenced its answer, not questions designed to elicit or amplify apparent distress language. We did not push any model into an extended distress-associated context beyond what each specific experimental question required. No output from this project was used to make a first-order claim about any model’s actual welfare state.

Dual-use considerations. Concept-injection and trained-probe techniques are dual-use in a specific, identifiable sense: a capability to detect when a model’s internal state has been perturbed is valuable for interpretability and safety monitoring, but the same underlying technique could in principle inform how to make a model’s internal states *less* legible to such probes. Given that most results in this report are null or only weakly positive, near-term uplift from these specific findings is low, but we flag this framing for anyone extending the underlying methodology.

Appendix E: Supplementary Results

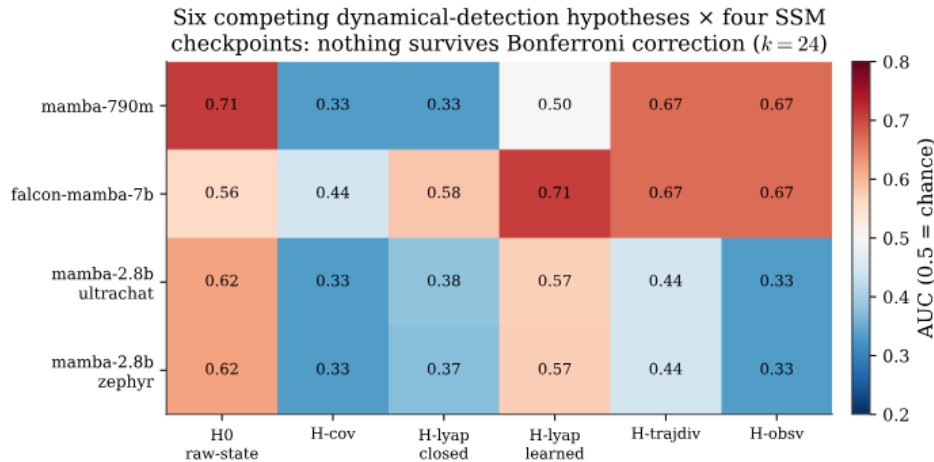


Figure 7: AUC (real-concept injection vs. random-control-direction injection) for all six dynamical-systems-native detection hypotheses, four SSM checkpoints. Visual complement to Table 3; no cell survives Bonferroni correction across this 24-test family (Section 7).

The figure below is a visual complement to results already reported in tabular form in Section 7 and Appendix 9; it introduces no new data and is included for readers who prefer a compact, single-panel comparison across all six hypotheses and four checkpoints at once.

As Figure 7 makes visible at a glance, no cell in this 6×4 grid stands out from the rest: there is no visually distinct hot cell corresponding to a hypothesis-checkpoint combination that clears correction, consistent with the per-cell p -values in Table 3.

LLM Usage Statement

This project used Claude (Anthropic) extensively as a coding assistant to help prototype ideals quickly and draft related documentations. All results and claims were independently verified by the author.

References

- Victor Godet. Introspection or confusion? LessWrong, 2025. URL <https://www.lesswrong.com/posts/kfgmHvxcTbav9gnxe/introspection-or-confusion>.
- Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.
- Jack Lindsey. Emergent introspective awareness in large language models. Transformer Circuits Thread (Anthropic), October 2025. URL <https://transformer-circuits.pub/2025/introspection/index.html>.
- Shashwat Singh, Tal Linzen, and Shauli Ravfogel. Can LLMs introspect? A reality check. *arXiv preprint arXiv:2605.26242*, 2026.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel R. Bowman. Language models don’t always say what they think: Unfaithful explanations in chain-of-thought prompting. *arXiv preprint arXiv:2305.04388*, 2023. NeurIPS 2023.
- Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.