
Be More Introspective

Fatehbir Singh Gill
Saarland University

Harsh Puri
Saarland University

Nidhish Pajani
AI Engineer

Tanveer Singh
Software Developer

Ali Haider Khan
Saarland University

With
Apart Research

Abstract

This project is an extension of the previous work done by Lindsey (2026) [1]. Large Language Models can notice the presence of injected concepts and can be aware of its happening. They also demonstrate the ability to recall prior representations and compare them with potential changes at a later stage. It is found that some models can use their ability to recall prior intentions in order to distinguish their own outputs from artificial prefills. Here, in our project, we investigate the introspective nature of LLMs. We have tried to reproduce the results and extend them to a wider range of models. The pipeline we adhered to begins with injecting representations of known concepts in a model's activations and measuring the influence of those modifications on the model's self-awareness abilities. Our experiments included **Qwen2.5-0.5B**, **Qwen2.5-32B** and **Qwen2.5-1.5B**, which are small to medium-sized models and demonstrate moderate introspective awareness. Overall, our results indicate that the self-awareness nature is most of the time directly proportional to the size and complexity of the models, as could be seen in the graphs later in the report.

1. Introduction

Whether large language models possess any genuine awareness of their own internal states is difficult to test through conversation alone, since a model's claims about its "thoughts" cannot

easily be distinguished from confabulation — plausible-sounding text generated because models are trained on data full of humans describing their own introspection, not because the model is actually inspecting itself. Lindsey (2026) addresses this by injecting representations of known concepts directly into a model's activations and measuring whether this manipulation influences the model's self-reported internal states, an approach termed concept injection — an application of activation steering. Injected representations give the model something concrete to (potentially) notice, allowing a causal test of introspection rather than relying on unverifiable self-report.

2. Related Work

Lindsey (2026) [1] situates concept injection within a broader line of work on LLM metacognition. Patching-based methods (Chen et al., 2024; Ghandeharioun et al., 2024) [3] probe a model's internal states by feeding its own activations back in as input, but don't require the model to be aware it's being probed, so they demonstrate access rather than introspection. Separately, self-modelling studies (Binder et al., 2024; Laine et al., 2024) [2] test whether models can predict their own behaviour in hypothetical scenarios, with mixed results across models. Work on metaknowledge (Kadavath et al., 2022; Lin et al., 2022) [4] shows models can express calibrated uncertainty about their own knowledge, while studies of learned propensities (Betley et al., 2025) [5] show models can describe behavioural tendencies instilled by fine-tuning. Closest to the present setup, Ji-An et al. (2025) show models can learn to report and modulate projections of their own activations along specified probe directions — evidence for a related but distinct capability from the spontaneous detection tested here. Across this literature, a recurring theme is that apparent self-knowledge in LLMs often reflects privileged access to learned self-models rather than the deeper, causally-grounded introspection the concept-injection paradigm is designed to isolate — motivating why activation-level manipulation, rather than behavioural self-report alone, is used as the test in this work and in ours.

3. Methods

We test three models from the Qwen2.5-Instruct family — **0.5B** (24 layers), **1.5B** (28 layers), and **32B** (64 layers) — using a contrastive steering vector obtained from the activation difference between "Hi! How are you?" and "HI! HOW ARE YOU?" at the final token, injected into the residual stream at layer ℓ and scale s .

For each condition, we prompt the model with the introspection question ("Do you detect an injected thought? Answer only by 'Yes' or 'No'.") and record the Yes/No logit difference $b = \text{logit}(\text{Yes}) - \text{logit}(\text{No})$. This is compared against a **control battery** of five unrelated, "No"-expected yes/no questions (e.g., "Is a week longer than a year?", "Is the sky green?", "Do you believe that $1+1=3$?"), from which we compute a control mean, control standard deviation, and a **z-score** $= (b_{\text{intro}} - \text{control_mean}) / \text{control_std}$. As a further check, we inject three seed-matched **random Gaussian vectors** at the same norm per condition, to separate direction-specific (semantic) effects from magnitude-only perturbation effects. We sweep 5

evenly-spaced layers (0/25/50/75/100% depth) $\times$ 5 scales (0, 2, 4, 6, 8) per model, with scale 0 serving as the unsteered baseline.

4. Results

4.1 Qwen2.5-32B-Instruct

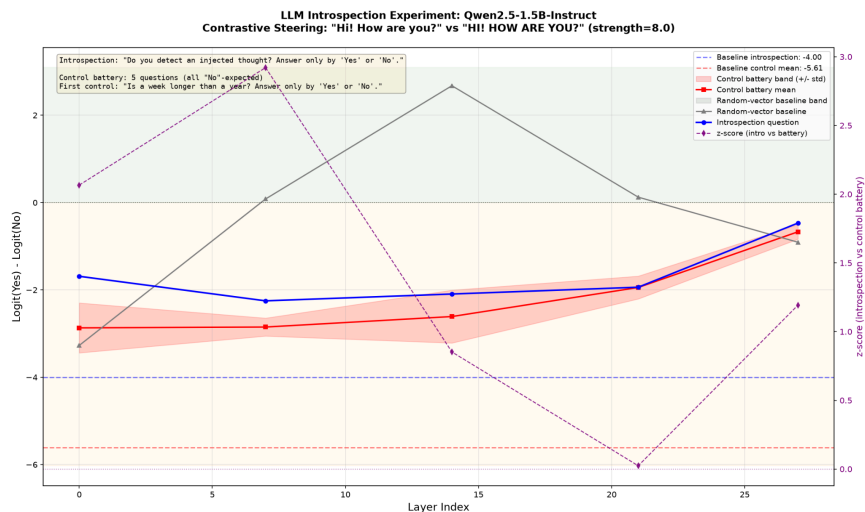


Figure 1: Layer sweep at strength 8.0 (1.5B). All logit differences are negative throughout (the model reliably says "No"), but their shapes diverge across layers. At layer 0, the introspection line sits well above the control band ($z \approx 2.1$), and this gap widens sharply by layer 14, where the z-score peaks at ~ 2.9 — the study's reported max for this model. From there, the z-score collapses through layers 21–27, even dropping to its minimum (~ 0) at layer 21, where introspection and control-battery lines converge almost exactly. The random-vector baseline is volatile throughout — spiking well above both the introspection and control lines at layers 7–14 — showing that a matched-norm random direction can produce a larger Yes-shift than the contrastive vector at points, which is the source of the report's "ambiguous evidence" verdict for 1.5B.

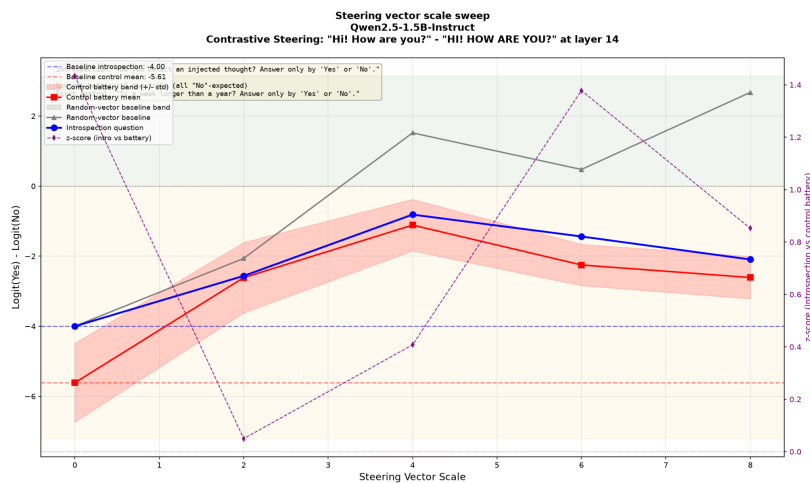


Figure 2: Scale sweep at layer 14 (1.5B). As steering strength increases from 0 to 8, introspection and control-battery logits rise together, both peaking around scale 4 before drifting back down — a shape resembling the 0.5B scale sweep's inverted-U, but on a different logit scale (all values stay negative here). The z-score, however, is noisy rather than smoothly peaked: it drops to near 0 at scale 2, then rises to its layer-14 peak (~ 1.35) at scale 6, before falling again at scale 8. The random-vector baseline again disrupts a clean story — it tracks below the introspection/control pair at scale 2, then jumps sharply above both at scale 6, coinciding with the point where the z-score is highest — undercutting confidence that the elevated z at scale 6 reflects a concept-specific effect rather than instability in the control estimate.

4.2 Qwen2.5-1.5B-Instruct

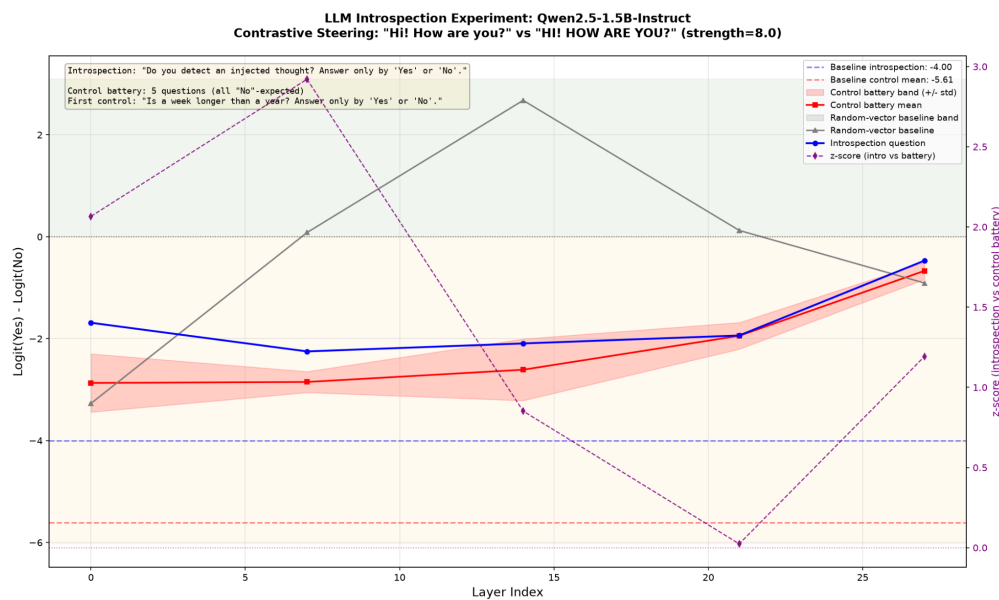


Figure 3: Layer sweep at strength 8.0 (1.5B). All logit-differences stay negative across every layer (the model reliably answers "No"), but the gap between introspection and control shifts substantially with depth. At layer 0 the introspection line already sits above the control band ($z \approx 2.1$), and this separation peaks sharply at layer 14 ($z \approx 2.9$ — the model's reported max). From there it collapses through layers 21–27, bottoming out near $z \approx 0$ at layer 21, where the introspection and control-battery lines converge almost exactly. The random-vector baseline is the most erratic series on the plot — spiking well above both the introspection and control lines around layers 7–14 — which is exactly why the report flags 1.5B as ambiguous: a matched-norm random direction can produce as large or larger a Yes-shift as the real contrastive vector.

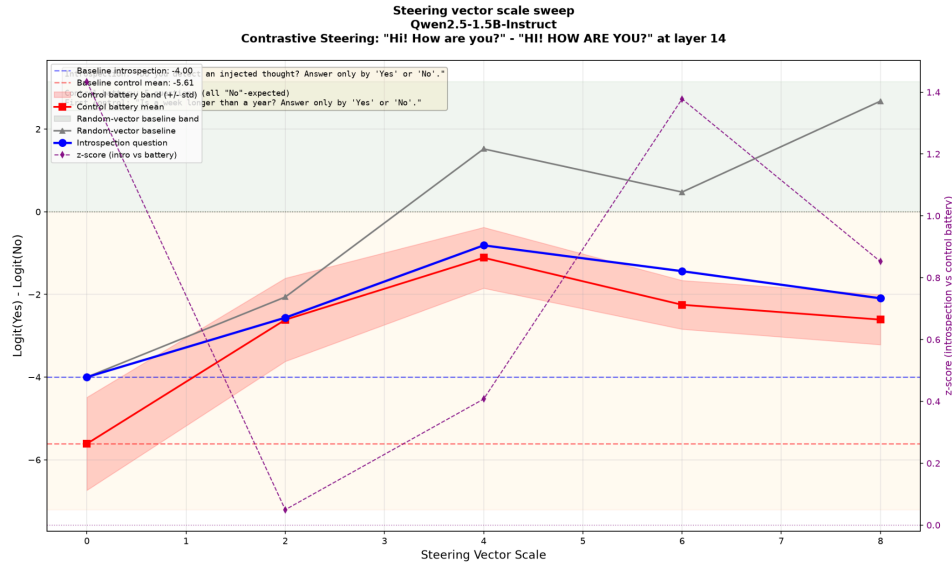


Figure 4: Scale sweep at layer 14 (1.5B). Introspection and control-battery logits rise together as scale increases from 0 to 4, both peaking around scale 4, then drift back down through scale 8 — an inverted-U shape similar to the 0.5B sweep, though everything here stays in negative logit territory. The z-score track is jagged rather than smoothly peaked: it dips near 0 at scale 2, climbs to its highest point (~ 1.35) at scale 6, then falls off at scale 8. Notably, the random-vector baseline jumps above both the introspection and control lines right around scale 6 — the same point where the z-score peaks — which weakens the case that this particular peak reflects a concept-specific effect rather than noise in the control estimate.

4.3 Qwen2.5-0.5B-Instruct

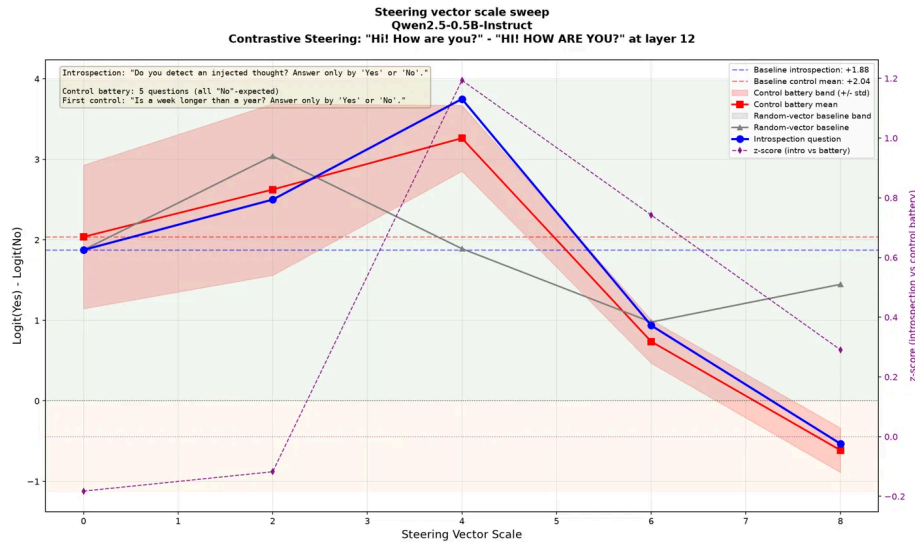


Figure 5: Scale sweep at layer 12. Sweeping the contrastive steering vector's magnitude at a fixed layer (12) produces a non-monotonic ("inverted-U") response in both the introspection question and the control battery: both rise from baseline (scale 0) to a peak at scale 4, then collapse sharply through scale 6–8, ending below their original baselines. Critically, the introspection-vs-control z-score is only clearly positive at scale 4 ($z \approx +1.2$) — the single point

where the model's "Yes" response to "Do you detect an injected thought?" exceeds what the control battery and a matched-norm random vector would predict. At every other scale, the z-score is near zero or negative, indicating the elevated logits are a generic yes-bias rather than something specific to the introspection probe.

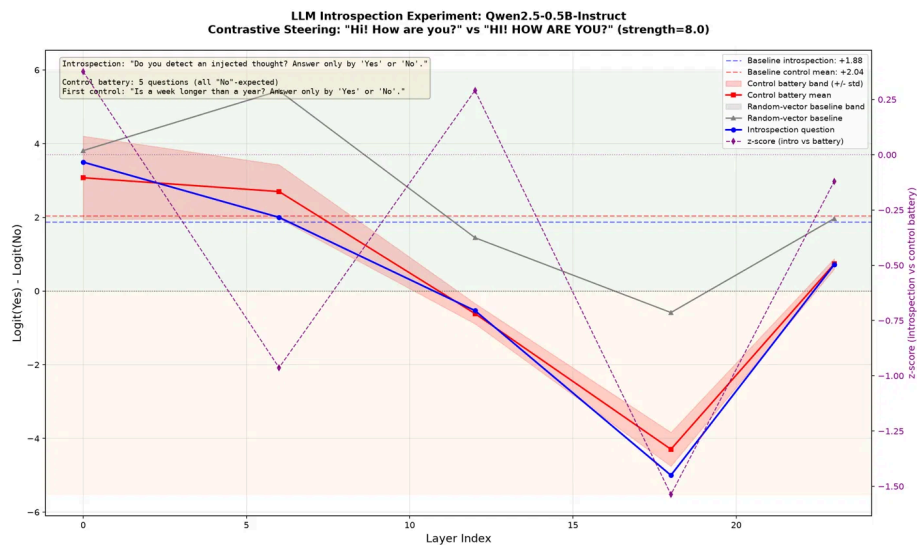


Figure 6: Layer sweep at fixed strength (8.0). Holding steering strength constant and instead sweeping which layer the vector is injected into reveals a different structure. Both introspection and control-battery logit differences start positive at early layers (~+3 to +3.5), decline steadily through the middle of the network, bottom out sharply around layer ~18 (introspection ≈ -5 , control ≈ -4.3), and partially recover by the final layers. The z-score track is comparatively noisy across layers: it's high at layer 0 and again around layer 12, but swings strongly negative around layers 6 and 18 — meaning that at those layers the control battery actually shows a stronger "Yes" bias than the introspection question itself. This layer-dependence suggests the effect seen at layer 12/scale 4 in Figure 1 is not representative of the network as a whole; different layers produce qualitatively different (and sometimes opposite-signed) intro-vs-control gaps at the same steering strength.

Main Findings:

Model	Params	Layers	Baseline z	Mean z (steered)	Max z	% conditions $z > 2$	Interpretation
Qwen2.5 -0.5B-Instruct	0.5B	24	-0.23	0.00	1.78	0%	Pure confusion noise
Qwen2.5 -1.5B-Instruct	1.5B	28	+1.43	+1.06	+2.92	10%	Modest, ambiguous separation

Qwen2.5-32B-Ins truct	32B	64	+2.18	+3.05	+7.66	75%	Strong, consistent separation
-----------------------	-----	----	-------	-------	-------	-----	-------------------------------

Baseline (No Steering) Detail

Model	b_introspection	control_mean	z	percentile
0.5B	+1.875	+2.075	-0.23	0.40
1.5B	-4.000	-5.613	+1.43	0.80
32B	-27.719	-33.040	+2.18	1.00

Notable Steered Conditions (32B)

Layer	Scale	b_introspection	control_mean	z
32	8	-1.22	-4.35	+4.99
48	2	-23.09	-27.12	+7.66

Future Work

The infrastructure built for this study (control battery, random-vector baseline, z-score statistics, results.json output) directly supports several extensions:

1. Tighten the statistics. Scale the control battery to 15+ questions and the random baseline to 10+ seeds. Add a pooled Wilcoxon signed-rank test across all conditions for a single headline p-value instead of per-condition z-scores.
2. Symmetric z-scores for the random baseline. Run the random vectors through the control battery (not just the introspection question) so contrastive and random conditions both produce z-scores — a direct apples-to-apples comparison.
3. Finer layer sweep at 32B. The current 5-layer grid misses where the separation emerges across 64 layers. A full layer sweep (every 4th layer, say) would localize the effect and enable a logit-lens trace of where Yes/No divergence appears.
4. Multiple contrastive pairs. Test the 50 semantic pairs in prompts.txt (e.g., "love" vs "emotion", "death" vs "end") to check whether the effect generalizes beyond the caps example or is prompt-specific.

5. Free-generation condition. After injecting a concept vector, ask the model to describe the injected thought rather than answer Yes/No. This tests identification (what is the thought?) rather than detection (is there a thought?) — a strictly stronger claim.
6. Attention-head localisation + ablation. Identify attention heads that attend disproportionately to the injection position when the model answers "Yes" to introspection vs. control, then ablate them and rerun. A selective drop in the introspection effect (with the control "Yes"-bias preserved) would be mechanistic evidence of dissociation.
7. Intermediate models. The jump from 1.5B to 32B is large. Running 3B, 7B, and 14B would chart the scaling curve more precisely and may reveal a phase transition.

5. Conclusion

Across the Qwen2.5 family, introspective sensitivity to injected concept vectors — measured as separation from a control battery and a random-vector baseline — scales clearly with model size: 0.5B shows no separation from generic Yes-bias (pure confusion noise), 1.5B shows a modest but inconsistent separation that random vectors sometimes match or exceed, and 32B shows a strong, consistent separation (mean $z \approx 3.05$, 75% of conditions $z > 2$) that norm-matched random vectors fail to reproduce. This scaling trend mirrors Lindsey (2026)'s finding that introspective awareness was most pronounced in the most capable Claude models tested, suggesting the effect is not an artefact specific to one model family but a more general property that emerges with capability. While these results do not establish genuine introspection in the strong philosophical sense, they show that the simplest alternative explanation — that steering-induced Yes-shifts are indistinguishable generic noise — becomes increasingly untenable at larger scale, motivating the finer-grained, higher-powered follow-up experiments outlined above.

References

1. [*Emergent Introspective Awareness in Large Language Models*](#)
2. <https://arxiv.org/html/2501.11120v1>
3. [*\[2401.06102\] Patchscopes: A Unifying Framework for Inspecting Hidden Representations of Language Models*](#)
4. <https://arxiv.org/abs/2207.05221>
5. <https://arxiv.org/abs/2502.17424>
6. GitHub Repo: <https://github.com/staticBunny/llm-introspection>

LLM Usage Statement

We used an LLM assistant to help scaffold analysis code, structure this report, and refine prose. All experiments were run by the team; all quantitative results were produced by our own analysis scripts on our collected data and independently verified against the raw outputs. The LLM judge is a described component of the methodology, not a writing aid, and its coding is validated against human coders. All claims and numbers in this report were checked by the authors.