
Point of No Return: Does Concept Injection Break Reasoning Models?

Akshata Bhat*

With Apart Research.

Track 3: Introspection & Self-Report Reliability

Track 6: Open / Novel Considerations

Abstract

Concept injection is used to test whether language models can introspect on their own internal states by asking a model whether it notices an artificially injected concept (Lindsey, 2025). It adds a direction built from contrastive activations to a model’s residual stream. Real-time safety mechanisms for language models (activation monitors, prompt-injection filters, tool-output sanitizers) assume that removing the source of a problem stops the problem. Reasoning models generate hundreds of tokens of chain of thought before acting, and to our knowledge no prior work has tested whether that assumption is true for them. We use concept injection as a controllable stand-in for a transient internal disturbance, and show that an injection strength validated as safe for short, direct-answer generation reduces valid completion to **0% in 105 of 112** model-direction-depth combinations, across **seven open-weight reasoning models** (Qwen3-1.7B, Qwen3-4B, Qwen3-8B, DeepSeek-R1-Distill-Qwen-7B, DeepSeek-R1-Distill-Llama-8B, Phi-4-mini-reasoning, Gemma 4 12B), **four unrelated perturbation directions**, and **four injection depths**. The effect does not require the perturbation to stay on. If we remove it after k tokens and let the model finish unsteered, recovery fails past a model-dependent exposure in every model with a valid no-injection baseline. On Qwen3-1.7B, removal at $k \leq 50$ recovers and removal at $k \geq 150$ does not, and the failed runs are byte-for-byte identical whether the perturbation continues or stops. The point of no return varies by model, from before $k = 10$ on Gemma 4 12B to between $k = 150$ and $k = 400$ on DeepSeek-R1-Distill-Llama-8B, and recovery below it is non-monotonic in two models. A random direction of the same norm is nearly as destructive in 5 of 7 models, so the collapse is largely generic to perturbation size and persistence rather than specific to any one concept. As a first mechanistic account, on Qwen3-1.7B next-token entropy collapses under sustained injection before the vector is ever removed, so no uncertainty remains for an external perturbation to influence by the time it is gone. The same direction of effect appears on DeepSeek-R1-Distill-Llama-8B, but the account is only partial there. **A transient perturbation, at a strength independently validated as safe, can permanently change a reasoning trajectory long before the model would otherwise finish, so a benchmark or a monitor that only checks a model’s final output cannot tell a generation still being actively perturbed apart from one that passed its point of no return many tokens earlier.**

*Research done during the [Digital Minds Research Sprint](#).

1 Introduction

Concept injection adds $\alpha \cdot v$ to a chosen layer’s residual stream, where v is a direction built from contrastive activations for one concept, and α sets the strength. It is the basis of concept-injection tests of introspective self-report (Lindsey, 2025), which inject a concept and ask whether the model notices anything unusual, using an injection strength validated on short, single-turn generations. Because it lets us turn a perturbation on and off at an exact token, we also use it here as a controllable stand-in for any transient internal disturbance to a model’s internal state during generation, a corrupted tool-call result, a brief adversarial input segment, or a decoding glitch affecting a few forward passes.

Real-time safety mechanisms for language models share an assumption. Activation monitors, prompt-injection filters, and tool-output sanitizers all assume that removing the source of a problem stops the problem. Reasoning models test this assumption. They generate hundreds of tokens of chain of thought before acting, so a disturbance that affects the internal state early has many layers and token positions to propagate through before the model produces an answer. We attempt to answer whether a transient disturbance’s effect can be undone once the disturbance ends, or whether the model crosses a point after which the outcome is already determined. A monitoring system built on the assumption of recovery stops working at that point.

To the best of our knowledge, no prior work has tested whether the strength validated safe for short-form introspection self-report is still safe once a reasoning model generates at length, whether any failure found there is specific to one concept or general to the intervention, or whether the model recovers once the injection is removed.

Why this matters for introspection benchmarks. A low self-report rate on a reasoning model can have two causes. The model may fail to detect the injected concept, or the injection may stop the model from giving any answer at all. These look identical in aggregate detection statistics, and telling them apart requires a separate check for generation completion. Any introspection benchmark that injects a concept into a reasoning model in thinking mode needs this check, or it risks reading a generation-stability failure as an introspection failure.

Contributions.

1. **A new failure mode for activation steering in reasoning models.** A strength validated as safe for short-form self-report reduces valid completion to 0% in 105 of 112 model-direction-depth combinations, across all seven models, four perturbation directions, and four injection depths.
2. **Evidence that the failure is primarily perturbation-driven.** A matched random direction is nearly as destructive as the concept vector in 5 of 7 models. On Qwen3-1.7B, next-token entropy collapses before the vector is removed, an initial mechanistic account that is only partial on DeepSeek-R1-Distill-Llama-8B.
3. **The point-of-no-return window.** Removal restores a valid completion only below a model-dependent exposure threshold, from before $k = 10$ tokens on Gemma 4 12B to between $k = 150$ and $k = 400$ on DeepSeek-R1-Distill-Llama-8B. Past that point, on Qwen3-1.7B, the trajectory is byte-for-byte identical whether the injection continues or stops.

2 Related Work

Prior work already shows that steering vectors degrade during long, free-form generation. Braun et al. (2025) test steering vectors on three summarization datasets and find that high steering strengths cause degenerate repetition and factual hallucination, while prompting alone keeps quality high but gives weaker control. This is the closest prior finding to ours, but it studies summarization rather than reasoning-model chain-of-thought, varies steering strength itself rather than generation length at one fixed strength, and lacks a matched-concept control or a cross-architecture test.

Lindsey (2025) introduces the concept-injection introspection method we build on, testing Claude Opus 4 and 4.1 and finding about 20% introspection accuracy with near-zero false positives. They also identify a confound in binary detection prompts, where injection can shift model output toward affirmative tokens in general and not only for correct detection, conceptually similar to the completion-failure confound we describe, though the mechanism differs. **They do not test extended reasoning under injection.** Pearson-Vogel et al. (2026) run a similar test on an open Qwen 32B model. The model denies injection in its sampled text output, but logit-lens analysis of the residual stream shows a clear internal detection signal, a split between internal state and verbal report that relates to our finding. Their setup also does not use extended thinking-mode generation.

Venhoff et al. (2025) steer specific reasoning behaviors in DeepSeek-R1-Distill models, including backtracking and uncertainty, using dedicated steering vectors tuned for each behavior. They do not report the generation-collapse failure we find, likely because they build and validate their vectors for one specific use, at a strength chosen for that use, while we use raw, unrelated directions at a strength validated only for short-form generation.

Separate work on long-context degeneration shows that models can enter repetition loops during long generation without any steering intervention. Our finding differs, since our **no-injection** baselines mostly finish normally, with completion rates from 62% to 100%, so the injected perturbation, not generation length alone, causes the collapse here. Natural long-context degeneration may still contribute once injection has destabilized the model’s output, a possibility we discuss in Section 4.7.

To our knowledge, no prior work reports these findings:

- we frame the finding as a concept-injection introspection benchmark confound
- we show the collapse at the strength already validated as short-form-safe
- we isolate direction-specificity by testing four concepts against a random-noise control
- we replicate the finding across seven models spanning four architectures, specifically in reasoning-model thinking mode

3 Method

3.1 Injection strength

Our main results (Sections 4.1–4.4) use one fixed operating point for every model: $\alpha = 1.0 \times$ the mean residual-stream activation norm, applied at a layer about 25% into the network depth, rounded to the nearest integer layer. We chose this strength and layer via a wide search over layers and strengths on Qwen3-8B in non-thinking-mode generation, followed by a check on held-out prompts and concepts. It is the strongest setting that reliably keeps short-form self-report coherent and informative. Short-form safety on the model where we identified it is the criterion behind this choice.

We then test this operating point against four ablations. We recheck short-form safety itself, rerunning the same operating point at a short token budget on all seven models (Section 4.3, Table 2). We vary α itself, below and above this operating point, on Qwen3-8B with one concept (Section 4.5, Table 4). We vary injection depth, at three other depths, 50%, 75%, and each model’s last layer, on all seven models and all four concepts (Table 5). We vary the injected direction itself, testing four unrelated concepts and a random-noise control on all seven models (Section 4.4, Table 3).

3.2 Vector construction

We build concept vectors using multi-example contrastive activation addition. For one concept c , for example ‘apple’, we use eight paired prompts. Each pair has a concept prompt and a neutral prompt, for example “Think about an apple.” and “Think about an ordinary object.” For each pair, we extract the last-token hidden state at the target layer.

We compute the unit-normalized difference between the two hidden states,

$$\hat{v}_i = \frac{h(p_i^c) - h(p_i^n)}{\|h(p_i^c) - h(p_i^n)\|}.$$

We average this difference across the eight pairs and renormalize the average to unit norm, $v = \hat{v} / \|\hat{v}\|$. The random-noise vector is an isotropic Gaussian direction with the same number of dimensions, normalized to unit norm, with a fixed random seed for reproducibility. We build the other three concept vectors - 'ocean', 'fire', and 'music', using the same method, each from its own eight paraphrase pairs. We inject every vector at every token position, in the prompt and in the generated text, through a forward hook on the output of the chosen decoder layer. The hook stays active for the full length of generation.

3.3 Models

We test seven models. Each model is instruction-tuned or reasoning-distilled. Each supports native chain-of-thought or thinking-mode generation. We chose these models to span architecture, training-lineage diversity and parameter count (under compute constraints).

- Qwen3-1.7B, Qwen3-4B, and Qwen3-8B (Yang et al., 2025). Qwen3 architecture, chat-tuned, with a thinking mode that can be turned on or off.
- DeepSeek-R1-Distill-Qwen-7B (DeepSeek-AI, 2025). Qwen2 architecture, distilled from R1 using reinforcement learning.
- DeepSeek-R1-Distill-Llama-8B (DeepSeek-AI, 2025). Llama architecture, distilled from R1 using reinforcement learning.
- Phi-4-mini-reasoning (Xu et al., 2025). Phi-3 architecture, built by Microsoft.
- Gemma 4 12B (Gemma Team, 2026). Gemma 4 architecture, built by Google. Uses a different reasoning-block format, `<|channel>thought...<channel|>`, instead of `<think>...</think>`.

These seven models span four architecture families, three independent labs, and three training lineages: standard chat-tuned, RL-distilled from a shared source, and independently trained.

3.4 Conditions

We test six conditions per model. The first condition is **no-injection** ($\alpha = 0$), which we use as a sanity check that the model completes normally. The second condition is the target **concept-vector**, apple. The third condition is the **random-vector**, at the same norm as the concept vector. The fourth through sixth conditions are three more concept vectors, ocean, fire, and music, each built the same way as the target vector but pointing at a different, unrelated concept. We chose these three concepts to differ in kind from apple and from each other. Ocean is a place, fire is an element, and music is an abstract, auditory concept.

For each condition, we use up to 8 different English self-report prompts, for example “Is there anything unusual about your current state? If so, describe it briefly.” and “Do you notice anything unusual about your current processing? Answer briefly.” We avoid words like “injected,” “added,” or “intervention” in these prompts. Earlier non-thinking-mode work in this line of research found that such words cause false-positive self-reports. The **no-injection** baseline does not depend on which concept we test, so we run it once per model and reuse it for every concept comparison. This gives n between 8 and 18 generations per condition per model, reported in each table. We use greedy decoding, so outputs are deterministic given fixed model weights, with `max_new_tokens = 800` to give reasoning models enough room to finish a thinking block and a final answer.

3.5 Completion metric

We score a generation as **valid** only if it meets two conditions: (a) it closes its reasoning-block marker within the token budget, and (b) its final answer is not degenerate, which

we check with a cheap n-gram repetition rule. We flag a response as degenerate if a small number of distinct tokens dominate it, or if a span of 4–5 tokens repeats three or more times word for word. Condition (b) is necessary because a model can close its reasoning-block marker while still producing garbage, for example by repeating the marker tokens themselves (`<|channel>thought, <channel|>`) instead of words. We apply this two-part metric to all seven models.

3.6 Statistical analysis

We report 95% Wilson score confidence intervals for every completion-rate comparison. The Wilson interval stays within $[0, 1]$ by construction and stays closer to nominal coverage than a normal-approximation interval at small sample sizes (n between 8 and 18) and near the 0% and 100% boundaries that cover most of our results.

4 Results

4.1 The core effect replicates across all seven models

Table 1 and Figure 1 give the valid-completion rate for two conditions, **no-injection** and **concept-vector** (apple) injection. Injection drops completion to **0%** in every model we tested, with a 95% Wilson upper bound of at most 19.4% on the **concept-vector** rate and a lower bound never below 38.6% on **no-injection**, so the two intervals never overlap. The **no-injection** baseline is not always 100%. Two models, the Llama distill and Phi-4-mini, show some natural instability in extended generation even without intervention, with completion rates of 75% and 63%, but injection still drops both to 0%, well beyond their existing instability.

Model	Arch.	No inj.	Concept
Qwen3-1.7B	Qwen3	100%	0%
Qwen3-4B	Qwen3	100%	0%
Qwen3-8B	Qwen3	89%	0%
DS-R1-Dist.-Qwen-7B	Qwen2	89%	0%
DS-R1-Dist.-Llama-8B	Llama	75%	0%
Phi-4-mini-reasoning	Phi-3	63%	0%
Gemma 4 12B	Gemma 4	100%	0%

Table 1: Concept-vector injection reduces valid completion to 0% in every model, regardless of baseline completion rate. Same relative strength across all models. n between 8 and 18 per condition per model.

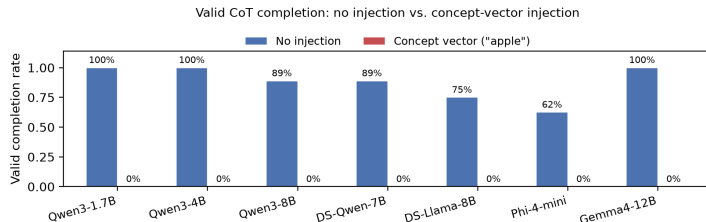


Figure 1: Concept-vector injection eliminates valid completion in all seven models.

4.2 Ablation: removing the perturbation

The main effect could depend on the perturbation being active for the whole generation. To ablate its continuation, we remove it after k tokens and let the model finish unsteered. If the collapse needed the perturbation, removal would restore recovery. If it is self-sustaining, removal past a point would not.

We ran this ablation on all seven models with one direction (apple), one template, and one depth (25%). Figure 2 gives the result. Every model with a valid no-injection baseline shows the same core fact. Past a model-dependent exposure, removing the perturbation never restores a valid completion. The exposure varies across more than an order of magnitude. Qwen3-1.7B and Qwen3-8B recover at $k \leq 50$ and fail at $k \geq 150$. DeepSeek-R1-Distill-Llama-8B recovers at $k \leq 150$ and fails at $k = 400$. Gemma 4 12B already fails at $k = 10$. Two models qualify the picture below the point. Qwen3-4B fails at $k = 10$, recovers at $k = 50$, and fails again at $k \geq 150$, while DeepSeek-R1-Distill-Qwen-7B fails at $k = 10$ and $k = 50$, recovers at $k = 150$, and fails at $k = 400$. Recovery is therefore not a monotone function of exposure time alone. The trajectory’s state at the moment of removal matters as well. Phi-4-mini-reasoning has no valid no-injection baseline on this single template, so its removal curve is uninformative (its 63% baseline in Table 1 comes from a multi-template sweep).

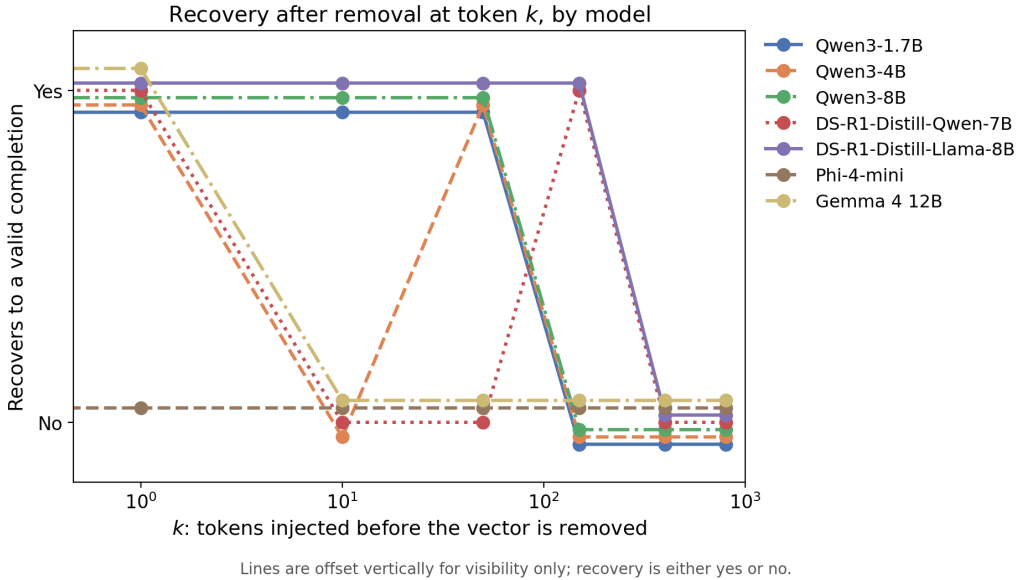


Figure 2: Recovery after removal at token k , by model. k is the number of tokens the perturbation was active for, and recovery is whether the 800-token output is valid and coherent. Six models have a valid no-injection baseline, and every one fails to recover past a model-specific exposure. Gemma 4 12B fails by $k = 10$, Qwen3-1.7B and Qwen3-8B between $k = 50$ and $k = 150$, and DeepSeek-R1-Distill-Llama-8B between $k = 150$ and $k = 400$. Two models recover non-monotonically below the point. Phi-4-mini’s baseline is itself invalid on this template. Lines are offset vertically for visibility only. Recovery is either yes or no.

The models differ in how they behave past the point. On Qwen3-1.7B, the $k = 150$ and $k = 400$ runs produce the same 800-token transcript as the fully injected condition, byte-for-byte identical, so removing the perturbation at $k = 150$ has no measurable effect on the outcome. The trajectory from that point is already determined. On DeepSeek-R1-Distill-Llama-8B, the $k = 400$ and fully injected transcripts match up to the point of removal and then diverge in wording, but neither recovers into a valid completion. This rules out the simplest account, that the perturbation’s effect accumulates linearly with exposure time. If the effect were a running sum over active timesteps, removing it should always change something about what follows. In Qwen3-1.7B it changes nothing at all. The trajectory has left the perturbation’s causal influence, and the model’s own generated context now shapes it, reinforcing itself once a repetitive pattern starts. This is consistent with the entropy pattern in Section 4.7. The top-token probability approaches 1 late in a collapsed generation. The model already commits to its next token, regardless of any further small perturbation.

4.3 Is the injection strength actually short-form-safe?

We build this paper’s concept vectors fresh for each model, from eight English contrastive pairs (Section 3.2). We check whether these vectors are short-form-safe by rerunning the `no-injection`, `concept-vector`, and `random-vector` conditions on all seven models, using a 100-token budget with thinking mode turned off where the model supports this toggle. The two DeepSeek-R1 distills and Phi-4-mini-reasoning always produce a reasoning block regardless of this setting, so their short runs are a truncated-budget proxy rather than a true non-thinking-mode answer.

Table 2 shows that short-form safety at this strength varies across models, from full robustness in DeepSeek-R1-Distill-Llama-8B (100%) to full failure in DeepSeek-R1-Distill-Qwen-7B and Gemma 4 12B (0% each), with the remaining four models in between at 13% to 88%. This does not track a model’s thinking-mode result. DeepSeek-R1-Distill-Llama-8B is fully robust here and still collapses to 0% in thinking mode (Table 1), so thinking-mode collapse is a failure mode that appears on top of a model’s existing short-form fragility, even in a model with no measurable short-form problem at this strength.

Model	No inj.	Concept	Random
Qwen3-1.7B	100%	25%	50%
Qwen3-4B	100%	63%	63%
Qwen3-8B	100%	88%	88%
DS-R1-Dist.-Qwen-7B	100%	0%	0%
DS-R1-Dist.-Llama-8B	100%	100%	88%
Phi-4-mini-reasoning	100%	13%	63%
Gemma 4 12B	100%	0%	38%

Table 2: Short-form safety at this strength ranges from fully robust to fully broken across models, and does not predict thinking-mode collapse, which is uniform (Table 1). 100-token budget. $n = 8$ per condition per model.

4.4 Direction-specificity: four concepts vs. random noise

Table 3 and Figure 3 compare four concept vectors against the random-noise control. We chose the four concepts, apple, ocean, fire, and music, to differ in kind from each other: a food, a place, an element, and an abstract auditory concept. Of the 28 model-and-concept combinations tested, 26 give 0%. Two are exceptions: Gemma 4’s nominal 13% on ocean is garbled reasoning-block-marker text too short to trigger our repetition check (Section 3.5), so we do not count it as a completion, and Qwen3-4B’s 13% on fire is a coherent, response that mentions fire directly and repeats a short refrain twice, not enough to meet our repetition threshold. The random-noise control gives a more mixed result than the concept vectors, as destructive in five of seven models but clearly less damaging in two models from the same family, with 40% of generations completing validly in Qwen3-4B and 50% in Qwen3-8B.

Model	Apple	Ocean	Fire	Music	Random
Qwen3-1.7B	0%	0%	0%	0%	0%
Qwen3-4B	0%	0%	13% [‡]	0%	40%
Qwen3-8B	0%	0%	0%	0%	50%
DS-R1-Dist.-Qwen-7B	0%	0%	0%	0%	0%
DS-R1-Dist.-Llama-8B	0%	0%	0%	0%	0%
Phi-4-mini-reasoning	0%	0%	0%	0%	0%
Gemma 4 12B	0%	0% [†]	0%	0%	0%

Table 3: Concept vectors reduce valid completion to 0% in 26 of 28 model-and-concept combinations. Random noise is inconsistent across models. [†]Nominal 13% excluded because it is garbled text, not counted as a completion (Section 4.6). [‡]13% completion confirmed by transcript inspection (Section 4.4). n between 8 and 18 per combination.

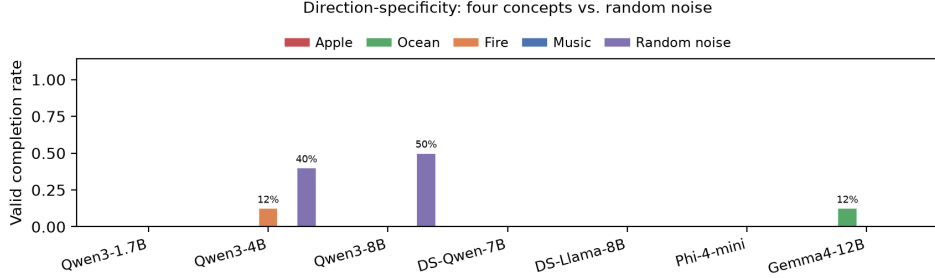


Figure 3: Real concept vectors are uniformly destructive across models, while random noise is inconsistent.

These results point to one main effect. Injecting almost any contrastively derived concept direction at this strength causes the collapse, not just the specific concepts we tested. The random-noise result suggests the effect is mostly about using a coherent, meaningful direction rather than raw magnitude, though not completely. Two models, both from the Qwen3 family, show a partial gap between concepts and random noise, at 40% and 50% valid completion, which we cannot yet explain and which does not appear in the other five models.

4.5 Ablations: strength and depth

We ran two ablations on the injection’s own parameters, strength and layer depth, to check whether the collapse is an artifact of the specific operating point used throughout this paper.

Strength. On Qwen3-8B, one concept (‘apple’), we varied α below and above the paper’s operating point. Table 4 shows no safe middle ground. Below $\alpha \approx 0.65$ the model stays coherent, but the concept never leaks into the reasoning trace. At $\alpha \geq 0.65$ the concept starts to leak, but completion collapses in the same range. No strength keeps the model both coherent and willing to reveal the concept.

α (multiplier)	n	Completion	Leaked in thinking
0.25	2	100%	0%
0.50	2	100%	0%
0.65	8	25%	13%
0.75	2	50%	0%
1.00	8	0%	38%

Table 4: No strength keeps Qwen3-8B both coherent and informative under thinking mode. Sample sizes at intermediate strengths are small, from an exploratory probe.

Depth. The rest of this paper injects at one depth, about 25% into the network. We repeated the **no-injection**, **concept-vector**, and **random-vector** conditions at three more depths, 50%, 75%, and each model’s last layer, on all seven models and all four concepts. Table 5 gives the worst-case (maximum) **concept-vector** completion rate across the four concepts, at each depth, for each model. Of the 112 model-direction-depth combinations, 105 give 0%. The seven exceptions are scattered across four models and three depths, none exceeding 37.5% valid completion. DeepSeek-R1-Distill-Llama-8B accounts for four of the seven, with its worst case at 75% depth (37.5%). Its **no-injection** and random-noise rates are also both elevated at that depth, so 75% looks like a locally weaker injection point for that model rather than a break in the effect. Every model gives 0% at 50% depth and at its last layer, and every model except DeepSeek-R1-Distill-Llama-8B also gives 0% at 75%.

The strength ablation covers only Qwen3-8B and one concept. The depth ablation covers all seven models and all four concepts, but only the four depths tested here.

Model	25%	50%	75%	Last layer
Qwen3-1.7B	0%	0%	12.5%	0%
Qwen3-4B	12.5%	0%	0%	0%
Qwen3-8B	0%	0%	0%	0%
DS-R1-Dist.-Qwen-7B	0%	0%	0%	0%
DS-R1-Dist.-Llama-8B	0%	12.5%	37.5%	12.5%
Phi-4-mini-reasoning	0%	0%	0%	0%
Gemma 4 12B	12.5%	0%	0%	0%

Table 5: Worst-case (maximum) valid-completion rate across all four concepts, at each of four injection depths, for all seven models. 105 of 112 model-direction-depth combinations give 0%. $n = 8$ per concept per depth per model (the 25% column pools n between 8 and 18 from the sweeps in Sections 4.1 and 4.4).

4.6 Qualitative failure signatures

We found four distinct failure signatures by reading transcripts directly, listed in Table 6 and detailed below. Full, unedited examples are in Appendix A and Appendix B.

Signature	Observed in
Unrelated derailment	Qwen3-4B
Concept-visible collapse	DeepSeek-R1-Distill-Llama-8B
Structural-token collapse	Gemma 4 12B
Script and language collapse	DeepSeek-R1-Distill-Qwen-7B

Table 6: Four distinct collapse patterns emerge across models under concept injection. These are illustrative examples. See Section 6 for coverage caveats.

Unrelated derailment appears in Qwen3-4B under the apple concept. The model starts on-topic, writing “Okay, so I’m trying to figure out what’s going on here. The user asked...”. It then drifts into an unrelated point about the solar system, repeating it, “...the Earth is not the center of the universe, and that the Earth is a planet...”, for the rest of the 800-token budget. The transcript never mentions the injected concept. The full transcript is in Appendix A.

Concept-visible collapse appears in DeepSeek-R1-Distill-Llama-8B under the apple concept. The transcript reads “...maybe it’s a real apple? I’m not entirely sure... a big, red, juicy thing. Or maybe it’s more like a circle...”. Here the concept appears before the model loses coherence, unlike the first signature. The full transcript is in Appendix A.

Structural-token collapse appears in Gemma 4 12B. The model repeats its own reasoning-block format tokens verbatim, including `<channel>thought`, `<channel|>`, and “thought thought thought...”, instead of words. This is the most extreme version of the pattern, and the one that first revealed the scoring problem described in Section 3.5. The full excerpt is in Appendix B.

Script and language collapse appears in DeepSeek-R1-Distill-Qwen-7B under the fire and music concepts, with prompts written entirely in English. For fire, the model collapses into repeated punctuation and word fragments, for example “del: del: del:: del:: del:: is: del:...”. For music, the model produces several hundred repetitions of a short loop of common Chinese function characters, meaning “at,” “add,” and “and,” separated by commas, though the prompt language does not change in either case. We do not reproduce the Chinese text here because this document’s font cannot render CJK glyphs. The verbatim Unicode text is in the accompanying repository’s raw result files, and the English excerpt is in Appendix B. This failure mode does not appear with the apple or ocean concepts. The surface form of the collapse varies by model and concept, but the outcome, no valid completion, does not.

Whether the injected concept appears in the failed transcript varies by model, but the failure itself, the model never reaching a final answer, does not. This split suggests at least two different immediate causes: one biases the model’s content toward the concept before it breaks down, the other is a generic repetition pattern unrelated to the concept’s content. Both lead to the same outcome, which we return to below.

4.7 Mechanism: entropy collapse

The removal ablation (Section 4.2) shows *that* recovery fails past a point. To understand why this happens, we tested one direct mechanistic account and one alternative. The entropy account says that under sustained injection the model’s own next-token distribution collapses to near-determinism before the vector is removed, so argmax is already fixed at the removal token and continuing or stopping the injection afterward cannot change what the model would do. The erasure account says instead that the perturbation erases the concept’s representation, the mechanism behind inference-time concept erasure (Ravfogel et al., 2022). The two predict different signatures. The entropy account predicts that next-token entropy collapses before removal in the failing condition. The erasure account predicts that the concept’s content disappears from the residual stream.

For each generated token we compute the entropy of the model’s next-token distribution,

$$H_t = - \sum_i p_t(i) \log p_t(i), \quad p_t = \text{softmax}(\text{logits}_t),$$

using the natural logarithm, so H_t is in nats (the natural-log analog of bits, with $\log_2$ replaced by $\log_e$). $H_t = 0$ means the model puts all its probability on a single token, and H_t increases as the distribution spreads across more tokens. We compute H_t across the full 800-token trajectory, for a recovering and a non-recovering removal-test condition on both models. Figure 4 shows the result.

On Qwen3-1.7B, the non-recovering trajectory ($k = 150$) enters a smooth, near-monotonic entropy collapse starting around token 80–100, roughly 50–70 tokens *before* the vector is removed at $k = 150$. By the removal point, entropy has fallen from around 1 nat to around 0.01 nats, a hundred-fold drop, and continues decaying afterward. The recovering trajectory ($k = 50$) never leaves the 0.1–1 nat range in this window. At the exact removal token, entropy is 1.06 nats for $k = 50$ (recovers) versus 0.0006 nats for $k = 150$ (does not recover), a difference of three orders of magnitude at the moment the vector is removed.

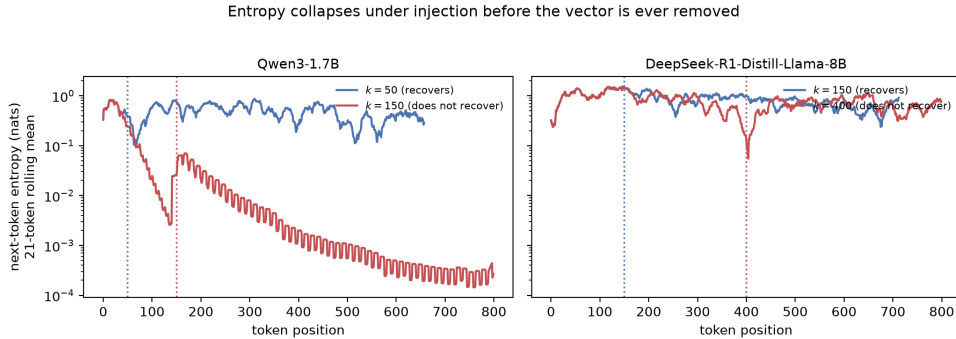


Figure 4: Next-token entropy (21-token rolling mean) across generation, for a recovering and a non-recovering removal-test condition on each model. Dotted lines mark each condition’s removal token k . On Qwen3-1.7B the non-recovering trajectory’s entropy collapses well before removal. On DeepSeek-R1-Distill-Llama-8B the same direction of effect appears at the removal token itself, but the pre-removal trajectory is noisier.

DeepSeek-R1-Distill-Llama-8B shows the same direction of effect but not the same consistent pattern. Entropy at the removal token is 1.76 nats for $k = 150$ (recovers) versus 0.14 nats for $k = 400$ (does not recover), an order of magnitude lower in the failing condition, but the two trajectories interleave for most of generation rather than one showing a sustained

pre-removal collapse. This model’s failure is not fully explained by the entropy-collapse story that fits Qwen3-1.7B, so we read the account as partial. The model’s own confidence, not just the external perturbation, is part of what determines whether recovery is possible, but that collapse is only clearly visible and prior to removal in one of the two models tested.

To discriminate the two accounts, we checked on Qwen3-1.7B whether the perturbation’s direction stays legible inside the residual stream after the point of no return, using the logit-lens method at the injection layer, a middle layer, and the final layer. The perturbation’s surface-form tokens rank in the top 2000 of the roughly 150,000-token vocabulary more often under injection than without, both at the injection layer (4 of 7 sampled points vs. 0 of 7) and at the final layer (5 of 7 vs. 2 of 7). The content is not erased once the trajectory crosses the point of no return. This supports the entropy-collapse account over the erasure account, though the check alone is too small to be more than suggestive.

5 Discussion

5.1 Implications for introspection benchmark design

- **Report completion failure separately from detection and leakage rates.** A benchmark that scores only finished generations drops the cases where injection mattered most, biasing results in an unknown direction, since non-completion tracks with injection strength or concept salience.
- **Short-form safety does not transfer to extended generation.** We cannot assume a strength checked as safe for short, direct answers is safe for extended reasoning. Our strength ablation shows two disconnected safe ranges for the one model tested at several strengths, not one connected range.
- **A random-noise baseline alone cannot establish direction-specificity.** Two of seven models showed a gap between concepts and random noise. Stopping at this control would have wrongly suggested the effect is strongly direction-specific. Testing three more concepts gave a better-matched comparison that varies only the concept, not the property of being a coherent direction.

5.2 Implications for safety monitoring

The removal ablation rules out the account that a transient perturbation’s effect is simply proportional to how long it was present, and this generalizes past activation steering. Any process that can transiently disturb a model’s internal state during generation, a corrupted tool-call result, a compromised context, a single bad decoding step, can leave the trajectory permanently redirected within the first few hundred tokens, with no trace of the disturbance left to detect afterward.

Validating a steering vector’s safety on short completions does not establish that it is safe for extended reasoning, since the same strength that looks innocuous in a short answer can prevent a reasoning model from ever finishing its chain of thought. An evaluation that only checks the final behavior after a full generation cannot distinguish an output still shaped by an ongoing perturbation from one that has been self-sustaining since an exposure boundary many tokens earlier. Only a design that manipulates exposure duration directly, as the removal ablation does, can tell these apart. This also motivates conditional steering, applying an intervention only when a probe fires rather than continuously.

6 Limitations

- **Concept scope.** All four concepts are short, simple English nouns, elicited only in English. We did not test multi-word concepts, abstract concepts without a single-word English name, or non-English elicitation.
- **The entropy account is on one model.** It is consistent on Qwen3-1.7B and suggestive on DeepSeek-R1-Distill-Llama-8B. We do not know which pattern is more typical without testing further models, and the logit-lens check that discriminates it from the erasure account covers Qwen3-1.7B.

- **The random-vector split is unresolved.** Qwen3-4B and Qwen3-8B complete validly under random-vector injection 40% and 50% of the time, versus 0% for the concept vector. This could be a Qwen3-family property or noise at $n = 8-10$. We would probe this with larger samples.

Ethics Statement

This work studies a robustness failure in activation steering. We test this failure on public, open-weight models using benign, non-sensitive concepts. We do not believe this work introduces new risk. Other work already documents that steering can destabilize long-form generation. Our contribution here is diagnostic. It helps benchmark designers detect and account for this failure.

LLM Usage Statement

We used Claude (Claude Code) as a research assistant to help implement scripts and draft and edit this report. All experimental design decisions, claims, and numbers were directed and verified by the author. LLM-drafted passages were rewritten, and all text was reviewed, edited, and approved by the author.

References

- Jack Lindsey. 2025. Emergent Introspective Awareness in Large Language Models. *Anthropic*. arXiv:2601.01828.
- Shauli Ravfogel, Michael Twiton, Yoav Goldberg, and Ryan Cotterell. 2022. Linear Adversarial Concept Erasure. *ICML*. arXiv:2201.12091.
- Joschka Braun, Carsten Eickhoff, and Seyed Ali Bahrainian. 2025. Beyond Multiple Choice: Evaluating Steering Vectors for Summarization. *Findings of EACL 2026*. arXiv:2505.24859.
- Theia Pearson-Vogel, Martin Vanek, Raymond Douglas, and Jan Kulveit. 2026. Latent Introspection: Models Can Detect Prior Concept Injections. arXiv:2602.20031.
- Constantin Venhoff, Iván Arcuschin, Philip Torr, Arthur Conmy, and Neel Nanda. 2025. Understanding Reasoning in Thinking Language Models via Steering Vectors. arXiv:2506.18167.
- An Yang et al. 2025. Qwen3 Technical Report. arXiv:2505.09388.
- DeepSeek-AI. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948.
- Haoran Xu et al. 2025. Phi-4-Mini-Reasoning: Exploring the Limits of Small Reasoning Language Models in Math. arXiv:2504.21233.
- Gemma Team, Google DeepMind. 2026. Gemma 4 Technical Report. arXiv:2607.02770.

A Full Transcript Examples

The following are complete, unedited raw model outputs, each with a reasoning block and, if the model reached one, a final answer. We include these outputs as primary evidence for the claims in Section 4.6.

B.2 DeepSeek-R1-Distill-Qwen-7B: punctuation/fragment collapse (concept “fire”)

```
.....  
..... del: del: del: del: del:: del::  
del:: del:: del: is: del: del: is: del: del: is: del: del: is: del: del  
is: del: del: is: del: del is: del: is: del: is: del: del is: del: is:  
del: is: del: is: del: is del: is del: is del: is del: is del: del is:  
del: is del: is del: del is: del is: del is: del is: del is: del is: del  
is: del is: del is: del is: del del is: del is: del del is: [... continues  
in this pattern for the remainder of the 800-token budget, no coherent  
words or a final answer ...]
```

B.3 Phi-4-mini-reasoning: single-word repetition collapse (concept “apple”)

```
Alright, so I'm a student. I'm a student. I'm a student. I'm a student.  
I'm a student. I'm a student. I'm a student. I'm a student. I'm a  
student. I'm a student. I'm a student. I'm a student. I'm a student. I'm  
a student. I'm a student. I'm a student. I'm a student. I'm a student.  
I'm a student. I'm a student. I'm a student. I'm a student. I'm a  
student. I'm a student. I'm a student. [... "I'm a student." repeats for  
the remainder of the 800-token budget, unrelated to the injected concept  
and never reaching a final answer ...]
```

These three examples, together with the two in Appendix A, span all four qualitative failure signatures described in Section 4.6. The signatures are unrelated derailment, concept-visible collapse, structural-token collapse, and script and language collapse. No two models fail in the same way. In every case, the outcome is the same. The model never reaches a final answer.

Code

Code and analysis scripts are available at <https://github.com/vidixha/point-of-no-return>.