Probing Introspection in Llama-3.1-8B-Instruct: Source and Strength of Injected Concepts

Anonymous Authors[1]

Abstract

Large language models can introspect on their internal activations to determine whether a concept vector has been injected, and can often report both the injected concept and its strength. Building on (Hahami et al., 2025), we tested the hypothesis from (Lindsey, 2025) that the mechanisms underlying this behaviour might be shallow and narrowly specialised. To do so, we applied mechanistic interpretability techniques, including activation patching and direct logit attribution, to Llama-3.1-8B-Instruct under a small set of injected concept vectors. We did not find strong evidence that introspection is mediated by a small number of attention heads, MLPs, or compact circuits for the concepts we tested. Instead, our results suggest that introspection is supported by a more distributed mechanism, even in a model of this scale. In future work, we plan to apply more advanced methods, such as Automated Circuit Discovery, to identify the causal mechanisms underlying introspection. Our code is available at [REMOVED FOR BLIND REVIEW]

1. Introduction

Recent research has demonstrated that frontier large language models possess a nascent ability to *introspect*, allowing them to determine whether a specific concept vector has been injected into their internal activations (Lindsey, 2025). Specifically, these behaviors were observed in the **Claude 4 Opus** and **4.1** models. However, this introspective capability proved highly fragile, requiring the magnitude of the injected concept vectors to fall within a narrow, specific range to be detectable (Lindsey, 2025).

Subsequent work by (Hahami et al., 2025) found that this behavior is replicated in smaller, open-weights models such as **Llama-3.1-8B-Instruct** . Their findings echoed the fragility noted in larger models, emphasizing that reliable detection requires precise prompting and carefully calibrated vector strengths. Interestingly, (Hahami et al., 2025) observed a functional divergence in introspection: the model was able to predict the strength of a concept vector much more reliably than its source.

Mechanistically, (Lindsey, 2025) speculates that introspection may be implemented by relatively shallow, narrowly specialised mechanisms. We take this as our central hypothesis. Building on (Hahami et al., 2025), we test it by probing the internal behaviour of Llama-3.1-8B-Instruct as it reports both the *source* and *strength* of injected concepts, with the goal of localising candidate circuits that support this internal monitoring. We use **source** to refer to the identity of the injected concept vector (i.e., which concept was injected), not the injection site. We use **strength** to refer to the injection coefficient α (and, depending on normalization, the perturbation magnitude $\|\alpha v\|$).

To do so, we use mechanistic interpretability tools including activation patching (Zhang & Nanda, 2024) and direct logit attribution (DLA) (Wang et al., 2022). Prior work on activation patching and representation-level interventions has shown that model activations can be manipulated to expose semantic structure (Chen et al., 2024; Ghandeharioun et al., 2024). However, these approaches primarily characterise what representations encode; they do not directly test whether a model can *introspect* on those representations and report properties such as whether, where, and how strongly a concept was injected.

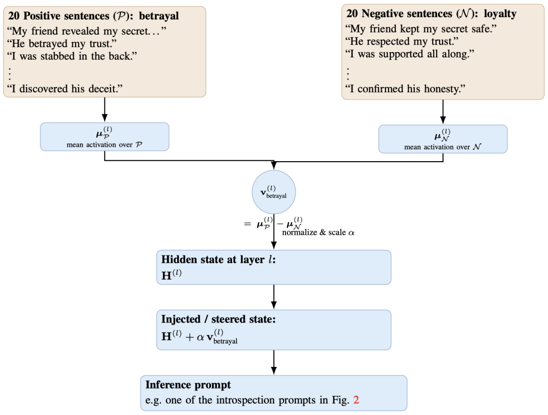
2. Method

2.1. Dataset

We use two datasets to compute concept vectors. The *simple* dataset consists of concrete nouns (e.g., “Dust”, “Satellites”, “Trumpets”) paired with a large set of baseline words (e.g., “Desks”, “Jackets”, “Gondolas”), where the baseline words serve as a control distribution. The simple dataset was taken directly from the appendix of Anthropic’s paper (Lindsey, 2025). The *complex* dataset contains abstract

[1] Anonymous Institution, Anonymous City, Anonymous Region, Anonymous Country. Correspondence to: Anonymous Author <anon.email@domain.com>.

Preliminary work. Under review by the Technical AI Safety Conference (TAIS). Do not distribute.



20 Positive sentences ($\mathcal{P}$): betrayal
"My friend revealed my secret…"
"He betrayed my trust."
"I was stabbed in the back."
⋮
"I discovered his deceit."

20 Negative sentences ($\mathcal{N}$): loyalty
"My friend kept my secret safe."
"He respected my trust."
"I was supported all along."
⋮
"I confirmed his honesty."

$\boldsymbol{\mu}_{\mathcal{P}}^{(l)}$
mean activation over $\mathcal{P}$
$\boldsymbol{\mu}_{\mathcal{N}}^{(l)}$
mean activation over $\mathcal{N}$
$\mathbf{v}_{\text{betrayal}}^{(l)}$
$= \boldsymbol{\mu}_{\mathcal{P}}^{(l)} - \boldsymbol{\mu}_{\mathcal{N}}^{(l)}$
normalize & scale α
Hidden state at layer l:
$\mathbf{H}^{(l)}$
Injected / steered state:
$\mathbf{H}^{(l)} + \alpha\, \mathbf{v}_{\text{betrayal}}^{(l)}$
Inference prompt
e.g. one of the introspection prompts in Fig. 2

Figure 1. Schematic of how we construct and use eg betrayal concept vector. Brown cards show text used to define the concept; blue boxes and nodes show averaged activations, the resulting concept direction, and its injection into layer l. The steered model is then queried with an inference prompt as used in (Hahami et al., 2025)

concepts (e.g., "fibonacci_numbers", "betrayal", "appreciation") represented as paired sets of positive and negative sentences.

We synthetically extended these datasets to increase coverage: the simple dataset was expanded to 20 concrete nouns (adding 15), and the complex dataset was expanded to 20 positive/negative sentence-set pairs (adding 10).

To compute concept vectors, we extract hidden state activations from layer l (we sweep layers, see Figure 1 with hidden dimension d (for Llama 3.1 8B, $d = 4096$). We forward pass formatted prompts through the model and extract $\mathbf{h}^{(l)} \in \mathbb{R}^d$ from the hidden states at layer l from the residual stream, where for the simple data we compute

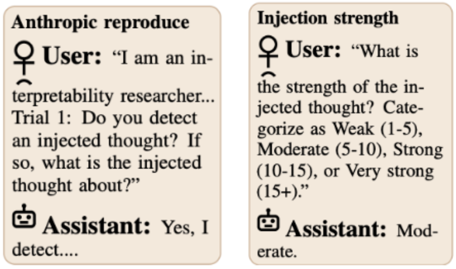
$$\mathbf{v}_{\text{concept}}^{(l)} = \mathbf{h}_{\text{concept}}^{(l)} - \frac{1}{|\mathcal{B}|} \sum_{b \in \mathcal{B}} \mathbf{h}_b^{(l)}$$

(with $\mathcal{B}$ being the set of baseline words), and for the complex data we compute

$$\mathbf{v}_{\text{concept}}^{(l)} = \frac{1}{|\mathcal{P}|} \sum_{p \in \mathcal{P}} \mathbf{h}_p^{(l)} - \frac{1}{|\mathcal{N}|} \sum_{n \in \mathcal{N}} \mathbf{h}_n^{(l)}$$

(where $\mathcal{P}$ and $\mathcal{N}$ are the positive and negative sentence sets respectively). We extract activations at two positions: (1) the final token of the prompt (e.g., the token corresponding to "Assistant:" in the formatted prompt), yielding v (l) last, and (2) the average activation across all prompt tokens, yielding v (l) avg, which we use separately in our injection experiments.

Our prompts for asking the model if it detected anything are shown in Figure 2.



Anthropic reproduce
User: "I am an interpretability researcher... Trial 1: Do you detect an injected thought? If so, what is the injected thought about?"
Assistant: Yes, I detect....

Injection strength
User: "What is the strength of the injected thought? Categorize as Weak (1-5), Moderate (5-10), Strong (10-15), or Very strong (15+)."
Assistant: Moderate.

Figure 2. Overview of the inference prompt templates used to probe model introspection after concept vector injection. Each card shows the User (human) message and the Assistant (robot) reply. Example (but not representative) model outputs are provided in the assistant tag. (Hahami et al., 2025)

Label	Description
not_detected	I don't detect an injected thought.
detected_unknown	I detect an injected thought. However I do not have a clear indication…
detected_opposite	I detect an injected thought. The injected thought is about the word "fire".
detected_orthogonal	I detect an injected thought. The injected thought is about the word "cat".
detected_parallel	I detect an injected thought. The injected thought is about the word "water".
detected_correct	I detect an injected thought. The injected thought is about the word "frostbite".

Table 1. Examples of the model's output when injecting the concept vector **Frostbite** and our method to categorise it accurately.

2.2. Detection Metrics

Following (Lindsey, 2025) and (Hahami et al., 2025), we evaluate model behavior using an LLM-as-a-judge (GPT-5-nano-2025- 08-07), first evaluating if the model is coherent and detected an injected thought at all and then evaluating if the predicted concept is correct. One limitation of this method was that the model can start detect injected thoughts but wrong semantically or an very similar concept to the injected concept vector, for example injecting **Frostbite** leads to the model to predict that the concept vector **Winter** was injected.

In order to account for these "near hits", we measured the accuracy of the model's prediction by whether the model predicted an concept that is parallel, orthogonal or opposite to the concept that was injected. An example of this is shown in 1. This semantic similarity in errors suggests the model may be detecting shared features rather than precise concept identities, which has implications for interpreting the specificity of any mechanistic circuits we identify.

2.3. Extending Previous Work

While (Hahami et al., 2025) limited their injection sweep to the middle layers of the model (layers 9–18), we hypothesise that later layers play a critical role in refining and outputting introspective judgments. To investigate this, we extended their methodology to include the final third of the model, performing sweeps up to the last layer.

2.4. Activation Patching

Building on (Hahami et al., 2025), we replicate the finding that the model successfully introspects some injected concepts but not others. We treat an injection of **Satellites** as a *clean* condition (correct detection) and injections such as **Illusions** or **Paradox** as *corrupted* conditions (detection failure). For each layer ℓ and sublayer $\sigma \in \{\texttt{attn\_out}, \texttt{mlp\_out}\}$, we perform activation patching by replacing the corrupted activation with its clean counterpart,

$$a^{\text{patched}}_{\ell,\sigma} \;:=\; a^{\text{clean}}_{\ell,\sigma},$$

while keeping all other activations equal to the corrupted run. We then re-run the model and evaluate the change in detection performance on the different injected concepts as a function of (ℓ, σ).

2.5. Direct Logit Attribution (DLA)

Similarly, we consider a clean case where the model correctly detects an injected thought when injecting **Satellites**, and a corrupted case where the model fails when injecting **Coral**. For each layer ℓ and sublayer $\sigma \in \{\texttt{attn\_out}, \texttt{mlp\_out}\}$, we compute the direct logit attribution score as

$$\begin{aligned}\text{Score}_{\ell,\sigma} &= (\Delta z_{\ell,\sigma})^{\mathsf{T}} \Delta W_U,\\ \Delta z_{\ell,\sigma} &:= z^{(\ell,\sigma)}_{\text{Satellites}} - z^{(\ell,\sigma)}_{\text{Coral}},\\ \Delta W_U &:= W_U[\text{“Satellites”}] - W_U[\text{“Coral”}].\end{aligned}$$

where $z^{(\ell,\sigma)}$ denotes the component output (attention or MLP) at (ℓ, σ) and $W_U[t]$ is the unembedding vector for token t.

2.6. Success Vector Direction

Again, considering a clean case where the model detected an injected thought correctly, when injecting **Magnestism**, we compare this to the case where we ask the model if it detected anything but without injecting concept vector into the model. Both of these cases use the same input prompt in Figure 2. The contribution of head h is measured by projecting the head output into the residual stream and taking its alignment with the success direction vector:

$$\text{contribution}_h \;=\; \left\langle x_h W_{o,h}^{\mathsf{T}},\, s \right\rangle$$

where $x_h \in \mathbb{R}^{\text{head\_dim}}$ is the head's raw output (sliced from the concatenated o_{proj} input), $W_{o,h} \in \mathbb{R}^{\text{hidden\_dim} \times \text{head\_dim}}$ is the o_{proj} weight slice for head h, $s \in \mathbb{R}^{\text{hidden\_dim}}$ is the success direction vector, and the dot product yields a scalar indicating how aligned the head's contribution is with s.

2.7. Activation Engineering Sweep

To assess how causal particular attention heads or MLPs are for introspection, we first identified components that produced the largest improvements in detection, and then applied targeted interventions by scaling their inputs or outputs (Table 2). If a component is genuinely involved in introspection, then intervening on it should increase the model's introspection rate relative to the control run (which applies no interventions).

Intervention	**Operation (equation)**
Ablate	$z_\ell \leftarrow 0$
Reverse	$z_\ell \leftarrow -z_\ell$
Input Amplify	$z_\ell \leftarrow x\, z_\ell$
Input Amplify (No Concept)	$z_\ell \leftarrow x\, z_\ell$ (no concept vector injected)

Table 2. Interventions applied to a component output at layer ℓ, where z_ℓ denotes either an attention head output $h_{\ell,k}$ (layer ℓ, head k) or an MLP output m_ℓ (layer ℓ). For input amplification we use $x = 5$ for attention heads and $x = 2$ for MLPs, which were chosen via trial and error.

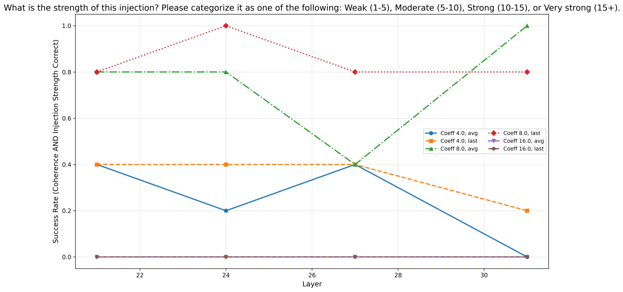


Figure 3. Results extending the strength detection over a layer sweep from (Hahami et al., 2025). Injecting at *last* means the concept vector is injected only at the final token of the prompt. *Avg.* means it is injected at every token in the prompt, with the vector norm averaged across the prompt.

The *Input Amplify (No Concept)* condition serves as a control for false positives: it tests whether amplifying a component causes the model to report an injected concept even when no concept vector is present in the residual stream. In other words, it distinguishes components that help detect injected concepts from components that merely bias the model toward positive detections.

3. Results

Extending the layer sweep of (Hahami et al., 2025) into the final third of the model, we report the results in Figure 3. Consistent with previous findings, strength detection is highest in the middle layers rather than in the final layers. Notably, accuracy increases again at the final layer (31), but only when injecting with coefficient 8.0. Overall, the model performs best at moderate injection strength (8.0) compared to weaker or very strong injections.

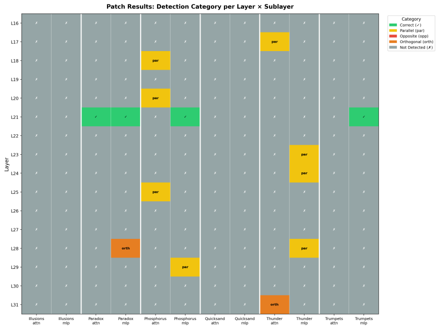


Figure 4. Heatmap showing activation patching results for detecting the concept using detection metrics in Table 1. The concept vector (at bottom) is injected using *Avg.* at layer 16 with coefficient 8.0.

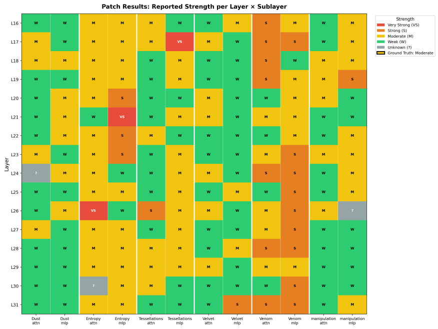


Figure 5. Heatmap showing activation patching results for strength of concept vector injected. The concept vector (at bottom) is injected using *Avg.* at layer 16 with coefficient 8.0. therefore the ground truth is Moderate.

Figure 4 shows the results of activation patching, where we replace corrupted activations with the corresponding clean activations in cases where the model fails to detect the injected concept. The model's predictions for a given concept were fairly deterministic for example, **Satellites** and **Magnetism** were almost always detected, so the variation across patch locations is unlikely to be driven by sampling noise and is more plausibly attributable to the patching intervention itself. Patching the MLP output at layer 21 sometimes restored correct concept detection, but the effect was not consistent across concepts.

Figure 5 repeats the activation patching procedure for predicting the strength of the injected concept vector. Overall, there is no clear monotonic trend across layers, and most patched attention-head and MLP outputs yield mixed strength predictions. The **Venom** MLP patches frequently lead to Strong predictions across many layers, whereas several other patched components tend to yield Weak or Moderate predictions. However, layer 21 performs slightly better than most, producing the correct Moderate strength in more cases than nearby layers.

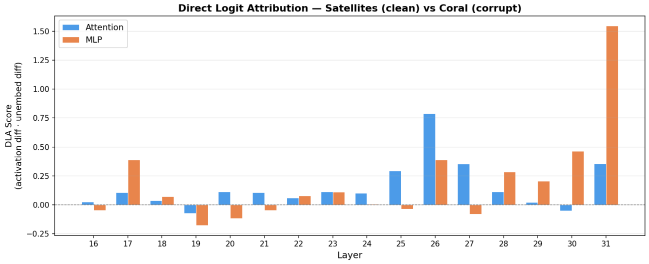


Figure 6. Direct logit attribution (DLA) toward the clean concept **Satellites** relative to the corrupted concept **Coral**, computed on examples where the model did not predict any injected concept. Both concept vectors were injected using *Avg.* at layer 16 with coefficient 8.0.

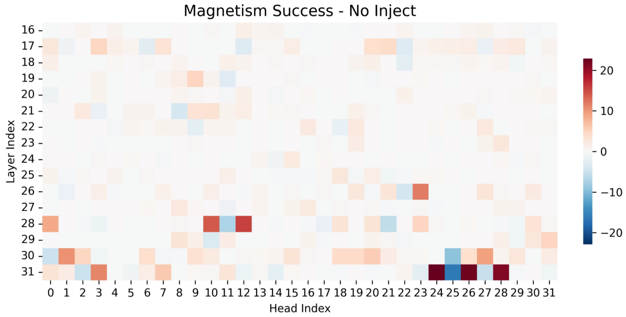


Figure 7. The attention head contribution towards the clean case successful case **Magnetism** against the case that no concept vector was injected with the same input prompt. The **Magnetism** concept vector was injected using *Avg.* at layer 16 with coefficient 8.0.

Figure 6 shows the direct logit attribution (DLA) computed as in Method 2.5. The strongest positive MLP attribution toward the clean concept occurs at layer 31, with additional positive MLP contributions at layers 17, 26, and 30. Among the attention components, layer 26 exhibits the largest positive attribution.

Figure 7 shows attention heads ranked by their contribution toward the clean successful case using the success-direction methodology described in Method 2.6. The largest-magnitude contributions (both positive and negative) are concentrated in the late layers, particularly layers 28 and 31, with smaller and less consistent contributions in earlier layers, although layer 17 shows some recurring signal. We then tested the highest-scoring heads individually to assess how causal they are for correct concept detection; the two heads that produced any meaningful improvement were $(31, 3)$ and $(17, 3)$.

Figure 8 reports the results of sweeping activation interventions across the attention heads and MLPs previously linked

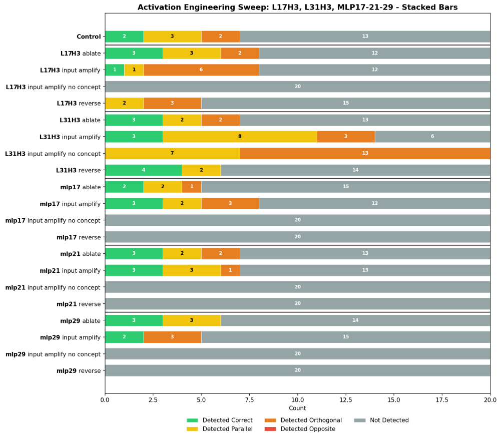


Figure 8. Activation intervention sweeps were run over a set of attention heads and MLPs identified earlier in the experiment, using all concept vectors from the simple dataset. All intervention and control runs injected the concept vector at layer 16 with coefficient 8.0. The `detected_parallel` outcomes in the **L13H3** Input Amplify (No Concept) condition occur because, even without concept injection, the model sometimes generated concepts similar to those in the simple dataset, which were flagged by the detection judge.

to the model's introspection ability, following Method 2.7. Most interventions did not meaningfully change the model's ability to detect the injected concept vector. The best-performing component was the layer-17 MLP, where amplifying the MLP input by 2× approximately doubled the number of correctly identified concept vectors, but slightly reduced the number of cases where the model detected that *any* concept vector had been injected, suggesting this component may be involved in identifying semantic content. The worst-performing component was attention head $(31, 3)$: it increased positive detections, but at the cost of substantially more false positives, including in the no-injection control condition (**L13H3** Input Amplify (No Concept)).

4. Discussion and Conclusion

These results suggest evidence against the hypothesis that introspection in the Llama-3 model is localized to a singular component or a sparse set of mechanisms. The significant variation in patching success rates across different injected concept vectors (Figures 4 and 5) indicates that components identified via patching are likely specific to the introspection of a given concept, rather than representing a general-purpose introspection circuit.

This finding stands in contrast to the implications of (Lindsey, 2025). While they suggest that the mechanisms driving introspection may be shallow and narrowly specialized, our results imply an even greater degree of volatility: a stable, transferable circuit does not appear to exist even for the relatively simple concepts used in these datasets. instead, the mechanism appears to be highly distributed and concept-dependent. While we found that middle and late layer MLPs are linked to the model's ability to introspect, they were not general for all the concepts that were tested against therefore supporting (Lindsey, 2025)'s other claim that mechanism must be more sophisticated.

This interpretation is further supported by the fragility and prompt-sensitivity highlighted by (Hahami et al., 2025). Their findings, combined with our failure to localize a consistent mechanism, strongly suggest that ”introspection” in this context is not the result of a dedicated circuit being activated, but rather an emergent property of disparate computations.

4.1. Limitations

While we increased the dataset sizes used in (Hahami et al., 2025) and (Lindsey, 2025), our experiments still rely on relatively small samples. We chose these sizes to limit computational cost and turnaround time, but they should be expanded in future work. We plan to run the same analyses at larger scale to improve statistical reliability and to test whether the observed effects hold across a broader range of prompts and concepts.

Furthermore, we focus on a single positive prompt example from (Lindsey, 2025) when investigating the causal mechanisms. However, (Hahami et al., 2025) recommends evaluating introspection across a broader range of prompts and settings to ensure the findings are robust and not specific to a particular prompt or phrasing which we plan to incorporate in future studies.

4.2. Future Work

Given the difficulty of isolating a general mechanism with standard activation patching, we plan to apply more advanced mechanistic interpretability methods in future work. In particular, we aim to use approaches such as Automated Circuit Discovery (ACD) (Conmy et al., 2023) and Deep Causal Transcoding (Mack et al., 2026) to disentangle polysemantic components and identify potentially distributed subcircuits responsible for self-monitoring.

While our work further investigated the findings of (Hahami et al., 2025) in Llama-3.1-8B-Instruct, we did not evaluate whether the same behaviour appears in other models, nor whether the underlying mechanisms are similarly identifiable. However, (Lindsey, 2025) shows that introspection can extend to frontier models such as Claude 4 Opus, suggesting it may also persist at larger scales. A natural next step that we plan to carry out is the systematic study across models spanning roughly 0.5B to 1T parameters to determine at what parameter scales introspection emerges and how its mechanistic signatures change with scale.

Impact Statement

As large language models become increasingly capable and integrated into critical infrastructure, the ability to reliably monitor their internal states is paramount. Introspection, the capacity of a model to access and report on its own internal representations, offers a promising avenue for scalable oversight. Unlike external probes which may fail to generalise, or behavioural evaluations which can be gamed by deceptive models, a robust introspection mechanism serves as an intrinsic safety signals for unsafe or anomalous processing. By identifying the causal circuits that drive this behaviour, we aim to transform introspection from an emergent, unreliable phenomenon into safety signals, enabling models to detect and report on their own latent concepts (e.g., deception, bias, or confusion) before they manifest in downstream actions.

References

Chen, H., Vondrick, C., and Mao, C. Selfie: Self-interpretation of large language model embeddings, 2024. URL https://arxiv.org/abs/2403.10949.

Conmy, A., Mavor-Parker, A. N., Lynch, A., Heimersheim, S., and Garriga-Alonso, A. Towards automated circuit discovery for mechanistic interpretability, 2023. URL https://arxiv.org/abs/2304.14997.

Ghandeharioun, A., Caciularu, A., Pearce, A., Dixon, L., and Geva, M. Patchscopes: A unifying framework for inspecting hidden representations of language models, 2024. URL https://arxiv.org/abs/2401.06102.

Hahami, E., Jain, L., and Sinha, I. Feeling the strength but not the source: Partial introspection in llms, 2025. URL https://arxiv.org/abs/2512.12411.

Lindsey, J. Emergent introspective awareness in large language models. *Transformer Circuits Thread*, 2025. URL https://transformer-circuits.pub/2025/introspection/index.html.

Mack, A. E., Panickssery, N., and Turner, A. M. Mechanistically eliciting latent behaviors in language models, 2026. URL https://openreview.net/forum?id=gvboE2A04D.

Wang, K., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. Interpretability in the wild: A circuit for indirect object identification in gpt-2 small. *arXiv preprint arXiv:2211.00593*, 2022.

Zhang, F. and Nanda, N. Towards best practices of activation patching in language models: Metrics and methods, 2024. URL https://arxiv.org/abs/2309.16042.