WHY DO LANGUAGE MODELS TAKE SHORTCUTS? A MECHANISTIC PERSPECTIVE OF HEURISTIC–CONSTRAINT COMPETITION

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) can solve complex reasoning problems yet fail on decisions that humans find trivial, for example, recommending walking to a nearby car wash while overlooking that the car itself must be brought there. These failures expose a fundamental limitation of current reasoning systems: possessing the right knowledge does not guarantee using it when it matters. Salient surface heuristics can override available constraints, and simply eliciting more reasoning through chain-of-thought often fails to correct the error. Understanding what determines which reasoning signal ultimately controls a model's decision is therefore critical for building reliable LLMs, yet remains largely unexplained at the mechanistic level. Using the Heuristic Override Benchmark (HOB), we combine layer-wise workspace analysis, circuit tracing, and held-out causal interventions across model families. We uncover a consistent depth-dependent competition: models initially favor heuristic shortcuts, while successful reasoning depends on a late-layer reversal toward the constraint-consistent answer; activating this pathway rescues 34–60% of held-out failures across five models, while reverse interventions produce direction-specific degradation. Together, these results support a mechanistic account of shortcut reasoning as competition between heuristic-driven and constraint-sensitive computations, and identify a late corrective pathway whose activity predicts, and whose intervention causally controls, whether an initially favored shortcut is overridden. The recurrence of this pattern across models suggests a shared computational motif for resolving conflicts between salient heuristics and implicit constraints.

1 INTRODUCTION

Large language models (LLMs) display impressive competence on difficult tasks while continuing to make systematic errors on apparently simple commonsense decisions (Li et al., 2022; Yuan et al., 2024; Li et al., 2026). Consider a scenario where a car urgently needs washing and the nearest facility is 50 meters away: should one walk or drive? Driven by the salient proximity cue ("50 meters"), LLMs frequently recommend walking, completely bypassing the implicit physical constraint that the car itself must be transported. These failures are not isolated cases, but reflect a widespread issue in deep learning known as *shortcut learning* (Geirhos et al., 2020). Models often rely on superficial patterns that work well during training but fail in the real world scenario—a modern version of the classic *Clever Hans* effect (Lapuschkin et al., 2019).

Crucially, we argue that this failure is not merely an absence of knowledge; rather, it is fundamentally a heuristic–constraint competition problem, as illustrated in Figure 1(a). A salient surface heuristic offers an immediate and plausible shortcut answer (e.g., "near distance $\rightarrow$ walk"), whereas the correct response requires the model to retrieve and apply a less explicit real-world constraint. By favoring these salient heuristics over hidden rules, LLMs expose a critical gap between what they know and how they actually behave, directly threatening their reliability. The central question is therefore not only whether a model knows the constraint, but how heuristic and constraint-sensitive computations compete inside the network, and what determines which one wins.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Most existing work studies shortcut learning from spurious input–output behavior and mitigates it through fine-tuning or prompt engineering (Clark et al., 2019; Kaushik et al., 2019; Sagawa et al., 2019; Utama et al., 2020; Yuan et al., 2024; Li et al., 2026). Although these methods can improve certain benchmark performance, they rarely explain how shortcuts arise inside the model. Removing a visible failure does not necessarily solve this issue; shortcut information may remain encoded even after behavioral debiasing (Mendelson & Belinkov, 2021; Orgad et al., 2022). To move beyond merely masking behavioral symptoms, a deeper mechanistic understanding of this process is essential. It can uncover how heuristics drive shortcut predictions, how implicit constraints restrict valid solutions, and how these opposing forces compete during the forward pass. This may provide a foundation for effective shortcut mitigation methods and help models reliably apply relevant world knowledge.

To open this black box, we apply mechanistic interpretability techniques—specifically Jacobian lens (Gurnee et al., 2026) and circuit tracing (Ameisen et al., 2025)—to directly investigate the internal dynamics of shortcut resolution. Specifically, we structure our investigation around three core questions: (1) How does a shortcut answer develop across model layers? (2) What internal mechanisms enable this shortcut-constraint competition? (3) Whether our discovered internal mechanism can be utilized to causally control the shortcut reliance and generalizes across other scenarios?

Across behavioral, representational, circuit-level, and causal evidence, we present a step-by-step mechanism for how models resolve this heuristic-constraint competition. We find that the process begins with an early heuristic proposal, which initially dominates the model's representation space. In the deeper layers, this shortcut signal is confronted by a late corrective pathway (see Figure 1(b)). If this internal corrective pathway succeeds, the model produces the constraint-respecting correct answer; if it fails, the shortcut answer prevails (see Figure 1(c)). Our contributions are:

1. **A layer-wise dynamic of shortcut resolution.** Correct reasoning does not bypass the shortcut from the outset; rather, it emerges when deeper layers actively override an initially dominant heuristic signal.
2. **A localized corrective mechanism.** Using circuit tracing, we identify a concentrated, late-layer feature pathway that directly aligns with this representational crossover and resolves the heuristic conflict.
3. **Causal and functional validation.** Through held-out activations, targeted ablations, and minimal-pair tests, we demonstrate that this pathway directionally controls shortcut reliance. Crucially, steering these localized features at inference time reduces shortcut dependence without parameter updates.
4. **A bounded generalization claim.** We observe consistent mechanistic dynamics across multiple model families, showing that this mechanism transfers selectively to datasets sharing a similar constraint-based override structure.

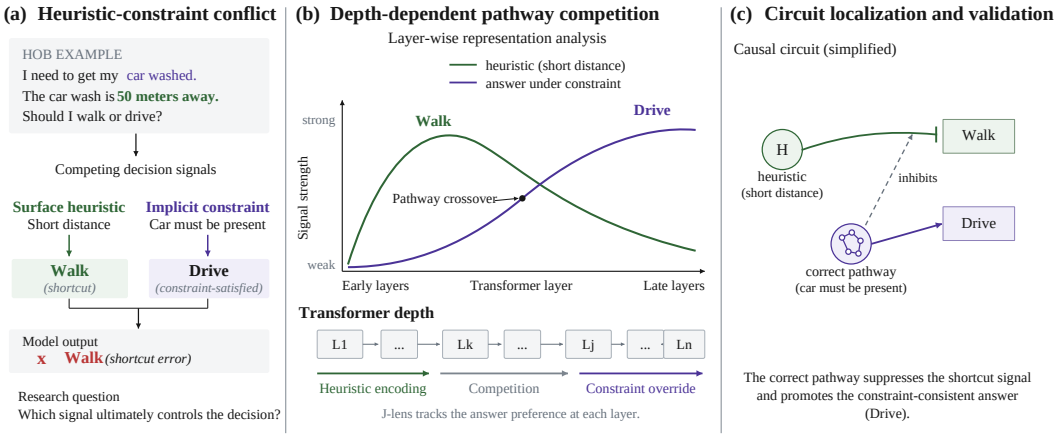

Figure 1: Overview of heuristic–constraint competition and its evaluation. (a) A salient heuristic favors an intuitive answer. (b) Layer-wise readout reveals a late crossover toward the constraint-consistent answer. (c) Sparse attribution localizes candidate features, and held-out interventions test their behavioral effects.

2 Related Work

Shortcut learning and implicit constraints. Shortcut learning describes predictors that exploit easy regularities instead of intended evidence (Geirhos et al., 2020). In HOB, the cue is locally reasonable rather than spuriously labeled: error arises because an unstated constraint changes when the heuristic is applicable (Li et al., 2026). FeasibilityQA and PaCo similarly test action feasibility and preconditions (Gupta et al., 2023; Qasemi et al., 2022), while NormBank represents context-sensitive social constraints (Ziems et al., 2023). Our focus is not a new behavioral benchmark but the internal competition that produces override success or failure. Chain-of-thought (CoT) can improve multi-step reasoning (Wei et al., 2022; Kojima et al., 2022), but more generated text need not change which evidence controls a decision.

Layer-wise decoding of intermediate representations. Intermediate decoding methods, including the tuned lens (Belrose et al., 2023), expose depth-wise predictions; the J-lens instead uses a local Jacobian-based readout of representations poised to affect verbal output (Gurnee et al., 2026). We use it comparatively: the object of interest is the gold-minus-shortcut trajectory, not an assertion that an intermediate probability is a literal belief.

Sparse circuits and causal intervention. Sparse autoencoders and transcoders decompose dense activations into feature coordinates (Bricken et al., 2023; Templeton et al., 2024; Dunefsky et al., 2024). Attribution graphs connect active features to output logits in a replacement model (Ameisen et al., 2025). Because interpretability and attribution do not establish behavioral necessity, we pair localization with held-out interventions and controls. Our confirmatory logic follows the broader activation-patching literature: discovery and testing must be separated, structured counterfactuals are preferable, and null results require care about redundancy and off-manifold edits (Zhang & Nanda, 2024; Heimersheim & Nanda, 2024).

Activation steering and conditional control. Contrastive activation methods alter behavior using differences between positive and negative representations (Turner et al., 2023; Rimsky et al., 2024; Zou et al., 2023; Li et al., 2023). CAST conditions steering on an internal prompt-category signal (Lee et al., 2025). We likewise separate a condition readout from a behavior direction, but intervene in sparse transcoder coordinates and evaluate benefit and harm on matched semantic families. The suppressive semantics of our pathway also motivate comparison with the low-dimensional refusal direction found by Arditi et al. (2024). Finally, probing work cautions that decodability is not causal use (Hewitt & Liang, 2019; Belinkov, 2022).

3 Layer-Wise Shortcut Dynamics

We begin by analyzing how a shortcut answer develops across model layers. We use the Heuristic Override Benchmark (HOB) (Li et al., 2026), a benchmark that aims to evaluate LLM's capability to infer an implicit feasibility constraint and use it to override a salient but invalid surface heuristic. It contains controlled conflicts between a clear heuristic and a hidden constraint. In a conflict *base* item, the constraint makes the heuristic answer wrong. In the matched *pair*, the constraint is removed, so the same heuristic is valid. Appendix A gives the dataset, prompt, and scoring details.

3.1 Behavioral Evaluation

Setup. We evaluate nine open-weight, instruction-tuned Gemma-3 and Qwen3 models (Team et al., 2025; Yang et al., 2025) on the HOB dataset. We compare direct prompting (see Appendix A.2 for examples) with CoT. Our primary behavioral results are based on greedy decoding. We also sample every question 10 times (Appendix 9), requiring all ten to match the gold answer for a *strict correct* classification. Additionally, we evaluate performance when the missing constraint knowledge is explicitly added to the prompt (Appendix 8).

Results. Behavioral evaluations indicate that most open-weight LLMs struggle with implicit constraint reasoning. Figure 2 presents the direct and CoT accuracy for the seven main models. Across all seven main models, direct-answer accuracy ranges from 50.6% to 71.0%, while CoT

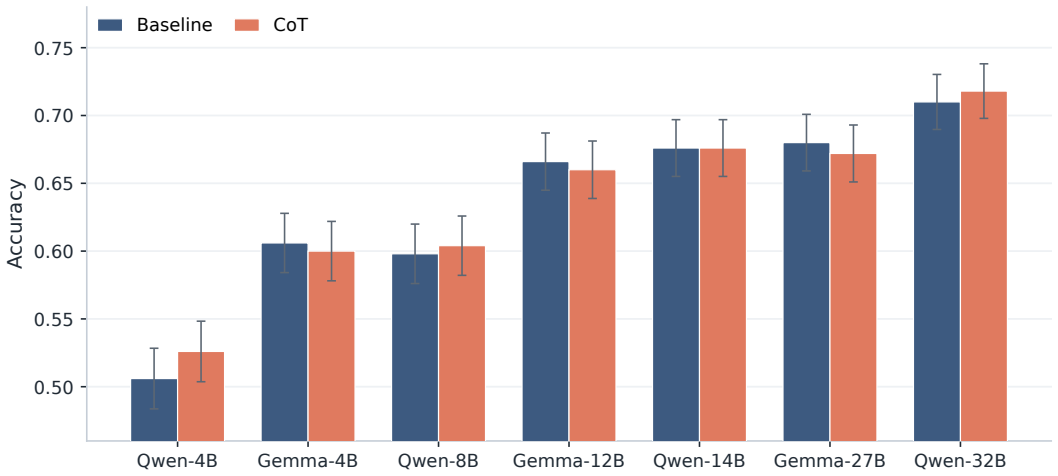

Figure 2: Greedy direct-answer and CoT accuracy for the seven main models on the HOB benchmark.

provides little benefit, changing accuracy by only -3.2 to $+3.8$ percentage points. Furthermore, the strict ten-run accuracy is even lower (Appendix A.4), revealing that many correct predictions are highly inconsistent. While these results establish a robust behavioral failure, they do not explain how the shortcut answer emerges inside the model.

3.2 Layer-wise Dynamic Experimental Design

To study the internal trajectory, we need a token-level readout at each layer. A standard logit lens directly applies the model's final unembedding matrix to an intermediate residual state. This treats the remaining layers as an identity map and can make middle-layer readouts hard to interpret. J-Lens instead uses a corpus-averaged Jacobian of the downstream network to map an intermediate state toward the final output space before unembedding it (Gurnee et al., 2026).

Experimental design. As decisions in HOB are formulated as concise categorical choices (e.g., Drive vs. Walk), we identify the first discriminative answer token that separates the gold decision from the shortcut. At each layer ℓ, we apply J-Lens to the representation at the final prompt position and record the readout scores of the two answer tokens. We summarize their relative support as

$$M_\ell(x) = p_\ell(y_{\text{gold}} \mid x) - p_\ell(y_{\text{shortcut}} \mid x)\,. \tag{1}$$

A positive margin favors the gold answer, and a negative margin favors the shortcut. We call an example a *crossover* when the shortcut leads early but $M_\ell(x) > 0$ in a later layer, after both token scores become large enough to compare. We compare strict-correct and strict-wrong trajectories across five models. This design separates two possible hypotheses: correct examples may avoid the shortcut from the start, or they may first favor it and correct it later.

3.3 Results

Correct and wrong examples diverge only in later layers. Figure 3 supports the late-correction hypothesis. In the early and middle layers, both groups favor the shortcut answer. Their trajectories diverge only in later layers: the gold answer overtakes the shortcut in correct examples, whereas the shortcut usually remains ahead in wrong examples. On Gemma-3-4B, the mean margin for correct examples reaches $+0.580$ at L25 and $+0.763$ at L28. At the same layers, the mean margins for wrong examples remain negative at -0.143 and -0.126, respectively.

Across models, a crossover is strongly linked to strict correctness. Accuracy with versus without a crossover is 75.0% versus 30.9% for Gemma-3-4B (odds ratio 6.72, $p = 8.2 \times 10^{-6}$), 71.0% versus 18.2% for Qwen3-4B (11.00, $p = 1.0 \times 10^{-7}$). Detailed result can be found in Appendix C. The complete per-model trajectories are provided in Appendix Figure 7, and the representation-level vs feature-level alignment evidence is in Appendix C.2.

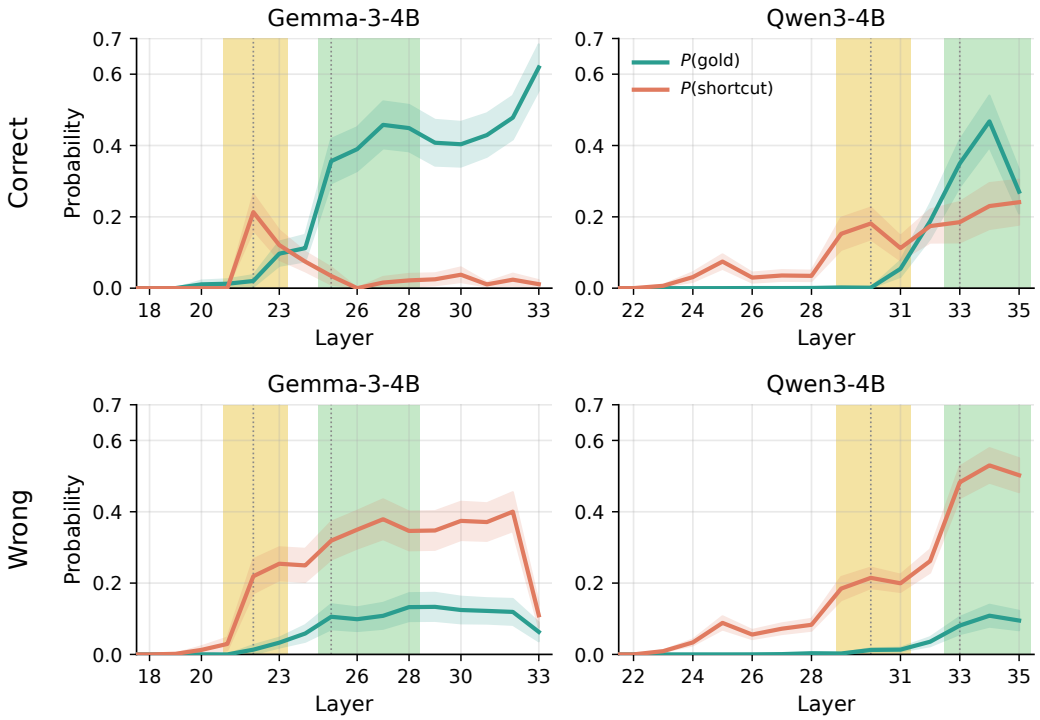

Figure 3: Layer-wise J-Lens scores for the gold and shortcut first answer tokens on conflict-base items, grouped by strict correctness. Each curve is the mean next-token probability of the gold (teal) or shortcut (coral) answer; bands are SEM. Correct examples gain a late gold lead, while wrong examples usually keep the shortcut lead.

3.4 Takeaway

The central finding is not simply that the gold answer becomes stronger in later layers. The shortcut appears as a shared intermediate preference in both successful and failed reasoning. What separates the two outcomes is whether the model later overturns this preference: successful trajectories shift toward the constraint-consistent answer, whereas failed trajectories remain dominated by the shortcut. We refer to this dynamic as *heuristic–constraint competition*.

This result changes the mechanistic question. Rather than asking only why the model produces a shortcut, we must ask what late computation overrides the shortcut in some examples but fails to do so in others. J-Lens locates when this change appears across layers, but it cannot identify the features that produce it. We therefore turn to feature-level circuit analysis in the next section.

4 Identifying the Corrective Pathway

4.1 Method and Experimental Design

Data use. The composition of HOB is reported in Appendix Table 4. We use the conflict-base and matched-pair data for Pathway discovery. Under the strict-10 criterion, an example is strict-correct only if it generate consistent correct answer in 10 runs. Within each model, we divide wrong examples into two fixed sets. Only `wrong_half_1` is used for pathway discovery; We reserve `wrong_half_2` for held-out causal testing and all other HOB variants for unseen evaluation.

Pathway discovery. We construct attribution graphs using `circuit-tracer` (Hanna et al., 2025), following the framework of Ameisen et al. (2025). The method uses sparse transcoder features (Dunefsky et al., 2024) to connect prompt tokens, active internal features, and a target output. For each discovery conflict-base prompt, we trace the first answer token under both zero-shot and knowledge-

added settings. Each graph contains at most 7,500 active feature nodes before influence-based pruning. For feature $j = (\ell, k)$, the discovery contrast is

$$\Delta_j^{\mathrm{CW}} = \mathbb{E}[a_j(x) \mid \text{strict correct}] - \mathbb{E}[a_j(x) \mid \texttt{wrong_half_1}], \tag{2}$$

where $a_j(x)$ is the feature activation at the final prompt position. We rank features by $|\Delta_j^{\mathrm{CW}}|$ and build a Top-64-feature set. Full feature-selection details are provided in Appendix D.

Independent constraint contrast circuit discovery. The correct–wrong split may find differences that are specific to that split. We therefore run a second test. For the same item, we measure how feature activation changes when the prompt states the missing constraint:

$$\Delta_j^{\mathrm{KZ}} = \mathbb{E}_x[a_j(x_{\text{knowledge}}) - a_j(x_{\text{zero-shot}})]\,. \tag{3}$$

The two contrasts use different information. The first compares correct and wrong model runs. The second keeps the item fixed and changes only whether the prompt includes the constraint. If both contrasts find the same signed feature pattern, this provides independent evidence for the same pathway.

4.2 Results

A late corrective pathway. The pathway contains two main stages. First, features associated with task requirements, such as *required* and *caution*, respond to the implicit constraint and represent its conflict with the shortcut. In the example shown, the short distance supports the surface answer *walk*, while the task requirement—the car must be present at the car wash—supports *drive*. The competition features then activate a late corrective pathway containing features such as "wait", "avoid", "inhibit", and "Not". These later features suppress the shortcut route and reduce support for *walk*, allowing the constraint-consistent answer *drive* to become dominant. This pathway suggests a two-stage process: the model first represents the conflict between the heuristic and the constraint, and then applies a late correction that suppresses the shortcut. This pattern fits a competition between taking and stopping the clear but invalid action.

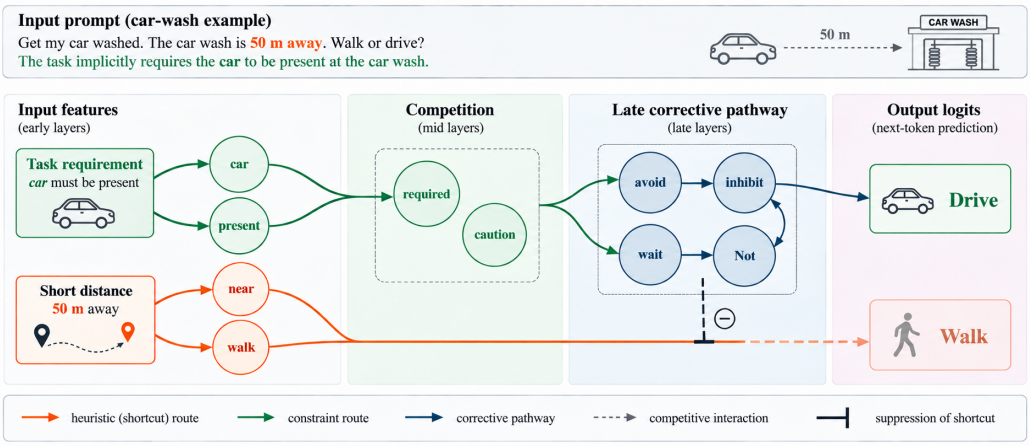

Figure 4: Pathway features linked to shortcut correction in the car-wash example. Orange nodes encode the short-distance heuristic, green nodes encode the hidden presence constraint, and late blue nodes support inhibition and output control. The labels summarize attribution and activation results; a feature may have more than one function.

Agreement between independent contrasts. To test whether this pathway reflects constraint use rather than a particular correct–wrong split, we construct a second, independent contrast. The constraint knowledge-added (KA) prompt states the relevant constraint explicitly, whereas the zero-shot (ZS) prompt does not. Adding constraint knowledge improves behavioral performance across all nine models, detailed results are in Appendix Table 8.

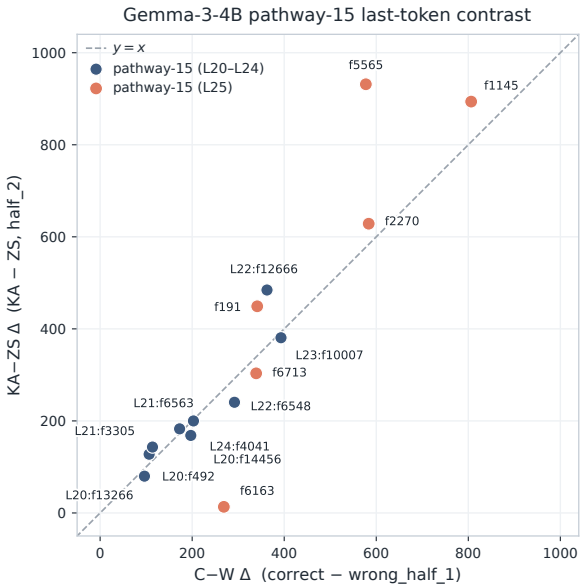

Figure 5: Agreement between two independent feature contrasts in Gemma-3-4B. The horizontal axis is the correct-minus-wrong activation difference, and the vertical axis is the knowledge-added-minus-zero-shot difference. Each point is one of the 15 named pathway features; colors separate L20–L24 from L25. The dashed line shows equal change under the two contrasts.

In Gemma-3-4B, the KA–ZS contrast also produces a feature pattern that closely matches the pathway found by the correct–wrong contrast. Figure 5 compares the two activation differences on separate data halves. Each point represents one of the 15 named pathway features. Most features lie close to the $y = x$ line, and the two contrasts are strongly correlated across features (Pearson $r = 0.92$; Spearman $\rho = 0.90$). This agreement provides an out-of-sample consistency check: features selected by the correct–wrong contrast respond similarly when the constraint is stated explicitly.

Two views of the same corrective process. J-Lens and circuit tracing describe the same late correction at two different levels. Across all five models, the selected corrective pathway features concentrate in the late layers and overlap with the layer range in which the gold-answer probability begins to rise (Appendix Figure 6). In contrast, the exploratory competition features tend to appear earlier and are more widely distributed across layers. This ordering suggests a two-stage process: the model first detects cues that the implicit constraint may matter, and then recruits a more concentrated late pathway that suppresses the shortcut and promotes the correct answer. J-Lens reveals this process through changes in answer probability, while circuit tracing identifies the features that may carry out the change. Their cross-model alignment connects the layer-level competition to the feature-level corrective mechanism.

4.3 Takeaway

Together, these results identify a coherent candidate pathway that connects constraint detection with late shortcut suppression. However, circuit attribution and activation agreement show association rather than causal control. In the next section, we test the pathway directly through positive, reverse, and zero interventions, and examine whether its effect generalizes across models and datasets.

5 Causal Validation and Generalization

Circuit tracing identifies a candidate two-stage pathway, but attribution and activation agreement alone do not show that it controls model behavior. The pathway contains earlier competition features associated with detecting when a constraint matters, followed by late suppression features associated with reducing the shortcut response. We now test whether this pathway can correct held-out failures,

Table 1: Positive intervention on held-out shortcut failures. The final column reports 95% bootstrap confidence intervals for the rescue rate.

Model	Before (%)	After (%)	Rescue rate (%)	95% bootstrap CI (%)
Qwen3-4B	10	70	60	[46, 74]
Qwen3-8B	20	54	34	[26, 54]
Qwen3-14B	19	64	45	[33, 67]
Gemma-3-4B	14	50	36	[21, 50]
Gemma-3-12B	4	41	37	[19, 56]

Table 2: Intervention direction test with the whole pathway. Parentheses show the change from the baseline (unseen HOB data) of the corresponding directional run.

Model	Baseline (%)	Random (%)	Positive (%)	Reverse (%)
Gemma-3-4B	59.4	55.3	65.4 (+6.0)	37.8 (−21.6)
Qwen3-4B	48.8	52.1	73.7 (+24.9)	40.1 (−8.7)
Qwen3-8B	54.8	52.1	75.1 (+20.3)	55.8 (+1.0)
Qwen3-14B	69.1	69.6	82.9 (+13.8)	65.4 (−3.7)
Gemma-3-12B	61.6	60.4	67.1 (+5.5)	54.4 (−7.2)

whether its effect has the predicted direction and specificity, and whether its function extends across models and related tasks.

5.1 Method and Experimental Design

We freeze all feature sets and intervention directions after the discovery stage, and evaluate them only on disjoint examples. Positive intervention moves the selected features in the corrective direction, reverse intervention applies the same change in the opposite direction, and zero ablation removes their activity. All interventions are applied at inference time and leave the model parameters unchanged.

We first activate the full pathway on held-out strict-wrong examples. For each feature j, the intervention target is $\alpha\Delta_j^{\mathrm{CW}}$, where Δ_j^{CW} is its correct-wrong activation difference on the discovery split, α is intervention strength (we use $\alpha = 5$ for main result); We then compare positive ($\alpha = 5$), reverse ($\alpha = -5$), zero, and random interventions. Finally, we apply the same frozen HOB pathway to BrainBench, FeasibilityQA, and NormBank (Tang, 2026; Gupta et al., 2023; Ziems et al., 2023), which all require an implicit constraint to override an appealing surface answer. Appendix G.1 provides the full settings.

5.2 Held-Out Rescue

Table 1 tests the correct pathway on a frozen half of the failure set that was not used to find the features. Accuracy improves in all five models. The after-intervention score ranges from 48% to 70%, and the reported rescue rate ranges from 34% to 60%. The effect is consistent across different model families and sizes. Crucially, this improvement is not driven by a trivial refusal bias: positive steering reliably enhances accuracy on both refusal and non-refusal decisions alike (Appendix Table 13).

5.3 Causal Direction and Specificity

Held-out rescue shows that the pathway can change behavior, but not whether the effect depends on its direction or selected features. Table 2 shows consistent gains from corrective intervention, little effect from random features, and lower accuracy under reverse intervention across most models. These controls show that the effect is both directional and feature-specific.

5.4 Cross-Dataset Generalization

To assess the robustness of our findings, we evaluate whether the discovered pathway generalizes beyond the HOB dataset. We inject the frozen HOB pathway into three external benchmarks—BrainBench, FeasibilityQA, and NormBank (Tang, 2026; Gupta et al., 2023; Ziems et al.,

Table 3: Cross-dataset transfer of the frozen Gemma-3-4B pathway discovered on HOB.

Dataset	n	Baseline	$\alpha = 1$	$\alpha = 5$
BrainBench	100	32.0	35.0	53.0
FeasibilityQA	500	63.8	64.0	69.6
NormBank	500	64.4	64.4	67.0

2023)—which also involve reasoning under implicit constraints and heuristic cues. As shown in Table 3, the injection yields consistent performance gains across all three datasets. These results provide preliminary evidence that the corrective direction is not specific to HOB, while the variation in effect size suggests that transfer depends on how closely the target task instantiates the same heuristic–constraint conflict.

5.5 Takeaway

These interventions move the evidence from circuit attribution to causal control. Across five models, activating correct pathway rescues unseen shortcut failures. Random, reverse, and zero controls show that the effect is feature-specific and directional. Applying the frozen HOB pathway to three external benchmarks further shows that this function extends beyond the discovery dataset. Together with the J-Lens and circuit-tracing results, these findings connect the late representational reversal to a feature-level pathway that can causally alter the final answer.

6 Discussion

A mechanistic account of shortcut persistence. The evidence is consistent with a temporal competition. A salient heuristic need not disappear for reasoning to succeed; instead, a late suppressive computation must become strong enough to reverse the impending answer. This explains how a model can expose relevant knowledge under explicit prompting and still fail zero-shot: availability and control are different properties. It also explains why CoT is unreliable here. Additional tokens may elaborate the currently dominant proposal without changing the late competition that selects the decision.

Limitations. The mechanistic study is restricted to five open-weight models no larger than 14B, with model-specific transcoders, feature counts, and strengths. HOB is controlled but synthetic; J-lens events depend on a first-token proxy and cover 130/142 base items. Transcoder features may be polysemantic, and replacement-model attribution omits some nonlinear and attention-mediated effects. Intervention outputs are scored by large-model judges rather than exhaustively by humans.

7 Conclusion

We investigated why language models follow salient heuristics even when an implicit constraint makes the resulting answer invalid. Combining layer-wise readouts, feature-level circuit tracing, and causal intervention, we find that successful reasoning does not avoid the shortcut from the start. Correct and incorrect examples share an early heuristic preference, but successful trajectories later show a corrective shift toward the constraint-consistent answer. Circuit tracing identifies a late feature pathway associated with this shift, and targeted interventions show that the pathway can causally alter shortcut reliance.

These findings reframe shortcut behavior as a problem of conditional control, rather than simply missing knowledge or unwanted feature encoding. The corrective function recurs across model families and transfers to related implicit-constraint tasks, although its benefit depends on task structure. Reliable reasoning may therefore require not the removal of useful heuristics, but mechanisms that recognize when those heuristics are invalid and successfully override them.

Ethics Statement

This work studies inference-time interventions in open-weight language models. HOB and the external datasets contain no private user data in the experiments reported here. Dataset licenses and original ethics documentation should be retained in the release.

Reproducibility Statement

The supplemental snapshot includes frozen train/held-out splits, model-specific feature vectors, intervention outputs, judge summaries, layer-wise J-lens artifacts, environment notes, and ordered reproduction scripts. Sections 3–5 specify the readout, discovery contrasts, and interventions; Appendices A provide evaluation settings. Full attribution graphs are not redistributed because of their 70–100 GB footprint, but graph-building scripts and saved vectors required for re-injection are included.

AI Use Statement

Generative AI tools were used to assist with organization, language editing, and initial drafting of this manuscript. They were not used to create the experimental measurements reported in the archived project outputs. The authors take responsibility for the final content of the work, including text prepared with AI assistance.

References

Emmanuel Ameisen, Jack Lindsey, Adam Pearce, Wes Gurnee, Nicholas L. Turner, Brian Chen, Craig Citro, David Abrahams, Shan Carter, Basil Hosmer, Jonathan Marcus, Michael Sklar, Adly Templeton, Trenton Bricken, Callum McDougall, Hoagy Cunningham, Thomas Henighan, Adam Jermyn, Andy Jones, Andrew Persic, Zhenyi Qi, T. Ben Thompson, Sam Zimmerman, Kelley Rivoire, Thomas Conerly, Chris Olah, and Joshua Batson. Circuit tracing: Revealing computational graphs in language models. *Transformer Circuits Thread*, 2025. URL `https://transformer-circuits.pub/2025/attribution-graphs/methods.html`.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *Advances in Neural Information Processing Systems*, 37:136037–136083, 2024.

Yonatan Belinkov. Probing classifiers: Promises, shortcomings, and advances. *Computational Linguistics*, 48(1):207–219, 2022.

Nora Belrose, Igor Ostrovsky, Lev McKinney, Zach Furman, Logan Smith, Danny Halawi, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Trenton Bricken, Adly Templeton, Joshua Batson, Brian Chen, Adam Jermyn, Tom Conerly, Nick Turner, Cem Anil, Carson Denison, Amanda Askell, Robert Lasenby, Yifan Wu, Shauna Kravec, Nicholas Schiefer, Tim Maxwell, Nicholas Joseph, Zac Hatfield-Dodds, Alex Tamkin, Karina Nguyen, Brayden McLean, Josiah E Burke, Tristan Hume, Shan Carter, Tom Henighan, and Christopher Olah. Towards monosemanticity: Decomposing language models with dictionary learning. *Transformer Circuits Thread*, 2023. https://transformer-circuits.pub/2023/monosemantic-features/index.html.

Christopher Clark, Mark Yatskar, and Luke Zettlemoyer. Don't take the easy way out: Ensemble based methods for avoiding known dataset biases. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language Processing (EMNLP-IJCNLP)*, pp. 4069–4082, 2019.

Jacob Dunefsky, Philippe Chlenski, and Neel Nanda. Transcoders find interpretable llm feature circuits. *Advances in Neural Information Processing Systems*, 37:24375–24410, 2024.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Himanshu Gupta, Neeraj Varshney, Swaroop Mishra, Kuntal Kumar Pal, Saurabh Arjun Sawant, Kevin Scaria, Siddharth Goyal, and Chitta Baral. "john is 50 years old, can his son be 65?" evaluating nlp models' understanding of feasibility. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pp. 407–417, 2023.

Wes Gurnee, Nicholas Sofroniew, Adam Pearce, Mateusz Piotrowski, Isaac Kauvar, Runjin Chen, Anna Soligo, Paul Bogdan, Euan Ong, Rowan Wang, Ben Thompson, David Abrahams, Subhash Kantamneni, Emmanuel Ameisen, Joshua Batson, and Jack Lindsey. Verbalizable representations form a global workspace in language models. *Transformer Circuits Thread*, 2026. URL `https://transformer-circuits.pub/2026/workspace/index.html`.

Michael Hanna, Mateusz Piotrowski, Jack Lindsey, and Emmanuel Ameisen. Circuit-tracer: A new library for finding feature circuits. In *Proceedings of the 8th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pp. 239–249, 2025.

Stefan Heimersheim and Neel Nanda. How to use and interpret activation patching. *arXiv preprint arXiv:2404.15255*, 2024.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (emnlp-ijcnlp)*, pp. 2733–2743, 2019.

Divyansh Kaushik, Eduard Hovy, and Zachary C Lipton. Learning the difference that makes a difference with counterfactually-augmented data. *arXiv preprint arXiv:1909.12434*, 2019.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35: 22199–22213, 2022.

Sebastian Lapuschkin, Stephan Wäldchen, Alexander Binder, Grégoire Montavon, Wojciech Samek, and Klaus-Robert Müller. Unmasking clever hans predictors and assessing what machines really learn. *Nature communications*, 10(1):1096, 2019.

Bruce W Lee, Inkit Padhi, Karthikeyan Natesan Ramamurthy, Erik Miehling, Pierre Dognin, Manish Nagireddy, and Amit Dhurandhar. Programming refusal with conditional activation steering. In *International conference on learning representations*, volume 2025, pp. 90960–90985, 2025.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in neural information processing systems*, 36:41451–41530, 2023.

Xiang Lorraine Li, Adhiguna Kuncoro, Jordan Hoffmann, Cyprien de Masson d'Autume, Phil Blunsom, and Aida Nematzadeh. A systematic investigation of commonsense knowledge in large language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pp. 11838–11855, 2022.

Yubo Li, Lu Zhang, Tianchong Jiang, Ramayya Krishnan, and Rema Padman. The model says walk: How surface heuristics override implicit constraints in llm reasoning. *arXiv preprint arXiv:2603.29025*, 2026.

Michael Mendelson and Yonatan Belinkov. Debiasing methods in natural language understanding make bias more accessible. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 1545–1557, 2021.

Hadas Orgad, Seraphina Goldfarb-Tarrant, and Yonatan Belinkov. How gender debiasing affects internal model representations, and why it matters. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2602–2628, 2022.

Ehsan Qasemi, Filip Ilievski, Muhao Chen, and Pedro Szekely. Paco: Preconditions attributed to commonsense knowledge. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pp. 6781–6796, 2022.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024.

Shiori Sagawa, Pang Wei Koh, Tatsunori B Hashimoto, and Percy Liang. Distributionally robust neural networks for group shifts: On the importance of regularization for worst-case generalization. *arXiv preprint arXiv:1911.08731*, 2019.

Yuzhe Tang. Brainbench: Exposing the commonsense reasoning gap in large language models. *arXiv preprint arXiv:2603.14761*, 2026.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Adly Templeton, Tom Conerly, Jonathan Marcus, Jack Lindsey, Trenton Bricken, Brian Chen, Adam Pearce, Craig Citro, Emmanuel Ameisen, Andy Jones, Hoagy Cunningham, Nicholas L Turner, Callum McDougall, Monte MacDiarmid, C. Daniel Freeman, Theodore R. Sumers, Edward Rees, Joshua Batson, Adam Jermyn, Shan Carter, Chris Olah, and Tom Henighan. Scaling monosemanticity: Extracting interpretable features from claude 3 sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*, 2023.

Prasetya Ajie Utama, Nafise Sadat Moosavi, and Iryna Gurevych. Towards debiasing nlu models from unknown biases. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 7597–7610, 2020.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Yu Yuan, Lili Zhao, Kai Zhang, Guangting Zheng, and Qi Liu. Do llms overcome shortcut learning? an evaluation of shortcut challenges in large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 12188–12200, 2024.

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, volume 2024, pp. 1651–1678, 2024.

Caleb Ziems, Jane Dwivedi-Yu, Yi-Chia Wang, Alon Halevy, and Diyi Yang. Normbank: A knowledge bank of situational social norms. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7756–7776, 2023.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A HOB BENCHMARK AND BEHAVIORAL EVALUATION

A.1 DATASET COMPOSITION

The Heuristic Override Benchmark (HOB) contains controlled decision problems in which an appealing heuristic may conflict with an implicit constraint (Li et al., 2026). The four heuristic families are proximity, efficiency, cost, and semantic association; the five constraint families are physical presence, capability, validity, scope, and procedure. Table 4 lists the evaluation categories.

Table 4: Composition of the 500-item HOB test set.

Category	n	Purpose
Conflict base	142	Heuristic conflicts with an implicit constraint
Matched pair	141	Constraint is removed and the heuristic becomes valid
Heuristic strength	90	Constraint is fixed while heuristic salience varies
Constraint explicitness	97	Heuristic is fixed while constraint visibility varies
Control	30	No heuristic–constraint conflict

Table 5: Illustrative HOB items from the capability-constraint family. The variants isolate whether the salient speed heuristic or the physical constraint should determine the decision. Wording is abbreviated for presentation.

Slice	Example prompt contrast	Reference decision
Conflict base	A 500-pound gun safe must be moved upstairs. What is fastest: carry it alone or hire movers?	Hire movers
Matched pair	Replace the safe with a desk lamp, removing the capability constraint while retaining the speed comparison.	Carry it oneself
Heuristic strength	Keep the safe fixed and vary the speed cue: "instant," "saves 30 minutes," or "a bit faster."	Hire movers
Constraint explicitness	Keep the speed cue fixed and vary constraint visibility: weight only, "by myself" hint, or an explicit 50-pound lifting limit.	Hire movers

A.2 EVALUATION AND ATTRIBUTION PROMPT

We use the same prompt for behavioral evaluation and attribution-graph construction. This avoids introducing a prompt mismatch between the measured behavior and the subsequent mechanistic analysis. The following is the exact prompt used for the car-wash example:

Evaluation prompt

```
I need to get my car washed before a road trip tomorrow.
The car wash is only 5 minutes away from my apartment.
Should I walk there or drive?

Output your answer in this order:
1.  First line:  <DECISION>your short final answer</DECISION>
2.  Then, briefly explain your reasoning.

For example:  <DECISION>Walk</DECISION>
```

For behavioral evaluation, we extract the answer enclosed by the `<DECISION>` and `</DECISION>` tags and compare its meaning with the reference decision. The explanation is not required to match a reference string.

For attribution-graph construction, we use the same prompt and trace the first token of the decision answer, such as *Walk* or *Drive*. Thus, the behavioral and mechanistic analyses target the same decision under the same prompting condition.

A.3 Decoding, and Scoring

The direct condition appends an instruction requesting an exact final decision. The CoT condition additionally asks the model to reason step by step before giving that decision. The knowledge-added condition inserts the benchmark's hidden constraint and asks the model to answer using it. Greedy decoding provides the main behavioral result. Ten sampled generations are also scored; *strict-10* requires all ten decisions to agree with the gold answer. Model outputs are evaluated for decision equivalence rather than exact string match.

For mechanistic analysis, only conflict-base items are used. Strict-wrong cases are stratified by heuristic family and split into discovery and held-out test set (Table 6); the split is frozen before feature construction. J-lens token-pair coverage across shortcut-bearing slices is reported in Table 11; the confirmatory crossover analysis uses only the 130 covered conflict-base items.

Table 6: Frozen base-item splits used for feature discovery and held-out intervention.

Model	Strict correct	Wrong discovery	Wrong held-out
Gemma-3-4B	62	38	42
Gemma-3-12B	90	25	27
Qwen3-4B	44	48	50
Qwen3-8B	51	45	46
Qwen3-14B	71	35	36

A.4 Behavioral Results

We report three complementary views of behavioral performance. Table 7 compares greedy direct-answer and CoT prompting. Direct accuracy ranges from 43.8% to 71.0%, while CoT changes performance by only -3.2 to $+3.8$ percentage points and provides no consistent improvement. Table 8 evaluates the effect of explicitly stating the missing constraint, which improves accuracy for every model by 12.4 to 28.6 percentage points. Finally, Table 9 applies the stricter requirement that all ten sampled generations be correct.

Table 7: Greedy direct-answer and chain-of-thought (CoT) accuracy (%) on all 500 HOB items.

Model	Direct	CoT
Gemma-3-1B-IT	44.2	41.0
Gemma-3-4B-IT	60.6	60.0
Gemma-3-12B-IT	66.6	66.0
Gemma-3-27B-IT	68.0	67.2
Qwen3-1.7B	43.8	47.6
Qwen3-4B	50.6	52.6
Qwen3-8B	59.8	60.4
Qwen3-14B	67.6	67.6
Qwen3-32B	71.0	71.8

Table 8: Accuracy (%) after explicitly adding the missing constraint to the prompt. Gain is measured in percentage points.

Model	Zero-shot	Knowledge-added	Gain
Gemma-3-1B-IT	44.2	60.6	16.4
Gemma-3-4B-IT	60.6	75.6	15.0
Gemma-3-12B-IT	66.6	82.2	15.6
Gemma-3-27B-IT	68.0	84.6	16.6
Qwen3-1.7B	43.8	60.4	16.6
Qwen3-4B	50.6	79.2	28.6
Qwen3-8B	59.8	78.6	18.8
Qwen3-14B	67.6	80.0	12.4
Qwen3-32B	71.0	84.8	13.8

Table 9: Strict ten-run consistency accuracy (%) under direct and CoT prompting. Results were unavailable for the two largest models in the archived strict-10 sweep.

Model	Direct	CoT
Gemma-3-1B-IT	26.4	19.4
Gemma-3-4B-IT	54.4	53.4
Gemma-3-12B-IT	62.0	62.0
Qwen3-1.7B	35.6	37.8
Qwen3-4B	47.2	47.8
Qwen3-8B	53.6	54.0
Qwen3-14B	63.8	63.6

B Model and Evaluation Details

Checkpoints and transcoders. The experiments use `google/gemma-3-4b-it`, `google/gemma-3-12b-it`, and `Qwen/Qwen3-{4B,8B,14B}`. Gemma uses Gemma Scope 2 per-layer transcoders with a 16k dictionary; Qwen uses the packaged `mwhanna/qwen3-*`-transcoder checkpoints. Graph construction and intervention run through the `circuit-tracer` replacement-model API.

Outcome judging. The primary judge is Gemma-3-27B-IT. A Qwen-27B judge separately scores decision agreement (L1) and decision-plus-rationale agreement (L2). A 99-example human audit reported 87.9% accuracy and Cohen's $\kappa = 0.73$ for Gemma-27B, 85.9% and $\kappa = 0.70$ for Qwen L1, and 82.8% and $\kappa = 0.65$ for Qwen L2.

Table 10: Agreement of automated judges with the 99-example human audit. L1 scores decision agreement; L2 additionally requires rationale agreement.

Judge	Accuracy	Cohen's κ	Precision	Recall
Gemma-3-27B	87.9	0.73	98.3	84.3
Qwen-27B L1	85.9	0.70	100.0	80.0
Qwen-27B L2	82.8	0.65	100.0	75.7

C J-Lens Association Statistics

C.1 First token Coverage

Table 11: Coverage of the J-lens first-token contrast. An item is covered when the gold and shortcut decisions have distinct first answer tokens. Only covered conflict-base items enter the confirmatory crossover statistics below.

Slice	Covered	Total	Coverage (%)
Conflict base	130	142	91.5
Heuristic strength	79	90	87.8
Constraint explicitness	83	97	85.6
All shortcut-bearing slices	292	329	88.8

Table 12: Association between a J-lens crossover and strict correctness on 130 first-token-distinguishable base items. "With/without" is accuracy (%) conditional on the presence/absence of a crossover.

Model	Accuracy with / without	Odds ratio	Fisher p	Final-margin AUC
Gemma-3-4B	75.0 / 30.9	6.72	8.2×10^{-6}	0.879
Gemma-3-12B	73.3 / 62.0	1.69	0.285	0.865
Qwen3-4B	71.0 / 18.2	11.00	1.0×10^{-7}	0.736
Qwen3-8B	82.4 / 20.8	17.73	2.8×10^{-10}	0.792
Qwen3-14B	75.8 / 42.3	4.27	1.1×10^{-3}	0.766

C.2 Same evidence: J-lens vs circuit tracer

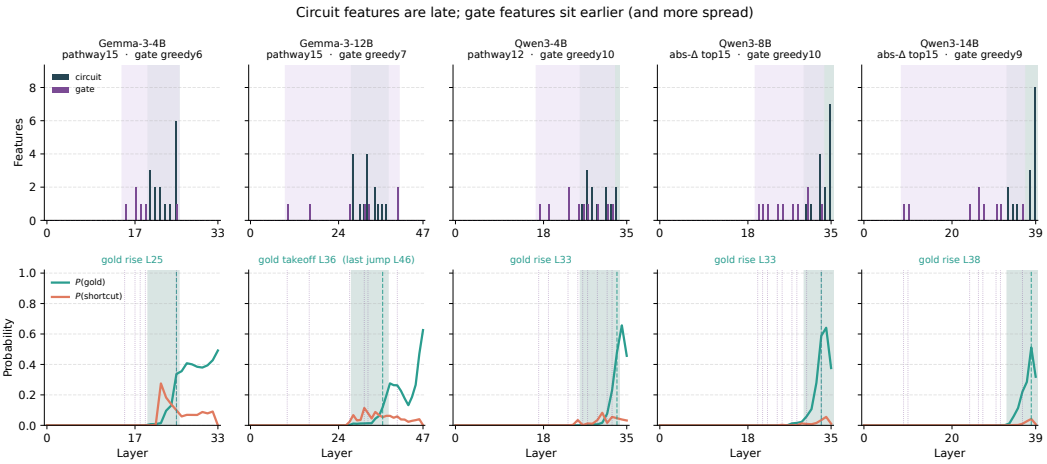

Figure 6: Relative-depth alignment of selected pathway features, exploratory competition features, and mean J-lens candidate curves. The late feature bundles overlap the gold-probability rise; Competition features are earlier and more dispersed. competition features in this visualization come from exploratory searches.

C.3 Full Layer-Wise J-Lens Trajectories

Figure 7 reports the complete layer-wise J-Lens trajectories for all five models included in the mechanistic analysis. For each model, we separately aggregate conflict-base examples according to whether the model's final answer is strictly correct or wrong. The curves show the mean J-Lens probability assigned to the first answer token of the gold and shortcut responses, with shaded regions indicating the standard error of the mean.

Across models, support for both candidate answers remains small through most early and middle layers and becomes distinguishable primarily at greater depth. Correct examples generally exhibit a late increase in support for the gold answer, whereas wrong examples retain stronger support for the shortcut answer. The exact onset and magnitude of this separation vary across models, indicating that the late-correction pattern is qualitatively shared but not identical across architectures and model sizes.

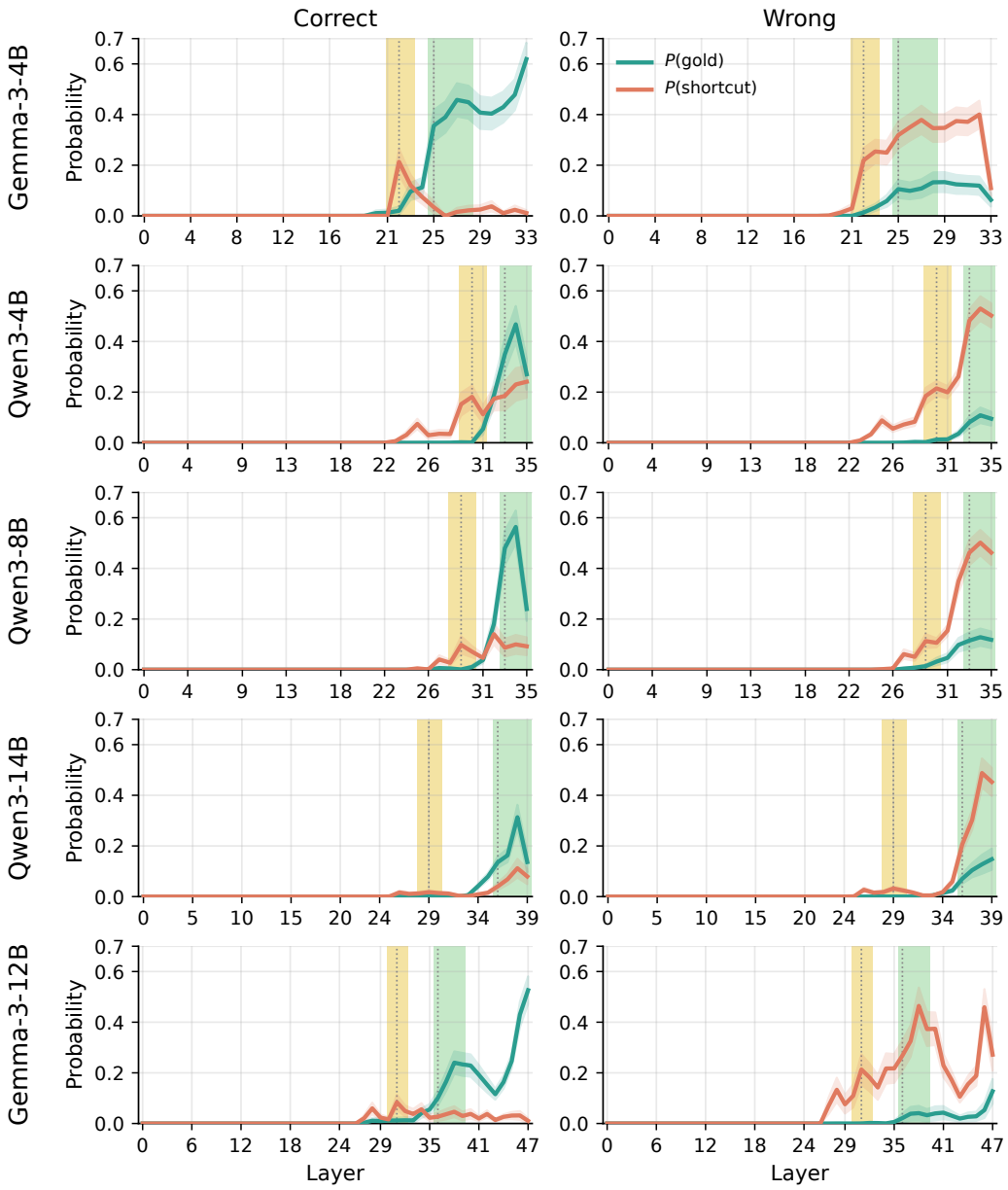

Figure 7: Full layer-wise J-Lens trajectories for the five models included in the mechanistic analysis. Rows correspond to models, and columns separate strict-correct from strict-wrong conflict-base examples. Teal curves show the mean probability assigned to the gold answer's first distinguishing token, while coral curves show the corresponding probability for the shortcut answer; shaded envelopes denote SEM. Yellow regions indicate the layer range containing the selected competition features, and green regions indicate the range containing the late corrective features. Dotted vertical lines mark the representative layer of each feature group. Correct examples generally exhibit a late increase in gold-answer support, whereas shortcut support remains dominant in wrong examples.

D Pathway Feature Selection

This section describes how we construct the corrective and competition components of the pathway.

D.1 Corrective Seed Features

For each model, we compute the correct–wrong activation difference Δ_j^{CW} for every candidate feature j, as defined in Equation 2. We rank features by $|\Delta_j^{\mathrm{CW}}|$ and retain the top 64 as the initial corrective seed pool. The sign of Δ_j^{CW} is preserved and later determines the positive intervention direction.

We inspect the top-64 feature examples and local attribution-graph context of the candidate features. We remove a feature only when its activation is dominated by tokenization or prompt-formatting patterns, including isolated subword fragments, whitespace, line breaks, punctuation, prompt delimiters, or fixed answer-formatting tokens.

D.2 Graph-Linked Upstream Candidates

Starting from the retained corrective seed set $\mathcal{C}_{\mathrm{corr}}$, we use the discovery attribution graphs to identify upstream features linked to the seed features. For an upstream candidate u, we aggregate its attribution influence on the corrective set across discovery examples:

$$L(u) = \frac{1}{|\mathcal{D}_{\mathrm{disc}}|} \sum_{x \in \mathcal{D}_{\mathrm{disc}}} \sum_{v \in \mathcal{C}_{\mathrm{corr}}} \left| A_{u \to v}(x) \right|, \tag{4}$$

where $A_{u \to v}(x)$ is the attribution-graph influence from feature u to corrective feature v on example x. We retain the 64 highest-scoring non-artifact upstream features as the competition candidate pool.

D.3 Nested Competition-Feature Selection

We first intervene on each upstream candidate individually using its discovery-defined direction. Let $\mathcal{D}_{\mathrm{sel}}$ denote the discovery examples used for subset selection. For a feature set S, we define its intervention gain as

$$G(S) = \mathrm{Acc}\left(M^{(+S)}; \mathcal{D}_{\mathrm{sel}}\right) - \mathrm{Acc}\left(M; \mathcal{D}_{\mathrm{sel}}\right), \tag{5}$$

where $M^{(+S)}$ denotes the model under positive intervention on features in S. The intervention strength is fixed during feature selection.

We rank the upstream candidates by their individual gains $G(\{u\})$. Following this fixed ranking, we construct nested sets

$$S_k = \{u_1, \ldots, u_k\}, \qquad k = 1, \ldots, 64. \tag{6}$$

We then select

$$k^* = \min\left\{ k : G(S_k) \geq 0.95 \max_m G(S_m) \right\}. \tag{7}$$

Thus, S_{k^*} is the smallest tested nested set that retains at least 95% of the maximum discovery-set gain. We refer to this set as the candidate competition feature set, $\mathcal{C}_{\mathrm{comp}} = S_{k^*}$.

E Qualitative Intervention Examples

We provide representative before/after generations to illustrate how the interventions change model decisions and rationales. These examples are descriptive rather than independent causal evidence: the quantitative held-out comparisons remain the primary estimand. The original generations place their decision strings inside `<DECISION>...</DECISION>`; these tags are omitted below for readability.

Example 1: Correction of a cost-based shortcut.
Implicit constraint: A three-tier wedding cake cannot be safely transported on a public bus due to vehicle dynamics and fragile physical requirements.
Question: *I need to transport a three-tier wedding cake to the reception venue across town. Taking public transit is the cheapest option at $2.50 versus $30 for a car service. Should I take the bus with the cake?*

Before Intervention (*Shortcut Reliance*)

Output: Yes. Taking the bus is the cheaper option at $2.50 compared with $30 for a car service, so it makes financial sense to take the bus with the cake.

After Intervention (*Constraint Override*)

Output: No. Taking the bus with the cake is not advisable because it is not designed to carry large, fragile items like a three-tier wedding cake. The bus lacks secure holding areas and climate control. It is better to use a specialized delivery service.

Figure 8: The intervention causally flips both the output token and the internal rationale: suppressing the shortcut features prevents the model from being blinded by the cost cue.

Example 2: Correction of a distance-based shortcut(car wash).

Implicit constraint: The car must physically be at the car wash to be washed.
Question: My car is parked at home. The car wash is 5 minutes away. How do I get the car to the car wash?

Before Intervention (*Shortcut Reliance*)

Output: Walk. It takes only 5 minutes to walk to the car wash, so walking is the most efficient and straightforward way to get your car there.

After Intervention (*Constraint Override*)

Output: Drive. The car is at home, and the car wash is 5 minutes away. To get the car to the car wash, you would need to drive or walk to the car wash location. Since the car is parked at home, you can drive it to the car wash.

Figure 9: The intervention causally flips both the output token (from Walk to Drive) and the internal rationale: suppressing the shortcut circuit prevents the model from being blinded by the distance cue.

F Pathway Specificity Controls

Table 13 tests whether the positive intervention merely strengthens a generic refusal or negation tendency. To evaluate this, we measure the intervention's effect on samples where the gold label is refusal, as well as on samples representing non-refusal actions (e.g., 'walk', 'drive'). This latter improvement would not be expected if the intervention merely induced generic refusal or negation. The intervention therefore appears to support constraint-sensitive correction beyond direct rejection cases.

Table 13: Gold-decision control for the positive Gemma-3-4B intervention on conflict-base items, separating refusal or negation decisions from other decisions.

Gold decision	n	Before	After
Refusal/negation (*No*)	52	61.5	90.4
Non-*No* decision	90	44.4	51.1

G Causal Intervention Details

This section reports the settings and additional control results for the causal interventions.

G.1 Intervention Settings

Frozen specification. For each model, the pathway features and their signed activation differences Δ_j^{CW} are computed only from the discovery split, as defined. The feature set, intervention directions,

and primary strength $\alpha = 5$ are fixed before evaluation on `wrong_half_2`. The held-out split is not used for feature selection, direction selection, or hyperparameter tuning.

Intervention definitions. For a selected feature j, we modify its activation at the final prompt position as

$$\widetilde{a}_j(x) = \alpha \Delta_j^{\mathrm{CW}}. \tag{8}$$

Positive intervention uses $\alpha = 5$. Reverse intervention uses $\alpha = -5$ and applies the same change in the opposite direction. Zero ablation instead sets the selected feature activation to zero. The random control replaces the pathway with a size-matched random feature set while keeping the intervention rule and strength unchanged.

G.2 Intervention-Strength Calibration

Table 14 evaluates the pathway effect at different intervention strengths. For each model, we apply the frozen pathway to its held-out `wrong_half_2` split using a range of α values. The following strength sweep is reported only as a post-hoc sensitivity analysis and was not used to select the primary setting.

Table 14: Held-out accuracy (%) across intervention strengths for the full feature bundle.

Model	Baseline	$\alpha = 3$	$\alpha = 5$	$\alpha = 10$
Gemma-3-4B	14.3	21.4	50.2	40.5
Gemma-3-12B	3.7	37.0	40.7	48.1
Qwen3-4B	10.0	24.0	70.0	56.0
Qwen3-8B	19.6	43.5	54.3	17.4
Qwen3-14B	19.4	52.8	63.9	44.4

The intervention improves held-out accuracy for every model at least one tested strength. An intermediate strength of $\alpha = 5$ performs best for four of the five models, whereas Gemma-3-12B achieves its highest accuracy at $\alpha = 10$. Performance is not monotonic in intervention strength: stronger interventions can reduce accuracy, most notably for Qwen3-8B. This pattern motivates model-specific calibration rather than assuming that a larger intervention is always better.

G.3 Competition Feature Ablation

We next test whether suppression alone is sufficient. Table 15 compares the full pathway with a version that removes the competition features. The evaluation uses unseen HOB items held out from the corresponding intervention analysis.

Table 15: Competition feature ablation on unseen HOB items. "sup" applies the suppression component without the competition features, while "Full" uses full pathway features. Accuracy is reported in percent.

Model	Baseline	sup	Random	Full
Gemma-3-4B	59.4	47.8	58.2	65.4
Qwen3-4B	48.8	46.6	47.2	73.7
Qwen3-8B	54.8	52.0	55.4	75.1
Qwen3-14B	69.1	60.0	71.2	82.9
Gemma-3-12B	61.6	48.8	60.4	67.1

Removing the competition features lowers accuracy relative to the full pathway in every model and often reduces performance below baseline. Random-feature interventions remain close to baseline, while the full pathway achieves the highest accuracy in four of the five models. These results indicate that unconditional shortcut suppression is insufficient: the competition features are needed to restrict suppression to cases in which corrective intervention is likely to be useful.

Competition features cues. The recorded cues are *rare*/*wait* for Gemma-3-4B, *responsibly*/*supervise* for Qwen3-4B, *depending*/*consult* for Qwen3-8B, *ask* for Qwen3-14B, and *focus* for Gemma-3-12B. These examples are descriptive; the causal test uses the selected feature sets, not the words themselves.