WHEN NEURONS COLLIDE: WHY INDEPENDENTLY SUCCESSFUL COMPUTATIONS FAIL TOGETHER

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models routinely carry out multiple computations within a single forward pass. Because these computations read from and write to the same residual stream, they can interact with one another. We study *neural collisions*: cases in which computations that succeed independently fail when processed together, even when both share the same answer. Across LLAMA-3.1-8B-INSTRUCT, GEMMA-4-31B-IT, and QWEN3-32B, we find that neural collisions are the extreme case of a broader phenomenon: concurrent computations routinely influence one another during inference, sometimes enough to overturn an otherwise correct prediction. In a collision, one computation *corrupts* another, pushing its internal representation away from the state it reaches in isolation. Using activation patching, we show that this corruption emerges abruptly, at a small number of token positions over just a few layers. To quantify this corruption, we introduce *residual drift*, the distance between a computation's joint and isolated trajectories. Across all three models, residual drift strongly predicts how much a collision changes the model's answer.

1 INTRODUCTION

Large language models routinely carry out multiple computations within a single forward pass (Feng & Steinhardt; Xiong et al., 2024). Because these computations read from and write to the same residual stream, they can interact with one another. We study cases where this interaction becomes destructive.

To make this concrete, Figure 1 shows LLAMA-3.1-8B-INSTRUCT evaluating the truthfulness of the statement *"Madison is the capital of Wisconsin"*. Asked on its own, the model correctly answers `Yes`. We then prepend a second, unrelated statement, and ask exactly the same question. The model now incorrectly answers `No`.

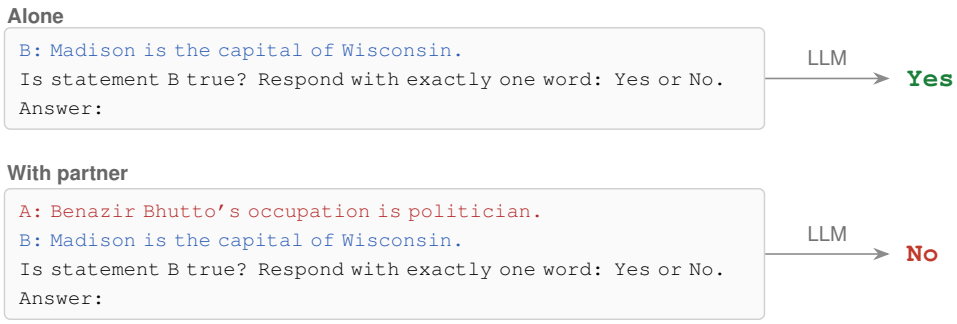

Figure 1: Independent computations can collide when processed together.

We call such failures *neural collisions*: cases in which two computations that succeed independently can fail when run together. This leads to the central question of the paper: what determines whether concurrent computations coexist successfully or collide?

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.

Related failures are well documented: irrelevant or conflicting context can degrade performance (Shi et al., 2023), with several studies tracing these failures to competition between opposing candidate answers (Ortu et al., 2024; Jin et al., 2024; Niu et al., 2025; Cheng et al., 2025). Neural collisions expose a complementary failure mode. Because we pair statements with the same ground-truth label, there is no competition between conflicting answers.

We find neural collisions in every model tested: LLAMA-3.1-8B-INSTRUCT, GEMMA-4-31B-IT and QWEN3-32B. To understand how they arise, we reconstruct these collisions inside the network, localizing where the failure first appears, down to individual tokens and layers, and tracing how it propagates to the model's answer.

Our contributions are as follows:

- We introduce *neural collisions* as a controlled setting for studying how concurrent computations disrupt one another.
- We show that neural collisions arise in all models tested, and form the extreme cases of a more general phenomenon in which concurrent processing routinely alters model predictions.
- We show that in a collision, one computation corrupts another, pushing its internal representation away from the state it reaches in isolation. This corruption emerges *abruptly*, at a few token positions over a small number of layers, and then propagates to the final answer position, where it changes the model's prediction.
- We introduce *residual drift* to measure how far corruption displaces a computation from its isolated trajectory, and show that it strongly predicts how much a collision changes the model's answer.

2 RELATED WORK

Concurrent computation in shared representations. Language models can maintain and manipulate multiple representations within a single forward pass. Models can perform several in-context tasks simultaneously (Xiong et al., 2024) and keep distinct entities separated within shared internal state (Feng & Steinhardt). More generally, work on superposition shows how neural networks can represent many features within a limited representational space, while also exposing the possibility of interference between them (Elhage et al., 2022). These results establish that multiple computations can coexist within the same network. We, on the other hand, study what happens when that coexistence breaks down.

Failures induced by irrelevant context. A substantial literature shows that irrelevant context can alter otherwise-correct predictions. For example, Shi et al. (2023) show that adding a single irrelevant sentence to a grade-school mathematics problem can substantially reduce accuracy. Mechanistic studies have since identified several ways in which such context can influence the final answer.

One route is direct competition between candidate answers. In studies of knowledge conflict, the prompt asserts something that contradicts a fact the model has memorized, leaving the model with two competing candidates: one recalled from memory and one supplied by the context. Ortu et al. (2024) trace how these competing mechanisms evolve and which comes to dominate the prediction. Similarly, Jin et al. (2024) identify attention heads that carry the competing candidates and find that the conflict emerges when their outputs are combined.

Another mechanism is that irrelevant context can supply material that the model incorporates into its answer. Cheng et al. (2025) show that models first infer the class of answer required (for example, a country) and then combine it with features drawn from the irrelevant context rather than from the query. Similarly, Niu et al. (2025) find that merely placing a token in the context can increase its probability of being generated later, even when it is irrelevant to the question.

Neural collisions complement this work by studying failures that arise even when the concurrent computations support the same answer. This rules out competition between distinct candidate answers as the source of the failure. Instead, we find that one computation corrupts the internal representation of another before that representation is used to form the answer.

3 WHAT ARE NEURAL COLLISIONS?

We now formalize the setting in which we study neural collisions.

3.1 SETUP

Let $\mathcal{S}$ be a set of *statements*: factual sentences, such as *Madison is the capital of Wisconsin*, each of which is either true or false. Each statement $X \in \mathcal{S}$ has a ground-truth label $y(X) \in \{\texttt{Yes}, \texttt{No}\}$, the correct answer to the question "Is X true?".

Prompts. We consider two prompting scenarios for the model: the *solo* scenario and the *joint* scenario. In the the solo scenario, the prompt contains a single statement and asks whether it is true:

```
B: [statement B]
Is statement B true?  Respond with exactly one word:  Yes or No.
Answer:
```

In the joint scenario, the prompt contains two statements but asks about only one of them. The example below asks about statement B:

```
A: [statement A]
B: [statement B]
Is statement B true?  Respond with exactly one word:  Yes or No.
Answer:
```

The labels A and B refer only to position: A appears first and B second. The statement being evaluated, $T \in \{A, B\}$, is the *target*; the other is its *partner*. For example, in the joint prompt above, B is the target and A is the partner.

Margin. We measure the model's answer by its *margin* m: the logit of the correct answer minus the logit of the incorrect one, so the model is correct when $m > 0$. For a target T, we write $m^{\text{solo}}(T)$ for its margin when evaluated alone, and $m^{\text{joint}}(A, B, T)$ for its margin in the joint prompt containing A and B. When the pairing is clear from context, we abbreviate these as m^{solo} and m^{joint}.

3.2 DEFINITIONS

With this notation in place, we now proceed to presenting the main definitions.

Definition (Neural Collision). For $A, B \in \mathcal{S}$, let $T \in \{A, B\}$ denote the target.

A *destructive neural collision* occurs when T is evaluated correctly in isolation but incorrectly in the joint prompt:

$$m^{\text{solo}} > 0 \qquad \text{and} \qquad m^{\text{joint}} < 0.$$

A *constructive neural collision* is the reverse: T is evaluated incorrectly in isolation but correctly in the joint prompt,

$$m^{\text{solo}} < 0 \qquad \text{and} \qquad m^{\text{joint}} > 0.$$

Neural collisions are the extreme cases, where the partner pushes the target across the decision boundary. More often, a partner strengthens or weakens the target's margin without changing the answer. To capture this broader effect, we also measure the *margin shift*.

Definition (Margin Shift). We define the target's *margin shift* as $\Delta m = m^{\text{solo}} - m^{\text{joint}}$.

A destructive neural collision occurs precisely when the margin shift is large enough to overcome the model's original confidence, i.e., $\Delta m > m^{\text{solo}}$.

4 Detecting neural collisions

We focus on *destructive* neural collisions to isolate computations that are reliable in isolation but fail under concurrent processing. To identify such cases, we proceed in two steps:

Step 1: Constructing a confident set. Because the task is binary, a model that does not know the answer will still be correct about half the time by chance. We want to exclude these guesses and focus on cases where the model genuinely knows the answer, but joint processing breaks that computation. We therefore fix a confidence threshold $\alpha > 0$ and keep only statements whose solo margin satisfies $m^{\text{solo}} \geq \alpha$. We call this the confident set $\mathcal{C}$.

Step 2: Searching for collisions We now ask whether placing two confident statements in the same prompt causes the model to fail on either of them. From the confident set $\mathcal{C}$, we sample two disjoint subsets, $\mathcal{A}$ and $\mathcal{B}$, each containing N statements, and place each pair $(A, B) \in \mathcal{A} \times \mathcal{B}$ together in a joint prompt. Since either statement could be the one that fails, we evaluate each pair twice: once asking about A, with B as its partner, and once asking about B, with A as its partner.

Our question is whether concurrent computations can disrupt one another even when they share the same ground-truth answer. If A were true and B false, the partner could simply pull the model toward the opposite answer. We therefore keep only pairs whose statements share the same ground-truth label (Yes–Yes or No–No), so that both support the same output token.

Our experiments therefore consider the subset of destructive neural collisions for which the target was confidently correct in isolation, $m^{\text{solo}} \geq \alpha$ and $m^{\text{joint}} < 0$.

5 Experiments and Results

We evaluate Qwen3-32B (Team, 2025), Gemma-4-31B-it (Team, 2026), and Llama-3.1-8B-Instruct (Grattafiori et al., 2024). These models test whether neural collisions arise across model families and model scales, rather than being specific to a single architecture. We combine PopQA facts (Mallen et al., 2023) with synthetic arithmetic to create a heterogeneous proposition bank spanning many semantic categories and types of computation.

For each model, we construct two disjoint sets, $\mathcal{A}$ and $\mathcal{B}$, each containing $N = 100$ statements using a confidence threshold of $\alpha = 2$. Of the $10{,}000$ pairs in $\mathcal{A} \times \mathcal{B}$, we keep the $5{,}000$ whose statements share a label. Each retained pair is evaluated twice, once with A as the target and once with B as the target, yielding $10{,}000$ target evaluations per model.

5.1 Results

Destructive collisions occur in all three models. For each target statement T, we compare its margin when evaluated alone, $m^{\text{solo}}(T)$, with its margin when evaluated alongside a partner, $m^{\text{joint}}(A, B, T)$. Figure 2 plots both distributions over all $10{,}000$ target evaluations. Because every target belongs to the confident set, all solo margins lie above $\alpha = 2$. Under joint processing, however, the distribution develops a left tail that crosses the decision boundary at $m = 0$. In these cases, a statement the model answers confidently on its own is answered incorrectly once another statement is present. These are destructive neural collisions.

Figure 2 shows that neural collisions occur. But collisions capture only cases in which the partner-induced change is large enough to flip the answer. To characterize the broader effect of joint processing, we therefore examine the margin shift, $\Delta m = m^{\text{solo}} - m^{\text{joint}}$, across all target–partner pairings.

Figure 3 reveals that neural collisions form the destructive tail of a broader distribution of partner-induced changes. Most partners shift the target margin without changing the answer, while a small number push it far enough to cross the decision boundary.

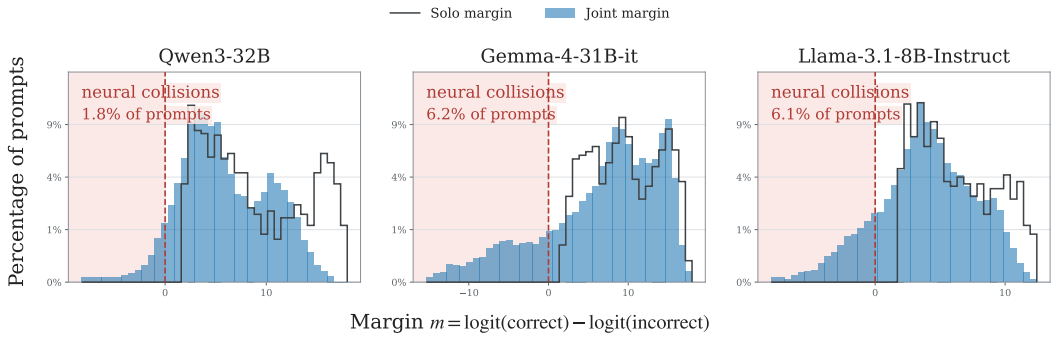

Figure 2: **Joint processing induces destructive neural collisions.** Colored histograms show margins under joint processing, m^{joint}, while black outlines show the corresponding solo margins, m^{solo}, for the same targets. The vertical axis uses a *square-root scale* to expose the tails.

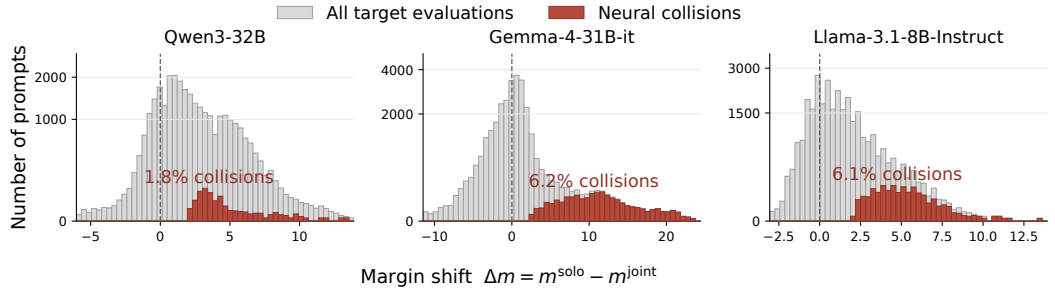

Figure 3: **Neural collisions are the extreme cases of a broader phenomenon of margin shifts.** Grey histograms show the distribution of margin shifts. Red bars highlight the subset of those same shifts that become destructive neural collisions. The vertical axis uses a *square-root scale* to expose the tails.

These results show that neural collisions occur, but not where the damage first arises or how it reaches the final answer. We next trace the collision through the network.

6 Reconstructing collisions inside the network

Experimental setup. In all mechanistic analyses that follow, the target is always statement B, and only the partner A varies. Because causal interventions across every token and layer are expensive, we restrict the mechanistic analyses to $n = 100$ destructive neural collisions sampled at random per model from the collision cases identified in the behavioural sweep.

The readout. We call the residual-stream state at the final token of the prompt the *readout*. This is the state from which the model's answer is decoded; in our prompts, it is the state at the colon in `Answer:`. Because the answer margin is determined by the final readout state, any collision must ultimately appear as a change in this state. The readout therefore shows where the damage ends up, but not where it began.

6.1 Activation patching

We distinguish these possibilities with activation patching, which tests whether a particular internal state is sufficient to reproduce a behaviour (Meng et al., 2022; Zhang & Nanda, 2024). We transplant internal states from the failed joint computation into the successful solo computation: if a transplanted state carries the collision, the solo computation should now fail too.

Let $h_{j,\ell}$ denote the residual-stream state at token position j at the input to decoder block ℓ.

Procedure (activation patching).

1. Run the solo and joint computations and record their answer margins, m^{solo} and m^{joint}.
2. Run the solo computation again. At layer ℓ, replace the residual-stream states at positions S with their values from the failed joint computation: $h^{\text{solo}}_{j,\ell} \leftarrow h^{\text{joint}}_{j,\ell}$ for all $j \in S$.
3. Continue the forward pass normally and record the resulting answer margin $m^{\text{patch}(S,\ell)}$.

We want to know how much of the collision is reproduced by transplanting these states alone. We measure this using *collision transfer*.

Definition (Collision transfer). The *collision transfer* is the fraction of the original solo-to-joint margin shift reproduced by the patch:

$$\tau(S,\ell) = \frac{m^{\text{solo}} - m^{\text{patch}(S,\ell)}}{m^{\text{solo}} - m^{\text{joint}}}.$$

A value of $\tau(S,\ell) = 0$ means that the patch reproduces none of the collision, while $\tau(S,\ell) = 1$ means that it reproduces the full solo-to-joint margin shift.

Where does the collision first leave a trace? At every layer, we compare two interventions:

- **B-token patch:** transplant all B-token states, $S = B$.
- **Readout patch:** transplant only the readout state, $S = \{\text{readout}\}$.

Together, these interventions let us track how collision damage moves from the target representation to the readout across network depth.

6.2 Results

Figure 4 reveals a two-stage pattern: at early and intermediate depth, collision transfer is carried mainly by the B-token states; over a narrow range of layers, this effect collapses as transfer through the readout rises, after which the readout carries the collision.

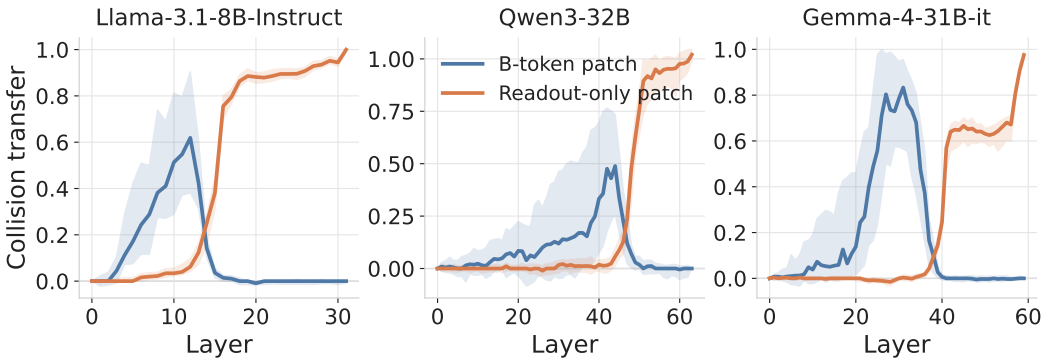

Figure 4: **Reconstructing a collision: damage is handed off from the target tokens to the readout.** Collision transfer for B-token patches (blue) and readout patches (orange). Shading shows the interquartile range.

7 Locating the Point of Impact

The transfer curves pool across target tokens and across collisions, so they cannot reveal whether the damage forms gradually or is localized to a particular token and depth.

To resolve this, we repeat the intervention one target token at a time. For each token $j \in B$ and layer ℓ, we write $\tau_j(\ell)$ for the collision transfer produced by patching that token alone. If the damage is localized, a single token should carry most of it:

Definition (Collision token). The *collision token*, $j^\star$, is the target token whose single-token transplant produces the largest collision transfer at any layer:

$$j^\star = \arg\max_j \max_\ell \tau_j(\ell).$$

The collision token marks where the damage becomes concentrated. To measure how abruptly it becomes concentrated there, we follow the collision token's transfer $\tau_{j^\star}(\ell)$ through depth and count how many layers it takes to rise:

Definition (Impact width). The *impact width*, $W_{20\text{–}80}$, is the number of layers over which $\tau_{j^\star}(\ell)$ rises from 20% to 80% of its peak value. Smaller values indicate that the collision emerges over a narrower range of layers.

7.1 Results

Table 1 shows that the impact is sudden. Across all three models, the dominant token leaps from 20% to 80% of its peak causal effect in a median of just 2 to 4.5 layers. Appendix B shows that this conclusion is robust to the choice of rise thresholds, including 10–90% and 30–70%.

Table 1: Depth localization of the collision token across 100 destructive collisions per model. Values report the median impact width $W_{20\text{–}80}$, with the interquartile range (IQR).

Model	Layers	Median $W_{20\text{–}80}$	[IQR]
Llama-3.1-8B-Instruct	32	3.0	[1.0, 4.0]
Gemma-4-31B-it	60	2.0	[1.0, 3.8]
Qwen3-32B	64	4.5	[3.0, 7.0]

Figure 5 visualizes what this sharp onset looks like in individual cases. For most token–layer positions, transplanting a single target-token state $h^{\text{solo}}_{j,\ell} \leftarrow h^{\text{joint}}_{j,\ell}$ has little effect; collision transfer then rises sharply over a small number of layers at one or a few target tokens.

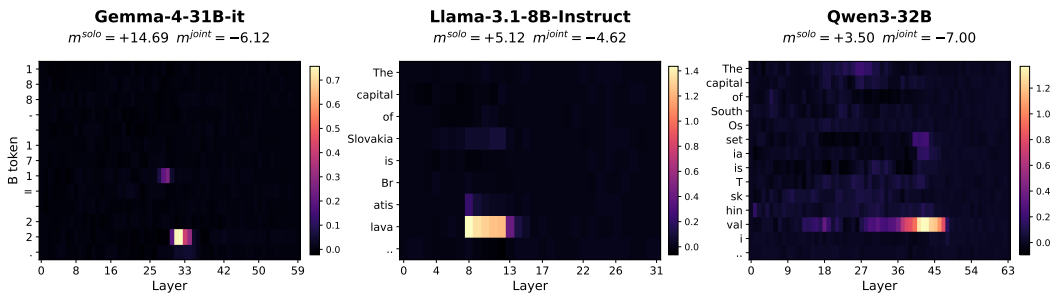

Figure 5: **Neural collisions are abrupt.** Single-token activation patching reveals that the state sufficient to induce a collision emerges sharply at a specific token and depth.

8 What determines the severity of a collision?

Why do some partners overturn the target, while others leave it intact? In the joint prompt, statement A can write information into the representation of B through attention, knocking B's computation

off the trajectory it would follow in isolation. A natural hypothesis is therefore that the farther the computation is displaced from its solo trajectory, the more severe the resulting collision.

8.1 Residual drift.

Let h_ℓ^{solo} and h_ℓ^{joint} denote the readout states at layer ℓ in the solo and joint computations, respectively.

Definition (Residual drift). The *residual drift* at layer ℓ is the relative displacement between the joint and solo readout states:

$$D_\ell = \frac{\left\| h_\ell^{\text{joint}} - h_\ell^{\text{solo}} \right\|_2}{\left\| h_\ell^{\text{solo}} \right\|_2}.$$

An illustration. Figure 6 illustrates residual drift for the same target under two different partners: one that produces a destructive neural collision and one that leaves the target intact.

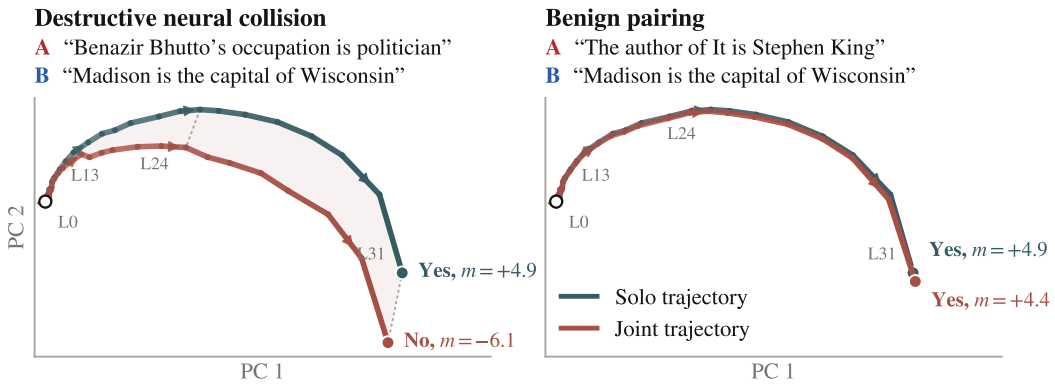

Figure 6: **Destructive partners drive the target off its solo trajectory.** Each panel compares the solo and joint readout trajectories, projected onto a shared PCA basis (Appendix C.1). *Left:* The trajectories separate from the middle layers onward and end with opposite answers. *Right:* The joint trajectory closely tracks the solo trajectory and the answer is unchanged. The projection is for illustration only; residual drift is computed in the full hidden space.

This single example is only illustrative, but it suggests a general hypothesis:

Residual Drift Hypothesis. The farther the target's internal state drifts from its solo trajectory, the larger the resulting behavioural effect, measured by the absolute margin shift $|\Delta m_B|$.

8.2 Experimental setup

We test whether pairs with greater residual drift also show larger margin shifts. For a sample of 100 destructive collisions, we record the readout state at every layer and compute D_ℓ. We then measure the Spearman correlation across pairs between D_ℓ and $|\Delta m_B|$ as a function of depth.

8.3 Results

We find that collisions with larger margin shifts are correlated with greater residual drift. Across all three models, this relationship is initially weak but rises sharply at intermediate-to-late depth (Figure 7). A permutation test confirms that the correlation peaks are statistically significant (Appendix C.2).

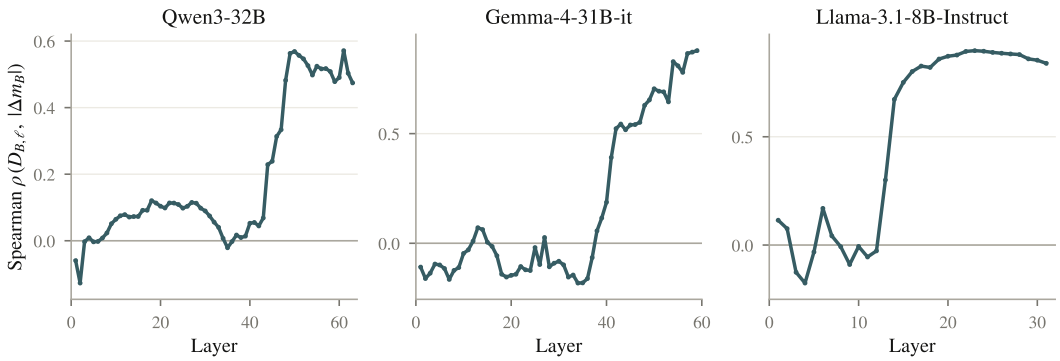

Figure 7: **Residual drift predicts collision severity in later layers.** Spearman correlation between residual drift D_ℓ and absolute margin shift $|\Delta m_B|$ across collisions at each layer.

Why does residual drift become predictive only at intermediate-to-late depth? The appendix shows that its onset closely tracks the point at which collision damage reaches the readout, where residual drift is measured (Appendix C.3).

9 Conclusion

We introduced *neural collisions*: cases in which computations that succeed independently fail when carried out together. By reconstructing collisions inside the network, we find that the failure does not arise only at the final answer. The partner abruptly corrupts the target's representation over just a few layers; that corrupted state is then propagated to the readout and shapes the final answer. How far the partner knocks the target off its solo trajectory, predicts how severe the collision will be.

Implications and future work. These findings raise questions for model reliability and safety. A model that makes a correct prediction in isolation can fail when required to make that same prediction alongside other computations. This raises a central question for future work: can collisions be predicted before they happen? Residual drift predicts the magnitude of a partner's effect, but not its direction, and becomes informative only once the damage has reached the readout. An important next step is therefore to ask whether properties of the target and partner representations earlier in the network can predict which interactions will become destructive.

AI use statement

We used generative AI tools to assist with writing the manuscript, searching for and checking related work, and developing experimental code. We did not use generative AI tools to generate the research ideas, design the experiments, or analyse and interpret the results. All AI-assisted code was reviewed and tested by the authors, and all AI-assisted text was reviewed by the authors. The authors take full responsibility for the content and conclusions of this work.

References

Ziling Cheng, Meng Cao, Marc-Antoine Rondeau, and Jackie CK Cheung. Stochastic chameleons: Irrelevant context hallucinations reveal class-based (mis) generalization in llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 30187–30214, 2025.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, et al. Toy models of superposition. *arXiv preprint arXiv:2209.10652*, 2022.

Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context?, 2024. *URL https://arxiv. org/abs/2310.17191*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Zhuoran Jin, Pengfei Cao, Hongbang Yuan, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. Cutting off the head ends the conflict: A mechanism for interpreting and mitigating knowledge conflicts in language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1193–1215, 2024.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 9802–9822, 2023.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in gpt. *Advances in neural information processing systems*, 35:17359–17372, 2022.

Jingcheng Niu, Xingdi Yuan, Tong Wang, Hamidreza Saghir, and Amir H Abdi. Llama see, llama do: A mechanistic perspective on contextual entrainment and distraction in llms. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16218–16239, 2025.

Francesco Ortu, Zhijing Jin, Diego Doimo, Mrinmaya Sachan, Alberto Cazzaniga, and Bernhard Schölkopf. Competition of mechanisms: Tracing how language models handle facts and counterfactuals. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8420–8436, 2024.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. Large language models can be easily distracted by irrelevant context. In *International conference on machine learning*, pp. 31210–31227. PMLR, 2023.

Gemma Team. Gemma 4 technical report, 2026. URL `https://arxiv.org/abs/2607.02770`.

Qwen Team. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

Zheyang Xiong, Ziyang Cai, John Cooper, Albert Ge, Vasilis Papageorgiou, Zack Sifakis, Angeliki Giannou, Ziqian Lin, Liu Yang, Saurabh Agarwal, et al. Everything everywhere all at once: Llms can in-context learn multiple tasks in superposition. *arXiv preprint arXiv:2410.05603*, 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.

A Appendix

B Additional analysis of collision localization

The main text measures the abruptness of collision onset using $W_{20\text{–}80}$, the number of layers over which the collision token rises from 20% to 80% of its peak causal effect. Table 2 shows that the sharp onset is not an artifact of the 20–80% definition. Wider rise intervals naturally produce larger onset widths, while narrower intervals produce smaller ones, but in all cases the collision still emerges over relatively few layers.

C Additional residual-drift analyses

This section provides additional analyses supporting the residual-drift results in the main text.

Table 2: Robustness of damage onset width to the choice of rise thresholds. Values are median numbers of layers, with interquartile ranges in brackets.

Model	Layers	Median $W_{10\text{–}90}$ [IQR]	Median $W_{20\text{–}80}$ [IQR]	Median $W_{30\text{–}70}$ [IQR]
Llama-3.1-8B-Instruct	32	4.0 [3.0, 5.8]	3.0 [1.0, 4.0]	1.0 [0.0, 3.0]
Gemma-4-31B-it	60	3.0 [3.0, 7.5]	2.0 [1.0, 3.8]	1.0 [1.0, 2.0]
Qwen3-32B	64	7.5 [4.2, 15.2]	4.5 [3.0, 7.0]	2.0 [1.0, 3.0]

C.1 Projection used in Figure 6

For the target B, let

$$\mathcal{T}^{\text{solo}} = \left(h_0^{\text{solo}}, \ldots, h_L^{\text{solo}}\right),$$

$$\mathcal{T}^{\text{destructive}} = \left(h_0^{\text{destructive}}, \ldots, h_L^{\text{destructive}}\right),$$

and

$$\mathcal{T}^{\text{benign}} = \left(h_0^{\text{benign}}, \ldots, h_L^{\text{benign}}\right)$$

denote the layerwise readout trajectories for the solo run, the joint run with the destructive partner, and the joint run with the benign partner, respectively. Each state $h_\ell^r \in \mathbb{R}^d$.

We pool all $3(L+1)$ readout states into a single matrix

$$H = \begin{bmatrix} h_0^{\text{solo}} \\ \vdots \\ h_L^{\text{solo}} \\ h_0^{\text{destructive}} \\ \vdots \\ h_L^{\text{destructive}} \\ h_0^{\text{benign}} \\ \vdots \\ h_L^{\text{benign}} \end{bmatrix} \in \mathbb{R}^{3(L+1)\times d}.$$

We center the rows of H by their mean and fit a single PCA to this pooled set. Let $v_1, v_2 \in \mathbb{R}^d$ denote the first two principal directions. Each readout state is then plotted at

$$z_\ell^r = \left(v_1^\top (h_\ell^r - \mu),\ v_2^\top (h_\ell^r - \mu)\right),$$

where $r \in \{\text{solo}, \text{destructive}, \text{benign}\}$.

Thus all three trajectories are projected into the same two-dimensional coordinate system. In particular, the solo trajectory is identical in the two panels of Figure 6, allowing the destructive and benign joint trajectories to be compared directly. The first two principal components explain 70.7% and 8.5% of the variance in the pooled states, respectively (79.2% in total).

C.2 Permutation test for residual-drift correlations

We test whether the layerwise correlations could arise by chance with a permutation test. For each model, we randomly shuffle $|\Delta m_B|$ across pairs, recompute the correlation at every layer, and record the largest absolute correlation across the network. Repeating this procedure $10{,}000$ times produces a null distribution for the strongest correlation that could arise by chance when searching across all layers.We use the 95th percentile of this distribution as the significance threshold.

Table 3 shows that the observed peak correlation comfortably exceeds this threshold in all three models.

Table 3: Permutation test with 10,000 shuffles. The 95% threshold is the 95th percentile of the largest absolute Spearman correlation observed anywhere in the network under 10,000 random permutations of $|\Delta m_B|$. "First significant layer" is the first layer at which the observed correlation exceeds this threshold. "Peak ρ" gives the largest observed correlation and the layer at which it occurs. p is the probability, under the permutation null, of obtaining a network-wide maximum at least as large as the observed peak.

Model	95% threshold	First significant layer	Peak Spearman ρ	Permutation p-value
Llama-3.1-8B-Instruct	0.282	13 / 32	0.898 (23)	$< 10^{-3}$
Gemma-4-31B-it	0.290	41 / 60	0.878 (59)	$< 10^{-3}$
Qwen3-32B	0.286	46 / 64	0.571 (61)	$< 10^{-3}$

C.3 Why residual drift becomes predictive late

In the main text, we found that residual drift at the readout predicts collision severity, but only in the later layers of the network. Residual drift is measured at the readout. The earlier patching analysis showed that collision damage reaches the readout only at intermediate-to-late depth. Before that transition, displacement of the readout need not reflect the eventual severity of the collision. Consistent with this account, the rise in the layerwise correlation between residual drift and margin shift closely tracks the rise in readout-patch collision transfer across all three models (Figure 8).

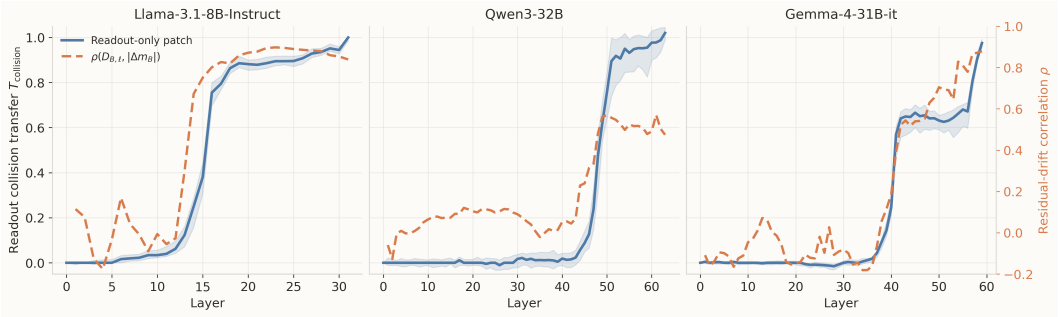

Figure 8: Solid lines show readout-patch collision transfer across layers. Dashed lines show the layerwise Spearman correlation between readout residual drift and absolute margin shift.