FALSE WITNESSES: WHEN IRRELEVANT FACTS CHANGE TRANSFORMER ANSWERS

Anonymous authors
Paper under double-blind review

ABSTRACT

We identify and study a failure mode in which a true relation about the wrong entity (*irrelevant fact*) steers a transformer toward a wrong answer. Specifically, we study a controlled two-step reasoning task that composes independent random functions: the first maps the queried entity to an intermediate value, and the second maps this value to the final answer. This defines *relevance* exactly, avoiding the interpretive ambiguity of natural-language contexts. An optional suggestion in the input prompt proposes the intermediate value. We call a *true* first-step relation about *another* entity that matches an *incorrect suggestion* a **false witness**. Our **equal-function test** changes whether this relation is present while preserving every entity's correct final answer, isolating its effect on suggestion following. Across two-layer transformers trained from different seeds, adding a false witness doubles predictions following the incorrect suggestion, from 27.4% to 54.8%. Causal interventions via activation patching identify the underlying attention mechanism: query–key matching (Q/K) favors the irrelevant entity and its value vector (V) supplies the answer. Replacing Q alone can recover the queried entity's answer. Training dynamics reveal a gap between solving the task and resisting misleading suggestions: models first become accurate only without suggestions, while resistance to incorrect suggestions develops later. This gap closes when each entity can look up its own final answer in the first attention layer; accuracy under incorrect suggestions then rises from 38.3% to 98.9%, nearly matching accuracy obtained without suggestions. In sum, we establish *false witnesses* as a failure of *relevance*, identify the attention mechanism responsible, and show how training and information access shape resistance to false witnesses.

1 INTRODUCTION

Reasoning from context requires more than finding a true fact: that fact must help answer the question being asked. A relation about one entity can resemble the relation needed for another. If a model uses this resemblance to select its evidence, it may retrieve accurate information and still answer the wrong question. Work on entity binding and retrieval explains how transformers associate entities with their attributes (Feng & Steinhardt, 2024; Variengien & Winsor, 2025; Gur-Arieh et al., 2026; Hoenig et al., 2026). We ask *when a true relation about an irrelevant entity can redirect that retrieval*, connecting these mechanisms to reasoning shortcuts and sensitivity to irrelevant context (Geirhos et al., 2020; Shi et al., 2023).

We study two-step reasoning through two independently sampled random functions, f and g. Given a query x, the model must first find $f(x)$ and then return $g(f(x))$ (Guo et al., 2025). The functions are supplied in each problem, so the answer must be computed from the current input. Randomizing them prevents any particular fact from remaining useful across training examples. We use this controlled setting to specify exactly which relations determine an answer and which transformations preserve it. This lets us vary suggestion matching while keeping every entity's answer fixed, then use donor inputs with known answers to test record selection and answer retrieval through activation patching.

An optional suggestion h proposes the intermediate value $f(x)$. Following it means returning $g(h)$. When h is wrong, another entity $j \neq x$ may nevertheless satisfy $f(j) = h$. The model could treat this match as support for the suggestion, even though it does not establish $f(x) = h$. We call this relation a **false witness**: it is true in the input, but concerns the wrong entity. Does the presence of such a relation make the model more likely to follow the suggestion?

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.

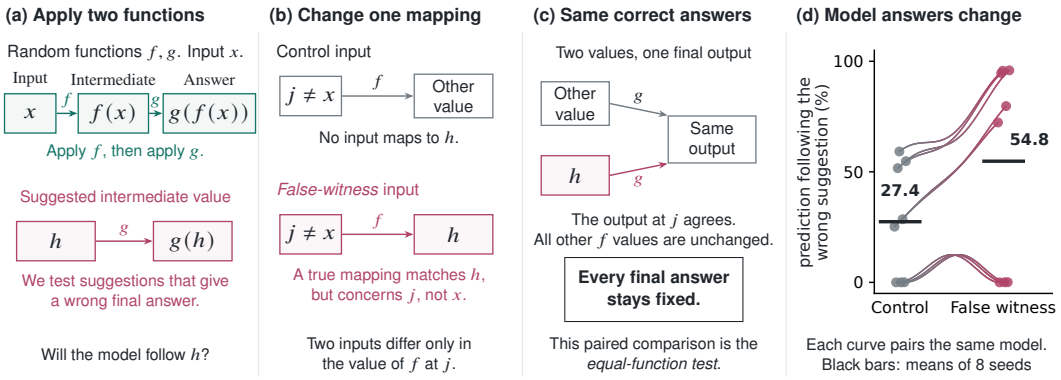


Figure 1: **An irrelevant match increases suggestion following while preserving every two-step answer.** (a) The query asks for $g(f(x))$; following suggestion h gives a wrong answer, $g(h)$. (b,c) Changing only unqueried record j introduces a match to h while preserving every entity's answer. (d) Mean suggestion following rises from 27.4% to 54.8%. Curves connect the two input conditions for each model; black bars show means. Eight two-layer models, g first and suggestion last, at 2,000 updates; 4,096 input pairs per model.

Our **equal-function test** isolates the effect of this match by comparing two inputs that differ only in a relation about an unqueried entity. The edit introduces a false witness while preserving the correct two-step answer for every entity, not just the one being queried. This is possible because different intermediate values can lead to the same final answer under g. We choose suggestions whose implied final answer is wrong. Across trained models from eight seeds, adding the false witness doubles predictions following that wrong answer, from 27.4% to 54.8% (Figure 1). These same models achieve 99.8% mean accuracy in separate tests without suggestions (Table 1), showing that accurate lookup in that setting can coexist with substantial sensitivity to an irrelevant match.

We then trace how the edited relation changes the answer. We use activation patching to test which record the model selects and which answer it retrieves from that record. Further interventions change which entity's information is used. Following the same models through training reveals a shift toward using the queried entity, while separate experiments identify input layouts in which the false-witness effect persists. Finally, we test whether allowing the model to combine the two functions in its first attention layer improves accuracy under incorrect suggestions.

Our contributions are:

- ***False witness* effect and *equal-function test* diagnostic.** We identify a failure mode in which a true relation about another entity makes an incorrect suggestion more influential. Our paired test changes this match while preserving the entire two-step answer function, isolating sensitivity that ordinary accuracy measurements can miss. It provides a controlled way to test whether models distinguish supporting a suggestion from answering the query.
- **A causal account of irrelevant retrieval.** Separate interventions on record matching and answer information show how an unqueried record gains influence and supplies its output. Patching the attention query can recover the correct answer or redirect the model to another entity's answer.
- **Learning dynamics and information-access interventions.** We show that accurate lookup without suggestions emerges before resistance to incorrect suggestions, with substantial variation across models and input layouts. In matched training experiments, allowing the model to combine the two functions in its first attention layer raises accuracy under incorrect suggestions from 38.3% to 98.9%. A separate matched control shows that the early benefit depends on information from the current problem.

Appendix A compares these contributions with prior work on how models retrieve and combine facts, how models respond to false claims and irrelevant context, and how internal interventions reveal the computations behind these behaviors. Figure 7 summarizes these connections.

2 Experimental Setup

The functions act on a finite set of symbols: $f, g : [N] \to [N]$, where $[N] = \{0, \ldots, N-1\}$ and we set $N = 8$. Each function assigns an output symbol to each input symbol. We sample f and g

independently and uniformly from all such functions, allowing repeated outputs. For query x, write $y = g(f(x))$ for the correct answer and $y_h = g(h)$ for the answer obtained by following suggestion h. A wrong intermediate suggestion, $h \neq f(x)$, can still give the correct final answer because $g(h)$ may equal $g(f(x))$. Our equal-function comparisons require $y_h \neq y$ so that the two answers can be distinguished.

Input representation and training. A finite function can be encoded as a lookup table. We supply its eight values in index order: a token's position identifies the input symbol, and the token gives its output. We call $i \mapsto f(i)$ the first-step *record* for entity i, and $u \mapsto g(u)$ the second-step record for intermediate value u. Each record occupies one position in its table. The remaining input contains the query, a task token, and an optional suggestion. We write t_1 and t_2 for the one-step and two-step task tokens, requesting $f(x)$ and $g(f(x))$, respectively. For example, a two-step question with f first and the suggestion last is encoded as $[f(0), \ldots, f(7),\ g(0), \ldots, g(7),\ x,\ t_2,\ h]$. Each entry is a single token, giving 19 input tokens (see Appendix C.1 for the full encoding). The training target is $y = g(f(x))$. Fresh functions are sampled for every training example. One-step questions make up 25% of training and two-step questions 75%. Suggestions occur in half of the examples and are correct 75% of the time. The training target is always the true answer.

Our two-layer transformers predict the answer using the final hidden state of the last input token. In the example above, this token is the suggestion h. By default, attention is *causal*: each position can read only itself and earlier positions. Table order therefore matters. When g comes first, a position in the f table can read the g table in the first layer; when f comes first, it cannot. We also vary whether inputs with a suggestion end in query x or suggestion h. Without a suggestion, both layouts end in [absence token, task token, x]. Removing a final suggestion therefore also changes the last token and the query position. We address this difference with fixed-position controls in Section 5.

The main experiment comprises eight training initializations for each table order and final-token layout, totaling 32 models. All training seeds are retained in the reported results, regardless of performance. We measure accuracy at nine training checkpoints and conduct internal interventions at 2,000 and 4,000 updates. Unless stated otherwise, the results below use models trained with the suggestion last when present. Architecture and training details are in Appendix C.

Equal-function test. We now construct the equal-function pair. Write f for the control function and $\bar{f}$ for its edited counterpart. Choose an unqueried entity $j \neq x$ and an intermediate value $r \neq h$ such that $g(r) = g(h)$. Set $\bar{f}(j) = r$ and $\bar{f}(j) = h$, keeping all other records identical. No other first-step record maps to h or r. The edited input contains a false witness, while the control has no first-step record matching the suggestion. Since the two intermediate values have the same second-step output,

$$g(f(i)) = g(\bar{f}(i)) \quad \text{for every } i \in [N]. \tag{1}$$

This equality preserves every two-step answer, although the edited first-step relation itself changes. The query x, suggestion h, and function g stay fixed, and we require $y_h \neq y$. Every model in a group receives the same paired inputs.

We measure *accuracy* as the fraction of predictions equal to y, and *suggestion following* as the fraction equal to y_h. The false-witness effect is the increase in suggestion following from the control to the edited input, measured in percentage points. Suggestion following is not the total error rate: a model can produce other wrong answers, and in naturally sampled tests a wrong intermediate suggestion can still lead to y.

Diagnostic interpretation. The control fixes more than the answer to one question: it preserves the answer to every two-step query. The paired inputs have the same composed function $g \circ f$ but differ in whether an irrelevant first-step record matches the suggestion. The sequence length, query, suggestion, and all but one function entry are identical. A change in prediction cannot be explained by a change in any correct two-step answer. When that change systematically favors y_h, the test identifies reliance on a query-irrelevant match. This is stronger than measuring a model's accuracy in either input alone, where identical aggregate accuracy could conceal which relations influence its predictions.

An observer uncertain about the queried entity can find this match predictive: the edit makes suggestion h more likely for entity j, whose answer is y_h in both inputs. Appendix G.1 illustrates this dependence on query information through a Bayesian observer model. The internal interventions in Section 4 test how the influence is implemented in the trained models.

Table 1: **Answer-preserving edits expose sensitivity to irrelevant matches.** Eight models per table order, 2,000 updates, suggestion last when present. Effects are increases in suggestion following. The equal-function test uses 4,096 pairs, natural accuracy tests use 4,096 examples per condition, and the same-row comparison uses 3,600 examples with distinct true and suggested answers.

	f table first	g table first
Equal-function effect (percentage points)	37.9	27.4
Models with a positive equal-function effect	8/8	7/8
Accuracy without a suggestion (%)	99.9	99.8
Accuracy with a wrong suggestion (%)	19.6	55.5
False witness vs. same-row control (percentage points)	51.9	41.1

3 Behavioral Results

The equal-function edit has a large effect in both table orders (Table 1). With g first, suggestion following rises from 27.4% to 54.8% on average, and seven of eight models have a positive effect. With f first, the mean increase is 37.9 percentage points and all eight effects are positive. A model whose predictions depended only on the composed function, query, and suggestion would give the same predictions for both inputs. These models remain sensitive to a local relation even when all three are held fixed.

The models can also perform accurate lookup. In the query-last layout, all 16 models answer every test question correctly at 2,000 updates. The suggestion-last models are nearly perfect when tested without a suggestion, but become less accurate when given an incorrect one (Table 1).

Specificity controls. To test whether the effect also occurs without the equal-function constraint, we use a same-row control that preserves the queried answer. Relative to this control, adding a false witness increases suggestion following by 51.9 and 41.1 percentage points for f-first and g-first models, respectively. Repeating the suggestion symbol in an unused second-step record has a much smaller effect. On the same 3,385 eligible examples for f-first models, the repetition and false-witness effects are 1.25 and 51.88 points, respectively. To test whether the match requires identical tokens, three additional models express suggestions with different tokens from those used for the same values in the functions. They still show equal-function effects of 32.8–45.8 points. These comparisons implicate the first-step relation matching h, rather than token repetition alone (Appendix D). We next examine how that relation changes the information used for the answer.

4 Mechanistic Analysis

We focus on the eight g-first, suggestion-last models at 2,000 updates. Their input order permits a simple route: in the first layer, record i in the f table can read the g table and acquire information about $g(f(i))$. In the second layer, the final position can read a first-step record and use its composed answer. On this route, successful reasoning requires both computing the record's answer and choosing a record relevant to x.

A database analogy. Interpret the finite functions as two tables: f maps people to employers, and g maps companies to countries. Combining the tables gives each person's country, an operation known in databases as a *join* (Codd, 1970; Selinger et al., 1979). The remaining question is whose country to return. Suppose Alice works at Harbor in Japan, while the suggestion names Cedar in France. A record saying that Bob works at Cedar matches the suggested company but does not establish Alice's employer. Table 2 connects the symbols to this example.

The intended selection is "person = Alice." Selecting "company = Cedar" instead can return Bob's country. In the equal-function pair, Bob's employer is Maple in one input and Cedar in the other, with both companies mapped to France by the same g. Alice's answer remains Japan and Bob's remains France, and every other person's country also agrees across the pair. What changes is whether the suggestion finds a matching employer at Bob's record. The analogy separates combining the relations from selecting whose answer to read. Appendix B develops this distinction through a worked database example. Table 3 summarizes the correspondence between database concepts, the equal-function test, and the activation-patching experiments.

Activation patching. Attention gives us separate components to intervene on. At a position that reads information, a *query vector* (Q) is compared with the *key vectors* (K) of available positions

Table 2: **Symbols and the person–company–country analogy.** Alice is the queried person. In the false-witness input, Bob works at Cedar; in its paired control, he works at Maple. Cedar and Maple both map to France. These are two hypothetical inputs with identical person–country answers.

Symbol	Computational role	Example
x	Queried entity	Alice
$j \neq x$	Another entity	Bob
f	First-step function	Person $\rightarrow$ employer
g	Second-step function	Company $\rightarrow$ country
$f(x)$	Queried intermediate value	Harbor
h	Suggested intermediate value	Cedar
r	Control value at j	Maple
$y = g(f(x))$	Correct output	Japan
$y_h = g(h) = g(r)$	Suggestion-implied output	France
$\bar{f}(j) = h$	False-witness relation	Bob works at Cedar

to determine their weights. The corresponding *value vectors* (V) carry the information combined using those weights (Elhage et al., 2021). The learned attention query is distinct from the task's query symbol x, and attention keys and values are distinct from the table's indices and entries. We use *activation patching*: copy a chosen internal vector from one input's run into another run of the same model, then run the rest of the model with that vector replaced (Geiger et al., 2021; Zhang & Nanda, 2024; Heimersheim & Nanda, 2024). The source is the donor and the modified run is the recipient. Unpatched vectors retain their recipient values. This lets us test what makes a record influential separately from what answer it supplies. Appendix E gives the donor constructions and full patching results across table orders and training checkpoints.

An initial experiment in three separately trained models tests whether first-step records carry composed answers. We change g in a donor input and copy first-layer record states into the original run. In the two models that already answer accurately, copying the queried record transfers its answer under the donor function, whereas copying the other first-step records retains the original answer. The third model follows the wrong suggestion. Copying records that match it transfers the donor answer associated with that suggestion. These transfers support distinct roles for answer availability and record selection. The following experiments return to the main eight-model group and intervene on second-layer attention vectors (Appendix E).

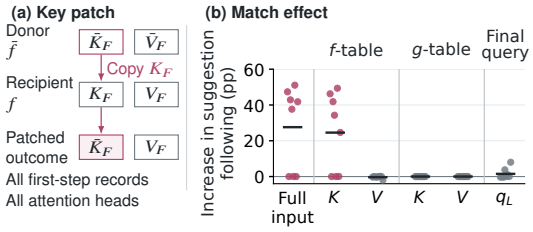


Figure 2: **Patching first-step keys increases suggestion following in g-first models.** (a) Copy second-layer keys at all first-step records (F) from the false-witness run into the control, keeping values fixed. (b) Increase in suggestion following under the full input edit or the indicated vector patches, applied across all heads. q_L is the final-position attention query; pp means percentage points. Black marks show means. Eight g-first, suggestion-last models at 2,000 updates; 2,048 pairs per model.

Key interventions. For an equal-function pair, take the control input as recipient and copy only the second-layer keys at all first-step records from the false-witness input. Patching these keys produces most of the increase in suggestion following (Figure 2). Replacing their values has a much smaller effect. The matching relation therefore changes the weight given to records whose composed outputs were already available. In the database analogy, the change affects whose record contributes to the answer.

Crossed key and value interventions. We next ask whether a matching key can make an irrelevant record supply a separately chosen output. This test patches only record j. Its key comes from the false-witness input, while its value comes from a different donor designed to carry a third distinguishable answer. The recipient retains $f(j) = r$ and its original g table.

Specifically, the key donor is $(\bar{f}, g)$, where $\bar{f}(j) = h$. The value donor is $(f, \bar{g})$, where $\bar{g}(r) = \bar{g}(h) = \bar{y}_h$. We choose $\bar{y}_h \notin \{y, y_h\}$ and preserve the queried answer by requiring $\bar{g}(f(x)) = y$. Here the overbar identifies the specified alternative function or output. We copy only the second-layer key and value at j, with the donors providing these two components separately. An increase in predictions of $\bar{y}_h$ indicates that the model uses the patched value at record j.

Figure 3b compares baseline, single, and combined patches. The combined patch elicits $\bar{y}_h$ in 31.35% of predictions. Its increase exceeds the sum of the individual increases by 25.95 percentage points.

This output interaction could arise within attention or in subsequent nonlinear computation. To distinguish these contributions, we replace the combined patch's second-layer attention output at the final position with the unpatched output plus the two single-patch changes. This additional intervention preserves the separate K and V effects at that output and reduces predictions of $\bar{y}_h$ from 31.35% to 3.28%. The decrease identifies a contribution from the interaction between the attention weight and the changed value; Proposition G.2 gives the exact decomposition.

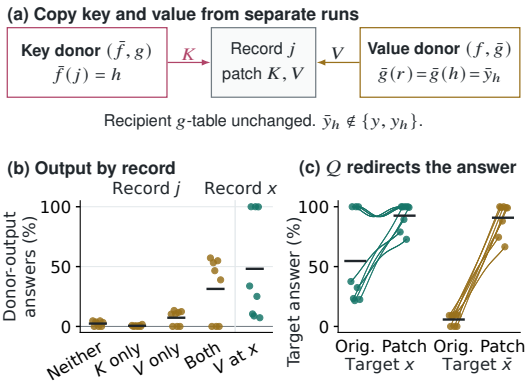


Figure 3: **Matching and answer retrieval can be varied separately.** (a) Separate donors supply a key matching h and a value carrying a third answer $\bar{y}_h$; recipient g stays fixed. (b) Predictions of $\bar{y}_h$ under key and value patches at unqueried record j, or the same value alone at queried record x. (c) Replacing only the final-position query vector with one from a run without a suggestion redirects predictions toward the answer for x or another entity $\bar{x}$. Baselines show how often the model predicts each target before patching. Curves in (c) connect each model's two measurements; black marks show means. Eight g-first, suggestion-last models at 2,000 updates; 2,048 pairs in (b) and 4,046 examples in (c) per model.

A fifth condition in Figure 3b patches the same donor value at the queried record x, without replacing the key at j, and measures how often the model predicts $\bar{y}_h$. We compare the patches at j and x again later in training.

Query interventions. We now hold every table vector fixed and replace only the final position's attention query. A donor run with the same task query x and no suggestion raises accuracy from 54.7% to 92.5%. A donor with another task query $\bar{x}$ and no suggestion makes 90.8% of predictions equal to that entity's answer, $g(f(\bar{x}))$ (Figure 3c). These results show that replacing the final query alone can redirect which entity's answer the model produces. Models with other table orders can also use second-step positions to supply answers (Appendix E).

5 Learning Dynamics

We first follow the same suggestion-last models through training. Accuracy is measured at nine checkpoints, while the equal-function and internal interventions compare 2,000 with 4,000 updates (Figure 4a–d). These measurements let us distinguish learning to compute an answer from learning to resist a misleading suggestion.

Accuracy trajectories. The trajectories reveal an early improvement in lookup and a later improvement in resistance to incorrect suggestions (Figure 4a,b). By 1,000 updates, mean accuracy without suggestions is already 98.7% with f first and 99.9% with g first, while accuracy under incorrect suggestions is only 16.1% and 18.9%, respectively. The models can therefore solve the composition task in the no-suggestion layout well before they reliably answer when the final token supplies a misleading intermediate value. Because removing the suggestion also changes the main input layout, this comparison motivates the fixed-position experiments below.

The later improvement differs substantially across models. With g first, three models become highly accurate by 2,000 updates and four more exceed 99% at 2,500; the remaining model still has only 24.0% accuracy at 4,000. With f first, all eight models remain below 23% at 2,000 updates, and only three exceed 92% by 4,000. Thus, the average curves combine improvements occurring at different times with trajectories that remain poor throughout the observation period.

Changes in record selection. A small false-witness effect alone cannot establish correct selection, because the effect can also vanish if a model follows every suggestion regardless of the records. We therefore examine the controlled input comparison together with task performance and the record interventions.

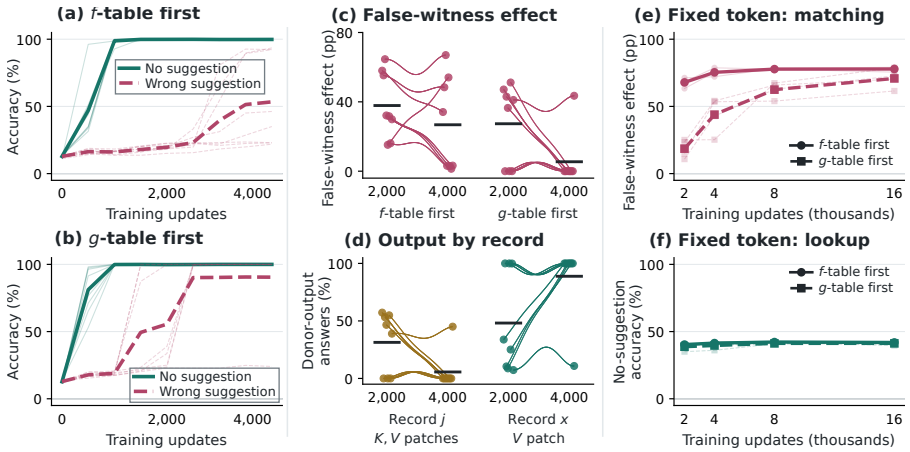


Figure 4: **Training changes record selection, with different outcomes across layouts.** (a,b) Accuracy without suggestions (solid) and with wrong suggestions (dashed), for eight main suggestion-last models per table order. (c,d) The same models at 2,000 and 4,000 updates: (c) false-witness effects; (d) predictions of the donor answer after patching key and value at unqueried record j, or the same value alone at queried record x (g first only; Figure 3). (e,f) Four additional models per order use a constant final answer token and 75%-reliable suggestions. Effects remain large through 16,000 updates while no-suggestion accuracy stays near 41%. Thin and thick lines in (a,b,e,f) show individual models and means. Curves in (c,d) join each model's two checkpoints; black bars show means.

At 4,000 updates, seven of eight g-first models have an equal-function effect within 0.1 percentage points of zero; one retains a 43.5-point effect. Among f-first models, four effects remain between 34 and 67 points, while four fall to 1.4–3.2 points. In the g-first group, replacing key and value at the unqueried record j elicits the third donor output in 31.4% of predictions at 2,000 updates and 5.6% at 4,000. Placing the same value at the queried record x instead increases donor-output predictions from 48.2% to 88.8%. At 4,000 updates, the same seven g-first models that become accurate give no donor-output responses when K and V are patched at j; the remaining model retains both a large false-witness effect and a strong response to that patch. Together, these two measured states show increased use of the queried record as performance improves.

Suggestion reliability and fixed input positions. Next, we vary suggestion reliability in new models with fixed query positions to test when they follow suggestions or check the records. One layout replaces h with an absence token in the same final slot; another appends a constant answer token after that slot. The latter always produces its answer from the same token. Figure 5a shows both layouts. For each layout and table order, we train four models at each of three suggestion reliabilities: chance level (12.5%), usually correct (75%), and always correct (100%). This gives 48 models. The constant-answer-token subset trained at 75% reliability is continued to 16,000 updates (Figure 4e,f).

Figure 5b–e shows how these fixed-position models use incorrect test suggestions. With chance-reliable suggestions, adding a matching record has little effect. With always-correct training suggestions and the suggestion/absence slot last, all g-first models follow every incorrect test suggestion at 4,000 updates, with or without a false witness. This gives a direct example of a negligible false-witness effect despite failure to select the queried record.

At 75% reliability, the g-first models with the suggestion/absence slot last have a mean false-witness effect of 10.2 points at 2,000 updates. With further training, those models become accurate and the effect disappears at the group level. The constant-answer-token layout instead retains a large effect. Partly reliable suggestions give models a reason to check the records, but checking whether a suggestion has a match is not sufficient to identify the queried entity's answer.

Extended training. We test whether the effects diminish with further training by continuing the eight constant-answer-token models trained at 75% reliability. At 16,000 updates, mean equal-function effects are still 77.8 and 70.8 percentage points for the f-first and g-first orders, respectively (Figure 4e). Their no-suggestion accuracies remain near 41% (Figure 4f). Thus, persistent sensitivity here accompanies a partly learned two-step lookup.

The results suggest that checking for any first-step match can be an accessible early strategy, while learning to use the queried record depends on both the input layout and the information available to it.

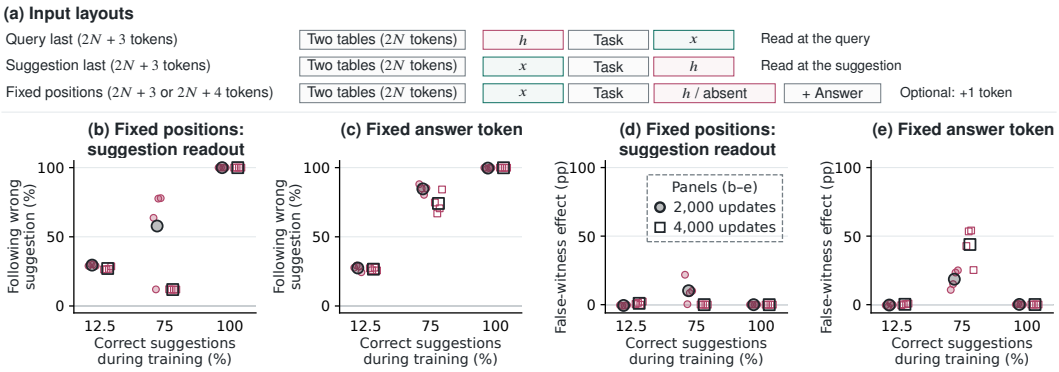


Figure 5: **Suggestion reliability changes suggestion following and sensitivity to false witnesses.** (a) Fixed-position layouts keep the query in the same slot when suggestions are removed. They predict from either the suggestion/absence slot or an appended constant answer token. (b,c) Suggestion following on natural tests with wrong intermediate suggestions. (d,e) Its increase in equal-function pairs. Small colored marks show four g-first models per layout and reliability; black marks show means. Filled circles and open squares denote 2,000 and 4,000 updates. Per model: 4,096 natural examples and 2,048 equal-function pairs.

The patching results link improved accuracy to greater reliance on the queried record. We next test the role of information access directly, using newly trained, matched model groups.

6 Information-Access Interventions

The route identified in Section 4 requires second-step information to reach first-step records before the answer position reads them. We therefore change first-layer access to the g table during training (Biran et al., 2024). These are fresh suggestion-last groups, separate from the fixed-answer-token continuation above. The intervention tests whether making that route available helps models learn to answer accurately despite wrong suggestions.

Opening and closing access. With f first, each of its eight record positions is normally unable to read the later g table. We open all 64 connections from these positions to the eight second-step positions in layer one. With g first, those connections are normally available; we close them (Figure 6a). Each comparison uses eight matched model pairs with identical initializations, training examples, parameter counts, and token positions within a pair.

Opening access raises accuracy under wrong suggestions from 38.3% to 98.9% after 4,000 updates and reduces the equal-function effect from 24.73 to -0.05 percentage points. Closing access lowers accuracy from 99.95% to 37.99% and increases the effect from -0.01 to 15.13 points (Figure 6b,c). All eight paired accuracy comparisons move in the predicted direction in each experiment.

Independent-information control. To isolate useful information from added connections, eight further matched pairs keep all 64 connections open. One condition sends first-step positions information from the current g table; the other sends information computed from an independently sampled table. The actual input and training answers remain the same, and layer two uses the actual input in both conditions. After 2,000 updates, current-table information gives 99.8% accuracy under wrong suggestions, compared with 21.0% for independent information. Both groups exceed 99.88% accuracy without suggestions. Their equal-function effects are 0.04 and 46.32 points, respectively (Figure 6d,e). Open connections help when they carry information needed for the current problem.

To test how the improved models answer, we change g in a donor input and copy only the second-layer value at the queried first-step record. For the opened-access and informative-message groups at 2,000 and 4,000 updates, the mean donor-answer following rate across eight models ranges from 98.5% to 99.8%. Evaluation uses 780 examples in which the true and suggestion-implied answers from the original and donor inputs are all distinct. Patching the value at the queried f-table position transfers the donor answer. Some models trained with independent messages improve later through second-step positions (Appendix E). First-layer access shapes which route is learned and how quickly it becomes useful.

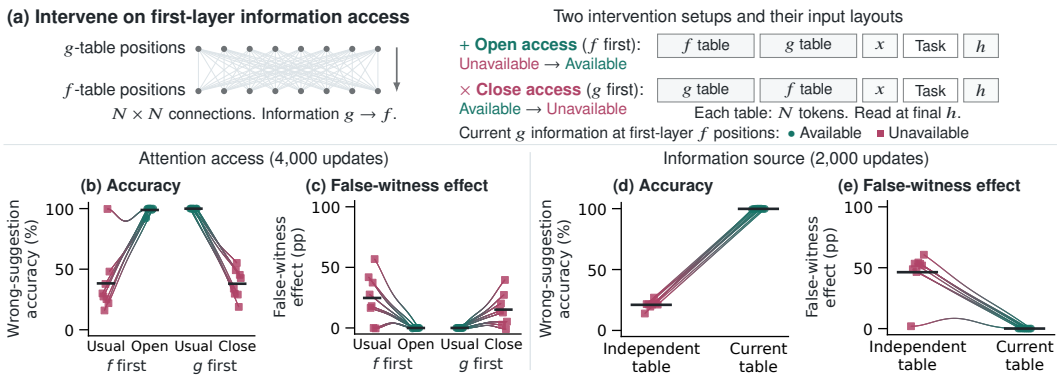


Figure 6: **First-layer access to the current g table improves mean accuracy under wrong suggestions.** (a) Open or close connections allowing f-table positions to read g-table positions. (b,c) Accuracy under wrong suggestions and false-witness effects after 4,000 updates. (d,e) The same measures after 2,000 updates with connections open in both conditions, carrying current or independently sampled g-table information. Teal circles mark access to the current table through these connections; berry squares mark closed connections or independent information. Curves connect eight matched model pairs per comparison; black bars show means. Per model: 4,096 natural examples for accuracy and 4,096 equal-function pairs for effects.

7 Discussion

A false witness can lead a model to retrieve a true fact about another entity and use it to answer the wrong question. Our experiments connect this failure to record selection and show how training and information access affect it. Our equal-function test provides a precise behavioral diagnostic that is reusable to various other applications. It introduces a false witness while preserving every entity's correct final answer. Any resulting increase in predictions of the suggestion-implied wrong answer therefore isolates the effect of the irrelevant match on suggestion following.

Experiments with pretrained models would test whether the same failure mode and retrieval mechanism arise in natural-language settings. To transfer the method to other applications, e.g., document assistants, one can construct paired contexts from linked records, retrieved passages, or tool outputs. The pair would differ in whether an irrelevant relation matches an incorrect suggestion, while preserving the correct answers to a specified family of questions. Comparing responses to the same question and suggestion would then test whether that match shifts answers toward the suggestion-implied output, without requiring access to internal activations. We anticipate that these targeted matches will have a larger effect than comparable random distractors.

Beyond evaluation, these paired contexts suggest complementary training and prompting experiments. Training could encourage stable answers across equal-function pairs and appropriate changes across edits that change the correct answer. Prompting could ask the model to identify the queried entity's intermediate value and its supporting record before taking the second step. These experiments would test ways to reduce sensitivity to false witnesses by strengthening the connection between the question and the records used to answer it.

Our results identify and analyze how irrelevant facts can steer a transformer toward an incorrect answer. The equal-function test makes this false-witness effect measurable, activation patching makes the underlying attention mechanism transparent, and training interventions show how resistance to such failures develops. In sum, these findings provide a concrete way to evaluate and improve how models select information relevant to the question.

AI use statement

We used generative AI tools for providing feedback on research methodology and experiments, and implementing methods. We have not used generative AI tools for generating synthetic data sets, helping develop theoretical models or conceptual frameworks, formulating mathematical claims, providing ingredients for proving mathematical claims, assisting in the writing of proofs, proposing or refining hypotheses, assisting with translation, cleaning and reformatting datasets, supporting qualitative and thematic data analysis, or interpreting results. We have reviewed all AI-assisted work.

We checked LLM-generated text and code, and evaluated LLM feedback on the methodology. We take responsibility for the final content of this work.

Ethics statement

All experiments use synthetic data. No human participants or personal data are involved.

Reproducibility statement

Experimental specifications and intervention procedures are provided in Appendices C and E. Code and supporting data will be released upon publication and can be shared privately with reviewers upon request during the discussion period.

References

D. Bahdanau, S. Murty, M. Noukhovitch, T. H. Nguyen, H. de Vries, and A. Courville. Systematic Generalization: What Is Required and Can It Be Learned? In *International Conference on Learning Representations*, 2019. URL https://arxiv.org/abs/1811.12889.

P. W. Battaglia, J. B. Hamrick, V. Bapst, et al. Relational inductive biases, deep learning, and graph networks. *arXiv preprint arXiv:1806.01261*, 2018. URL https://arxiv.org/abs/1806.01261.

E. Biran, D. Gottesman, S. Yang, M. Geva, and A. Globerson. Hopping Too Late: Exploring the Limitations of Large Language Models on Multi-Hop Queries. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 14113–14130, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.781. URL https://aclanthology.org/2024.emnlp-main.781/.

T. Bricken, A. Templeton, J. Batson, et al. Towards Monosemanticity: Decomposing Language Models With Dictionary Learning. *Transformer Circuits Thread*, 2023. URL https://transformer-circuits.pub/2023/monosemantic-features/index.html.

L. Chan, A. Garriga-Alonso, N. Goldowsky-Dill, et al. Causal scrubbing, a method for rigorously testing interpretability hypotheses. *AI Alignment Forum*, 2022. URL https://www.alignmentforum.org/posts/JvZhhzycHu2Yd57RN/causal-scrubbing-a-method-for-rigorously-testing.

Z. Chen, H. Lin, Z. Chen, et al. Knowing but Not Correcting: Routine Task Requests Suppress Factual Correction in LLMs. *arXiv preprint arXiv:2605.05957*, 2026. URL https://arxiv.org/abs/2605.05957v2.

E. F. Codd. A Relational Model of Data for Large Shared Data Banks. *Communications of the ACM*, 13(6): 377–387, 1970. doi: 10.1145/362384.362685. URL https://doi.org/10.1145/362384.362685.

A. Conmy, A. N. Mavor-Parker, A. Lynch, S. Heimersheim, and A. Garriga-Alonso. Towards Automated Circuit Discovery for Mechanistic Interpretability. In *Advances in Neural Information Processing Systems*, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Paper-Conference.pdf.

Q. Dai, B. Heinzerling, and K. Inui. Representational Analysis of Binding in Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17468–17493, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.967. URL https://aclanthology.org/2024.emnlp-main.967/.

N. Elhage, N. Nanda, C. Olsson, et al. A Mathematical Framework for Transformer Circuits. *Transformer Circuits Thread*, 2021. URL https://transformer-circuits.pub/2021/framework/index.html.

J. Feng and J. Steinhardt. How Do Language Models Bind Entities in Context? In *International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=zb3b6oKO77.

A. Geiger, H. Lu, T. Icard, and C. Potts. Causal Abstractions of Neural Networks. In *Advances in Neural Information Processing Systems*, 2021. URL https://papers.nips.cc/paper/2021/file/4f5c422f4d49a5a807eda27434231040-Paper.pdf.

A. Geiger, Z. Wu, C. Potts, T. Icard, and N. D. Goodman. Finding Alignments Between Interpretable Causal Variables and Distributed Neural Representations. In *Proceedings of the Third Conference on Causal Learning and Reasoning*, 2024. URL https://proceedings.mlr.press/v236/geiger24a/geiger24a.pdf.

R. Geirhos, J.-H. Jacobsen, C. Michaelis, et al. Shortcut Learning in Deep Neural Networks. *Nature Machine Intelligence*, 2:665–673, 2020. doi: 10.1038/s42256-020-00257-z. URL https://arxiv.org/abs/2004.07780.

M. Geva, R. Schuster, J. Berant, and O. Levy. Transformer Feed-Forward Layers Are Key-Value Memories. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2021. URL https://aclanthology.org/2021.emnlp-main.446.pdf.

M. Geva, J. Bastings, K. Filippova, and A. Globerson. Dissecting Recall of Factual Associations in Auto-Regressive Language Models. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, 2023. URL https://aclanthology.org/2023.emnlp-main.751.pdf.

J. Goldstein and P.-Å. Larson. Optimizing Queries Using Materialized Views: A Practical, Scalable Solution. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 331–342, 2001. doi: 10.1145/375663.375706.

T. Guo, H. Zhu, R. Zhang, et al. How Do LLMs Perform Two-Hop Reasoning in Context? *arXiv preprint arXiv:2502.13913*, 2025. URL https://arxiv.org/abs/2502.13913.

Y. Gur-Arieh, M. Geva, and A. Geiger. Mixing Mechanisms: How Language Models Retrieve Bound Entities In-Context. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2eeff35664016c7f0f8aa704f0d9a83e-Abstract-Conference.html.

S. Gururangan, S. Swayamdipta, O. Levy, R. Schwartz, S. R. Bowman, and N. A. Smith. Annotation Artifacts in Natural Language Inference Data. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pp. 107–112, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-2017. URL https://aclanthology.org/N18-2017/.

M. Hanna, O. Liu, and A. Variengien. How Does GPT-2 Compute Greater-Than?: Interpreting Mathematical Abilities in a Pre-trained Language Model. In *Advances in Neural Information Processing Systems*, 2023. URL https://arxiv.org/pdf/2305.00586.

S. Heimersheim and N. Nanda. How to Use and Interpret Activation Patching. *arXiv preprint arXiv:2404.15255*, 2024. URL https://arxiv.org/pdf/2404.15255.

S. Hoenig, S. J. Park, K. Jain, B. Bussmann, and P. Leask. Additive Relational Bindings in Transformers: What Sparse Autoencoders Miss. In *Second Workshop on Compositional Learning: Safety, Interpretability, and Agents at ICML*, 2026. URL https://openreview.net/forum?id=UHqGHqDD6y.

R. Jia and P. Liang. Adversarial Examples for Evaluating Reading Comprehension Systems. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pp. 2021–2031, Copenhagen, Denmark, September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1215. URL https://aclanthology.org/D17-1215/.

Y. Jiang and M. Bansal. Avoiding Reasoning Shortcuts: Adversarial Evaluation, Training, and Model Development for Multi-Hop QA. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2726–2736, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1262. URL https://aclanthology.org/P19-1262/.

D. Keysers, N. Schärli, N. Scales, et al. Measuring Compositional Generalization: A Comprehensive Method on Realistic Data. In *International Conference on Learning Representations*, 2020. URL https://arxiv.org/abs/1912.09713.

N. Kim and S. Schuster. Entity Tracking in Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 3835–3855, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.213. URL https://aclanthology.org/2023.acl-long.213/.

B. Lake and M. Baroni. Generalization without systematicity: On the compositional skills of sequence-to-sequence recurrent networks. In *Proceedings of the 35th International Conference on Machine Learning*, volume 80, pp. 2873–2882. PMLR, 2018. URL https://proceedings.mlr.press/v80/lake18a.html.

B. Z. Li, M. Nye, and J. Andreas. Implicit Representations of Meaning in Neural Language Models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1813–1827, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.143. URL https://aclanthology.org/2021.acl-long.143/.

K. Li, A. K. Hopkins, D. Bau, F. Viégas, H. Pfister, and M. Wattenberg. Emergent World Representations: Exploring a Sequence Model Trained on a Synthetic Task. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.13382.

B. Liu, J. T. Ash, S. Goel, A. Krishnamurthy, and C. Zhang. Transformers Learn Shortcuts to Automata. In *International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=De4FYqjFueZ.

N. F. Liu, K. Lin, J. Hewitt, et al. Lost in the Middle: How Language Models Use Long Contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173, 2024. doi: 10.1162/tacl_a_00638. URL https://aclanthology.org/2024.tacl-1.9/.

S. Longpre, K. Perisetla, A. Chen, N. Ramesh, C. DuBois, and S. Singh. Entity-Based Knowledge Conflicts in Question Answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7052–7063, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.emnlp-main.565. URL https://aclanthology.org/2021.emnlp-main.565/.

S. Marks, C. Rager, E. J. Michaud, Y. Belinkov, D. Bau, and A. Mueller. Sparse Feature Circuits: Discovering and Editing Interpretable Causal Graphs in Language Models. In *International Conference on Learning Representations*, 2025. URL https://belinkov.com/assets/pdf/iclr2025-sfc.pdf.

R. T. McCoy, E. Pavlick, and T. Linzen. Right for the Wrong Reasons: Diagnosing Syntactic Heuristics in Natural Language Inference. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 3428–3448, Florence, Italy, July 2019. Association for Computational Linguistics. doi: 10.18653/v1/P19-1334. URL https://aclanthology.org/P19-1334/.

I. R. McKenzie, A. Lyzhov, M. Pieler, et al. Inverse Scaling: When Bigger Isn't Better. *Transactions on Machine Learning Research*, 2023. URL https://arxiv.org/abs/2306.09479.

K. Meng, D. Bau, A. Andonian, and Y. Belinkov. Locating and Editing Factual Associations in GPT. In *Advances in Neural Information Processing Systems*, 2022. URL https://arxiv.org/pdf/2202.05262.

W. Merrill and A. Sabharwal. The Expressive Power of Transformers with Chain of Thought. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2310.07923.

N. Nanda, L. Chan, T. Lieberum, J. Smith, and J. Steinhardt. Progress Measures for Grokking via Mechanistic Interpretability. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/pdf/2301.05217v1.

S. Oh and V. Demberg. A Retrieval Conditioned Rebinding Circuit for Dynamic Entity Tracking in Large Language Models. *arXiv preprint arXiv:2606.08644*, 2026. URL https://arxiv.org/abs/2606.08644v2.

C. Olah, A. Satyanarayan, I. Johnson, et al. The Building Blocks of Interpretability. *Distill*, 2018. doi: 10.23915/distill.00010. URL https://distill.pub/2018/building-blocks.

C. Olah, N. Cammarata, L. Schubert, G. Goh, M. Petrov, and S. Carter. Zoom In: An Introduction to Circuits. *Distill*, 2020. doi: 10.23915/distill.00024.001. URL https://distill.pub/2020/circuits/zoom-in.

C. Olsson, N. Elhage, N. Nanda, et al. In-context Learning and Induction Heads. *Transformer Circuits Thread*, 2022. URL https://arxiv.org/pdf/2209.11895.

E. Perez, S. Ringer, K. Lukosiute, et al. Discovering Language Model Behaviors with Model-Written Evaluations. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 13387–13434, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.findings-acl.847. URL https://aclanthology.org/2023.findings-acl.847/.

F. Perez and I. Ribeiro. Ignore Previous Prompt: Attack Techniques For Language Models. *arXiv preprint arXiv:2211.09527*, 2022. URL https://arxiv.org/abs/2211.09527.

N. Prakash, N. Shapira, A. S. Sharma, et al. Language Models Use Lookbacks to Track Beliefs. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2505.14685v3.

C. Sanford, D. Hsu, and M. Telgarsky. Transformers, parallel computation, and logarithmic depth. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 43276–43327. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/sanford24a.html.

P. G. Selinger, M. M. Astrahan, D. D. Chamberlin, R. A. Lorie, and T. G. Price. Access Path Selection in a Relational Database Management System. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 23–34, 1979. doi: 10.1145/582095.582099. URL https://doi.org/10.1145/582095.582099.

F. Shi, X. Chen, K. Misra, et al. Large Language Models Can Be Easily Distracted by Irrelevant Context. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 31210–31227. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/shi23a.html.

A. Templeton, T. Conerly, J. Marcus, et al. Scaling Monosemanticity: Extracting Interpretable Features from Claude 3 Sonnet. *Transformer Circuits Thread*, 2024. URL https://transformer-circuits.pub/2024/scaling-monosemanticity/index.html.

M. Turpin, J. Michael, E. Perez, and S. R. Bowman. Language Models Don't Always Say What They Think: Unfaithful Explanations in Chain-of-Thought Prompting. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://arxiv.org/abs/2305.04388.

A. Variengien and E. Winsor. Look Before You Leap: Universal Emergent Mechanism for Retrieval in Language Models. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/ad36c2cfc423e75c6d68d751a955b22e-Abstract-Conference.html.

J. Vig, S. Gehrmann, Y. Belinkov, et al. Investigating Gender Bias in Language Models Using Causal Mediation Analysis. In *Advances in Neural Information Processing Systems*, 2020. URL https://papers.nips.cc/paper/2020/file/92650b2e92217715fe312e6fa7b90d82-Paper.pdf.

K. Wang, A. Variengien, A. Conmy, B. Shlegeris, and J. Steinhardt. Interpretability in the Wild: A Circuit for Indirect Object Identification in GPT-2 Small. In *International Conference on Learning Representations*, 2023. URL https://arxiv.org/pdf/2211.00593.

J. Wei, D. Huang, Y. Lu, D. Zhou, and Q. V. Le. Simple synthetic data reduces sycophancy in large language models. *arXiv preprint arXiv:2308.03958*, 2023. URL https://arxiv.org/abs/2308.03958.

Y. Wu, A. Geiger, and R. Millière. How Do Transformers Learn Variable Binding in Symbolic Programs? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67284–67299. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/wu25j.html.

X. Xie, J. W. K. Ho, C. Murphy, G. Kaiser, B. Xu, and T. Y. Chen. Testing and Validating Machine Learning Classifiers by Metamorphic Testing. *Journal of Systems and Software*, 84(4):544–558, 2011. doi: 10.1016/j.jss.2010.11.920.

H. Yuan, P. Cao, Z. Jin, et al. Whispers that Shake Foundations: Analyzing and Mitigating False Premise Hallucinations in Large Language Models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 2670–2683, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.155. URL https://aclanthology.org/2024.emnlp-main.155/.

F. Zhang and N. Nanda. Towards Best Practices of Activation Patching in Language Models: Metrics and Methods. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/pdf/2309.16042.

APPENDIX CONTENTS

A Related work . . . **15**

B A database view of relational relevance . . . **16**

B.1 Joining records and selecting the requested person . . . 17

B.2 A locally supported but query-irrelevant predicate . . . 17

B.3 A concept map for the experiments . . . 17

B.4 The equal-function test preserves the answer view . . . 17

B.5 From the analogy to activation patching . . . 18

C Task, inputs, and training . . . **18**

C.1 Random tables and target answers . . . 18

C.2 Model and optimizer . . . 19

C.3 Training groups and paired comparisons . . . 19

C.4 Access masks and the independent-information control . . . 19

D Paired input tests and measurements . . . **20**

D.1 One matching record and a same-row control . . . 20

D.2 Equal-function pairs . . . 20

D.3 Repeating the symbol in the second-step table . . . 20

D.4 A separate suggestion vocabulary . . . 21

E Activation patching and what it measures . . . **21**

E.1 Patching one part of attention . . . 21

E.2 Output information at first-step record positions . . . 21

E.3 Equal-function changes in individual projections . . . 21

E.4 Crossing independently changed keys and values . . . 22

E.5 Replacing the query vector to recover or redirect the answer . . . 23

E.6 The computation learned after access interventions . . . 23

F Calculations behind the controlled tests . . . **23**

F.1 The complete answer function and a matching-record computation . . . 23

F.2 Information paths through two layers . . . 24

F.3 Using the records and suggestion without the queried entity . . . 24

G Query information and attention interactions . . . **24**

G.1 Single-edit sensitivity with a query cue . . . 24

G.2 Crossed attention interactions and mixed-term removal . . . 25

H Learning across reliability, layout, and task settings . . . **26**

H.1 Fixed positions and longer training . . . 26

H.2 Additional sizes and parameterizations . . . 27

H.3 Reading changes over time . . . 27

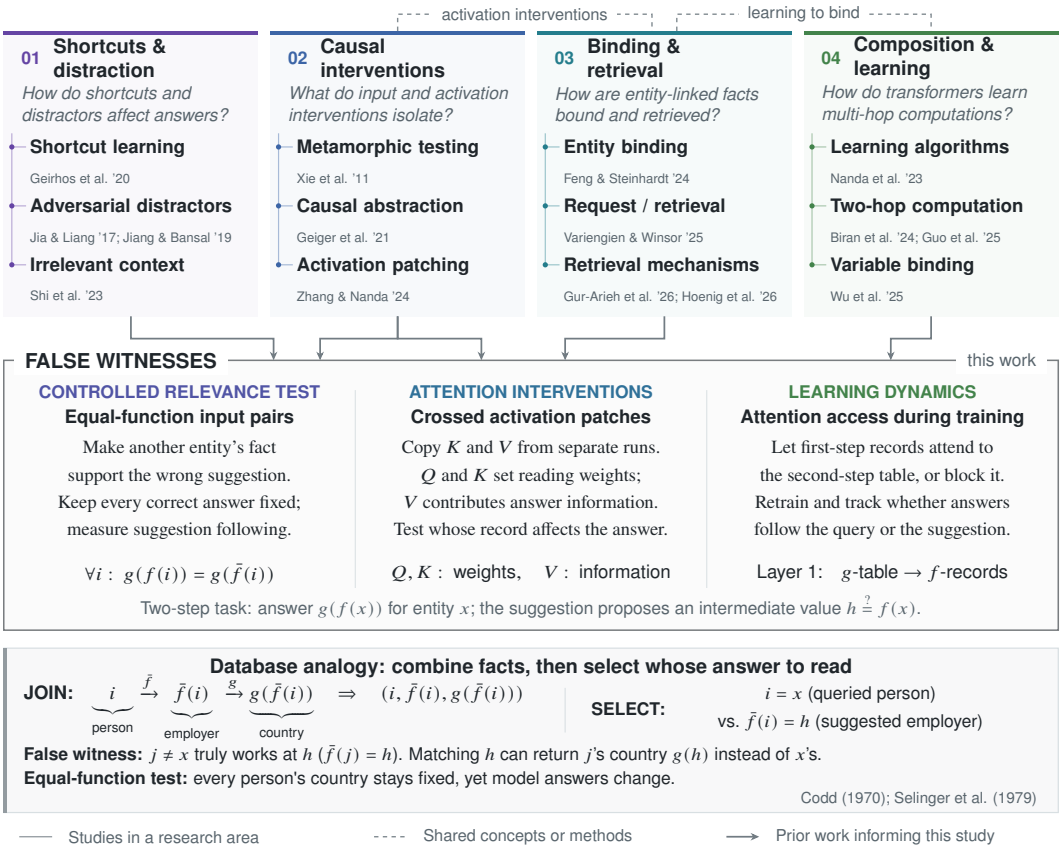


Figure 7: **Connections between prior work and the three contributions of this study.** The four upper branches group representative studies of shortcuts and contextual distraction, causal interventions, entity binding and retrieval, and compositional learning. Dashed links mark shared concepts or methods, and arrows connect these areas to our experiments. The middle panel summarizes our equal-function test, which changes an off-query suggestion match while preserving every entity's composed answer; crossed K/V patches and Q interventions, which separate record selection from retrieved content; and information-access interventions that probe how selection is learned. The database analogy below distinguishes composing the relations from selecting whose answer to read: a true record $\bar{f}(j) = h$ for $j \neq x$ can supply another person's country $g(h)$. Equal-function pairs preserve all person–country answers while changing whether that record matches the suggestion. Appendix B develops this analogy.

A Related work

Mechanistic interpretability and causal interventions. Mechanistic interpretability explains behavior through the representations and computations that produce it. Feature visualization and the circuits program established a vocabulary for such explanations (Olah et al., 2018; 2020). Transformer circuit analysis separates query–key matching from value transport (Elhage et al., 2021). Induction, indirect object identification, and numerical comparison illustrate how components implement algorithms (Olsson et al., 2022; Wang et al., 2023; Hanna et al., 2023). Automated circuit discovery, sparse autoencoders, and sparse feature circuits extend the analysis to computational subgraphs and learned features (Conmy et al., 2023; Bricken et al., 2023; Templeton et al., 2024; Marks et al., 2025). Causal mediation links internal activations to behavioral effects (Vig et al., 2020; Meng et al., 2022). Causal abstraction, learned alignments, and causal scrubbing test hypotheses about higher-level computations (Geiger et al., 2021; 2024; Chan et al., 2022). Activation-patching methodology makes the input contrast, intervention site, and outcome metric explicit (Zhang & Nanda, 2024; Heimersheim & Nanda, 2024). Our crossed key/value interventions apply this logic to distinguish the relation that selects a record from the output retrieved through it.

Entity binding and relational computation. Binding research asks how models keep entity–attribute associations distinct and retrieve them in context. Feng & Steinhardt (2024) identify binding-ID vectors through causal interventions; Dai et al. (2024) analyze entity–attribute binding representations.

Entity tracking and representations of discourse and game states extend this question to evolving situations (Kim & Schuster, 2023; Li et al., 2021; 2023). Variengien & Winsor (2025) identify a decomposition between request processing and contextual retrieval. Gur-Arieh et al. (2026) separate positional, lexical, and reflexive retrieval mechanisms and explain how models combine them across context positions. Prakash et al. (2026) describe belief tracking through lookbacks: a query-side pointer matches an address to retrieve its associated payload. Oh & Demberg (2026) identify a retrieval-conditioned rebinding circuit for dynamic entity tracking. In a two-layer transformer, Hoenig et al. (2026) find an additive entity–relation address, a separate payload, and same-head query–key matching. Address patches redirect retrieval to a distractor. These studies provide precise accounts of binding and retrieval against which to examine query-relevant selection. In our task, a suggested intermediate value offers a competing criterion for selecting an entity's record. A true relation involving another entity satisfies this criterion, allowing an accurate local match to direct the computation toward an incorrect answer. The equal-function construction varies that match while keeping every entity's composed output unchanged. Independent key and value donors then identify the contributions of matching and retrieved content, and query patches test which entity the same contextual representations are used to answer about.

Compositional-generalization benchmarks and relational inductive biases place these operations within the broader problem of recombining learned structure (Lake & Baroni, 2018; Keysers et al., 2020; Bahdanau et al., 2019; Battaglia et al., 2018). Two-hop reasoning studies examine intermediate representations and composition (Biran et al., 2024; Guo et al., 2025). Analyses of automata, parallel computation, and chain-of-thought expressivity describe how transformers organize multi-step operations across depth and tokens (Liu et al., 2023; Sanford et al., 2024; Merrill & Sabharwal, 2024). Training-time analysis traces the acquisition of algorithms (Nanda et al., 2023). Particularly close to our setting, Wu et al. (2025) follow a transition from shallow heuristics to systematic variable dereferencing, with the residual stream serving as addressable memory. Our training experiments follow the acquisition of query-relevant record selection as suggestion reliability and access to intermediate information change.

Shortcut learning and misleading context. Shortcut learning connects predictive regularities to departures from an intended computation (Geirhos et al., 2020), including annotation artifacts and syntactic heuristics (Gururangan et al., 2018; McCoy et al., 2019). Targeted distractors, multi-hop question-answering shortcuts, irrelevant context, and context position expose sensitivity to surrounding evidence (Jia & Liang, 2017; Jiang & Bansal, 2019; Shi et al., 2023; Liu et al., 2024). Factual-recall mechanisms distinguish storing associations from retrieving them (Geva et al., 2021; 2023), while knowledge conflicts, false premises, and task-induced suppression of factual correction connect available information to response selection (Longpre et al., 2021; Yuan et al., 2024; Chen et al., 2026). Sycophancy, inverse scaling, biased chain-of-thought explanations, and prompt injection identify additional ways that suggestions or instructions steer outputs (Perez et al., 2023; Wei et al., 2023; McKenzie et al., 2023; Turpin et al., 2023; Perez & Ribeiro, 2022). A false witness specifies a relational shortcut: an off-query record matches a suggested intermediate value and supplies a coherent path to the suggested output. Controlling suggestion reliability during training connects this cue's predictive value to its subsequent use. Fresh random function tables make the relational structure, including the distinction between the queried entity and other entities, the recurring feature of the task. A same-row control measures the effect of editing that position without creating the suggested match.

B A DATABASE VIEW OF RELATIONAL RELEVANCE

A relational query separates the records being combined from the condition specifying which result is requested. This distinction is illustrated here using the person–company–country interpretation of f and g: a country can be correctly associated with an employee and still answer a question about someone else. We develop the analogy using relational operations (Codd, 1970), query execution (Selinger et al., 1979), and materialized views (Goldstein & Larson, 2001). The table-and-query description of contextual retrieval in Variengien & Winsor (2025) provides a direct connection to language-model interpretability.

B.1 JOINING RECORDS AND SELECTING THE REQUESTED PERSON

Here entities i are people, intermediate values $f(i)$ are companies, and outputs $g(f(i))$ are countries. Write the two input relations as

$$\mathcal{R}_F = \{(i, f(i)) : i \in [N]\}, \qquad \text{columns: } (\text{person}, \text{company}), \tag{2}$$

$$\mathcal{R}_G = \{(c, g(c)) : c \in [N]\}, \qquad \text{columns: } (\text{company}, \text{country}). \tag{3}$$

Their natural join on the company column is

$$\mathcal{J} = \mathcal{R}_F \bowtie \mathcal{R}_G = \{(i, f(i), g(f(i))) : i \in [N]\}. \tag{4}$$

Selecting the queried person and projecting the country column gives

$$\pi_{\text{country}}\big(\sigma_{\text{person}=x}(\mathcal{J})\big) = \{g(f(x))\}. \tag{5}$$

Here σ selects rows satisfying its predicate, and π retains the named columns. In SQL, the query is:

```
SELECT G.country
FROM Employees AS F JOIN Companies AS G
     ON F.company = G.company
WHERE F.person = :queried_person;
```

The join condition connects each employee to the correct company. **The selection predicate specifies whose connected record is relevant.** For Alice at Harbor and Bob at Cedar, the joined records include $(\text{Alice}, \text{Harbor}, \text{Japan})$ and $(\text{Bob}, \text{Cedar}, \text{France})$. Selecting Alice returns Japan.

B.2 A LOCALLY SUPPORTED BUT QUERY-IRRELEVANT PREDICATE

The suggestion h offers a different predicate, $\text{company} = h$. Applied to the same join, it gives

$$\pi_{\text{country}}\big(\sigma_{\text{company}=h}(\mathcal{J})\big) = \{g(h)\} \qquad \text{when some } i \text{ has } f(i) = h. \tag{6}$$

For the Cedar suggestion, Bob supplies a genuine joined record ending in France. The false witness is this off-query record. **Its company matches the suggestion, while its person differs from the queried person.** The two predicates therefore explain the competing answers:

$$\underbrace{i = x}_{\text{select the queried person}} \qquad \text{and} \qquad \underbrace{f(i) = h}_{\text{select a suggestion-matching person}} \quad . \tag{7}$$

The matching-record computation in Appendix F expresses the second operation as normalized weights over the matching employee rows. Every such row supplies the same suggested country.

B.3 A CONCEPT MAP FOR THE EXPERIMENTS

Table 3 connects the database vocabulary to the paper's task and interventions. The database key identifies a record in the symbolic input; an attention key is a projected activation compared with the query vector. Their relationship is tested by changing which record matches and by patching the corresponding activations.

B.4 THE EQUAL-FUNCTION TEST PRESERVES THE ANSWER VIEW

Return to the equal-function pair, with control table f and false-witness table $\bar{f}$. Project the control input's joined relation onto person and country:

$$\mathcal{H} = \pi_{\text{person,country}}(\mathcal{J}) = \{(i, g(f(i))) : i \in [N]\}. \tag{8}$$

The equal-function pair assigns different companies r and h to the same off-query employee j in two hypothetical inputs, with both companies mapping to the same country. Hence

$$\mathcal{H} = \bar{\mathcal{H}}, \qquad \mathbf{1}[f(j) = h] = 0, \qquad \mathbf{1}[\bar{f}(j) = h] = 1. \tag{9}$$

Here $\bar{\mathcal{H}} = \{(i, g(\bar{f}(i))) : i \in [N]\}$ is the view computed from $\bar{f}$. The person–country view is identical, and the company column determines which side of the pair contains the suggestion match. In one hypothetical input, Bob works at Maple; in the other, he works at Cedar. Both companies map to France. Alice's country is Japan in both inputs, Bob's is France in both, and every other person's country also agrees across them. The experiment measures the effect of making the company-based predicate succeed at Bob's record.

Table 3: Database concepts and their role in the task and mechanistic experiments.

Database concept	Meaning in the relational task	Connection to the experiments
Base relation	Employee–company table f; company–country table g.	Fresh tables supply the facts for each example.
Record key	Person i or company c indexing an input row.	Input positions identify table keys; tokens give their values.
Join	Connect $(i, f(i))$ with $(f(i), g(f(i)))$.	First-layer company-to-employee access lets employee positions obtain country information.
Projected view	$\mathcal{H} = \{(i, g(f(i))) : i \in [N]\}$.	The equal-function test preserves this complete person–country relation.
Materialized view	A previously computed relation available to later queries.	Provides the analogy for country information available at employee positions before final selection.
Selection predicate	$i = x$ selects the queried person; $f(i) = h$ selects a suggestion match.	Query and key activation patches change which employee position influences the answer.
Retrieved field	The country attached to the selected record.	Value patches transfer country information; crossed K/V patches test selection and transfer together.
False witness	A true off-query record satisfying $f(j) = h$.	Its presence can increase answers following the incorrect suggestion.
View-preserving edit	Change $f(j)$ between r and h with $g(r) = g(h)$.	Equal-function pairs change the suggestion match while preserving $\mathcal{H}$.
Execution plan	An arrangement of lookups and intermediate results used to answer the query.	Table order and attention masks change when company information can reach employee positions.

B.5 From the analogy to activation patching

The database description supplies a concrete mechanistic hypothesis: country information is prepared at employee positions, and the final attention computation selects among those positions. In the vocabulary of binding and retrieval, a representation used for matching serves as an address, while the information retrieved from the matched position serves as a payload (Prakash et al., 2026; Hoenig et al., 2026). The experiments operationalize these roles with projected query, key, and value activations.

In the crossed intervention, Bob's key comes from the input where his company matches Cedar. His value comes from an independently changed input carrying Canada. The recipient company records still associate Cedar with France. An answer of Canada therefore tracks the field supplied at the selected employee position. Patching the final query tests the complementary operation: changing whose record is requested while retaining the table keys and values. Finally, opening or closing first-layer company-to-employee attention connections changes when the country information can become available during learning. The analogy unifies the behavioral test, the activation interventions, and the training intervention around two operations: *preparing a value and selecting the record relevant to the query*.

C Task, inputs, and training

This appendix gives the input encoding, training settings, and populations behind the figures. In technical descriptions, F means f-table positions and G means g-table positions.

C.1 Random tables and target answers

Let $[N] = \{0, \ldots, N-1\}$. Independently sample $f, g : [N] \to [N]$ uniformly over all functions, and sample x uniformly. With a suggestion present, set $h = f(x)$ with probability α; otherwise choose h uniformly from $[N] \setminus \{f(x)\}$. The main setting is $N = 8$, $\alpha = 3/4$. The label is always $g(f(x))$ for a two-step question or $f(x)$ for a one-step question. A wrong intermediate suggestion

can still have $g(h) = g(f(x))$ because distinct intermediate values can map to the same output. We distinguish these output collisions when defining evaluation subsets.

For $N = 8$, the sequence has 19 tokens and the vocabulary has 11 entries. Each table is the list of its eight values in key order. Positions supply the keys. Reversing the two table blocks leaves the order of rows within each block unchanged.

Write $t \in \{t_1, t_2\}$ for the task token and a for the absence token. The primary query-last layout ends in (h, t, x) and the suggestion-last layout in (x, t, h) when a suggestion is present. Without a suggestion, both end in (a, t, x). The fixed-position experiments instead always use $(x, t, h$ or $a)$, or append a constant answer token to that suffix. These comparisons keep x at position 16. The appended token is 12, giving a 13-entry vocabulary and sequence length 20.

C.2 Model and optimizer

The primary transformer has two pre-normalized layers, model width 64, four attention heads, and zero dropout. Each layer's MLP has widths $64 \to 128 \to 64$, with exact GELU between the two linear layers. It uses token embeddings and learned absolute position embeddings, with the latter initialized with independent normal entries of standard deviation 0.02. A final layer normalization and linear eight-class head read the last position. Attention and feed-forward biases are included.

Training uses float32 AdamW with learning rate 0.003, weight decay 0.01, betas $(0.9, 0.999)$, epsilon 10^{-8}, batch size 128, and global gradient norm clipping at 1. Weight decay applies to all trainable parameters. The learning rate stays constant. Each example independently draws the tables, question, task type, suggestion presence, and suggestion validity. One-step questions occupy 25% of training examples and two-step questions 75%. Suggestions occur in half. Four thousand updates expose a model to 512,000 fresh examples.

C.3 Training groups and paired comparisons

The initial study uses three independent initializations in each of three configurations, giving nine models. The primary replication uses eight initializations for each combination of table order and final-token layout, giving 32 models. The access experiment uses eight initializations in each of four conditions, also giving 32 models. Both groups train for 4,000 updates and use 2,000 and 4,000 as intervention checkpoints. Nine saved natural-prediction evaluations cover updates $0, 500, \ldots, 4,000$ for the primary replication.

Matched conditions share initialization and the full sequence of training examples. Evaluations use fixed inputs shared across the compared models. Group statistics give equal weight to each model. All members of each specified group appear in the summaries, and checkpoint comparisons follow the same models through training.

C.4 Access masks and the independent-information control

Let $A_{ij} = 1$ indicate that query position i may attend to key position j. The ordinary mask is $A_{ij} = \mathbf{1}[j \le i]$. With the f table first, $F = \{0, \ldots, 7\}$ and $G = \{8, \ldots, 15\}$. Opening access sets $A_{ij} = 1$ for all $i \in F, j \in G$ in layer one. With the g table first, the blocks are reversed; closing access sets those 64 entries to zero. Self-attention, the table-to-suffix restriction, and the second-layer mask retain their original settings. The opened access uses the complete, already supplied table prefix.

The information control uses the opened f-first mask in both conditions. For first-layer queries at f-table positions only, g-position keys and values are computed either from the current g table or from an independent uniform table $\widetilde{g}$. If Z_j is the normalized embedding-plus-position state and $\widetilde{Z}_j$ is its independent-table counterpart, these messages are $\widetilde{Z}_j W_K + b_K$ and $\widetilde{Z}_j W_V + b_V$ in the independent condition. Every other query uses the original Z_j messages. Query vectors and receiver residuals remain those of the actual input, and layer two keeps its causal mask. *This fixes the graph and key count while changing whether messages contain the current problem's g-table information.*

We train eight matched pairs of models for 4,000 updates under the two message conditions. Both conditions share initialization, training examples, and auxiliary tables. Natural and equal-function

tests use separate 4,096-example populations. Each pair shares its independent messages across all edited inputs.

D PAIRED INPUT TESTS AND MEASUREMENTS

Let $y = g(f(x))$ be the true answer, $y_h = g(h)$ the suggestion-implied answer, and ℓ_c the output score for class c. The model predicts $\widehat{y} = \arg\max_c \ell_c$, with ties resolved by the smallest class index. We report accuracy $\Pr[\widehat{y} = y]$ and suggestion following $\Pr[\widehat{y} = y_h]$. On equal-function inputs, $y \neq y_h$ by construction. A false-witness effect is the change in suggestion following between the input containing the witness and its specified control, expressed in percentage points. This is the main figure metric. The appendix also reports changes in the score difference $\ell_{y_h} - \ell_y$ and in predicted probabilities.

D.1 ONE MATCHING RECORD AND A SAME-ROW CONTROL

To test whether the effect also occurs without the equal-function constraint, we compare a false-witness edit with a nonmatching edit at the same unqueried position. Sample a wrong suggestion absent from the image of f, and choose another row $j \neq x$ uniformly. The witness edit sets $f(j) = h$; the control sets that same entry to a value different from both h and its previous value. Both preserve $x, f(x), g, h, y, y_h$. Of the 4,096 examples, 3,600 have $y \neq y_h$; Table 1 uses those examples for this contrast. The primary eight-seed replication uses the same intervention examples as the initial study. **False witnesses also increase suggestion following without the equal-function constraint** (Table 1).

We also test whether multiple false witnesses affect models that initially answer accurately. We create 2, 3, or 7 matching rows by following a randomly ordered list of the seven nonqueried rows. The two initially accurate g-first models retain about 94% true-answer accuracy even when all seven nonqueried rows match h.

D.2 EQUAL-FUNCTION PAIRS

Choose distinct intermediate values h, r and distinct queried and edited entities x, j. Draw an initial table with no values equal to h or r, then set its entry at j to r to obtain the control table f. Construct $\bar{f}$ by changing that same entry to h. Set $g(r) = g(h)$ and require $y \neq y_h$. These two inputs share the full two-step answer function, the current question, the suggestion, and the one-step answer $f(x)$. **The edited record changes from $f(j) = r$ to $\bar{f}(j) = h$, while its two-step answer remains fixed.** The primary population contains 4,096 such pairs.

In the same spirit as metamorphic testing (Xie et al., 2011), our equal-function test compares model predictions before and after an input transformation that preserves the complete two-step answer function.

For pair e, let $\widehat{y}^{\text{ctrl}}_{m,e}$ and $\widehat{y}^{\text{fw}}_{m,e}$ denote model m's predictions on the control and false-witness inputs, and let $y_{h,e}$ be their shared suggestion-implied answer. The principal effect over n pairs is

$$\Delta_m = \frac{100}{n} \sum_{e=1}^{n} \left(\mathbf{1}[\widehat{y}^{\text{fw}}_{m,e} = y_{h,e}] - \mathbf{1}[\widehat{y}^{\text{ctrl}}_{m,e} = y_{h,e}] \right). \tag{10}$$

Figures show each model and the equally weighted group mean. The 4,096 examples estimate each model's behavior; the eight training seeds show variation across trained models. Score and probability changes provide additional resolution when the most likely class stays the same.

D.3 REPEATING THE SYMBOL IN THE SECOND-STEP TABLE

To distinguish relational matching from repetition of the suggestion symbol, we repeat that symbol at an unused second-step position. From the no-witness input, choose a g-table key k outside the image of f, with $k \neq h$ and $g(k) \neq h$. Change $g(k)$ to h, or to another nonmatching symbol as a same-position control. This keeps $g \circ f$ and $g(h)$ fixed while introducing the same literal symbol in a different table role. Among 4,096 inputs, 243 have no eligible g-table row and remain unchanged

in all-example results. The direct comparison with f-table edits uses the 3,385 examples that have an eligible row and $y \neq y_h$. At 2,000/4,000 updates, f-first witness effects are 51.88/36.19 points and g-symbol effects are 1.25/0.86 points. **Repeating the symbol has a much smaller effect than introducing a matching first-step relation.**

D.4 A SEPARATE SUGGESTION VOCABULARY

In three initial models, suggestions use a separate token vocabulary with its own embeddings, disjoint from the vocabulary used for table values. Equal-function effects are 32.84, 45.83, and 37.01 points. **The matching relation can therefore be learned across distinct token identities.** This extends the input test from literal repetition to the relationship between a suggestion and a table value.

E ACTIVATION PATCHING AND WHAT IT MEASURES

E.1 PATCHING ONE PART OF ATTENTION

Let $H^{(1)}$ be the state after layer one and let $Z = \mathrm{LN}(H^{(1)})$. Layer-two projections are $Q = ZW_Q + b_Q$, $K = ZW_K + b_K$, and $V = ZW_V + b_V$. For head a at final position L, attention computes

$$u_L^{(a)} = \mathrm{softmax}\left(\frac{q_L^{(a)} K^{(a)\top}}{\sqrt{16}} + M_{L,:}\right) V^{(a)}. \tag{11}$$

The output projection, residual addition, feed-forward block, final normalization, and answer head then proceed normally. A key replacement changes only selected rows of K; a value replacement changes only rows of V; a query replacement changes only q_L. We replace all heads together. Copying a full first-layer state instead changes its subsequent projections and, at the final position, the residual.

For each intervention, we compute vectors from explicit alternative inputs to the same model and retain every unselected vector from the original computation. The result measures the contribution of the copied component in that combination. Joint replacements reveal interactions between components. Comparing single and combined replacements is therefore more informative than adding their individual effects.

The final-position query attends to all 19 input positions.

E.2 OUTPUT INFORMATION AT FIRST-STEP RECORD POSITIONS

We test whether first-step record representations carry their composed answers. Changing g while fixing f, x, h creates original and alternative true and suggested answers $y, y_h, \bar{y}, \bar{y}_h$, where the donor outputs are $\bar{y} = \bar{g}(f(x))$ and $\bar{y}_h = \bar{g}(h)$. In the initial accurate g-first models, copying the first-layer state at the position for the queried entity x follows $\bar{y}$ on 99.00% and 97.95% of 1,705 examples where all four answers differ. Copying all other f-table positions follows $\bar{y}$ on 0.12% and 0.18%. Patching only V at f-table positions gives corresponding output transfer. In the initially failing g-first model, copying the f-table rows that match h transfers $\bar{y}_h$ on 85.49% of the 1,006 all-distinct examples with a matching row; copying the queried row retains y_h on 99.80%. **In these models, the accurate and suggestion-following computations draw their answers from different first-step records.** Query-last models instead transfer the changed answer through g-position values, with rates of 99.62–100% in the initial study.

E.3 EQUAL-FUNCTION CHANGES IN INDIVIDUAL PROJECTIONS

We patch projections separately to identify which attention components transmit the effect of the irrelevant match. The original computation uses f and the alternative input uses $\bar{f}$. We copy f-position keys, f-position values, g-position keys, g-position values, or the final query. The factorial tests use a fixed 2,048-pair population and all 64 model/checkpoint combinations. Figure 2 reports the class-proportion changes for g-first models at 2,000 updates; Figure 8 gives score changes for both orders.

For the initially failing g-first model, suggestion following is 58.42% in the original input and 98.97% in the witness input. Copying f-position keys gives 98.85%; copying f-position values gives 58.57%. The respective mean changes in $\ell_{y_h} - \ell_y$ are 2.005, 0.004, and 2.033 for keys, values, and the full input change. Across all 16 g-first suggestion-last replication checkpoints, f-key score changes are positive and exceed the absolute f-value change. One late model has full-input, f-key, f-value, and final-query changes of 1.774, 1.242, -0.112, and 0.078.

With the f table first, g-position keys and the final query carry the main positive group effects. In one late model, either single replacement has a negative effect while the complete input change has a positive effect. Thus, **table order changes which attention components carry the false-witness effect.**

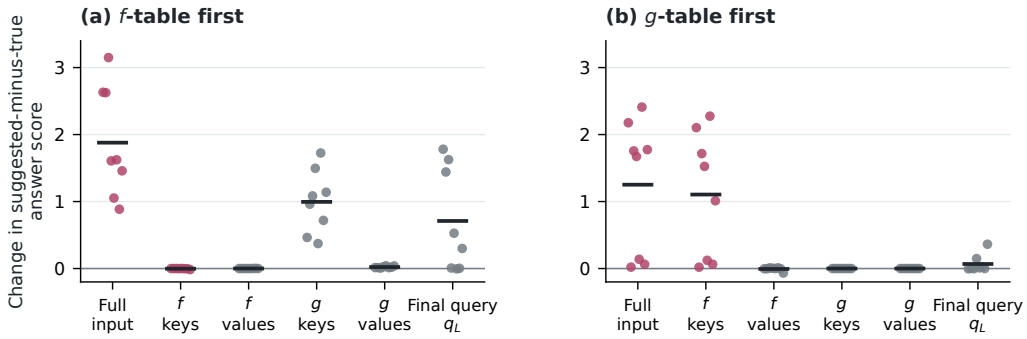


Figure 8: **Score changes reveal order-dependent contributions.** All eight suggestion-last models per order at 2,000 updates, on the same 2,048 input pairs. Each point is the change in the suggested-answer minus true-answer score under one replacement. Black marks show means. The labels f and g denote the corresponding table positions. This companion to Figure 2 retains information below the threshold needed to change the predicted class.

E.4 Crossing independently changed keys and values

We test whether a matching key can make an unqueried record supply an independently specified answer. We generate 2,048 fresh pairs. The recipient uses (f, g) with $f(j) = r$ and $g(r) = g(h) = y_h \neq y$. The key donor uses $(\bar{f}, g)$, where $\bar{f}$ differs from f only at j, with $\bar{f}(j) = h$. For the independent value donor $(f, \bar{g})$, choose a third output $\bar{y}_h \notin \{y, y_h\}$ and set $\bar{g}(r) = \bar{g}(h) = \bar{y}_h$, leaving all other entries equal to g. No other f-table row uses r or h, so the value donor changes only entity j's two-step answer. In particular, $\bar{y} = \bar{g}(f(x)) = y$: the value donor still has the same correct answer to the original query.

At row j, replace neither vector, K alone, V alone, or both. Every other key, value, query, and residual stays fixed. In particular, the original g-position values continue to encode y_h. The combination therefore tests whether the f-position value contributes the output. For donor-output answer rates p_0, p_K, p_V, p_{KV}, the interaction is

$$I_{\bar{y}_h} = 100\big(p_{KV} - p_K - p_V + p_0\big). \tag{12}$$

The initial failing model gives $p_{KV} = 72.12\%$, $p_V = 9.86\%$, and $I_{\bar{y}_h} = 65.97$ points; K alone selects the old suggested answer on 98.49%. Across the eight replication models, $I_{\bar{y}_h}$ averages 25.95 points at 2,000 updates (five positive, three zero), and 4.33 at 4,000 (one positive, seven zero). **A suggestion-matching key can make an unqueried record supply an independently specified answer.**

Another condition places the same V at the position for the queried entity x. This tests where the donor output can be used. It transfers that output on 98.83% and 98.63% in the two initially accurate g-first models; the combination at the position for j gives 0.15% and 0.49%. The late replication mean at the queried position is 88.82%. With the f table first, the layer-two value at an f-table position is structurally independent of the changed g-table information, and these value changes have exactly zero effect. We evaluate all nine initial models and all 64 replication checkpoints.

E.5 Replacing the query vector to recover or redirect the answer

One alternative input removes the suggestion and retains x; a second also changes the question to $\bar{x}$ with $g(f(\bar{x})) \notin \{y, y_h\}$. Tables cannot read the later question/suggestion tokens, so their first-layer states are identical in all these runs. Only the final projected query is copied. Of 4,096 inputs, 4,046 have an available $\bar{x}$; 3,555 also have $y \neq y_h$.

At 2,000 updates, same-entity query replacement raises mean accuracy from 18.98% to 90.65% with the f table first, and from 54.67% to 92.55% with the g table first. Alternate-entity query replacement follows the alternate answer on 87.94% and 90.82%, respectively. These are complete eight-model means on the 4,046-example subset. **Replacing the final query can redirect the answer while all table representations remain fixed.** At 4,000 updates, the same-entity replacement lowers accuracy in five suggestion-last model/checkpoint conditions. The largest decrease is 100% to 83.22%, even though the alternative input is answered perfectly. A query vector that is effective in the donor run need not be equally effective with the recipient's remaining computation.

E.6 The computation learned after access interventions

We test whether models retrieve the answer through first-step records or second-step positions after the access interventions. The tests cover all 96 access and message-control checkpoints. We generate 2,048 fresh triples $f, g, \bar{g}$, questions, suggestions, and auxiliary g tables. Changing only g produces $y, y_h, \bar{y}, \bar{y}_h$, with donor outputs $\bar{y} = \bar{g}(f(x))$ and $\bar{y}_h = \bar{g}(h)$. Of these examples, 780 have four distinct answers and 473 of those also have a record satisfying $f(i) = h$. Independent messages are shared between original and alternative inputs.

Value replacements target the position for x, positions satisfying $f(i) = h$, remaining f-table positions, all f-table positions, or all g-table positions. For opened f-first models, copying the value at the position for the queried entity x gives 98.48%/98.57% $\bar{y}$ following at 2,000/4,000 updates. With informative messages the corresponding rates are 99.82%/99.78%. Ordinary g-first models increase from 63.43% to 99.82% at the queried position; copying matching f-table positions transfers $\bar{y}_h$ on 14.81% at 2,000 updates and zero at 4,000.

For ordinary f-first, closed g-first, and independent-message models, first-layer states at f-table positions are independent of the actual g change. Every value replacement at f-table positions leaves logits unchanged at all 48 such checkpoints. Accurate models within these conditions transfer the alternative answer through g-position values. **Accurate models can retrieve answers through different table positions under different access conditions.**

F Calculations behind the controlled tests

F.1 The complete answer function and a matching-record computation

Represent deterministic lookups by one-hot matrices $\mathsf{F}_{ij} = \mathbf{1}[f(i) = j]$ and $\mathsf{G}_{jy} = \mathbf{1}[g(j) = y]$. Their product $\mathsf{H} = \mathsf{FG}$ lists the two-step output for every entity. Correct selection returns $e_x^\top \mathsf{H}$.

For a suggestion h, let $n_h = \sum_i \mathbf{1}[f(i) = h]$. When $n_h > 0$, the normalized weights $w_h = \mathsf{F}e_h/n_h$ select the entities satisfying $f(i) = h$. Since every selected entity has output $g(h)$,

$$(w_h^\top \mathsf{F})_j = \frac{1}{n_h} \sum_i \mathbf{1}[f(i) = h]\mathbf{1}[f(i) = j] = \mathbf{1}[j = h], \tag{13}$$

$$w_h^\top \mathsf{H} = e_h^\top \mathsf{G}. \tag{14}$$

The result is the suggested output $g(h)$, independently of the queried entity x. The condition $n_h > 0$ defines this computation. Behavior when there is no match is measured by the no-witness input tests.

For the equal-function pair, every unchanged f-table row has the same two-step output, and the changed row has $g(h) = g(r)$. This proves Equation (1). A minimal symbolic instance is $g = (0, 1, 1)$, $x = 0$, $h = 2$, with $f = (0, 2, 0)$ and $f = (0, 1, 0)$. Both complete output tables are $(0, 1, 0)$. Three symbols are needed for this construction with distinct true and suggested answers: with only two symbols, requiring the two distinct intermediate values to share an output makes g constant.

F.2 Information paths through two layers

When the last token contains x, one possible computation retrieves $f(x)$ in layer one and $g(f(x))$ in layer two. When the g table precedes the f table, each f-table position for entity i can independently retrieve $g(f(i))$ in layer one, while the final position reads x. Layer two can then select the record for x. These are different arrangements of the same logical lookups.

With the f table first, g-table positions can aggregate information about which inputs map to their key under f; the final position can use that information to answer. Representing such a set may require both its membership and the presence of matches: an average index alone can conflate different sets. These parallel possibilities motivate studying actual information paths rather than assigning each informal reasoning step to a fixed number of layers (Liu et al., 2023; Sanford et al., 2024). The mask intervention controls one specific path and tests its contribution to learning.

F.3 Using the records and suggestion without the queried entity

A useful comparison computes the answer distribution after averaging over the queried entity. This specifies what an idealized solver would do if it used f, g, h but discarded x. Let $\beta = (1-\alpha)/(N-1)$, $n_j = |f^{-1}(j)|$, and $S_y = |\{i : g(f(i)) = y\}|$. Summing over intermediate values gives

$$\Pr(Y = y, h \mid f, g) = \frac{\beta S_y + (\alpha - \beta) n_h \mathbf{1}[y = g(h)]}{N}. \tag{15}$$

For $\alpha < 1$ and positive evidence, define $R = \alpha(N-1)/(1-\alpha)$ and normalize to obtain

$$p(y \mid f, g, h) = \frac{S_y + (R-1) n_h \mathbf{1}[y = g(h)]}{N + (R-1) n_h}. \tag{16}$$

This calculation aggregates all intermediate values with the same output before choosing a most likely output. For $n_h > 0$, the suggested output $g(h)$ is the unique mode precisely when

$$R > 1 + \frac{\max_{y \neq g(h)} S_y - S_{g(h)}}{n_h}. \tag{17}$$

Equality gives a tie. At $N = 8, \alpha = 3/4$, $R = 21$ and one match suffices for a unique suggested-output mode. At $\alpha = 1/N$ the suggestion is uninformative. At $\alpha = 1, n_h = 0$ or $\alpha = 0, n_h = N$, the observed evidence has zero probability, so a conditional distribution is undefined.

Thus, ignoring the queried entity can make a locally valid but irrelevant match favor the suggested answer. This differs from inference using the full input, where x determines the correct output $g(f(x))$.

The initially failing g-first model agrees with the posterior's set of most likely answers on 61.77% of 4,096 no-witness examples. Mean total-variation distance is 0.349 and mean probability on classes assigned zero by the candidate is 16.79%. Agreement is 57.48% on 3,083 unique-mode examples and 74.83% on 1,013 tied examples. In a separate sweep over all eight queries for each of 512 fixed (f, g, h) triples, the predicted answer varies with the query in 126 triples. The tested model retains sensitivity to the query and differs from this query-averaged posterior.

G Query information and attention interactions

We first use an idealized Bayesian observer to illustrate how incomplete query information can make answer-preserving edits predictive. We then derive an exact decomposition of the crossed attention interventions and use it to isolate the contribution removed by the additional intervention in Section 4.

G.1 Single-edit sensitivity with a query cue

The full prompt determines the answer; the observer below instead has access to a partial cue about the queried entity.

Observer and suggestion channel. Condition on a two-step task with a suggestion present. Let $N \geq 3$, sample $f, g : [N] \to [N]$ independently and uniformly, and let X be the independent, uniform queried entity, with answer $Y = g(f(X))$. The suggestion h equals $f(X)$ with probability α, and is otherwise uniform on the other $N-1$ symbols. Assume $1/N < \alpha < 1$. The observer receives only (B, f, g, h), where cue B is generated from X independently of the tables and suggestion conditional on X. Fix a cue b of positive probability and write

$$\eta_i = \Pr(X = i \mid B = b), \qquad z_i = g(f(i)), \qquad \lambda = \frac{N\alpha - 1}{1 - \alpha}.$$

For each output label y, set $c_y = \sum_i \eta_i \mathbf{1}\{z_i = y\}$ and $\omega_h = \sum_i \eta_i \mathbf{1}\{f(i) = h\}$. Set $y_h = g(h)$ and $\beta = (1-\alpha)/(N-1)$. Conditional independence gives $\Pr(X = i \mid b, f, g) = \eta_i$, while query i has likelihood $\beta[1 + \lambda \mathbf{1}\{f(i) = h\}]$ for h. Summing weighted likelihoods and normalizing therefore gives

$$p_\eta(y; f) := \Pr(Y = y \mid b, f, g, h) = \frac{c_y + \lambda \omega_h \mathbf{1}\{y = y_h\}}{1 + \lambda \omega_h}. \tag{18}$$

This posterior uses the task's suggestion channel, before the evaluator conditions on an off-row query and a wrong suggested answer. An uninformative cue gives $\eta_i = 1/N$, $c_y = S_y/N$, and $\omega_h = n_h/N$; with $\lambda = R - 1$, this recovers Equation (16).

Theorem G.1 (Sharp single-edit sensitivity)**.** *For an equal-function edit $f(j) = r \mapsto \bar{f}(j) = h$, with $r \neq h$ and $g(r) = g(h)$,*

$$p_\eta(y_h; \bar{f}) - p_\eta(y_h; f) = \frac{\lambda \eta_j (1 - c_{y_h})}{(1 + \lambda \omega_h)(1 + \lambda(\omega_h + \eta_j))}. \tag{19}$$

For fixed η, the largest increase over all tables, suggestions, and such edits admitting an off-row query $x \neq j$ with $\eta_x > 0$ and $z_x \neq y_h$ is

$$\mathcal{S}_\lambda(\eta) = \max_j \frac{\lambda \eta_j (1 - \eta_j)}{1 + \lambda \eta_j}. \tag{20}$$

It is zero exactly when η is a point mass. The bound is attained with no other first-step row using either intermediate value r or h.

Proof. The edit fixes all z_i and c_y and increases ω_h by η_j. Subtracting (18) gives (19). Since $c_{y_h} \geq \eta_j$ and $\omega_h \geq 0$, this is at most $\lambda \eta_j (1 - \eta_j)/(1 + \lambda \eta_j)$. For equality, choose distinct intermediate symbols r, h, u, set $g(r) = g(h) = y_h$ and $g(u) = y_\star \neq y_h$, and let $f(j) = r$ and every other row map to u. Then $c_{y_h} = \eta_j$ and $\omega_h = 0$. When $0 < \eta_j < 1$, a supported off-row query has answer $y_\star$, so the pair is admissible. At a point mass, choosing j outside its support attains zero. Maximizing over j proves the claim; all terms vanish exactly at the simplex vertices. □

For $N = 8$, $\alpha = 3/4$, and uniform η, the largest change in posterior probability is $5/8$. The edit has zero posterior effect at $\alpha = 1/N$; the endpoint $\alpha = 1$ is excluded because a no-witness control has zero suggestion likelihood.

Within this posterior family, vanishing worst-case sensitivity therefore characterizes complete query information. The cue restriction is essential: a table-dependent cue $B = Y$ could instead supply the correct answer without identifying X.

G.2 Crossed attention interactions and mixed-term removal

To locate the interaction measured in Section 4, we analyze the attention operation under the crossed patches. Fix the final-position query, recipient residual, and all unpatched projections. In head a, let $\gamma_i^{(a)}$ be the attention weight. Replacing only the key at row j gives weights $\gamma_i^{(a)\prime}$; independently, replace its value by $v_j^{(a)} + \Delta v_j^{(a)}$. Let $W_O^{(a)}$ be the corresponding head block of the output projection, acting on row vectors as in Appendix E. Denote the projected final-position attention outputs for neither patch, K alone, V alone, and both by T_0, T_K, T_V, T_{KV}.

Proposition G.2 (Crossed difference and mixed-term removal)**.** *The attention mixed term is exactly*

$$C := T_{KV} - T_K - T_V + T_0 = \sum_a (\gamma_j^{(a)\prime} - \gamma_j^{(a)}) \Delta v_j^{(a)} W_O^{(a)}. \tag{21}$$

Define $T_{\mathrm{add}} = T_K + T_V - T_0$. Let $\Phi(T)$ be any scalar outcome after adding the fixed residual to T and running the remaining network, including an argmax-class indicator. Then

$$\begin{aligned} I_{\mathrm{total}} &:= \Phi(T_{KV}) - \Phi(T_K) - \Phi(T_V) + \Phi(T_0) \\ &= \underbrace{\Phi(T_{KV}) - \Phi(T_{\mathrm{add}})}_{I_{\mathrm{mixed}}} + \underbrace{\Phi(T_{\mathrm{add}}) - \Phi(T_K) - \Phi(T_V) + \Phi(T_0)}_{I_{\mathrm{suffix}}}. \end{aligned} \tag{22}$$

Here I_{mixed} is the contribution removed by replacing T_{KV} with T_{add} while retaining both individual patch effects and the fixed residual.

Proof. For the original keys, changing the value adds $\sum_a \gamma_j^{(a)} \Delta v_j^{(a)} W_O^{(a)}$; for patched keys, it adds $\sum_a \gamma_j^{(a)\prime} \Delta v_j^{(a)} W_O^{(a)}$. Subtracting gives (21), hence $T_{KV} = T_{\mathrm{add}} + C$. The fixed output bias cancels in these differences. Adding and subtracting $\Phi(T_{\mathrm{add}})$ proves (22). □

Output interaction and suffix nonlinearity. Output interaction alone does not locate the interaction in attention. Scalar states $T_0 = 0$, $T_K = T_V = 1$, and $T_{KV} = 2$ have $C = 0$, but $\Phi(t) = t^2$ gives $I_{\mathrm{total}} = 2$. Additive effects from different heads can thus interact downstream. Conversely, $C \neq 0$ need not change an argmax, and $C = 0$ for a donor can result from head cancellation or zero value changes.

Let $D_K = T_K - T_0$, $D_V = T_V - T_0$, and $\psi(s,t) = \Phi(T_0 + sD_K + tD_V)$. If Φ is continuously twice differentiable near the parallelogram $\{T_0 + sD_K + tD_V : 0 \le s,t \le 1\}$, then $I_{\mathrm{suffix}} = \psi(1,1) - \psi(1,0) - \psi(0,1) + \psi(0,0)$. Integrating $\partial_s \partial_t \psi$ over the unit square gives

$$I_{\mathrm{suffix}} = \int_0^1 \int_0^1 \nabla^2 \Phi(T_0 + sD_K + tD_V)[D_K, D_V]\, ds\, dt. \tag{23}$$

The Hessian acts bilinearly on the two directions. If its operator norm is at most κ throughout this parallelogram, the integrand has absolute value at most $\kappa \|D_K\| \|D_V\|$, giving the same bound on $|I_{\mathrm{suffix}}|$. For affine Φ, $I_{\mathrm{suffix}} = 0$. For argmax indicators, the algebraic decomposition (22) applies directly.

Connection to the measured interventions. For the indicator of predicting $\bar{y}_h$, averaging (22) over pairs and multiplying by 100 gives the percentage-point interaction in Equation (12). Applying mixed-term removal to all eight main g-first, suggestion-last models at 2,000 updates, on all 2,048 crossed pairs per model, reduces $\bar{y}_h$ predictions from 31.35% under T_{KV} to 3.28% under T_{add}. The decomposition is $25.95 = 28.07 - 2.12$ percentage points. The removed contribution is positive for five seeds and zero for three. It exceeds the total interaction because the suffix contribution is negative.

T_{add} preserves both individual patch effects, although it need not be realizable by an input. Its comparison with T_{KV} measures the contribution of the attention mixed term to these patched predictions.

These results connect the equal-function diagnostic to query information and the crossed interventions to attention interactions. The following experiments examine how diagnostic sensitivity and retrieval behavior vary with training and input layout.

H Learning across reliability, layout, and task settings

At both 2,000 and 4,000 updates, all 16 query-last models achieve 100% accuracy on one- and two-step questions with absent, correct, or incorrect suggestions, and have zero equal-function effects.

H.1 Fixed positions and longer training

Fixed positions separate the effect of a suggestion from changes in the input layout when it is removed. Varying suggestion reliability tests when models learn to follow suggestions or check them against the records. Longer training tests whether false-witness effects diminish with additional updates. The reliability experiments use both table orders, the two fixed-position suffixes in Appendix C, and

$\alpha \in \{0.125, 0.75, 1\}$. Four independent initializations per setting give 48 models in total. Each trains for 4,000 updates and is evaluated at 2,000 and 4,000. The eight models with a fixed answer token and $\alpha = 0.75$ continue to 8,000 and 16,000 updates using the same optimizer and random stream.

Within a layout and seed, table orders and reliabilities share the same initialization and underlying random draws. Only the reliability threshold changes the choice of a correct or incorrect suggestion. The two suffix layouts have different vocabulary and position-embedding shapes, as specified above. We evaluate 4,096 natural examples and a separate set of 2,048 equal-function pairs. All six task/suggestion conditions and three equal-function inputs are evaluated at every checkpoint.

At 16,000 updates the f-first equal-function effects are 79.10, 75.49, 78.12, and 78.61 points; g-first effects are 72.75, 61.47, 70.26, and 78.76. No-suggestion accuracies range from 39.60% to 42.65%. On the equal-function control input, mean accuracies are 39.26% and 36.02%. **In the constant-answer-token layout, substantial false-witness effects remain after 16,000 updates.** The 100%-reliability experiments measure how a model trained exclusively on correct suggestions responds when that reliability changes at test time. The lower-reliability experiments include incorrect suggestions during training.

H.2 Additional sizes and parameterizations

To test which changes preserve or alter the effect, we vary model size, suggestion encoding, and direct access to the query. Initial three-seed architecture experiments at 2,000 updates include model width 80 (MLP widths $80 \to 160 \to 80$), three layers, a separate suggestion vocabulary, and a direct learned query-embedding connection to the final position. Width 80 gives equal-function effects of 7.50, 46.53, and 54.61 points. Three layers give 51.42, 56.59, and -0.10. The third model achieves high wrong-suggestion accuracy. The separate suggestion vocabulary retains large effects, whereas the query-embedding connection gives zero class-proportion effects in all three models.

To test whether the effect extends beyond the primary task and training setting, a further 15-model experiment crosses domain sizes $N = 6, 10$ with learning rates 0.001 and 0.003, and also tests $N = 8$ at 0.001. Each configuration uses three independent training runs of 4,000 updates, evaluated at 2,000 and 4,000 updates. Models with the same N and seed share initialization and training examples across learning rates. Each domain has 2,048 natural examples and 2,048 separate equal-function pairs.

For $N = 10$ and learning rate 0.001 at 4,000 updates, all three models achieve at least 99.95% no-suggestion accuracy, with 17.43–18.07% wrong-suggestion accuracy and equal-function effects of 56.25, 45.65, and 50.15 points. **The effect also appears at $N = 10$ in models with near-perfect no-suggestion accuracy.** At $N = 6$, three late checkpoints give negative effects of -3.47, -6.45, and -2.00 points, with negative mean score changes as well. Some early, lower-learning-rate models have no-suggestion accuracy as low as 35.30%. Showing both task accuracy and signed paired effects distinguishes these different learning states.

H.3 Reading changes over time

The nine primary accuracy measurements show when performance changes; the paired tests at 2,000 and 4,000 identify how the same models respond to false witnesses and vector replacements. Figures display the two kinds of observation on their actual time grids. Finer measurements of record selection around the accuracy transitions would reveal more about how the changes develop. Longer training under each access condition would characterize which strategies remain stable.