A LANGUAGE MODEL'S SOURCE REPORT IS READ FROM A TRACE IN ITS KV CACHE

Anonymous authors
Paper under double-blind review

ABSTRACT

Language models can answer a question from a supplied document or from their parametric knowledge. When later asked where the answer came from, what determines their source report? We study this by intervening on the model's key–value (KV) cache after the answer is generated and before the source question is asked. On a controlled factual task with four models (7B–32B), changing or removing the document's cache never changes the report. In contrast, transplanting the question-and-answer (Q+A) cache between runs with identical question and answer tokens transfers it: every eligible report follows the donor, while the recipient's document is unchanged. On the two smaller models, steering Q+A values along directions fitted before any source question reverses 98.9–100% of reports across four document conditions at a fixed strength, while norm-matched random directions flip at most 1.1%. The components of keys and values along fitted directions carry the transplant effect: copying them alone nearly reproduces the report change; holding them fixed nearly prevents it. Applied during answer generation, the same value directions also change which source supplies the answer, though more weakly and depending on the prompt. Source reporting in this setting thus depends causally on cached state produced while answering. We do not establish general reporting accuracy or a complete interpretation of this state.

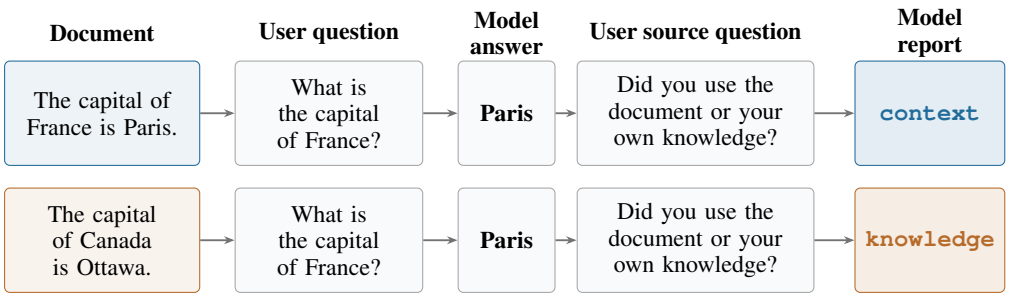

Figure 1: **Same answer, different source reports.** Each row is a separate dialogue. After answering, the model reports `context` for the document or `knowledge` for its own knowledge. We ask how this report is produced. Schematic examples; source-question wording shortened.

1 INTRODUCTION

When a language model answers a question with a document in context, a user may want to know where the answer came from. Did the model use the document, or what it already knew? This matters when the document updates or contradicts the model's knowledge. We can ask the model directly (Figure 1). Asked for France's capital, it answers "Paris" with either a supporting or an irrelevant document, but reports `context` in the first case and `knowledge` in the second. We call this reply the *source report*. Our research question is how the model produces it.

This question is distinct from asking whether the document supports the answer. A source report is a claim about *how the answer was produced*. In retrieval-augmented systems, both a retrieved

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.

document and parametric knowledge may support the same answer. Prior work on attribution evaluates whether generated content is supported by the available context (Rashkin et al., 2023); work on explanation faithfulness likewise shows that a model's verbal account of its computation need not directly expose the computation itself (Turpin et al., 2023; Lanham et al., 2023). Appendix A reviews related work in more detail. We instead ask what internal state the model uses when it produces a source report about an answer that has already been generated.

Our starting point is the transformer's key–value (KV) cache. During generation, previously processed tokens leave keys and values that later tokens can attend to. These entries contain more than the token identities themselves: the same question tokens can acquire different cached states depending on the document that was processed before them. This gives us a way to separate *what the model saw* from *the state produced by processing it*. We let the model answer normally, intervene on selected cache entries, and only then ask it about the source. This setup lets us distinguish three possible mechanisms for the source report:

H1: Re-read the document. The model may inspect the document again when asked about the source. Under this hypothesis, changing the document after answering should change the report.

H2: Judge the answer using parametric knowledge. The model may instead assess whether its answer is compatible with knowledge stored in its parameters. Under this hypothesis, the report can be determined from the answer's compatibility with parametric knowledge rather than from a record of how the answer was produced.

H3: Read a cached trace of the earlier computation. The model may use source-related information left in the cache while answering. If so, transplanting the Q+A cache between runs should transfer the report, even when the recipient's question, answer, and document stay unchanged.

Table 1: **Hypotheses for the source report and evidence from our interventions.** Blue ✓ marks evidence supporting the hypothesis; orange ✗ marks evidence against it. H1 and H2 are tested as sufficient explanations of the report. H3 represents the hypothesis that the report depends on source-related state produced while answering.

Hypothesis	Proposed mechanism	Evidence
H1: Re-read the document	The model re-inspects the document to produce the source report.	✗ **Not supported as a sufficient explanation.** Changing or removing the document after answering does not change the report (§3).
H2: Judge the answer against knowledge	The model determines the report by evaluating the answer against its parametric knowledge.	✗ **Not supported as a sufficient explanation.** Transplanting the Q+A cache changes the report while keeping the answer tokens and recipient document unchanged (§4).
H3: Read a cached trace	The report depends on source-related state produced while answering.	✓ **Supported by our interventions.** Q+A cache transplants transfer the report, and steering Q+A values changes it (§4, §5). Clamping fitted K+V projections nearly blocks transfer; transferring those projections alone nearly reproduces it (§6).

We test these mechanisms in sequence (Table 1). First, we change or remove the document's cache after the answer has been generated while preserving the Q+A cache. The report remains unchanged, even though the replacement document is still readable. This argues against determining the report from the document currently available at report time. Second, we transplant the Q+A cache between runs with identical question and answer tokens. The report follows the transplanted cache, while the recipient's document remains fixed. This establishes a causal dependence on the state produced during the earlier computation. Third, we fit linear directions in Q+A values and show that steering these values can change the source report across facts. Fitted K+V components also carry nearly the full transplant effect: transferring them nearly reproduces the report change, while holding them fixed nearly prevents it (Section 6). Finally, we apply the same fitted value directions during answer generation, without refitting. They change which source supplies the answer, linking source-related cached state to source selection even under opposing instructions.

Importantly, H1 and H2 are tested as *sufficient explanations* of the report, not as claims that document processing and parametric knowledge cannot contribute to the cached state. A successful Q+A transplant does not imply that the trace represents a single discrete "source label"; it shows

that the state produced during the earlier computation contains information that causally affects the later report. Hybrid mechanisms may therefore contribute to the computation as well.

Our main contribution is therefore a mechanistic explanation of source reporting in the controlled setting: the source report depends on source-related computational state inherited from the question-answering process, rather than simply on the document that remains available when the report is requested. We call this state a *source trace*; its role in reporting leaves open whether it encodes evidence availability, actual source use, or both. The experiments do not establish general reliability of source reports, unrestricted introspection, or a complete semantic interpretation of this state.

2 SETUP: HOLD THE WORDS FIXED, CHANGE THEIR CACHED STATE

Facts and documents. We start from 155 candidate facts across eight relations: capital, continent, official language, chemical symbol, author, painter, composer, and inventor. Each fact has seven one-sentence documents: a true fact, two counterfactuals, two irrelevant facts, and two neutral sentences. Document variants have equal token counts within each item under each model's tokenizer. Every later token therefore sits at the same position in each run of that fact, so cache entries can be copied between runs without padding.

Models. We test OLMo-3-7B and OLMo-3.1-32B (Team Olmo, 2025), and Qwen3-8B and Qwen3-32B (Yang et al., 2025), with Qwen's thinking disabled. Every dialogue is shorter than 300 tokens and fits each model's attention window. Steering experiments use the two smaller models.

One question, different routes to an answer (schematic)

Question (every run): What is the capital of France?

Type	Document	Answer	Intact report
True fact	"The capital of France is Paris."	Paris	`context`
Counterfactual	"The capital of France is Tokyo."	Tokyo	`context`
Irrelevant	"The capital of Canada is Ottawa."	Paris	`knowledge`
Neutral	"Snow melted slowly on the hills."	Paris	`knowledge`

Source question: "What is the source of the answer you just gave: the provided Document, or your own knowledge? Answer 'context' for the Document or 'knowledge' for your own knowledge. One word only."

Dialogue and source report. The model sees a document and a question, and generates its answer greedily. In Sections 3–6, the LLM is instructed to use a relevant document even when it conflicts with its knowledge, and otherwise to answer from knowledge. The fixed-document follow-up (Appendix C.8) and the experiment in Section 7 vary this instruction explicitly. After the answer, we ask the source question. We call the model's reply the *source report*: `context` for the document, `knowledge` for its own knowledge. The report is the label with the higher summed token log-probability. This agrees with the model's freely generated one-word reply in 99.6–100% of cases.

Eligible runs. The *intact report* is the report of an unedited run. We keep only *eligible* runs, whose answer and intact report match those expected for their document type (above), using rules fixed before the experiments. Eligibility is 92–100% by model and condition. The exception is irrelevant-fact runs on OLMo-3-7B (81–82%, mainly refusals). This filter ensures that every edited run starts from a known answer and report, so that a change can be attributed to the edit. Reported success rates are therefore conditional on eligibility.

Intervention timing. For every token and layer the KV cache stores the key and value vectors that later tokens attend to. We divide the dialogue into spans. The *document* span is the document tokens. The *question span* (Q) contains the question and the template tokens between the document and the answer. The *answer span* (A) contains the generated answer tokens. We edit the cache *after* the answer is generated and *before* the source question is processed. We use two types of edits:

(1) A *drop* removes the document's cache entries. Later tokens keep their original position indices.

(2) A *swap* (or *transplant*) replaces the keys and values of a span with those from another run of the same fact that saw a different document. We write "recipient ← donor": the recipient run receives the donor's cache. Swapping the answer span requires identical answer tokens in both runs. Our main transplant replaces Q and A together (Q+A). Entries we do not edit are kept, not recomputed. A token's text thus stays the same while its cached state can come from a different processing history.

Outcomes and rates. A *flip* is a change of the source report to the other label. A *source switch* (Section 7) is a change of the answer to the other source's value, e.g., from "Tokyo" to "Paris". Rates in Sections 3–4 are over eligible facts. Steering directions are fitted on some relations and evaluated on others. Report-steering rates pool two retained splits (Appendix C.5); steering rates are over *item–split pairs*, not independent facts. The examples are schematic illustrations, not quoted transcripts. Detailed rates appear in Appendix C.1 onward.

3 Changing the document leaves the report in place

Under H1, the report may compare the answer with the document currently in context. If so, changing the document after the model has answered should change the report. The model reads "The capital of France is Tokyo" and answers "Tokyo". We keep its question-and-answer cache and either remove the document's entries or replace them with an irrelevant fact. Only then do we ask the source question (Figure 2).

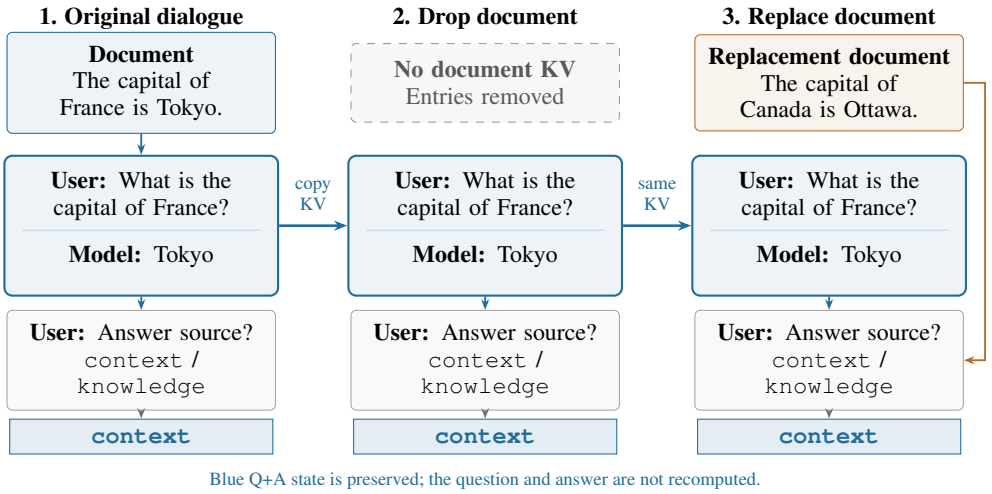

Figure 2: **The document changes; the source report does not.** The LLM answers "Tokyo" with a counterfactual document. Each edited branch keeps the original question-and-answer cache, and either removes the document entries or replaces them with an irrelevant fact, which remains readable. All three branches then report `context` to the source question, and the answer shown was generated before either edit. Schematic example; Table 2 gives results for all four models.

Table 2: **Zero source-report flips after document edits.** Cells are report-flip rates (%), with Q+A state retained. The first two edits try to remove a `context` report; the third tries to create one.

Document edit	OLMo-3-7B	Qwen3-8B	OLMo-3.1-32B	Qwen3-32B
Drop counterfactual	0%	0%	0%	0%
Counterfactual → irrelevant	0%	0%	0%	0%
Irrelevant → true fact	0%	0%	0%	0%

In this setting, neither edit flips a report on any model. The edit is available to the model: in every eligible run with a replaced document, the model can quote the new document. The reverse test leads

to the same conclusion. After the model answers "Paris" from memory, inserting a true fact stating "Paris" into the document never turns `knowledge` into `context`. The report therefore does not track whether the current document supports the answer. DynamicQA Wikipedia paragraphs show the same pattern, with rare exceptions (Appendix E). Table 7 gives the full templated-task results.

4 THE REPORT FOLLOWS THE QUESTION-AND-ANSWER CACHE

The previous experiment showed that the report does not change when we replace the document after the answer has been generated. This suggests that the information used for reporting may be retained in the model's internal state from the earlier question-answering computation. We test this by transplanting the KV cache of the question and answer (*Q+A cache*) between runs.

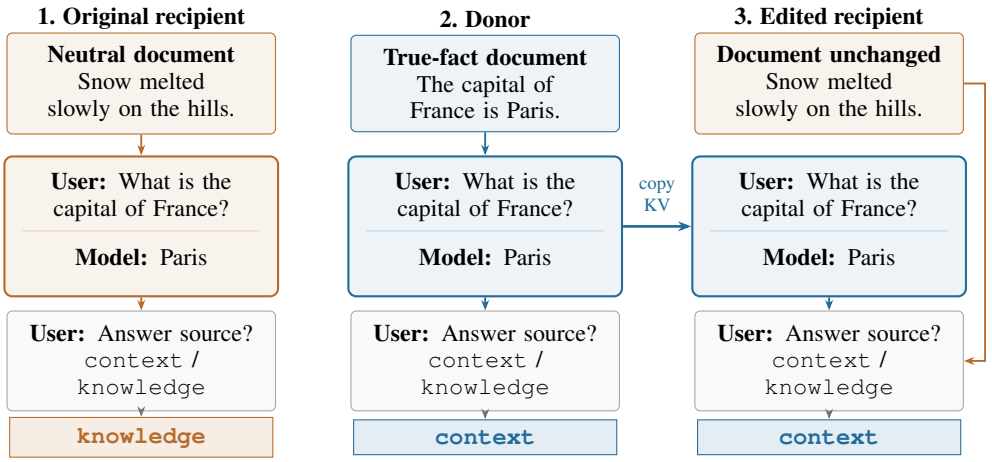

Figure 3: **Source report follows the transplanted cache.** Both runs answer "Paris", but report different sources. We copy the donor's Q+A cache into the recipient after answering, retaining the recipient's document. Asked anew, the source question now elicits `context`. Colours identify the origin of cached state; the original question and its answer are not regenerated.

We pair runs with identical question and answer tokens but different documents. For example, both runs may answer "Paris", one with a supporting document and a `context` report, the other with a neutral document and a `knowledge` report (Figure 3). We copy the Q+A cache from the *donor* run into the *recipient*; the recipient's document and all other cache entries remain unchanged.

After the transplant, the report consistently follows the donor's cache across all four tested models (Table 3). For example, replacing the Q+A cache of a neutral or irrelevant-document run with the cache from a supporting-document run changes the recipient's report to `context`, even though the recipient's original document is not relevant. Conversely, transplanting the Q+A cache from a neutral or irrelevant-document run causes a recipient that still contains supporting evidence to report `knowledge`. Control transplants, which preserve the source condition, produce no report flips. On DynamicQA Wikipedia paragraphs, joint Q+A transplants similarly produce 97.2–100% donor-following (Appendix E).

The intervention changes the report while leaving the recipient's document and answer unchanged. Thus, the report cannot be determined solely by whether the recipient's document supports the answer, because that document is identical before and after the transplant. Likewise, the answer itself is identical, so the report cannot be explained simply by the answer string or by a difference in the model's final answer. Instead, the intervention changes only the Q+A cache, indicating that source-related information carried in this state contributes to the report.

Why transplant the question and answer together? The source-related information may be distributed across the computation of both the question and the answer. We therefore transplant

Table 3: **Every eligible report follows the Q+A donor.** Cells are report-flip rates (%) among eligible facts with identical answer tokens. The document stays unchanged; only Q+A KV entries are replaced.

Recipient ← Q+A donor	OLMo-3-7B	Qwen3-8B	OLMo-3.1-32B	Qwen3-32B
True fact ← neutral	100%	100%	100%	100%
Neutral ← true fact	100%	100%	100%	100%
True fact ← irrelevant	100%	100%	100%	100%
Irrelevant ← true fact	100%	100%	100%	100%

Re-splicing the recipient's own cache or using a same-condition donor: zero flips on every model.

their KV entries jointly, which produces a consistent effect across conditions. Results for separate question-only and answer-only transplants are reported in Appendices C.2 and C.4.

The transplant experiment therefore provides causal evidence that the later source report depends on information encoded in the Q+A cache. We next ask whether this source-related component can be directly manipulated by a shared intervention in the cache.

5 Steering question-and-answer values changes the report

We next ask whether the source-related state identified by the transplant experiment can be directly manipulated. We construct directions in Q+A value space from the difference between runs with supporting and irrelevant documents.

For each model and relation split, we fit directions on one set of relations and evaluate them on the remaining relations. Fitting examples pair a *supporting-fact* document that supports the answer with an *irrelevant-fact* document that does not; answer-span fitting requires identical answer tokens. At each layer, we average the value vectors over the positions of the question or answer span and compute the mean difference between the two conditions:

$$d_\ell^s = \underset{i \in \mathcal{F}_s}{\text{mean}} \left(\overline{V}_{\ell,i}^{\text{true},s} - \overline{V}_{\ell,i}^{\text{irrelevant},s} \right), \quad V'_{\ell,t} = V_{\ell,t} \pm \alpha d_\ell^{s(t)}, \quad s \in \{Q, A\}. \tag{1}$$

Here, s specifies whether the intervention is applied to the question or answer span, and $\mathcal{F}_s$ is the corresponding fitting set. Values carry the information retrieved by attention; the key/value ablations motivating this choice are in Appendix C.3. Fitting uses states recorded before any source question. We then intervene on the Q+A values after the model has already produced its answer, leaving the keys and all other cache entries unchanged. Adding the direction moves the Q+A state toward the supporting-document condition; subtracting it moves the state in the opposite direction. At $\alpha = 1$, the intervention has the magnitude of the mean difference between the two fitting conditions.

As a control, we apply random directions with the same layerwise norms and the same intervention locations and strengths. Each random condition is averaged over three draws.

Changing the report without changing the answer (schematic)

Completed dialogue: the model has answered "Tokyo" to a question.
Intervention: after the answer is generated, subtract the learned source direction from the Q+A values.
Result: the source report changes from `context` to `knowledge`, while the previously generated answer remains "Tokyo".

The learned direction reliably changes the subsequent source report in both directions (Figure 4). With an irrelevant document, adding the direction produces a `context` report in 100% of evaluations at $\alpha = 2$ on both models. Conversely, with a counterfactual document, subtracting the direction removes the `context` report in 100% of evaluations by $\alpha = 3$ on both models. In all of these interventions, the answer tokens were generated before the edit and remain fixed while the subsequent source report changes.

The effect also generalises to neutral and true-fact documents. Across all four document conditions, $\alpha = 3$ flips 100% of reports except on OLMo true facts (98.9%). At $\alpha = 1$–3, random controls

produce no flips on Qwen and at most 1.1% on OLMo. Additional results, including the comparison of joint and separate-span interventions, are in Appendices C.5–C.6.

These results show that the Q+A cache contains a component that can be accessed by a shared linear intervention to change the source report. They do not imply that the entire source-related state is one-dimensional: the learned direction captures a manipulable component of the natural Q+A state.

For irrelevant ($\alpha = 2$) and counterfactual documents ($\alpha = 3$), every report flips while answer repetition and document quotation both succeed. Both models also retrieve the counterfactual value in every case. These checks use separate continuations of the same edited cache (Appendix C.7).

Interpretation of the contrast. This contrast does not separate evidence availability from source use. A fixed-document contrast also supports report steering (Appendix C.8), but neither contrast establishes a pure use signal. The intervention therefore establishes that the state produced during answering is sufficient to change the later source report.

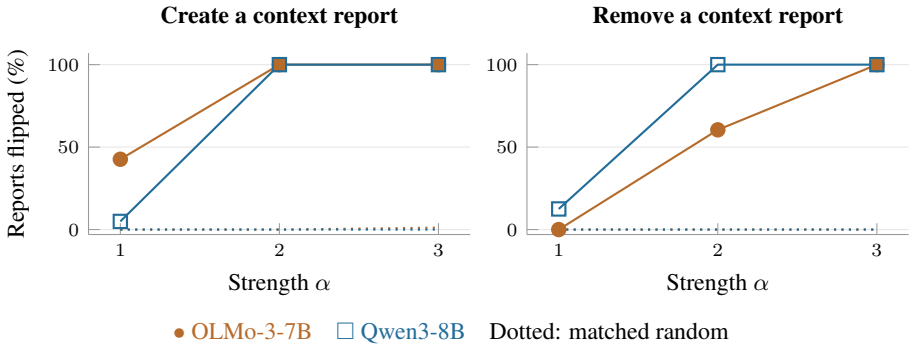

Figure 4: **Steering Q+A values changes the source report.** Left: addition makes irrelevant-document runs report `context`; both reach 100% at $\alpha = 2$. Right: subtraction makes counterfactual runs report `knowledge`; both reach 100% by $\alpha = 3$. The answer tokens remain fixed. Dotted curves show mean random-control rates over three draws, at most 1.1% on the left and zero on the right. Rates are over eligible item–split pairs. Other document conditions and span ablations appear in Tables 11–12.

6 Fitted directions capture the report-transfer effect

Copying the Q+A cache makes the source report follow the donor (Section 4). Steering also changes the report (Section 5). Do the fitted directions capture the information that makes the transplant work? We pair true-fact and irrelevant-fact runs with identical Q+A tokens (Figure 5). We reuse the value directions from Section 5 and fit key directions from the same contrast. All edits cover Q+A at every layer, after answering and before asking for the source. The document and all dialogue tokens stay fixed.

Are these components needed for transfer? Copy the donor's full Q+A cache, then restore its projections onto the fitted directions to those of the recipient. For a value vector, this *clamp* is

$$v_{\text{clamp}} = \underbrace{v_D}_{\text{copy donor}} - \underbrace{\langle v_D, u\rangle u}_{\text{remove donor projection}} + \underbrace{\langle v_R, u\rangle u}_{\text{restore recipient projection}} . \tag{2}$$

Here v_D and v_R are the original donor and recipient value vectors at the same layer and position; u is the unit direction fitted for that layer and span. The projection $\langle v, u\rangle u$ is the component of v along u, with $\langle \cdot, \cdot \rangle$ denoting the dot product. Keys use a norm-preserving rotation instead (Appendix D). Joint K+V clamps reduce flips from 100% to **1.1%** on OLMo and from 100% to **0%** on Qwen.

Is transferring these projections enough? Start from the recipient's cache and impose the donor's projections: for values, exchange D and R on the right of Equation 2. Without amplification, this flips **97.9% of OLMo and 100% of Qwen reports**, nearly reproducing the full transplant.

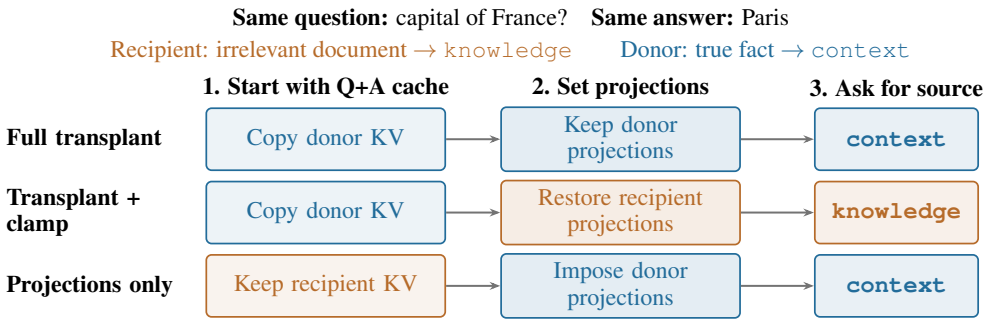

Figure 5: **Which part of the transplanted cache carries the report?** Each row independently edits the same recipient's Q+A cache. Projections are components along the fitted directions. Schematic example; Table 4 gives measured rates in both directions.

Table 4: **Projections onto fitted directions block and reproduce report transfer.** Report-flip rates (%) over directional evaluations for true-fact ↔ irrelevant-fact pairs, pooled over two held-out relation splits. Every row tests the same pairs in both directions.

Intervention	OLMo-3-7B	Qwen3-8B
Full Q+A KV transplant	100%	100%
+ restore recipient K+V projections	**1.1%**	**0%**
+ clamp energy-matched subspace	100%	94.7%
No transplant: donor K+V projections only	**97.9%**	**100%**

Nearly necessary and sufficient for report transfer. By contrast, clamping a control subspace with comparable donor–recipient difference energy on fitting data leaves 100% and 94.7% of reports flipped (Table 4). On these pairs, the fitted K+V components therefore largely account for the full transplant's effect on the report. Values-only tests and behavioural-check limits are in Appendix D.

7 Steering values during answering changes the chosen source

So far we edited the cache only after the answer was complete, which can change the report but not the answer. We now ask whether the same value-steering direction also affects which source the model answers from. To test this, we steer values while the model reads the question and generates its answer. This readout needs no source question. Given the document "The capital of France is Tokyo", the answer "Tokyo" comes from the document and "Paris" from memory.

Move the intervention before the answer is complete (schematic)

Document: "The capital of France is Tokyo." **Question:** capital of France?
Edit during answering: steer values as question and answer tokens are processed.
Readout: the answer switches between "Tokyo" and "Paris".

We reuse the fitted value direction of Equation 1 (true-fact minus irrelevant-fact values), without refitting for the instructions tested here. We apply it at $\alpha = 2$ at every layer, first to the question tokens and then to each generated answer token. To make the test stringent, the instruction always opposes the edit. We steer toward memory when the model is told to use the document, and toward the document when it is told the document may contain errors. Recipients are runs that initially answered from the source we steer away from.

Steering overrides the instruction in both directions on both models (Figure 6). Toward memory, it switches 71% of OLMo and 51% of Qwen answers; toward the provided context, 85% and 73%. In these two conditions, random directions of equal norm switch at most 4.1% of answers, except for OLMo toward the context (24%), where the fitted direction still switches 3.6 times as many. At most 9.3% of answers are damaged. Residual-stream steering is weaker than value steering on OLMo and stronger on Qwen (Appendix C.9). The same fitted value direction therefore changes

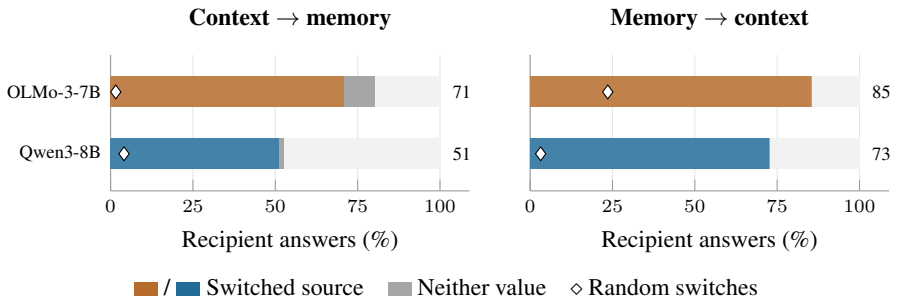

Figure 6: **The value intervention also changes the answer's source.** Value steering at $\alpha = 2$. Left: instructed to use documents, but steering favours memory. Right: the instruction favours knowledge when the document may be wrong; steering favours the document. Coloured bars are switches; grey are answers containing neither value; diamonds are mean random switches over three draws. Switch percentages are on the right. Rates are over eligible item–split pairs. The residual-stream comparison is in Table 15.

both the later source report and which source supplies the answer, even when the instruction opposes the intervention.

8 Discussion and Limitations

Together, the interventions give a consistent account of source reporting in this setting. Editing the document after answering leaves the report unchanged. Transplanting the question-and-answer cache moves the report while every recipient token stays fixed. A shared value direction in that cache can change the report. K+V projections onto fitted directions also nearly block or reproduce report transfer on true-fact–irrelevant-fact pairs. Applying the same value direction during answering changes which source supplies the answer. The report therefore depends on state computed before the source question is asked. These findings concern eligible runs and one source-question wording. Document edits and transplants extend to selected DynamicQA items; steering and projection tests use only 7–8B models. General attribution reliability in natural retrieval settings remains untested. Three questions remain open:

Meaning and formation of the directions. Our interventions establish a causal role in reporting without resolving whether the trace represents evidence availability, actual source use, or both. The fixed-document comparison leaves this distinction open (Appendix C.8). Future work should determine what these directions encode and how they form during computation. H1 and H2 fail as complete accounts, but hybrid mechanisms may contribute.

Roles of keys and values. Joint K+V edits nearly block or reproduce report transfer. Value-only clamps are less effective when keys are also transplanted (Appendix D). Explaining how keys and values contribute separately, and how they interact, is a next step. Answer repetition and document read-back remain untested for these joint K+V edits.

Roles of question and answer states. Our main interventions edit Q+A together. Separate-span ablations show that their contributions vary across models and conditions (Appendices C.4 and C.6). Future work should distinguish what is established while processing the question from what is added during answering. The fitted directions span layers and Q/A positions; they do not define one global source variable.

Reproducibility Statement

The appendix reports the joint steering sweep, compact span and fitting-contrast comparisons, controls and the mapping from experiments to scripts and result artifacts. Full sweeps, random controls, behavioural checks and the fitting protocol are preserved in the accompanying data.

AI USE STATEMENT

We used generative AI tools to assist with drafting and editing the manuscript and with writing code. The authors take responsibility for the interpretation of the results, the scientific conclusions, and the final content of this paper.

REFERENCES

Amos Azaria and Tom Mitchell. The internal state of an LLM knows when it's lying. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pp. 967–976, 2023. doi: 10.18653/v1/2023.findings-emnlp.68. URL https://aclanthology.org/2023.findings-emnlp.68/.

Max Belitsky, Dawid J. Kopiczko, Michael Dorkenwald, M. Jehanzeb Mirza, James R. Glass, Cees G. M. Snoek, and Yuki M. Asano. KV cache steering for controlling frozen LLMs. *arXiv preprint arXiv:2507.08799*, 2025. URL https://arxiv.org/abs/2507.08799.

Felix Jedidja Binder, James Chua, Tomek Korbak, Henry Sleight, John Hughes, Robert Long, Ethan Perez, Miles Turpin, and Owain Evans. Looking inward: Language models can learn about themselves by introspection. In *International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/0a6059857ae5c82ea9726ee9282a7145-Abstract-Conference.html.

Benjamin Cohen-Wang, Harshay Shah, Kristian Georgiev, and Aleksander Madry. ContextCite: Attributing model generation to context. In *Advances in Neural Information Processing Systems*, volume 37, pp. 95764–95807, 2024. doi: 10.52202/079017-3035. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/adbea136219b64db96a9941e4249a857-Abstract-Conference.html.

Zhuoran Jin, Pengfei Cao, Hongbang Yuan, Yubo Chen, Jiexin Xu, Huaijun Li, Xiaojian Jiang, Kang Liu, and Jun Zhao. Cutting off the head ends the conflict: A mechanism for interpreting and mitigating knowledge conflicts in language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 1193–1215, 2024. doi: 10.18653/v1/2024.findings-acl.70. URL https://aclanthology.org/2024.findings-acl.70/.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. URL https://arxiv.org/abs/2207.05221.

Tamera Lanham, Anna Chen, Ansh Radhakrishnan, Benoit Steiner, Carson Denison, Danny Hernandez, Dustin Li, Esin Durmus, Evan Hubinger, Jackson Kernion, et al. Measuring faithfulness in chain-of-thought reasoning. *arXiv preprint arXiv:2307.13702*, 2023. URL https://arxiv.org/abs/2307.13702.

Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36, pp. 41451–41530, 2023. doi: 10.52202/075280-1797. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/81b8390039b7302c909cb769f8b6cd93-Abstract-Conference.html.

Stephanie Lin, Jacob Hilton, and Owain Evans. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*, 2022. URL https://openreview.net/forum?id=8s8K2UZGTZ.

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

Shayne Longpre, Kartik Perisetla, Anthony Chen, Nikhil Ramesh, Chris DuBois, and Sameer Singh. Entity-based knowledge conflicts in question answering. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pp. 7052–7063, 2021. doi: 10.18653/

v1/2021.emnlp-main.565. URL https://aclanthology.org/2021.emnlp-main.565/.

Sara Vera Marjanovic, Haeun Yu, Pepa Atanasova, Maria Maistro, Christina Lioma, and Isabelle Augenstein. DYNAMICQA: Tracing internal knowledge conflicts in language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 14346–14360. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-emnlp.838. URL https://aclanthology.org/2024.findings-emnlp.838/.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35, pp. 17359–17372, 2022. doi: 10.52202/068431-1262. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/6f1d43d5a82a37e89b0665b33bf3a182-Abstract-Conference.html.

Francesco Ortu, Zhijing Jin, Diego Doimo, Mrinmaya Sachan, Alberto Cazzaniga, and Bernhard Schölkopf. Competition of mechanisms: Tracing how language models handle facts and counterfactuals. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8420–8436, 2024. doi: 10.18653/v1/2024.acl-long.458. URL https://aclanthology.org/2024.acl-long.458/.

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. Measuring attribution in natural language generation models. *Computational Linguistics*, 49(4):777–840, 2023. doi: 10.1162/coli_a_00486. URL https://aclanthology.org/2023.cl-4.2/.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522, 2024. doi: 10.18653/v1/2024.acl-long.828. URL https://aclanthology.org/2024.acl-long.828/.

Team Olmo. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025. URL https://arxiv.org/abs/2512.13961.

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

Miles Turpin, Julian Michael, Ethan Perez, and Samuel 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, pp. 74952–74965, 2023. doi: 10.52202/075280-3275. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/ed3fea9033a80fea1376299fa7863f4a-Abstract.html.

Jesse Vig, Sebastian Gehrmann, Yonatan Belinkov, Sharon Qian, Daniel Nevo, Yaron Singer, and Stuart Shieber. Investigating gender bias in language models using causal mediation analysis. In *Advances in Neural Information Processing Systems*, volume 33, pp. 12388–12401, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/92650b2e92217715fe312e6fa7b90d82-Abstract.html.

Zhengxuan Wu, Aryaman Arora, Atticus Geiger, Zheng Wang, Jing Huang, Dan Jurafsky, Christopher D. Manning, and Christopher Potts. AxBench: Steering LLMs? even simple baselines outperform sparse autoencoders. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 67035–67080. PMLR, 2025. URL https://proceedings.mlr.press/v267/wu25a.html.

Jian Xie, Kai Zhang, Jiangjie Chen, Renze Lou, and Yu Su. Adaptive chameleon or stubborn sloth: Revealing the behavior of large language models in knowledge conflicts. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/99261adc8a6356b38bcf999bba9a26dc-Abstract-Conference.html.

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

Fred Zhang and Neel Nanda. Towards best practices of activation patching in language models: Metrics and methods. In *International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/06a52a54c8ee03cd86771136bc91eb1f-Abstract-Conference.html.

Yu Zhao, Xiaotang Du, Giwon Hong, Aryo Pradipta Gema, Alessio Devoto, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. Analysing the residual stream of language models under knowledge conflicts. In *NeurIPS 2024 Workshop on Foundation Model Interventions*, 2024. URL https://arxiv.org/abs/2410.16090.

Yu Zhao, Alessio Devoto, Giwon Hong, Xiaotang Du, Aryo Pradipta Gema, Hongru Wang, Xuanli He, Kam-Fai Wong, and Pasquale Minervini. Steering knowledge selection behaviours in LLMs via SAE-based representation engineering. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5117–5136. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.naacl-long.264. URL https://aclanthology.org/2025.naacl-long.264/.

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

A Related work

Reports about internal computation. Prior work studies uncertainty, truth-related states and the faithfulness of explanations and self-reports (Kadavath et al., 2022; Lin et al., 2022; Azaria & Mitchell, 2023; Turpin et al., 2023; Lanham et al., 2023; Binder et al., 2025). In a close precedent, Lindsey (2025) injects a concept before an artificial prefill, making Claude more likely to accept it as intentional. We study naturally generated answers and test the later source report with document edits and Q+A transplants that preserve the question and answer tokens.

Knowledge conflicts and source selection. Contextual and parametric knowledge can compete (Longpre et al., 2021; Xie et al., 2024; Ortu et al., 2024; Jin et al., 2024). Zhao et al. (2024) predict source choice from residual activations; SpARE (Zhao et al., 2025) controls it through sparse autoencoder features. Our central contribution concerns the mechanism of the subsequent source report. Section 7 connects source-related state to answer selection.

Attribution to context. ContextCite (Cohen-Wang et al., 2024) estimates contextual contributions to generation using context ablations and a sparse linear surrogate. We instead intervene after answering, preserving downstream Q+A state. Invariance of the later report therefore does not imply that the document did not contribute to generation.

Patching and steering. We build on activation patching (Vig et al., 2020; Meng et al., 2022; Zhang & Nanda, 2024), mean-difference steering (Turner et al., 2023; Zou et al., 2023; Li et al., 2023; Rimsky et al., 2024), and KV-cache steering (Belitsky et al., 2025), with matched controls to assess intervention specificity (Wu et al., 2025).

B GENERATED FACTUAL ITEMS

Each candidate is a subject–relation–object statement, rendered as "The {phrase} of {subject} is {value}." and queried with "{Who/What} is the {phrase} of {subject}?" Table 5 describes the original pool of 155 candidates. Model- and intervention-specific eligibility checks determine which items enter each reported result (Section 2).

Table 5: **The eight relations in the candidate pool.** Counts precede the length-matching and eligibility requirements of the reported experiments.

Relation	Phrase	Candidates
capital	capital	30
symbol	chemical symbol	26
author	author	22
continent	continent	21
language	official language	20
painter	painter	12
composer	composer	12
inventor	inventor	12
Total		**155**

Counterfactual construction. Counterfactuals use other attested values of the same relation. For candidate i, let `others` list that relation's distinct values excluding the true value. The counterfactual is chosen deterministically as $\texttt{others}[i \bmod |\texttt{others}|]$; the second counterfactual uses the next value in the same cycle. A capital is thus replaced by a capital and an author by an author, preserving the answer type without inventing new strings. Values need not be unique across subjects: the continent relation has six distinct values across 21 candidates.

Parametric-knowledge check. Each candidate is queried with neutral filler in place of an informative document. The check accepts an answer if it contains the true value and does not contain the counterfactual, using normalised string matching over at most 12 greedily decoded tokens. It is applied per model, so eligible item sets may differ. This tests recall of the factual answer; the additional answer and intact-report requirements for each intervention are specified in Section 2.

Length matching. The seven document variants described in Appendix C.1, including neutral fillers, have equal token counts within an item under each model's tokenizer. This keeps the later question and answer positions aligned across conditions, allowing cache entries to be exchanged without a positional shift. Length matching is an additional constraint on the candidate inventory, separate from the model's ability to recall a fact.

Relation-based splits. Directions are fitted on DEV relations and tested on held-out relations. Two candidate splits exchange the groups {capital, continent, language, symbol} and {author, painter, composer, inventor}; the third uses a mixed DEV group. The exact memberships and retained splits for the reported steering experiments are given in Appendix C.5. Thus, evaluation relations are absent from the corresponding fitting set, while the extent of domain transfer depends on the split.

C ADDITIONAL RESULTS AND DETAILS

C.1 DOCUMENTS AND DOCUMENT INTERVENTIONS

The examples in Table 6 illustrate the conditions. Each item has seven documents: one true fact and two variants of each of the other conditions. Exact token-length equality is enforced within each item in the experimental dataset. Appendix B describes the candidate facts and their construction.

Table 6: The seven document variants for the question *What is the capital of France?* Each control variant is a second document of the same condition.

document	example	control variant	role
true fact	The capital of France is Paris.	—	relevant, true value
counterfactual	The capital of France is Tokyo.	… is Lima.	relevant, false value
irrelevant fact	The capital of Canada is Ottawa.	… of Egypt is Cairo.	true fact, other subject
neutral sentence	Snow melted slowly on the hills.	The library was open until late.	no fact at all

Table 7: Document edits leave the report fixed; question transplants often move it. Percentages of eligible items with the stated report. Bold: the weak direction on each 32B model (Section 4).

	OLMo-3-7B	Qwen3-8B	OLMo-3.1-32B	Qwen3-32B
Counterfactual run (read "Tokyo", answered "Tokyo"): reports `context`				
intact	100%	100%	100%	100%
DOCUMENT DROP	100%	100%	100%	100%
DOCUMENT SWAP → irrelevant fact	100%	100%	100%	100%
model quotes the new document	100%	100%	100%	100%
Irrelevant-fact run (answered "Paris" from memory): reports `context`				
intact	0%	0%	0%	0%
DOCUMENT SWAP → true fact (its own answer)	0%	0%	0%	0%
DOCUMENT SWAP → counterfactual	0%	0%	0%	0%
QUESTION SWAP *(document and answer stay): report becomes …*				
counterfactual ← irrelevant-fact: `knowledge`	97.9%	100%	100%	**14.2%**
irrelevant-fact ← counterfactual: `context`	86.6%	90.6%	**6.7%**	100%
controls: report changes	0%	0%	0%	0%

Read-back caveats. Qwen3, after receiving the question KV of an irrelevant-fact run, says the document does not mention the subject when asked to read it (77.4% on Qwen3-8B); on that model the clean dissociation is the counterfactual ← neutral question-swap test (Table 8). After the document's KV is dropped, Qwen3-32B reconstructs the document from its own answer when asked to repeat it (26.2%); the DOCUMENT SWAP is the clean version of that cell.

C.2 QUESTION AND ANSWER TRANSPLANTS

Question-only tests first located a signal downstream of the document, but were not uniformly effective. Figure 7 retains that comparison. The main Q+A result uses identical-answer pairs, a different eligible subset from the counterfactual/memory pairs below.

Rates are over facts for which both runs qualify; the identical-text intervention additionally requires identical answer tokens. The read-back row measures the joint event: the report changes to `knowledge` and the model can still read the false value from the document.

Table 8: The report follows the cache even when the answer, or all visible text, is fixed. Percentages of eligible items whose report became the stated one. Bold: Qwen3-32B with a false answer, where question-only transplants are weakest.

recipient ← donor: report becomes	OLMo-3-7B	Qwen3-8B	OLMo-3.1-32B	Qwen3-32B
The same true answer "Paris"; QUESTION SWAP				
true-fact ← neutral: `knowledge`	80.2%	100%	100%	95.8%
neutral ← true-fact: `context`	97.4%	100%	73.3%	100%
A false answer "Tokyo" in plain sight; QUESTION SWAP				
counterfactual ← neutral: `knowledge`	90.4%	100%	68.3%	**25.8%**
still reads "Tokyo" in the document	83.3%	97.2%	68.3%	25.8%
neutral ← counterfactual: `context`	89.5%	97.2%	49.2%	100%
IDENTICAL-TEXT SWAP*: question and answer swapped, visible text unchanged*				
true-fact ← neutral: `knowledge`	100%	100%	100%	100%
neutral ← true-fact: `context`	100%	100%	100%	100%
true-fact ← irrelevant-fact: `knowledge`	100%	100%	100%	100%
irrelevant-fact ← true-fact: `context`	100%	100%	100%	100%
controls: report changes	0%	0%	0%	0%

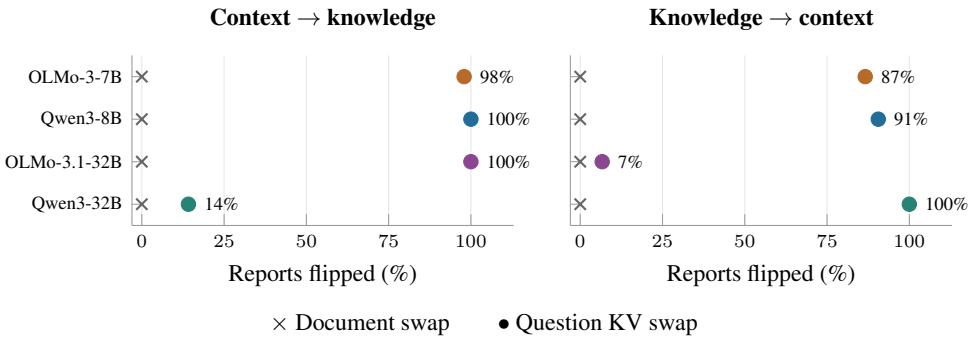

Figure 7: **The report moves with question state, not with the edited document.** Question swaps exchange counterfactual and irrelevant-fact runs; coloured dots show the fraction whose report follows the donor. Document swaps replace a counterfactual with an irrelevant fact (left), or an irrelevant fact with a true fact stating the answer (right): zero flips on every model. Document deletion also produces zero flips. Question swaps are asymmetric at 32B, motivating the answer-cache tests. Controls: Table 7.

C.3 KEYS AND VALUES IN QUESTION-ONLY TRANSPLANTS

Table 9: The question span swapped as keys+values, keys only or values only. Cells: OLMo-3-7B · Qwen3-8B, percentages of eligible items whose report flipped (remove or create the `context` report). Value-only transplants create the report in most cases; removing it is more effective with keys and values together.

swap of the question span	keys+values	keys only	values only
counterfactual ← irrelevant (remove)	97.9% · 100%	0% · 0%	6.2% · 0%
true-fact ← irrelevant (remove)	92.9% · 100%	0% · 0%	0% · 0%
irrelevant ← counterfactual (create)	86.6% · 90.6%	40.2% · 0.9%	99% · 97.2%
irrelevant ← true-fact (create)	90.9% · 98.3%	54.5% · 5.2%	100% · 100%

C.4 QUESTION AND ANSWER SPANS IN TRANSPLANTS

Using the pairs from the IDENTICAL-TEXT SWAP experiment (Section 4), we swap the question cache, the answer cache, or both. The joint result establishes sufficiency for transferring the report on these pairs; it does not establish that each span separately encodes an independent source variable.

Table 10: Percentages of eligible items whose report moved, by swapped span.

	question only	answer only	question + answer
OLMo-3-7B, Qwen3-8B, Qwen3-32B (all four directions)	75–100%	0–3%	100%
OLMo-3.1-32B, irrelevant-fact ← true-fact	30%	0%	100%
OLMo-3.1-32B, true-fact ← irrelevant-fact	100%	59%	100%

C.5 JOINT REPORT STEERING ACROSS STRENGTHS

The sweep uses the true-fact versus irrelevant-fact contrast of Section 5. Rates are over eligible item–split pairs pooled over splits 0 and 2, not independent facts. The full fitted-direction sweep and random controls are retained in the accompanying data.

Relation splits. The three candidate DEV sets are: split 0, {capital, continent, language, symbol}; split 1, {author, painter, composer, inventor}; and split 2, {capital, language, author, composer}. Evaluation uses the remaining relations. The reported Q+A sweep and the answer-only fitting-contrast comparison use splits 0 and 2. Exact prompts and fitting details are archived in `data/steering_protocol.json`.

Table 11: **Joint Q+A steering across all four document conditions.** Report-flip rates (%) over eligible item–split pairs pooled across the two retained relation splits. Every cell reaches 100% at $\alpha = 4$; at $\alpha = 3$, the only exception is OLMo on true facts (98.9%). Bold marks complete flipping. Random means are given below.

Document / direction	Model	$\alpha = 1$	$\alpha = 2$	$\alpha = 3$	$\alpha = 4$
Irrelevant fact: create	OLMo-3-7B	42.6%	**100%**	**100%**	**100%**
	Qwen3-8B	4.9%	**100%**	**100%**	**100%**
Neutral sentence: create	OLMo-3-7B	12.8%	88.5%	**100%**	**100%**
	Qwen3-8B	0%	2.4%	**100%**	**100%**
Counterfactual: remove	OLMo-3-7B	0%	60.5%	**100%**	**100%**
	Qwen3-8B	12.5%	**100%**	**100%**	**100%**
True fact: remove	OLMo-3-7B	0%	46.6%	98.9%	**100%**
	Qwen3-8B	0%	96.6%	**100%**	**100%**

Random means: all Qwen cells are zero. For OLMo, irrelevant-fact Q+A gives 1.1% at $\alpha = 3$ and 1.6% at $\alpha = 4$; neutral Q+A gives 2.2% at $\alpha = 4$. All other joint cells are zero.

C.6 Question and answer spans in report steering

Table 12 compares Q, A and Q+A at a common strength. The joint edit is more effective in every model–condition cell, supporting the choice of Q+A in the main text. It does not imply that either span has a single exclusive function.

Table 12: **Joint editing is more effective than either span alone.** Report-flip rates (%) over eligible item–split pairs at the same strength, $\alpha = 3$, for all three edits. This descriptive comparison does not establish that Q and A encode independent source variables.

Document / direction	Model	Q only	A only	Q+A
Irrelevant fact: create	OLMo-3-7B	62.3%	67.2%	**100%**
	Qwen3-8B	97.5%	3.7%	**100%**
Neutral sentence: create	OLMo-3-7B	42.3%	62.8%	**100%**
	Qwen3-8B	8.4%	0%	**100%**
Counterfactual: remove	OLMo-3-7B	0%	72.1%	**100%**
	Qwen3-8B	58%	85.2%	**100%**
True fact: remove	OLMo-3-7B	0%	37.5%	**98.9%**
	Qwen3-8B	40.9%	23.9%	**100%**

At $\alpha = 3$, random means are zero except OLMo irrelevant-fact Q (0.5%) and Q+A (1.1%). Answer-only random controls are zero at all four strengths. Across the full sweep, the maximum random rate is 2.2%.

The distinction persists at higher strength in the counterfactual setting: answer-only steering at $\alpha = 4$ flips 86% of OLMo and 90.9% of Qwen reports, whereas Q+A flips 100% on both models. Thus, the complete effect reported in the main text belongs to the joint intervention.

C.7 Content checks after report steering

These checks use the same edited Q+A caches and eligible item–split pairs as the report-steering sweep (block 3i). The report, answer repetition, document quotation and counterfactual-value read-back are separate continuations; no evaluations are filtered after intervention. Table 13 shows the operating points used for the two main conditions.

Table 13: **Report flips with successful answer repetition at the operating points.** Joint success rates (%) over eligible item–split pairs for Q+A steering. Every evaluation flips and passes repetition; document quotation also succeeds throughout.

Document / direction	α	OLMo-3-7B	Qwen3-8B
Irrelevant fact: create	2	100%	100%
Counterfactual: remove	3	100%	100%

For answer repetition, "Repeat the answer you just gave, exactly." succeeds if the response contains the original answer. For quotation, "Repeat the Document above word for word." succeeds if the response contains the full document, ignoring case and a final period. Both checks succeed on every unedited baseline. After steering, quotation succeeds in every evaluation in Table 13. For counterfactual documents, "According to the Document above, $\langle q \rangle$ Reply with the exact word from the Document." also retrieves the false value in every case; success requires that value and no refusal such as "not mentioned". These checks establish preservation at the stated settings, not across every strength or intervention. The full supplied checks are archived in `data/report_steering_checks.json`.

C.8 SUPPORTING VERSUS IRRELEVANT AND FIXED-DOCUMENT FITTING CONTRASTS

We compare two fitting procedures while editing only answer values. The original contrast is supporting versus irrelevant documents. The fixed-document contrast changes the instruction: P_D says to use a relevant document even against prior knowledge; P_M warns that the document may be wrong and favours memory.

Construction. Let CF and IRR denote counterfactual and irrelevant documents. At layer ℓ and span s, we fit

$$d_\ell^{s,\mathrm{FD}} = \underset{i\in\mathrm{DEV}}{\mathrm{mean}}\Big[\big(\overline{V}^s_{\ell,i}(\mathrm{CF}, P_D) - \overline{V}^s_{\ell,i}(\mathrm{CF}, P_M)\big) - \big(\overline{V}^s_{\ell,i}(\mathrm{IRR}, P_D) - \overline{V}^s_{\ell,i}(\mathrm{IRR}, P_M)\big)\Big]. \quad (3)$$

Here $\overline{V}$ averages values over span positions, retaining all KV heads. The second difference subtracts the prompt contrast measured with an irrelevant document. All four answers are generated greedily, up to eight tokens, and their caches are rebuilt from the prompt and generated answer. CF answers differ across prompts; IRR answers need not match either. Each answer is averaged over its own span, so answer content can enter this contrast. The original answer direction instead requires identical answer tokens.

Fitting and evaluation. We retain items that use CF under P_D and memory under P_M, and memory with IRR under both prompts, with matching source reports: 48.4% of OLMo and 77.9% of Qwen items. Both directions require at least ten DEV pairs. The answer-only comparison uses splits 0 and 2 (Appendix C.5); split 1 fails the DEV-pair minimum for the fixed-document contrast on OLMo and the original contrast on Qwen. Recipients are eligible P_D runs; edits cover answer values at every layer after generation. Exact prompts and selection rules are in `data/steering_protocol.json`.

Table 14: **Two fitting contrasts, the same answer-only intervention.** Report-flip rates (%) over eligible evaluations at $\alpha = 3$; remove/create refer to `context`. Supporting versus irrelevant is the original contrast; fixed document uses a prompt correction. This is a descriptive comparison at a common multiplier, not at matched edit norms or a separately optimised strength.

Document / direction	Model	Supporting vs. irrelevant	Fixed document
True fact: remove	OLMo-3-7B	37.5%	30.7%
	Qwen3-8B	23.9%	77.3%
Counterfactual: remove	OLMo-3-7B	72.1%	72.1%
	Qwen3-8B	85.2%	100%
Irrelevant fact: create	OLMo-3-7B	67.2%	37.7%
	Qwen3-8B	3.7%	2.5%
Neutral sentence: create	OLMo-3-7B	62.8%	23.1%
	Qwen3-8B	0%	0%

All norm-matched random controls are zero at every tested strength ($\alpha = 1, 2, 3, 4$), separately for each contrast. The full fitted-direction sweep is retained in the accompanying data.

The fixed-document direction removes `context` more effectively on Qwen. On OLMo, neither contrast dominates removal across the full sweep, while the original contrast creates `context` more effectively. Neither answer-only direction reliably creates `context` on Qwen. These names describe fitting conditions, not established semantic variables.

Geometry and scope. Norms are $\sqrt{\sum_\ell \|d_\ell\|^2}$ over all layers and heads, computed per split. Cosines are unweighted means of per-layer cosines, including a near-zero layer-0 contribution: 0.565 and 0.571 for OLMo, and 0.577 and 0.569 for Qwen, on splits 0 and 2, respectively. Per-split norms and full sweeps are archived with the protocol and comparison data. Answer repetition was not measured at $\alpha = 3$, so Table 14 establishes report control at that strength, not preservation of answer readout.

C.9 SOURCE-CHOICE STEERING: INSTRUCTIONS AND CONTROLS

We use the two settings in Section 7. To steer toward memory, we keep the document-priority instruction used in the preceding experiments. To steer toward the document, we instead warn that the document may contain errors and instruct the model to prefer its own knowledge. This provides recipients that initially answer from memory; otherwise, the model usually already uses the document. In both settings, recipients initially answer from the source we steer away from.

Value steering reuses the fitted directions of Section 5, at all layers and without refitting for the instruction. The residual-stream comparison adds a mean hidden-state difference fitted at question positions under a neutral instruction. It uses one layer selected on fitting relations: OLMo layers 12, 16 and 12 across the three splits, Qwen layer 20, out of 32 and 36 layers respectively. Both methods intervene during question processing and answer generation at $\alpha = 2$.

Table 15: SOURCE STEERING in the two main settings at $\alpha = 2$. All rates are percentages over eligible item–split pairs. *Switched*: the answer changes to the target source's value. *Random*: source switches under random directions, averaged over three draws. *Damaged*: the answer contains neither the document nor the memory value.

		OLMo-3-7B			Qwen3-8B		
Steer toward	Edit	Switched	Random	Damaged	Switched	Random	Damaged
Memory	Values	70.9	1.6	9.3	51.2	4.1	1.2
	Residual	16.3	0.2	0.6	75.6	3.7	2.4
Document	Values	85.3	23.6	0	72.8	3.2	0
	Residual	52.3	5.5	1.8	98	1.2	0

C.10 TERMINOLOGY USED IN THE TABLES

Flip: the report changed to the other source label. *Switched* (Section 7): the answer changed to the other source's value. *Damaged*: the answer contains neither value. *Random*: a random direction with the same norm per layer, same positions and α, averaged over 3 draws. *Held-out*: directions fitted on some relations and measured on others; retained splits depend on the experiment.

C.11 WHERE EVERY NUMBER COMES FROM

Table 16: Scripts and result artifacts behind each section (per model, under a directory named by the model tag).

section	script	artifacts
items	`90_clean_items.py`	`items.jsonl`
3–4, document and question swaps	`91_clean_kv_patching.py`	`kv_patching.json`, `kv_patching_items.jsonl`
4, IDENTICAL-TEXT SWAP	`93_clean_answer_patching.py`	`answer_patching.json`, `answer_patching_items.jsonl`
tables for 3–4	`92_clean_tables.py`	`tables.md`
5, report-steering sweep	author-supplied aggregate table	`data/report_steering_update.csv`
Appendix C.7	author-supplied block 3i (script 105)	`data/report_steering_checks.json`
7	`96_clean_source_steering.py`	`source_steering.json`, `source_steering_items.jsonl`
Appendix E	author-supplied aggregate tables	`data/dynamicqa_results.json`
Appendix C.8	author-supplied blocks 3e/3g (scripts 100 and 102)	`data/answer_direction_comparison.json`; `data/steering_protocol.json`
6, Appendix D	author-supplied blocks 3j–3l	`data/transplant_clamp_results.json`

D CLAMPS AND TRANSFERS ALONG FITTED DIRECTIONS

D.1 PAIRED RUNS, DIRECTIONS AND READOUT

Blocks 3j (values), 3k (joint K+V) and 3l (energy controls) use OLMo-3-7B-Instruct and Qwen3-8B with thinking disabled, bf16 and Transformers 4.57.1. OLMo and Qwen have 32 and 36 layers, respectively, with 32 and 8 KV heads of dimension 128.

Pairs follow Section 4: a true-fact run reports `context`, while its irrelevant-document partner gives identical answer tokens but reports `knowledge`. Equal document lengths align positions. The question span runs from the document's end to the prompt's end, including chat-template tokens; the answer span ends at the answer's last token. Edits cover Q+A at every layer after answering and before the fixed source question. Other cache entries remain unchanged.

Directions are fitted on DEV relations and evaluated on held-out relations in two splits. Rates pool both splits and both directions for each true-fact–irrelevant-fact pair; the unit is a *directional evaluation*, not a distinct fact. The readout is forced-choice $P(\texttt{context})$; a flip crosses 0.5 away from the recipient's original report. These results are not pooled with the report-steering sweep.

For each layer and span, take the mean over DEV items of the position-averaged true-fact minus irrelevant-fact difference, then normalise the resulting direction. Value directions are the same as in Section 5, flattened across KV heads: one coordinate per position in 4096 dimensions on OLMo and 1024 on Qwen. Key directions are fitted in the pre-RoPE, post-`k_norm`-gain space. Grouping follows the model's key normalisation: one 4096-dimensional group on OLMo, but eight 128-dimensional head groups on Qwen. Question and answer positions use their respective fitted directions. Layer 0 has zero difference and is left unchanged. This is a family of local coordinates, not a rank-one intervention on the entire cache.

D.2 TWO OPERATIONS WITH THE SAME COORDINATES

Let u be a fitted unit direction, x the vector being edited, and s the paired vector supplying the target scalar projection (called a *coordinate* here). A value edit is

$$x' = x + \big(\langle s, u\rangle - \langle x, u\rangle\big)u. \tag{4}$$

For a *clamp*, first transplant Q+A K+V, then use the recipient's original vector as s. For *donor coordinates only*, start from the recipient's cache and use the donor as s. The value component orthogonal to u is unchanged. Both operations use observed coordinates rather than an amplified mean-difference shift.

Keys require a norm-preserving edit. We first undo RoPE using the relevant layer's rotary cosine and sine, including OLMo's distinct sliding-window and full-attention rotary modules:

$$y = \frac{k \odot C - \mathrm{rot_half}(k) \odot S}{C^2 + S^2}. \tag{5}$$

Reported round-trip error is approximately 2×10^{-7}. Within each key group, write $c = \langle y, u\rangle$, $r = y - cu$, and set c' to the source coordinate, clipped to $\pm 0.999\|y\|$. The edit is

$$y' = c'u + r\,\frac{\sqrt{\|y\|^2 - (c')^2}}{\|r\|}. \tag{6}$$

We then reapply RoPE and store the result in bf16. This keeps the starting key norm while changing its orientation; unlike the value edit, it rescales the orthogonal component. It does not preserve attention weights. Earlier additive key interventions caused nonspecific report changes even for random directions, motivating the norm constraint.

The rotation is performed *after* the key-normalisation gain. OLMo has near-zero gains in approximately a quarter of key dimensions, making inversion of the gain numerically unstable. The edit preserves Euclidean norm in this post-gain space, but need not stay on the exact ellipsoid defined by the model's normalisation and gain.

D.3 VALUE AND KEY CONTRIBUTIONS

Table 17: **Value-coordinate interventions.** Author block 3j. Report-flip rates (%) over directional evaluations on true-fact–irrelevant-fact pairs. Create/remove refer to `context`. Random-control rates are means over three draws.

	OLMo-3-7B		Qwen3-8B	
	Create	Remove	Create	Remove
K+V transplant	100%	100%	100%	100%
V transplant	100%	10.6%	100%	47.4%
V transplant + V clamp	0%	0%	0%	0%
V transplant + random V clamp	100%	10.6%	100%	47.4%
K transplant	78.7%	0%	39.5%	0%
K+V transplant + V clamp	100%	0%	68.4%	0%
Donor V coordinate only	100%	2.1%	100%	31.6%

Value-coordinate clamps eliminate flips produced by value-only transplants: 0% of reports flip on either model, compared with 55.3% on OLMo and 73.7% on Qwen without the clamp. With donor keys also transplanted, a value-only clamp is less effective, leaving 50% and 34.2% of reports flipped, all when creating `context`.

Table 18: **Joint key/value interventions.** Author block 3k. Report-flip rates (%) over directional evaluations on true-fact–irrelevant-fact pairs. Create/remove refer to `context`. Random-control rates are means over three draws.

	OLMo-3-7B		Qwen3-8B	
	Create	Remove	Create	Remove
K+V transplant + V clamp	100%	0%	68.4%	0%
K+V transplant + K+V clamp	2.1%	0%	0%	0%
K+V transplant + V clamp + random K clamp	100%	0%	67.6%	0%
K transplant + K clamp	0%	0%	0%	0%
K transplant + random K clamp	76%	0%	38.7%	0%
Random donor coordinates only	0%	0%	0%	0%

Joint K+V clamps leave 1.1% of reports flipped on OLMo and 0% on Qwen. Residual OLMo flips occur only in the creation direction (2.1%); no flips remain in the removal direction on either model.

Transferring only the donor's K+V coordinates flips 97.9% of OLMo and 100% of Qwen reports, pooled over both directions (Table 19), matching the main result in Table 4.

D.4 ENERGY-MATCHED CONTROLS

A random one-dimensional subspace captures very little of the donor–recipient difference. To test whether the selected coordinates matter simply because they capture a large change, block 3l also fits a control subspace on DEV. After projecting out the selected direction, it takes the fewest leading principal directions of per-position DEV differences needed to capture at least as much squared difference energy as the selected direction. This choice is made separately for each layer, span and key group. Reported median subspace dimensions range from 3–72 on OLMo and 6–186 on Qwen, versus one selected direction per group. The top-PC control uses only the first such principal direction. Random controls use three draws and the same intervention locations and coordinate-replacement geometry.

Table 19: **Controls for coordinate transfer.** Author block 3l; true-fact–irrelevant-fact pairs only. All cells give report-flip rates (%); random conditions average three draws. Top PC and matched subspaces are fitted after removing the selected direction.

Intervention	OLMo-3-7B	Qwen3-8B
Full K+V transplant	100%	100%
+ selected-coordinate clamp	1.1%	0%
+ top-PC clamp	100%	100%
+ energy-matched clamp	100%	94.7%
+ random clamp	100%	100%
V transplant + selected-coordinate clamp	0%	0%
+ energy-matched clamp instead	51.1%	65.8%
+ random clamp instead	55.3%	74.1%
Donor K+V coordinates only: selected	97.9%	100%
top PC instead	0%	0%
energy-matched subspace instead	0%	2.6%
random directions instead	0%	0%

Table 20: **Held-out difference energy covered by each intervention subspace.** Fractions of squared donor–recipient differences, summed over Q+A positions and layers. Matching is on fitting data, so held-out fractions need not be equal.

Model	Cache	Selected	Matched	Random
OLMo-3-7B	V	11.6%	12.3%	0.0%
OLMo-3-7B	K	25.1%	21.0%	0.0%
Qwen3-8B	V	15.7%	18.5%	0.1%
Qwen3-8B	K	36.2%	49.3%	0.8%

For a subspace projector P, the reported held-out energy fraction is

$$\frac{\sum_{\ell,t} \|P_{\ell,s(t)}(x_{\ell,t}^{\text{donor}} - x_{\ell,t}^{\text{recipient}})\|^2}{\sum_{\ell,t} \|x_{\ell,t}^{\text{donor}} - x_{\ell,t}^{\text{recipient}}\|^2}, \tag{7}$$

with key-group contributions also summed. This measures projected difference energy, not the norm of the nonlinear key edit. Matching is enforced on DEV only: the held-out OLMo key control captures 21.0%, compared with 25.1% for the selected directions. Qwen's matched control captures more held-out energy in both values and keys, yet leaves 94.7% of reports flipped after clamping and flips only 2.6% when its donor coordinates are imposed alone.

Scope of the behavioural checks. Answer repetition and document read-back were reported as fully preserved for *value-only donor-coordinate* edits, but have not been supplied for the joint K+V conditions. Fixed dialogue tokens therefore do not establish preservation of subsequent content readouts. Nor do these interventions establish exclusive coordinate semantics or a complete account of ordinary source reporting.

E REPLICATION ON DYNAMICQA WIKIPEDIA PARAGRAPHS

We extend the document and cache-transplant experiments to a Wikipedia-paragraph adaptation of DynamicQA (Marjanovic et al., 2024). Each item has seven length-matched document variants. For each model, we restrict the item pool to questions it can answer from memory. This evaluates the interventions on richer, benchmark-derived passages while preserving a known memory answer.

Rates are over eligible items for each condition or pair, rather than item–split pairs or the full memory-known pools. "Uninformative" denotes the condition providing no answer information; "irrelevant" denotes a paragraph about a different fact. Identical-token transplants additionally require matching question and answer tokens. All four model columns use the same order as the main tables.

E.1 DOCUMENT INTERVENTIONS

We edit the document after answering and retain the Q+A cache. Table 21 reports the percentage of `context` reports, rather than using a single flip metric for interventions with different initial reports.

Table 21: **DynamicQA: document edits mostly preserve the source report.** Cells give `context` report rates (%) among eligible items, with Q+A retained. In the first two rows, `context` is the original report; in the last two, the original report is `knowledge`.

Document edit	OLMo-3-7B	Qwen3-8B	OLMo-3.1-32B	Qwen3-32B
Drop counterfactual	100%	98.4%	99.4%	100%
Counterfactual → irrelevant	100%	100%	99.4%	100%
Irrelevant → true fact	0%	0%	0%	0%
Irrelevant → counterfactual	0%	0%	0%	0%

After deletion or irrelevant replacement of a counterfactual paragraph, `context` is retained in 98.4–100% of items across model–edit cells. Inserting a true or counterfactual paragraph into an irrelevant-document run creates no `context` reports. Thus, the pattern from Section 3 extends to these Wikipedia paragraphs, with rare exceptions to exact invariance.

E.2 QUESTION-ONLY TRANSPLANTS

Table 22 combines the question-only tests with irrelevant and uninformative donors. These effects vary by model and direction. The joint read-back rate uses all eligible items; it is not a success rate conditional on the report having flipped.

Table 22: **DynamicQA: question-only transplants.** Cells give the stated report rate (%) over eligible items; the recipient keeps its document and answer cache. The read-back row measures the joint event: the report becomes `knowledge` and the model still reads the counterfactual value from its document.

Recipient ← Q donor: report	OLMo-3-7B	Qwen3-8B	OLMo-3.1-32B	Qwen3-32B
Counterfactual ← irrelevant: `knowledge`	54.5%	100%	95.5%	57.6%
Irrelevant ← counterfactual: `context`	78.8%	72.4%	15.2%	72.9%
True fact ← uninformative: `knowledge`	94.9%	100%	99.5%	70.8%
Uninformative ← true fact: `context`	84.7%	77.1%	25.9%	59.8%
Counterfactual ← uninformative: `knowledge`	79.7%	99.2%	94.5%	16.8%
+ reads the false document value	66.7%	90.7%	93.9%	11.9%
Uninformative ← counterfactual: `context`	84.1%	72.9%	12.3%	61.6%

E.3 Question and answer contributions with identical tokens

For these pairs, the recipient's document and all question and answer tokens stay fixed. Only the cached state of Q, A, or Q+A is replaced. Each three-row group uses the same item subset, allowing a direct comparison of the edited spans. These subsets differ from the broader question-only tests above.

Table 23: **DynamicQA: joint Q+A transplants transfer nearly every report with identical text.** Cells give donor-following report rates (%) among eligible items. Within each pair, Q, A and Q+A edits use the same subset with identical question and answer tokens. Bold highlights the joint intervention, including its exceptions.

Edited span	OLMo-3-7B	Qwen3-8B	OLMo-3.1-32B	Qwen3-32B
True fact ← uninformative: report becomes `knowledge`				
Q only	97.6%	100%	99.4%	77.6%
A only	10.6%	8.5%	51.9%	7.2%
Q+A	**100%**	**100%**	**100%**	**100%**
Uninformative ← true fact: report becomes `context`				
Q only	82.4%	84.5%	27.1%	72.4%
A only	1.2%	0%	0%	3.9%
Q+A	**100%**	**97.2%**	**99.4%**	**100%**
True fact ← irrelevant: report becomes `knowledge`				
Q only	60.9%	100%	100%	97.3%
A only	6.5%	19.2%	60.6%	19.8%
Q+A	**100%**	**100%**	**100%**	**100%**
Irrelevant ← true fact: report becomes `context`				
Q only	87%	79.8%	32.4%	80.2%
A only	28.3%	0%	0%	2.7%
Q+A	**97.8%**	**99%**	**100%**	**100%**

Joint Q+A transplants make 97.2–100% of reports follow the donor across the 16 model–direction cells. The effect is nearly complete, rather than universal as on the templated task. The separate-span results again depend on model and direction, supporting the main paper's use of the joint span. This replication tests document edits and transplants; steering on DynamicQA remains outside the reported experiments.