KNOWING WHAT WILL MATTER NARROWS WORLD-STATE COVERAGE IN TRANSFORMERS

Anonymous authors
Paper under double-blind review

ABSTRACT

Accurately predicting one object's state does not reveal how much of an evolving world remains accessible inside a Transformer. We test whether advance knowledge of the evaluated object changes this breadth of accessible state. In a multi-object state-tracking task, paired two-layer Transformers receive the same trajectories and one selected-object final-state label per example. *Announced* models know the selected identity before the trajectory; *Delayed* models never receive it during the forward pass. Across the 14 of 20 paired initializations meeting a development-only success criterion in both conditions, selected-state accuracy is closely matched (98.62% vs. 97.77%). Yet Delayed models show 32.49 percentage points higher non-target accuracy through native readouts and 25.38 points higher accuracy with independent linear probes. We then intervene on residual states, component updates, and attention-head contributions. Transplanting the second-layer attention update redirects selected-state predictions while preserving the recipient residual; development-ranked head sets outperform matched alternatives on held-out interventions, and the same ranked heads produce larger reversion effects on average. These results support distributed, non-uniform contributions to prediction transfer without claiming to explain the coverage gap itself. Even among models that answer the evaluated question accurately, knowing what will matter can narrow what remains accessible.

1 INTRODUCTION

A correct answer about an evolving world reveals less about a model's internal state than it may appear to. To report an object's location after a sequence of changes, a model could maintain all object states, selectively preserve only what it expects to matter, or reconstruct the relevant state from earlier context when needed. The same answer is compatible with each strategy. State tracking therefore raises two distinct questions: *how much additional state is accessible*, and *which internal computations support the selected prediction*.

Prior work has recovered structured world-state and entity-binding information from language-model activations and causally manipulated it in several settings (Li et al., 2021; 2023; Feng & Steinhardt, 2024; Feng et al., 2025). Other studies identify lookback or late-aggregation mechanisms that recover relevant information when a query makes it useful (Prakash et al., 2026; Tang et al., 2026), while state-abstraction work shows that learned representations can favor task goals over general world structure (Li et al., 2024). These findings motivate a more specific question: *does advance knowledge of what will be evaluated merely select from an already accessible world state, or can it change the breadth of state accessible at the point of prediction?*

We isolate this question in a controlled multi-object state-tracking task. Each example supervises one object's final state. In *Announced*, the selected object's identity precedes the trajectory; in *Delayed*, the input contains `TARGET UNKNOWN`. Delayed never receives the selected identity during the forward pass: it is used only externally to choose the output slot for loss and evaluation. Paired conditions use the same trajectories, selected objects, labels, architecture, readout design, training protocol, and initial parameters. Both receive one state-label loss per example; the controlled difference is whether trajectory processing can depend on which state variable will be scored.

Among the 14 of 20 paired initializations satisfying a pre-specified development-only success criterion in both conditions, selected-state accuracy is closely matched (98.62% Announced; 97.77% Delayed).

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.

Non-target native-readout accuracy, however, is 65.20% versus 97.69%, a Delayed advantage of 32.49 percentage points. Independently fitted linear probes recover the same selective pattern: the non-target gap is 25.38 points, while the selected-state gap is only 0.36 points. We call this *endpoint state coverage*: what state is accessible at a common final representation, without assuming continuous maintenance rather than retrieval or reconstruction.

We then study computation supporting the selected prediction. Donor second-layer attention updates substantially redirect recipient predictions while preserving the recipient incoming residual. Head sets ranked using development interventions outperform matched alternatives on held-out test interventions, and complementary reversion tests support the same non-uniform importance pattern; the DEV-ranked top head alone falls well short of the all-head transfer. These analyses localize selected-state computation without claiming to explain why Delayed training produces broader non-target coverage.

Our contributions are threefold: **(1)** a matched test of anticipated information demand with fixed selected-state supervision; **(2)** evidence that withholding target identity yields broader non-target endpoint coverage under native and independent decoding despite closely matched selected-state accuracy; and **(3)** component- and head-level intervention evidence for distributed, unequal contributions to selected-state prediction transfer.

2 Related Work

World-state representations and retrieval. Language-model activations can encode evolving entity properties, board states, and entity–attribute bindings, with several studies also demonstrating intervention effects (Li et al., 2021; 2023; Feng & Steinhardt, 2024; Dai et al., 2024; Feng et al., 2025). Strong task performance nevertheless need not imply a coherent or reusable world model (Vafa et al., 2024; 2025). Entity-tracking and mechanistic work further shows that models can learn different tracking algorithms (Kim & Schuster, 2023; Li et al., 2025), retrieve bound information through mixtures of mechanisms (Gur-Arieh et al., 2026), or aggregate query-relevant updates only when needed (Prakash et al., 2026; Tang et al., 2026). Our endpoint measurements remain compatible with retrieval and late aggregation; we instead manipulate whether the evaluated object's identity is available during trajectory processing.

Task relevance and anticipated demand. Representations are shaped by task relevance (Hermann & Lampinen, 2020), and goal-oriented state abstractions can be favored over general world abstractions (Li et al., 2024). Latent-learning work asks whether models acquire information not immediately useful (Lampinen et al., 2026), while sleep-time compute links predictable future queries to the value of anticipatory computation (Lin et al., 2025). Our contribution is narrower: within one state-tracking task, paired conditions receive the same trajectories, selected objects, labels, architecture, and single-slot supervision, but differ in advance access to the selected identity. We test whether this changes endpoint coverage beyond the state scored on each example.

Decodability and causal localization. Probe accuracy depends on both representations and decoders (Hewitt & Liang, 2019), and encoded information need not be behaviorally usable (Lepori et al., 2026). Causal-abstraction and circuit work therefore combines localization with intervention while distinguishing faithfulness, completeness, and minimality (Geiger et al., 2021; Wang et al., 2023; Conmy et al., 2023). Activation-patching conclusions also depend on the intervention and metric, and successful output manipulation need not identify a faithful natural mechanism (Zhang & Nanda, 2024; Makelov et al., 2024). We accordingly separate native coverage, independent linear accessibility, probe-aligned subspace edits, and component/head interventions.

3 State Tracking under Anticipated Information Demand

We test whether advance knowledge of the state variable that will matter permits a more selective endpoint representation. *State coverage* denotes object-state information accessible at a common final readout position; it does not by itself distinguish continuous maintenance from retrieval or reconstruction.

3.1 Environment and paired conditions

Each story contains three objects (`apple`, `book`, `key`) and three boxes; each object is in one box or `NONE`. Stories contain eight base operations: three initial `PUT`s followed by five valid `PUT`, `MOVE`, `REMOVE`, or `CHECK` operations. `CHECK` mentions an object–box pair without changing state, and an additional misleading `CHECK` may be appended. We generate 20,000/2,000/5,000 train/dev/test stories, reject exact duplicate operation sequences across splits, and quota-balance the selected object's final-state label. Full generation details are in Appendix A.

For trajectory tokens $x_{1:T}$ and selected object q, the conditions are

$$\textsc{Announced}: \quad \texttt{TARGET}\ q\ \texttt{;}\ x_{1:T}, \tag{1}$$

$$\textsc{Delayed}: \quad \texttt{TARGET UNKNOWN}\ \texttt{;}\ x_{1:T}. \tag{2}$$

The prefixes have equal token length. Delayed receives no later query; q is used only externally to select the scored output slot.

Both conditions use the same two-layer causal Transformer architecture (hidden size 128; four heads per layer) and identical object-by-location readout architecture, with separately trained parameters but paired identical initializations. From the final normalized activation h_T, a linear head produces four state logits for each object. Only the selected slot contributes to the current-example loss,

$$\ell_c(x, q) = -\log p_{\theta_c,q}(s_K(q) \mid x_{1:T}, c)\,. \tag{3}$$

Thus both conditions receive one state-label loss per example; only Announced can condition trajectory processing on which object is being scored. For Delayed, however, the hidden target varies across examples. Under the generator-induced data distribution, its expected risk mixes supervision across object slots,

$$\mathbb{E}_x\left[\sum_o p(q = o \mid x)\,\mathrm{CE}(f_o(x), s(x, o))\right].$$

The quota-constrained generator does not make $q \mid x$ uniform (Appendix B). Broad multi-slot pressure is therefore part of Delayed training; the empirical question is how much advance target information permits Announced models to specialize under the same one-slot-per-example supervision.

3.2 Coverage and paired evaluation

We report selected-state accuracy A_{sel} and non-target native-readout coverage

$$A_{\mathrm{nt}} = \mathbb{E}\left[\frac{1}{|\mathcal{O}| - 1}\sum_{o \neq q}\mathbf{1}\{\hat{s}_K(o) = s_K(o)\}\right].$$

No additional query is issued for A_{nt}; it measures the remaining object slots from the same endpoint readout. Because native coverage depends jointly on h_T and the learned decoder, we additionally use independent probes and interventions.

We train 20 paired initialization seeds (100–119). Before formal test evaluation, a development-only gate requires both conditions in a pair to reach at least 90% selected-state accuracy. Announced passes in 18/20 runs and Delayed in 14/20; all 14 Delayed successes are jointly eligible. This set is frozen before test, probe, and intervention analyses. All primary between-condition differences are Delayed minus Announced, with 95% paired-bootstrap intervals over the 14 pairs and exact two-sided sign tests. Full gate outcomes are reported in Appendix C; optimization and split diagnostics are in Appendix A.

4 Anticipated Information Demand Shapes Native State Coverage

We first compare accuracy on the selected object with coverage of the remaining object states. Mean selected-state accuracy is 98.62% for Announced and 97.77% for Delayed, a paired difference of -0.85 percentage points (pp; 95% CI $[-2.36, +0.46]$). This is not an equivalence test, but it establishes the supervised-task reference within the eligible pairs.

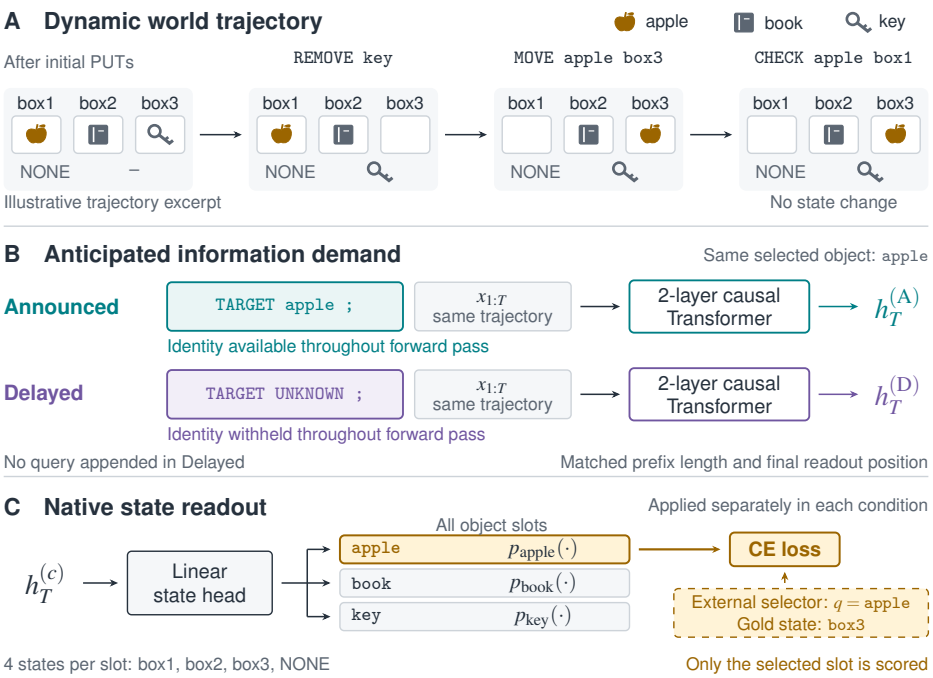


Figure 1: **Manipulating anticipated information demand while holding the state-tracking task fixed.** **(A)** A trajectory excerpt includes state changes and a misleading `CHECK` that leaves the state unchanged. **(B)** Paired inputs differ only in whether the prefix identifies the selected object ($q =$ `apple` here). Prefix lengths, trajectory tokens, and readout positions match; Delayed receives no later query. **(C)** The same readout architecture predicts all object states, but only the externally selected slot is scored; model parameters are trained separately.

Non-target native-readout accuracy differs sharply: 65.20% for Announced versus 97.69% for Delayed, a paired difference of $+32.49$ pp (95% CI $[21.13, 42.98]$), positive in 13/14 pairs (sign test $p = 0.0018$; Figure 2A,C). These predictions use the remaining object-specific slots from the same forward pass; non-target slots receive no loss on the current example, although each object can be selected on other examples. Announced models therefore still expose substantial non-target state, but the two conditions differ markedly in endpoint breadth outside the state known in advance to be scored.

Native-readout accuracy depends on both the hidden representation and learned output weights. We next ask whether the same selective gap persists under independently fitted decoders.

5 The Coverage Difference Persists under Independent Linear Decoding

For each frozen model, we extract the final normalized activation h_T and fit a separate four-class multinomial logistic-regression probe for each object. Probe weights are fit on training activations; inverse regularization strength $C \in \{0.1, 1.0\}$ is selected on development data; test data are used only for final evaluation. Announced and Delayed use the same protocol but independently fitted probe parameters.

Selected-state probe accuracy is 97.64% for Announced and 98.00% for Delayed, a paired difference of $+0.36$ pp (95% CI $[-0.62, +1.44]$). Non-target probe accuracy is 72.60% versus 97.98%, yielding $+25.38$ pp (95% CI $[16.13, 33.82]$), positive in 12/14 pairs (sign test $p = 0.0129$; Figure 2B,C). Thus, the gap persists after replacing the native readout weights with independently fitted linear decoders: non-target state is more linearly accessible from Delayed endpoint activations.

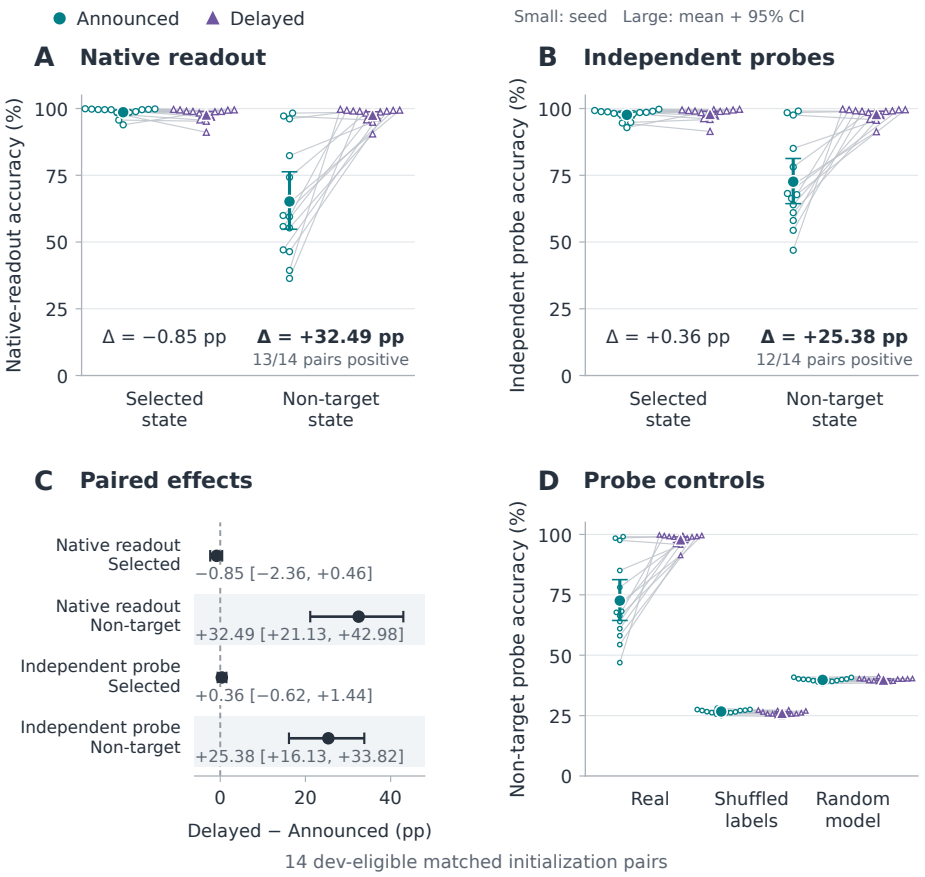


Figure 2: **The information conditions differ sharply in non-target endpoint coverage while selected-state performance remains closely matched.** **(A)** Native-readout accuracy on selected and non-target states. **(B)** Independent linear-probe accuracy on selected and non-target states. **(C)** Mean paired effects (Delayed minus Announced) with 95% paired-bootstrap intervals. **(D)** Non-target probe controls using shuffled labels and randomly initialized model features. Markers show the 14 development-eligible matched pairs and bootstrap summaries.

An exploratory post hoc difference-in-differences check directly compares the condition effect on non-target versus selected accuracy. It is $+33.35$ pp for the native readout and $+25.02$ pp for the independent probe, with paired-bootstrap intervals excluding zero in both cases (Appendix B). We report this as a sensitivity analysis rather than a pre-specified test.

We place this decodability against two controls (Figure 2D). Shuffling training/development labels yields 26.70%/26.16% non-target accuracy for Announced/Delayed, while probing fixed random-model features yields 39.78%/39.90%. Trained-model probes outperform both controls in every condition–seed run; the smallest advantages are 19.82 pp over shuffled labels and 6.71 pp over random-model features. These results establish linear accessibility beyond the native head, not task use: a decoder can recover information that the model need not use for its prediction.

6 Endpoint Patching as a Causal Reference

We next intervene on the model's selected-state computation. Recipient R and donor D share the selected object q but have different gold final states. At the final-block output, we replace the recipient's complete residual vector at the readout position with the donor's and then apply the common final normalization and state head. For fixed selected-slot readout g_q,

$$p_q^{\text{patch}} = g_q(r_D^{(L)}) = p_q^D.$$

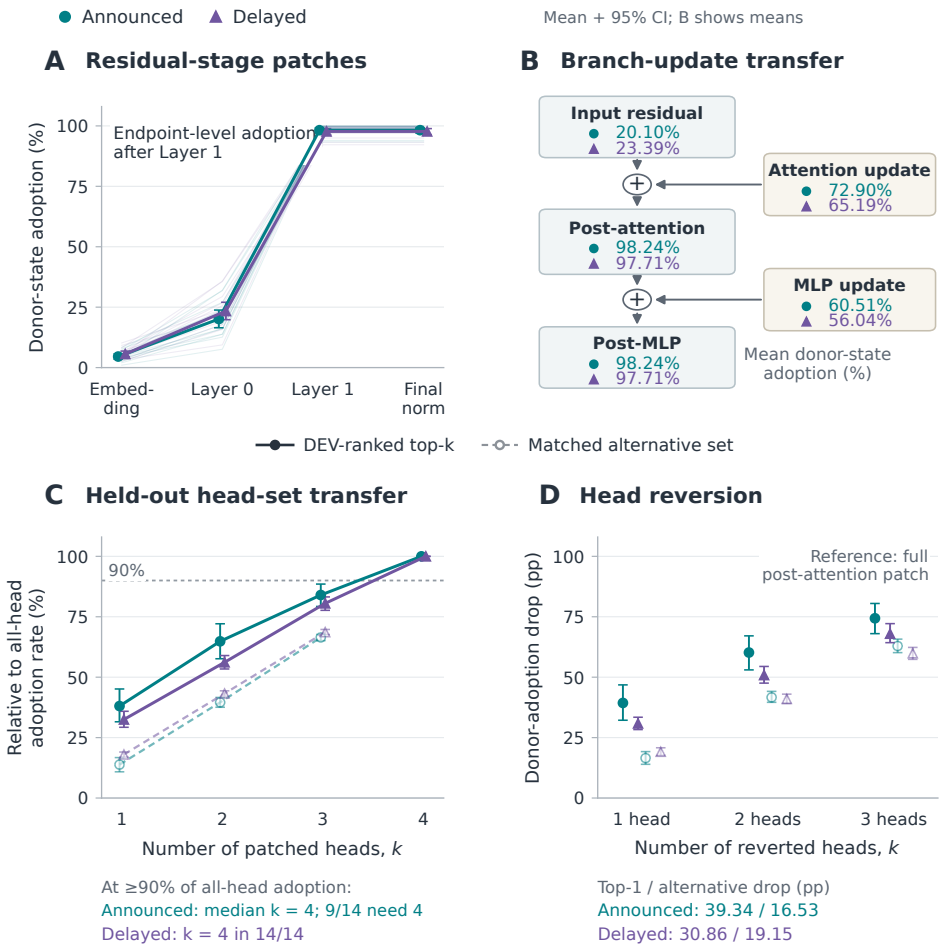


Figure 3: **Selected-state prediction transfer localizes to distributed, non-uniform final-layer attention contributions.** **(A)** Residual-stage patches bracket the final block. **(B)** Donor attention and MLP updates each transfer selected-state predictions. **(C)** DEV-ranked head sets generalize to held-out test interventions; normalization is by all-head attention-update adoption. **(D)** Reverting the same ranked heads in a donor background produces larger drops than matched alternatives.

Endpoint replacement therefore transfers the donor prediction by construction; donor-gold adoption (98.23% Announced, 97.71% Delayed) equals donor accuracy on the sampled pairs rather than independently measuring the strength of a particular state feature. We use this intervention only as a reference for patches placed earlier in the computation.

Same-state donor pairs, an earlier wrong-position intervention with a remaining downstream attention path, final-layer wrong-position patches, and self-patches calibrate the pipeline. Their interpretation and full results are given in Appendix D. The mechanistically informative question is how far upstream a *partial* donor computation can still redirect the selected prediction.

7 Localizing Selected-State Computation from Layers to Attention Heads

We move the counterfactual intervention upstream to ask which internal updates can redirect selected-state predictions. These analyses concern selected-state computation; they do not explain the between-condition non-target coverage gap.

7.1 FROM RESIDUAL STAGES TO BRANCH UPDATES

Whole-vector patches at the embedding, Layer 0 output, Layer 1 output, and final-normalized representation yield donor-state adoption of 4.57%, 20.10%, 98.23%, and 98.23% for Announced, and 5.63%, 23.39%, 97.71%, and 97.71% for Delayed (Figure 3A). Because late whole-vector replacement is endpoint-like, this only coarsely brackets the second and final Transformer block.

We therefore intervene on its branch updates. Let X denote the full incoming residual sequence, x_T its readout-position vector, and $a_T = [\mathrm{Attn}(\mathrm{LN}_1(X))]_T$ the attention update at that position. The MLP is position-wise, giving

$$z_{\mathrm{attn},T} = x_T + a_T, \qquad z_{\mathrm{out},T} = z_{\mathrm{attn},T} + m(z_{\mathrm{attn},T}).$$

An attention-update transplant keeps the recipient residual sequence but inserts the donor update at the readout position, $z'_T = x^R_T + a^D_T$. This yields 72.90% donor-state adoption for Announced and 65.19% for Delayed; donor MLP updates alone yield 60.51% and 56.04% (Figure 3B). These are separate interventions, not an additive decomposition. Complete post-attention residual replacement reaches the endpoint reference, so the nontrivial result is that branch updates alone already transfer substantial selected-state information.

7.2 DEV-RANKED HEAD SETS OUTPERFORM MATCHED ALTERNATIVES ON HELD-OUT INTERVENTIONS

We decompose the final-layer attention update into post-W_O head contributions c_h. For patched head set H,

$$z'_H = x^R_T + a^R_T + \sum_{h \in H} (c^D_h - c^R_h),$$

then recompute the downstream MLP and readout. Within each seed and condition, heads are ranked on development examples by single-head donor adoption; the DEV selection is frozen before test evaluation. On held-out pairs, the top-ranked head reaches 28.17% adoption versus 9.61% for matched alternatives in Announced, and 21.00% versus 11.74% in Delayed. For $k = 1, 2, 3$, DEV-ranked top-k sets outperform matched same-size alternatives in all 14 initialization pairs in both conditions (all sign tests $p = 1.22 \times 10^{-4}$).

Cumulative top-k adoption is 28.17/48.07/61.60/72.90% for Announced and 21.00/36.57/52.33/65.19% for Delayed as $k = 1 \ldots 4$ (Figure 3C). Relative to the all-head attention-update adoption rate, the corresponding mean ratios are 38.1/64.8/84.0/100% and 32.5/56.2/80.5/100%. The DEV-ranked top head alone falls well short of the all-head behavior; along the DEV-ranked sequence, 9/14 Announced and all 14 Delayed runs require all four heads to reach 90% of the all-head rate. This is a compactness statistic, not a globally minimal circuit claim.

A complementary reversion begins from the donor post-attention state and restores selected heads to recipient contributions. Reverting the top-ranked head drops donor adoption by 39.34 pp versus 16.53 pp for matched alternatives in Announced, and 30.86 versus 19.15 pp in Delayed (Figure 3D). Because this uses a donor residual background, it is an importance test rather than strict necessity under ordinary inference. Across $k = 1, 2, 3$, DEV-ranked reversions produce larger mean drops than matched alternatives, with positive paired differences in 12–14 of 14 seeds depending on condition and set size (Appendix E). Together, forward transfer and reversion support distributed but non-uniform head contributions to selected-state prediction transfer under the tested counterfactual backgrounds.

8 PROBE DIRECTIONS DO NOT CAPTURE FULL DONOR-STATE TRANSFER

To test whether decoder-aligned directions suffice to transfer the model's native output, we patch only the donor–recipient difference projected onto the span of centered probe weights (Appendix F). At full numerical rank ($r = 9$), donor-state adoption is 8.00% for Announced and 35.37% for Delayed, compared with 0.51% and 1.00% for same-rank random subspaces, but far below the 98.23% and 97.71% full-vector endpoint references. Probe-aligned directions therefore produce greater donor-state transfer than same-rank random controls under this edit, yet are not sufficient to reproduce near-complete native donor transfer. Linear accessibility and intervention efficacy are distinct from establishing a faithful natural mechanism (Makelov et al., 2024).

9 Discussion and Limitations

Our findings suggest that accessible world state depends on the information demands under which a model is trained. Advance target identity makes target-conditioned trajectory processing possible; withholding it removes that conditioning signal. One plausible interpretation is that advance information encourages more selective state processing, whereas uncertainty about which object will be scored favors broader endpoint coverage. We use *demand-adaptive* only in this across-training-conditions sense: we do not show that a fixed model dynamically changes its representational breadth, nor do endpoint measurements distinguish continuous maintenance from retrieval or reconstruction.

Closely matched accuracy on the evaluated state can coexist with large differences in access to the rest of the world. Selected-state accuracy therefore incompletely characterizes internal state. Conversely, lower native or probe accuracy does not establish absence under every decoder or representation site. Because Announced inputs contain q, target-conditioned reparameterization is also a possible contributor to the across-q probe gap; our probes establish accessibility under the tested common linear-decoding protocol, not absence under every target-conditioned decoder. The interpretability analyses answer complementary questions: native heads measure trained-readout accessibility; independent probes measure linear accessibility; probe-subspace edits test decoder-aligned directions; and component/head interventions identify updates that can transfer selected predictions under specified counterfactual backgrounds. Successful transfer alone does not establish faithful natural use, strict necessity, or a complete algorithm. In particular, explaining how the training manipulation produces the non-target coverage gap remains the central open mechanistic question.

The empirical scope is limited to a synthetic environment, two-layer Transformers with four heads per layer, and a binary manipulation of target availability. Main representation and causal analyses are conditional on the 14/20 initialization pairs that succeed in both conditions, so they characterize jointly successful training outcomes rather than the average effect across all initializations. We localize to residual stages, branch updates, and heads, not neurons or globally minimal circuits. Finally, we have not established algorithmic length generalization. A longer-trajectory stress test degrades sharply, but it also reaches learned absolute positional embeddings at positions never occupied by non-padding training tokens, so it is not a clean test of the tracking algorithm itself (Appendix G). Representational breadth, computational mechanism, and generalization should therefore be treated as distinct properties.

10 Conclusion

In our controlled state-tracking setting, models given advance knowledge of the evaluated object exhibit substantially narrower non-target endpoint coverage under both native readouts and independent probes, despite closely matched selected-state accuracy. Complementary interventions support distributed, non-uniform second-layer attention-head contributions to selected-state prediction transfer. These findings distinguish accessible-state breadth from the computation supporting a selected answer; explaining how anticipated demand produces the coverage gap remains open. More broadly, accurate prediction does not determine how much of a world remains internally accessible: knowing what will matter can shape what remains available.

AI use statement

Generative AI tools were used to assist with refining research hypotheses and experimental methodology, drafting and debugging research and analysis code, interpreting and cross-checking experimental outputs, searching and summarizing related literature, drafting and editing manuscript text, and preparing or checking tables and figures. The reported experimental measurements were produced by the authors' research pipeline rather than generated by an AI system. The authors reviewed AI-assisted methodological suggestions, code, interpretations, citations, and text; checked reported numerical claims against source outputs where available; and take responsibility for the final methodology, analysis, citations, artifacts, and claims in this submission.

REFERENCES

Arthur Conmy, Augustine N. Mavor-Parker, Aengus Lynch, Stefan Heimersheim, and Adrià Garriga-Alonso. Towards automated circuit discovery for mechanistic interpretability. In *Advances in Neural Information Processing Systems*, volume 36, 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/34e1dbe95d34d7ebaf99b9bcaeb5b2be-Abstract-Conference.html.

Qin Dai, Benjamin Heinzerling, and Kentaro Inui. Representational analysis of binding in language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 17468–17493, 2024. URL https://aclanthology.org/2024.emnlp-main.967/.

Jiahai Feng and Jacob Steinhardt. How do language models bind entities in context? In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/9d1b7fc578c0d2d6431fc26d736ecaf3-Abstract-Conference.html.

Jiahai Feng, Stuart Russell, and Jacob Steinhardt. Monitoring latent world states in language models with propositional probes. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/hash/3132d0405fabe24b2a7b6cd7ba9de6b5-Abstract-Conference.html.

Atticus Geiger, Hanson Lu, Thomas Icard, and Christopher Potts. Causal abstractions of neural networks. In *Advances in Neural Information Processing Systems*, volume 34, 2021. URL https://proceedings.neurips.cc/paper/2021/file/4f5c422f4d49a5a807eda27434231040-Paper.pdf.

Yoav Gur-Arieh, Mor Geva, and Atticus Geiger. Mixing mechanisms: How language models retrieve bound entities in-context. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/2eeff35664016c7f0f8aa704f0d9a83e-Abstract-Conference.html.

Katherine Hermann and Andrew Lampinen. What shapes feature representations? exploring datasets, architectures, and training. In *Advances in Neural Information Processing Systems*, volume 33, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/71e9c6620d381d60196ebe694840aaaa-Abstract.html.

John Hewitt and Percy Liang. Designing and interpreting probes with control tasks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing*, 2019. URL https://aclanthology.org/D19-1275/.

Najoung Kim and Sebastian 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, 2023. URL https://aclanthology.org/2023.acl-long.213/.

Andrew Kyle Lampinen, Martin Engelcke, Yuxuan Li, Arslan Chaudhry, and James L. McClelland. Latent learning: episodic memory complements parametric learning by enabling flexible reuse of experiences. *Transactions on Machine Learning Research*, 2026. URL https://arxiv.org/abs/2509.16189.

Michael A. Lepori, Tal Linzen, Ann Yuan, and Katja Filippova. Language models struggle to use representations learned in-context. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 14841–14857, 2026. URL https://aclanthology.org/2026.acl-long.676/.

Belinda Z. Li, Maxwell Nye, and Jacob 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, 2021. URL https://aclanthology.org/2021.acl-long.143/.

Belinda Z. Li, Zifan Carl Guo, and Jacob Andreas. (How) do language models track state? In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 34429–34452, 2025. URL https://proceedings.mlr.press/v267/li25r.html.

Kenneth Li, Aspen K. Hopkins, David Bau, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Emergent world representations: Exploring a sequence model trained on a synthetic task. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://arxiv.org/abs/2210.13382.

Zichao Li, Yanshuai Cao, and Jackie Chi Kit Cheung. Do LLMs build world representations? probing through the lens of state abstraction. In *Advances in Neural Information Processing Systems*, volume 37, pp. 98009–98032, 2024. URL https://proceedings.nips.cc/paper_files/paper/2024/hash/b1b16c4b875eb84d3585cb70d23970ca-Abstract-Conference.html.

Kevin Lin, Charlie Snell, Yu Wang, Charles Packer, Sarah Wooders, Ion Stoica, and Joseph E. Gonzalez. Sleep-time compute: Beyond inference scaling at test-time, 2025. URL https://arxiv.org/abs/2504.13171.

Aleksandar Makelov, Georg Lange, Atticus Geiger, and Neel Nanda. Is this the subspace you are looking for? an interpretability illusion for subspace activation patching. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/hash/70b8505ac79e3e131756f793cd80eb8d-Abstract-Conference.html.

Nikhil Prakash, Natalie Shapira, Arnab Sen Sharma, Christoph Riedl, Yonatan Belinkov, Tamar Shaham, David Bau, and Atticus Geiger. Language models use lookbacks to track beliefs. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0d1941f73833eb107939f62d323d6cc0-Abstract-Conference.html.

Zilu Tang, Qiao Zhao, Gabriel Franco, Derry Wijaya, Aaron Mueller, Sebastian Schuster, and Najoung Kim. Do language models track entities across state changes? In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2605.30233.

Keyon Vafa, Justin Y. Chen, Ashesh Rambachan, Jon Kleinberg, and Sendhil Mullainathan. Evaluating the world model implicit in a generative model. In *Advances in Neural Information Processing Systems*, volume 37, 2024. URL https://proceedings.nips.cc/paper_files/paper/2024/hash/2f6a6317bada76b26a4f61bb70a7db59-Abstract-Conference.html.

Keyon Vafa, Peter G. Chang, Ashesh Rambachan, and Sendhil Mullainathan. What has a foundation model found? Using inductive bias to probe for world models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 60727–60747, 2025. URL https://proceedings.mlr.press/v267/vafa25a.html.

Kevin Wang, Alexandre Variengien, Arthur Conmy, Buck Shlegeris, and Jacob Steinhardt. Interpretability in the wild: A circuit for indirect object identification in GPT-2 small. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=NpsVSN6o4ul.

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

A IMPLEMENTATION DETAILS FOR THE CONTROLLED STATE-TRACKING SETUP

A.1 DATA GENERATION AND TARGET SELECTION

The formal generator uses three objects and three boxes, with eight base operations per story. The first three operations are one `PUT` per object in random order. The remaining five base operations are sampled from the set of operations valid in the current state. Their configured relative weights are 0.30 for `PUT`, 0.40 for `MOVE`, 0.15 for `REMOVE`, and 0.15 for `CHECK`. A `MOVE` must change boxes, a `REMOVE` is valid only for a present object, and a later `PUT` is valid only for an absent object. `CHECK` never changes the simulator state. Base `CHECK`s are misleading with probability 0.75 conditional on being sampled; after the eight base operations, the generator additionally appends a misleading post-update `CHECK` with probability 0.60 whenever at least one object is present.

The selected final-state label is balanced by quota. After a story is generated, the target object is sampled uniformly from the subset of objects whose final-state label has not yet filled its quota for that split. Consequently, the selected label is exactly balanced, whereas the selected object itself is not forced to be uniform and target selection is not conditionally uniform given the trajectory. Under the fixed formal seed (20260915), the selected-object counts are $(6683, 6619, 6698)$ for (apple, book, key) in train, $(671, 649, 680)$ in development, and $(1730, 1604, 1666)$ in test. Each of the four selected final-state labels occurs exactly 5000, 500, and 1250 times in train, development, and test, respectively.

An operation sequence is assigned the SHA1-derived identifier `sha1(repr(operations))[:16]`. The generator rejects repeated identifiers within a split and carries all previously used identifiers forward when generating later splits. Regenerating the formal dataset yields 20,000, 2,000, and 5,000 unique identifiers and zero overlap between train/development, train/test, and development/test. This guarantees disjointness of exact serialized operation sequences; it does not rule out semantically similar but non-identical trajectories.

The observed operation-count distribution under the fixed seed is 8083/11917 stories of length 8/9 in train, 791/1209 in development, and 2104/2896 in test. The ninth operation, when present, is the appended misleading `CHECK`.

A.2 TOKENIZATION AND READOUT POSITION

The model uses a symbolic tokenizer with dedicated tokens for `TARGET`, `UNKNOWN`, `ALL`, operation names, objects, boxes, the separator "`;`", padding, and beginning-of-sequence. Hence `TARGET apple ;` and `TARGET UNKNOWN ;` each contain exactly three symbols before the trajectory. Each example is prepended with a beginning-of-sequence token. The readout index is set to the last actual token in that example before padding; padding positions are masked in attention. No post-trajectory query token is present in either condition.

A.3 ARCHITECTURE AND OPTIMIZATION

The Transformer has two pre-norm encoder blocks used with a causal attention mask, hidden dimension 128, four attention heads per block, feed-forward dimension 512, GELU activation, learned token and positional embeddings, maximum sequence length 128, and dropout 0. The final hidden sequence is layer-normalized before the last-token activation is passed to a linear $128 \to (3\times4)$ state head.

We optimize with AdamW using learning rate 3×10^{-4}, weight decay 0.01, batch size 256, and gradient clipping at 1.0. The learning-rate multiplier warms linearly for 500 steps and then decays linearly toward a floor of 0.1 by the 10,000-step maximum. Development evaluation is performed every 200 steps. Early stopping uses patience 20 evaluations and minimum improvement 0.001. A checkpoint is saved when development selected-state accuracy improves by more than this minimum delta. The formal pair-eligibility gate is computed separately from the complete recorded development history and uses the maximum recorded selected-state accuracy. These are distinct bookkeeping quantities: the step attaining the maximum recorded DEV score need not be the retained-checkpoint step under

the minimum-improvement rule. The fixed data-order seed is 20260915; model initialization seeds are 100–119.

A.4 Formal pair selection and uncertainty estimates

The development-only eligibility threshold is 0.90 in both Announced and Delayed. The 14 eligible initialization seeds are

$$100, 101, 103, 104, 105, 107, 110, 111, 113, 115, 116, 117, 118, 119.$$

Formal test evaluation begins only after this set is written by the gate stage. For a paired metric m, we form $d_i = m_i^{\text{Delayed}} - m_i^{\text{Announced}}$ for each eligible initialization pair and bootstrap the 14 values $\{d_i\}$ with 10,000 resamples using seed 20260915. We report the empirical mean and the 2.5th/97.5th percentiles of the bootstrap means. Exact two-sided sign tests count positive and negative paired differences and exclude ties.

A.5 Scope of the endpoint coverage measure

The experiment does not treat the non-selected slots as globally unsupervised. Only one slot contributes loss on any single example, but the selected object varies across the dataset. In Delayed, the selected object is hidden from the model's forward pass, whereas in Announced it is explicitly provided. Because the actual target-selection procedure is quota-constrained by final-state labels, we do not equate Delayed training with uniform full-state supervision in expectation. The empirical question is instead whether advance knowledge of the selected variable permits a more target-selective endpoint representation.

Likewise, endpoint coverage does not establish that all object states are continuously carried through every layer and token position. A model may retrieve or reconstruct information from earlier tokens before the final readout. Later analyses therefore separate independent endpoint decodability from counterfactual prediction transfer and component-level localization, without treating endpoint coverage alone as evidence for a unique online state-tracking algorithm.

B Target-selection and selectivity diagnostics

B.1 Target-selection distribution

The target variable is not sampled independently and uniformly from the three objects. The generator first enforces quotas over the selected object's final-state label and then samples the target uniformly among objects whose label quota remains available. Consequently, the selected label is exactly balanced but $q \mid x$ is not generally uniform. Table 1 gives the realized counts under the frozen data seed.

Table 1: Realized target-object counts and selected-label counts. The same stories and target identities are materialized for Announced and Delayed.

	Selected object q			Selected final-state label			
Split	apple	book	key	box1	box2	box3	NONE
Train	6683	6619	6698	5000	5000	5000	5000
Dev	671	649	680	500	500	500	500
Test	1730	1604	1666	1250	1250	1250	1250

For Delayed, q is hidden from the forward pass but varies across training examples. Its expected risk under the generator-induced distribution can therefore be written as a mixture over selected slots,

$$\mathbb{E}_x \left[\sum_o p(q = o \mid x)\, \mathrm{CE}(f_o(x), s(x, o)) \right],$$

where the actual $p(q = o \mid x)$ is induced by the quota-constrained generator rather than assumed uniform. Broad multi-slot pressure is therefore part of the Delayed training problem; our result should

not be interpreted as broad state emerging without supervision pressure. The controlled question is instead how much narrower the endpoint can become when the current target identity is available to the computation under the same one-slot-per-example supervision.

For reference, pooling the two non-target slots per example gives the following final-state label counts:

Split	box1	box2	box3	NONE
Train	11130	10826	10613	7431
Dev	1118	1065	1035	782
Test	2769	2748	2651	1832

These non-target labels are not balanced by construction, which is another reason to report model comparisons on the same stories and target assignments rather than interpret raw non-target accuracy as a class-balanced metric.

B.2 Exploratory selectivity interaction

The primary analysis separately reports the between-condition effect on selected and non-target accuracy. As a post hoc sensitivity check requested during paper auditing, we also compute, for each paired initialization,

$$I_i = \big(D - A\big)_{\mathrm{nt},i} - \big(D - A\big)_{\mathrm{sel},i}.$$

Using 100,000 paired-seed bootstrap resamples with NumPy `default_rng(20260926)`, the mean interaction is $+33.35$ percentage points for the native readout (95% percentile CI $[22.01, 44.16]$) and $+25.02$ points for the independent probe (95% CI $[16.20, 33.49]$). This analysis was added after the primary results were frozen and is therefore reported as exploratory rather than pre-specified evidence. It directly checks that the condition difference is larger for non-target than selected-state accuracy, rather than inferring an interaction from separate significance tests.

B.3 Decoder-conditioning limitation

The independent probes in the main analysis use a common fitting protocol across examples rather than separate decoders for each announced target identity. Because Announced inputs contain q, they could in principle use target-conditioned coordinate systems that are less well read by a single across-q linear probe. The present result therefore establishes a difference under the tested linear decoding protocol, not that non-target information is absent under every target-conditioned or nonlinear decoder. Target-conditioned probes are a useful sensitivity analysis for future evaluation.

C Training-success gate and all-run transparency

The formal protocol trains 20 paired initializations and freezes the eligible set using development selected-state accuracy before any formal test, probe, or intervention evaluation. Announced reaches the 90% development threshold in 18/20 runs and Delayed in 14/20; all 14 Delayed successes are paired with Announced successes, yielding the 14 jointly eligible pairs used in the main analyses. Table 2 reports every training run. In accordance with the frozen protocol, the six ineligible pairs were *not* evaluated on formal TEST at the gate stage; we therefore show TEST entries only for the eligible runs rather than retroactively treating excluded runs as part of the pre-specified test analysis.

The gate therefore changes the estimand: the main paired effects characterize outcomes conditional on successful training in both conditions, not the average effect across all 20 initializations. We keep this conditional estimand explicit throughout the paper. The table reports the gate statistic rather than checkpoint provenance: the exported gate summary available in this submission bundle does not contain the retained-checkpoint step or file hash, so we do not infer them from the maximum-DEV step.

Table 2: All 20 paired initializations. DEV is the maximum recorded selected-state accuracy used by the gate; Max-step is the development step at which that maximum was recorded. Because checkpoint retention uses a minimum-improvement rule, Max-step need not equal the retained-checkpoint step. TEST columns are selected/non-target native-readout accuracy for eligible runs only. "–" indicates that formal TEST was not evaluated under the frozen gate protocol.

	Announced DEV		Delayed DEV		Announced TEST		Delayed TEST		
Seed	Acc.	Max-step	Acc.	Max-step	Sel.	NT	Sel.	NT	Eligible
100	99.85	9800	96.55	10000	99.56	47.07	96.74	96.53	yes
101	100.00	4400	99.10	9800	99.92	46.39	98.90	99.03	yes
102	81.05	9600	76.80	6800	–	–	–	–	no
103	99.50	5600	98.30	9000	99.58	55.81	97.84	98.12	yes
104	99.85	4200	98.75	9200	99.68	98.29	98.80	98.71	yes
105	99.85	6400	99.15	9800	99.84	36.37	98.92	98.98	yes
106	98.35	9800	77.75	6800	–	–	–	–	no
107	99.80	6000	91.70	9000	99.62	55.30	91.12	90.49	yes
108	99.55	7200	88.25	5400	–	–	–	–	no
109	96.95	6400	76.10	7800	–	–	–	–	no
110	95.90	10000	95.05	9600	95.68	82.35	95.24	94.87	yes
111	97.80	7200	99.45	9800	98.44	64.81	99.40	99.40	yes
112	99.95	6800	88.05	6600	–	–	–	–	no
113	98.60	8600	98.85	10000	98.54	96.14	99.04	99.07	yes
114	85.85	9400	83.70	8600	–	–	–	–	no
115	99.85	6600	99.90	6000	99.58	39.37	99.68	99.71	yes
116	99.80	4000	99.70	5600	99.78	59.93	99.50	99.31	yes
117	98.15	9800	99.50	7600	98.86	59.47	99.50	99.35	yes
118	97.95	10000	95.75	9400	97.70	97.20	95.72	95.76	yes
119	94.05	8600	98.45	9800	93.90	74.25	98.32	98.29	yes

D ENDPOINT PATCHING CONTROLS

For each of the 14 eligible model pairs, we construct 500 recipient–donor test pairs with the same selected object and different gold final states. At the final-block output and final trajectory position, replacing the recipient residual vector by the donor vector transfers the donor prediction exactly because only the common final normalization and selected-slot readout remain. Patched predictions therefore match donor predictions in 100% of sampled pairs; donor-gold adoption is 98.23% for Announced (95% CI $[97.07, 99.17]$) and 97.71% for Delayed ($[96.33, 98.87]$), with recipient prediction-flip rates of 99.14% and 98.71%.

Same-state donors produce only 2.94%/4.34% flip rates for Announced/Delayed; the different-state minus same-state contrasts are 96.20/94.37 pp and positive in all 14 seeds (sign tests $p = 1.22 \times 10^{-4}$). Because full-vector endpoint replacement copies donor predictions, this contrast is a pairing/pipeline check rather than isolation of a state-specific feature. An early wrong-position intervention patches the preceding token one block earlier, retaining a downstream attention path, and yields 7.57%/7.30% flips with donor-gold adoption around 3.7%. This comparison changes both layer and position. Preceding-token patching after the final block and exact self-patching produce zero flips and serve as implementation checks.

E MECHANISM-ANALYSIS CONTROLS AND CROSS-VERSION CHECKS

E.1 MATCHED ALTERNATIVE HEAD SETS

The final attention layer has four heads. Within each seed and condition, the DEV single-head ranking is frozen before TEST. For cumulative top-k interventions, the matched alternative control enumerates every same-size head subset except the DEV-ranked top-k set: 3 alternatives for $k = 1$, 5 for $k = 2$, and 3 for $k = 3$. Because these counts are all below the implementation cap of 16, no random subsampling is used in the formal four-head models. Each alternative is evaluated on the same 500 TEST donor–recipient pairs and the control rate is the mean over all alternative subsets and pairs. Thus "matched alternative" is not a random-head draw and should not be read as one.

The DEV-ranked top-k forward patches exceed these matched alternatives in all 14 initialization pairs for both conditions at $k = 1, 2, 3$. Reversion is less uniform. Table 3 reports the exact seed counts for the ranked-minus-alternative drop from the full donor post-attention background.

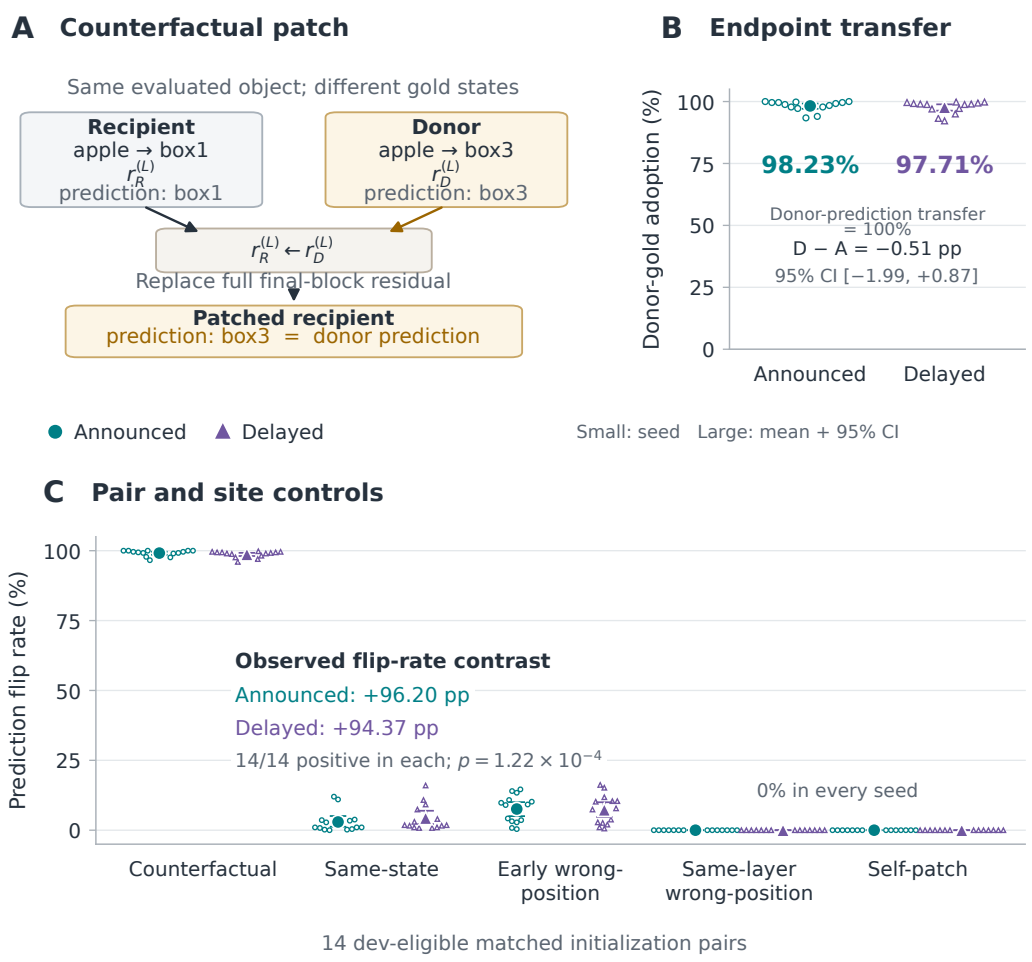


Figure 4: **Endpoint activation replacement provides a reference for upstream causal localization.** **(A)** Recipient and donor share the evaluated object but have different gold states; the donor's full final-block residual vector replaces the recipient's at the readout position. **(B)** Because the downstream normalization and readout are identical, endpoint replacement transfers the donor prediction by construction; donor-gold adoption therefore tracks donor accuracy on the sampled pairs (98.23% Announced; 97.71% Delayed). **(C)** Same-state pairing and alternative-site interventions calibrate the pipeline. The early wrong-position intervention preserves a downstream attention path but changes both layer and position; final-layer wrong-position and self-patch are implementation checks.

For completeness, the mean TEST donor-gold adoption of individual heads, grouped by their DEV rank, is:

	Rank 1	Rank 2	Rank 3	Rank 4
Announced	28.17	14.47	8.49	5.89
Delayed	21.00	15.00	11.81	8.40

These values support the narrower statement used in the main text: the DEV-ranked top head alone is far below the all-head attention-update adoption rate (72.90% Announced; 65.19% Delayed).

Table 3: Paired sign counts for DEV-ranked versus matched-alternative reversion drops. Ties are excluded from the exact two-sided sign test. These are within-condition counterfactual comparisons, not tests of ordinary-inference necessity.

Condition	k	Positive	Negative	Tie	Sign-test p
Announced	1	13	0	1	0.000244
Announced	2	13	1	0	0.001831
Announced	3	12	2	0	0.012939
Delayed	1	14	0	0	0.000122
Delayed	2	14	0	0	0.000122
Delayed	3	13	1	0	0.001831

E.2 Why 98.23% and 98.24% are kept distinct

Layer localization (v14) and component decomposition (v15) reuse the same frozen v10 checkpoint paths, the same 500-pair selection rule, and the same seed within each condition. For Announced seed 110, the exported pair IDs and recipient/donor indices are identical across v14 and v15, and the unpatched recipient and donor predictions agree on all 500 pairs. One patched example (pair ID 259) nevertheless differs between the v14 Layer-1 whole-vector path and the v15 manually decomposed post-attention path, changing that seed's donor-gold adoption from 93.4% to 93.6%. This produces the aggregate display values 98.23% and 98.24%, respectively.

The v14 implementation uses the model's native Transformer-layer forward, whereas v15 manually reconstructs the final block to expose branch-level intervention sites. The available artifacts localize the discrepancy to that single patched prediction, but they do not include the logit-level native/manual equivalence audit needed to attribute it to floating-point execution order or to rule out a lower-level reconstruction mismatch. We therefore treat this as an unresolved implementation-equivalence uncertainty and retain the two values as outputs of their respective intervention runs rather than silently harmonizing them. The difference is one example out of 7,000 Announced intervention pairs aggregated over the 14 seeds and does not affect the coverage results; none of our scientific claims relies on the 0.01-point difference.

E.3 Reference semantics

The v14 Layer-1/final-normalized whole-vector patches and the v15 complete post-attention residual patch are endpoint-like references, not theoretical ceilings on all partial interventions. Likewise, Figure 3C normalizes each seed by its *all-head attention-update* donor-adoption rate, whereas panel D measures a drop from the *full donor post-attention* background. These references are intentionally different and are not treated as a single baseline-subtracted causal-effect decomposition.

F Probe-defined subspace intervention details

The probe-subspace analysis uses the same final-normalized endpoint representation h_T used for independent probing in Section 5. For each object, the four-class multinomial probe weights are centered across classes; centered weights from all three objects are stacked and decomposed by singular value decomposition. The resulting numerical probe span has rank 9 in every eligible run. Probe fitting uses training activations, the inverse regularization strength is selected on development data, and test data are reserved for intervention evaluation.

For an orthonormal basis U with projector P_U, the probe-subspace donor patch is

$$h_{\text{probe}} = P_U h_D + (I - P_U)h_R = h_R + P_U(h_D - h_R),$$

whereas the complementary patch is

$$h_{\perp} = P_U h_R + (I - P_U)h_D.$$

The latter therefore retains the recipient's probe-subspace component and transfers only the orthogonal-complement component; it is not an ablation of U. A separate mean-ablation intervention removes centered variation in the probe subspace,

$$h_{\text{abl}} = h - P_U(h - \mu_{\text{train}}).$$

Matched random subspaces use the same rank.

Because the probe and intervention use the same final-normalized coordinate system, the full-rank construction has a simple decoder-specific interpretation. Let C stack the centered class-contrast rows of the fitted probes. By construction, the columns of U span the row space of C, so analytically $CP_U = C$ (up to floating-point error in an implementation). A full-rank probe-subspace patch therefore reproduces the donor's *probe logit contrasts*:

$$C(h_R + P_U(h_D - h_R)) = Ch_D.$$

The low native donor-state adoption after this patch should therefore be read as a mismatch between the external probe-aligned directions and the model's native readout transfer, not as evidence that every low-dimensional state subspace must be insufficient. The archived summary does not include a separate numerical residual for this logit-contrast identity, so we present the relation as an algebraic property of the stated construction rather than as an additional measured result. The centered row span of the native linear head would itself be an algebraic output-sufficiency control at this final site; such output sufficiency would still not identify a faithful semantic state mechanism.

At full rank, probe-subspace donor-state adoption is 8.00% for Announced and 35.37% for Delayed, compared with 0.51% and 1.00% for same-rank random subspaces. Complement-only patches retain substantial donor transfer (89.66% and 61.07%, respectively). These intervention outcomes are not additive components of the full-vector effect. In the complementary ablation analysis, removing the rank-9 probe subspace reduces non-target native-readout accuracy by 1.85 percentage points in Announced and 5.49 points in Delayed, compared with 0.10 and 0.07 points for matched random-subspace ablations. Together, these controls show that probe-aligned directions have non-random intervention effects while remaining insufficient to reproduce near-complete full-vector donor transfer.

G LENGTH STRESS TEST AND POSITIONAL-EMBEDDING CONFOUND

We evaluate frozen eligible checkpoints on longer trajectories with the same symbolic transition rules. In-distribution train/dev/test sequences occupy token lengths 36–45 (including the beginning-of-sequence token). The length-12, length-16, and length-24 stress sets occupy token ranges 52–65, 68–85, and 103–123, respectively. The model uses learned absolute positional embeddings with a table of length 128. Consequently, every longer-trajectory evaluation reaches positions that never occur as non-padding positions in formal training sequences and therefore receive no task gradient from those sequences.

Empirically, the Delayed-minus-Announced non-target native-readout gap shrinks from 32.49 percentage points in distribution to 3.17, 1.00, and 0.53 points at nominal operation lengths 12, 16, and 24, while selected-state accuracy in both conditions also drops sharply. Because this diagnostic confounds longer state-tracking trajectories with unseen learned positional embeddings, we do *not* interpret the collapse as evidence that a particular tracking algorithm fails to extrapolate, nor do we infer a mechanism from the shrinking near-floor gap. It is best viewed as a stress test showing that the present models do not provide a clean demonstration of length generalization. A controlled extrapolation study would require positional encodings or training coverage that separates trajectory length from unseen position parameters.