Causal Influence Control for Persistent Memory in Language Model Systems

Marvin Danig
Achiral Research
marvin@achiral.ai

Architecture and analysis preprint
July 2026

Abstract

Persistent-memory language model systems increasingly decide which prior user, project, or organizational records should influence present inference. Existing memory and retrieval systems usually expose relevance, recency, summarization, or tool-mediated persistence; they do not by themselves specify how a recalled memory's downstream behavioral effect should be predicted, constrained, verified, and reversed. We formulate memory recall as a controlled inference-time intervention. Each memory entity is associated with a causal influence signature: an estimate, in a canonical influence space, of the directional change in future model behavior produced by applying that memory through context, an external memory interface, attention, or latent state. A controller selects admissible memory interventions by target effect, side-effect budget, policy risk, and reliability; observes the realized effect; records lineage; and rolls back or quarantines interventions whose observed effect diverges from prediction. The contribution is an architecture-and-analysis proposal, not an empirical benchmark report. We state claim boundaries, falsification conditions, evaluation protocols, and deployment risks for future measurement.

1 Introduction

Long-lived language model systems now maintain state across conversations and tasks. Personal and agent memory systems retain preferences, project facts, files, summaries, tool outputs, and prior decisions, allowing future inference to start from accumulated context rather than an empty prompt [9, 10, 11]. Open-weight frontier models are also increasingly optimized for agentic tool use and long-context work [13]. These trends make memory a runtime control problem: the model does not merely retrieve information; it is shaped by what the system allows to become cognitively available.

Retrieval-augmented generation (RAG) selects external information for generation [1]. Graph, hierarchical, and agent-memory systems broaden the retrieval substrate and representation [2, 3, 4, 5]. Mechanistic interpretability and model-editing work shows that causal interventions on model internals can localize or alter behavior [7, 8]. Influence functions trace training examples to predictions [6]. These literatures supply useful tools, but a persistent organizational memory system needs a distinct runtime question:

> Which memory intervention should be permitted to affect this model now, what effect is expected, and what should happen if the effect is wrong?

This paper proposes *causal influence control* for persistent AI memory. The central claim is narrow: persistent memory should be governed as a reversible, observed, lineage-bearing intervention rather than as a relevance-ranked context append. The claim is falsified or narrowed if simpler retrieval, graph-reranking, or prompt-policy baselines match the proposed controller on effect-targeting accuracy, side-effect control, and rollback utility under equal model, memory, policy, and cost budgets.

Contributions. We (i) define the distinction between memory relevance and memory influence; (ii) specify causal influence signatures for persistent memory entities; (iii) formulate constrained selection and observed-effect

verification; (iv) describe rollback, quarantine, reliability, and lineage mechanisms; (v) derive implementation-level cost and controllability considerations; and (vi) provide a reproducible evaluation protocol. This architecture-and-analysis preprint reports no benchmark measurements.

2 Memory Relevance Is Not Memory Influence

Let $M = \{m_1, \ldots, m_n\}$ be a set of persistent memory entities and q a current request. Conventional retrieval ranks candidates using a relevance score

$$s_q(m_i) = \mathrm{sim}(f(q), f(m_i)), \tag{1}$$

possibly combined with lexical, graph, metadata, or learned reranking features. Relevance answers whether a memory appears useful for the request. It does not measure the behavioral direction in which the memory will move a model once it is included.

For a compliance recommendation, a semantically relevant pricing memory may shift generation toward cost reduction when the intended effect is caution. A stale project summary may match the prompt while suppressing newer evidence. A high-confidence memory may become risky when combined with a particular user, tool, or policy state. In each case, the system needs an influence estimate, not only a relevance score.

We model a memory intervention as an operator $I(m_i, a_t)$ that applies memory m_i through an application target a_t, such as the context window, a memory tool, an attention mechanism, or a latent-state interface. The intervention acts on a current model state and changes a future output distribution or a projected behavioral measurement. The object of control is therefore

$$\Delta y_{future}(m_i, a_t \mid h_t, q, \pi), \tag{2}$$

where h_t is a model or task state and π is the applicable authorization and policy regime.

Dimension	**Relevance retrieval**	**Influence-controlled memory**
Primary question	What matches the request?	What will this memory cause the model to do?
Selection target	Passage, file, node, or summary	Reversible memory intervention
Evidence object	Similarity, metadata, graph proximity	Predicted and observed behavioral effect
Safety boundary	Filter before or after retrieval	Hard admissibility plus side-effect budget
Runtime feedback	Optional click, rating, or trace	Prediction, observation, divergence, lineage
Failure response	Drop, rerank, or ignore	Roll back, suppress, quarantine, refresh

Table 1: Relevance is a candidate signal. Influence control treats recall as an intervention with expected and observed effects.

3 Architecture

The proposed controller contains six runtime components: a signature estimator, a signature store, an effect specification interface, an admissible-memory selector, an effect observer, and a lineage-governed rollback mechanism.

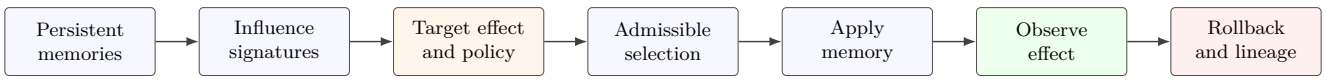

Figure 1: Causal influence control loop for persistent memory. The diagram omits product-specific coefficients, thresholds, and storage schemas; the research object is the observable control loop.

3.1 Canonical influence space

Predicted and observed effects must be comparable. We call their shared representation a *canonical influence space*. It may be a full output-logit space, a task-specific token subspace, a sparse-feature space, a probe-score vector, or

another projected measurement space. The choice is an engineering and evaluation decision. The publication-level requirement is that the system records the model version, prompt or probe protocol, projection, decoding policy, and measurement method used to produce each effect vector.

3.2 Causal influence signatures

For memory entity m, a causal influence signature ϕ_m estimates the directional shift caused by applying m through a memory intervention operator. In a differentiable implementation, a local linearization can be written

$$\phi_m \approx P J_t M_m(c_m), \tag{3}$$

where c_m is memory content, M_m maps the content into an intervention perturbation, J_t is a future-output Jacobian or integrated influence map at state h_t, and P projects the result into the canonical influence space. When internal gradients are unavailable, ϕ_m can instead be estimated through finite differences, controlled probes, or learned local maps. This paper does not require one estimator to be universal.

A signature record should be treated as provisional unless it carries reliability evidence: model version, computation method, prompt or probe cohort, confidence intervals or residuals, staleness status, and calibration history. The record belongs to the memory governance layer, not to the memory text alone.

3.3 Effect specification and selection

At runtime, a requester or policy process supplies a cognitive-effect specification containing a target influence vector ϕ_{target}, prohibited regions, side-effect budget, policy constraints, and intervention criticality. The selector evaluates only authorized candidates and may minimize

$$\mathcal{L}(m) = \|\phi_m - \phi_{\text{target}}\|^2 + \lambda \|\operatorname{proj}_{\perp}(\phi_m)\|^2 + \mu R_\pi(m) + \nu U(m), \tag{4}$$

where $R_\pi(m)$ is policy risk and $U(m)$ is signature uncertainty. The formula is illustrative of the control objective: target the intended shift, penalize off-target influence, respect policy, and prefer reliable signatures. Candidates that violate hard access, provenance, or prohibited-region constraints are excluded rather than softly downweighted.

3.4 Observation and rollback

After a memory intervention is applied, an observer measures the actual effect in the same canonical influence space. The system can compare prediction and observation with a normalized divergence score

$$D(m) = \frac{\|\phi_{\text{observed}} - \phi_m\|^2}{\|\phi_m\|^2 + \epsilon}. \tag{5}$$

If $D(m)$ exceeds a threshold determined by criticality and uncertainty, the controller executes a response appropriate to the application target: remove context, revoke a memory-tool result, discard an attention bias, restore a checkpoint, suppress future selection, or quarantine the memory pending review. The key point is not that rollback is always perfect; it is that memory influence should be observed, bounded, and recorded rather than silently trusted.

4 Analytic Consequences

C1: The retrieval unit changes. The selected object is no longer simply the most relevant document or passage. It is an admissible intervention expected to move model behavior in a desired direction.

C2: Side effects become measurable design objects. A memory can be on-topic and still harmful because it pushes the model toward an unwanted frame. Influence control makes off-target projection a first-class cost.

C3: Reliability is memory-specific and model-version-specific. A signature estimated for one checkpoint, prompt protocol, or tool interface may become stale when the model, decoding policy, or memory application target changes.

C4: Memory safety includes reversibility. Access control prevents unauthorized use; reversibility addresses authorized memories whose effects are inaccurate, stale, adversarial, or contextually unsafe.

C5: Lineage converts failures into calibration data. Every confirmed or rolled-back intervention provides evidence about estimator quality, memory staleness, multi-memory interaction, and policy adequacy.

5 Cost and Controllability

Influence estimation is more expensive than ordinary retrieval. A practical system therefore separates offline signature preparation, cheap candidate filtering, and high-fidelity validation for sensitive contexts. Full Jacobians are usually impractical at runtime for frontier-scale models; integrated gradients, low-rank approximations, probe-based estimates, and cached signatures are more plausible operational regimes.

For N memory entities and influence dimension d, dense signature storage costs $O(Nd)$ before compression. Candidate selection can be reduced using approximate nearest-neighbor search over signatures after authorization and metadata filters. Observed-effect measurement costs depend on probe count, output horizon, and whether the system can inspect logits or only scored responses.

Control-theoretic diagnostics are useful when planning multiple interventions. A reachability matrix or controllability Gramian over candidate signatures can identify target directions poorly reachable by memory injection alone:

$$W_c = \sum_{k=0}^{T} \phi_k \phi_k^T. \tag{6}$$

Small eigenvalue directions should be treated cautiously: failure to reach them may reflect the limits of the memory substrate rather than a selection bug. An observability analogue can test whether the proposed probe set can reliably distinguish the intended influence direction from noise.

6 Evaluation and Falsification

The proposed architecture should be evaluated against simpler baselines before deployment claims are made. We propose matched comparisons among:

1. dense or hybrid top-k retrieval;
2. graph retrieval or memory-tool recall without influence signatures;
3. relevance retrieval plus policy filtering;
4. influence reranking without observation;
5. full influence-controlled memory with observation, rollback, reliability, and lineage.

Tasks should include compliance-sensitive recommendations, temporal supersession, project-memory recall, contradiction handling, tool-use planning, multi-session agent work, and adversarial or stale memory injection. Primary metrics should include target-effect alignment, side-effect magnitude, answer accuracy, provenance completeness, stale-memory false-positive rate, rollback precision, post-rollback recovery quality, calibration error, latency, token cost, and human audit time.

The central claim is narrowed if relevance or graph baselines match target-effect alignment and side-effect control at lower cost. It is falsified if observed-effect measurement fails to predictably distinguish beneficial interventions from harmful ones, or if rollback and lineage do not improve post-failure recovery.

Reproducibility protocol. A future empirical paper should publish or escrow the corpus snapshot or synthetic generator, memory schema, authorization rules, signature-estimation protocol, projection definition, model and

decoding versions, random seeds, probe templates, per-task traces, selected and rejected memories, observed effects, rollback events, and aggregate plus per-stratum metrics. This preprint releases no benchmark dataset, production thresholds, or product implementation.

7 Failure Modes and Scope

Influence control creates its own risks. Stored signatures may leak information about private memories if not protected. Probe protocols may overfit to narrow behavior. Rollback can be incomplete once generated text, tool calls, or user decisions escape the model boundary. Multi-memory interactions can be nonlinear, making single-memory signatures unreliable. Attackers may craft memories with benign surface meaning and harmful influence signatures. Finally, a controller optimized for target effect can become manipulative if targets are not constrained by user intent, organizational policy, and law.

The intended scope is not short-lived factual question answering over a small corpus. The architecture is most relevant for long-running agents, enterprise memory, compliance-sensitive work, model-assisted decision processes, and settings where memory influence must be attributable and reversible.

8 Related Work and Claim Boundary

RAG and its graph or hierarchical variants improve the selection and organization of external information for generation [1, 2, 3]. Agent-memory systems such as MemGPT, temporal knowledge-graph memory, Claude memory features, and ChatGPT memory systems address persistence, summarization, and continuity across sessions [4, 5, 10, 9]. Mechanistic interpretability and model-editing work studies causal circuits and interventions within models [7, 8]. Influence functions attribute predictions to training examples [6]. Recent interpretability work on global workspace-like structures suggests that modern language models may expose controllable internal patterns relevant to deliberate reasoning [12].

This paper does not claim those components independently. The proposed contribution is the ordered runtime architecture: persistent memory entities carry causal influence signatures; a controller selects interventions by target effect, side-effect budget, policy risk, and uncertainty; observed effects are compared to predicted effects; and divergence triggers rollback, suppression, quarantine, and lineage updates.

9 Conclusion

Persistent memory changes the status of retrieval. A recalled memory is not just evidence placed near a prompt; it is an intervention that can steer a model's future behavior. Treating memory as a causal, observed, reversible control object gives long-lived language model systems a sharper contract: predict the influence of recall, constrain its side effects, verify its realized effect, and preserve a lineage when prediction fails. Whether this improves real systems is an empirical question, but the evaluation target is clear.

Declarations

Author contributions. Marvin Danig conceived the architecture, developed the analytic specification, and wrote the manuscript.

Funding. This research received no external funding.

Competing interests. The author is affiliated with Achiral Research. Certain aspects of the described architecture may be covered by patent applications associated with Achiral.

Data, code, and materials availability. This paper reports an architecture and analytic results and therefore introduces no experimental dataset or benchmark code. A companion research essay and future artifact links are maintained at achiral.ai/research/jacobian-causal-memory-control.

Ethics and limitations. The proposed system could be misused to manipulate model behavior through persistent memory. Deployment requires user-intent constraints, authorization before scoring, privacy protection for signatures, robust audit logs, calibrated rollback, and human review for high-stakes memory effects.

References

[1] Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., and Kiela, D. Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474, 2020. arXiv:2005.11401.

[2] Edge, D., Trinh, H., Cheng, N., Bradley, J., Chao, A., Mody, A., Truitt, S., Metropolitansky, D., Ness, R. O., and Larson, J. From Local to Global: A Graph RAG Approach to Query-Focused Summarization. 2024. arXiv:2404.16130.

[3] Sarthi, P., Abdullah, S., Tuli, A., Khanna, S., Goldie, A., and Manning, C. D. RAPTOR: Recursive Abstractive Processing for Tree-Organized Retrieval. *International Conference on Learning Representations*, 2024. arXiv:2401.18059.

[4] Packer, C., Wooders, S., Lin, K., Fang, V., Patil, S. G., Stoica, I., and Gonzalez, J. E. MemGPT: Towards LLMs as Operating Systems. 2023. arXiv:2310.08560.

[5] Rasmussen, P., Paliychuk, P., Beauvais, T., Ryan, J., and Chalef, D. Zep: A Temporal Knowledge Graph Architecture for Agent Memory. 2025. arXiv:2501.13956.

[6] Koh, P. W. and Liang, P. Understanding Black-box Predictions via Influence Functions. In *Proceedings of the 34th International Conference on Machine Learning*, PMLR 70:1885–1894, 2017. PMLR.

[7] Wang, K. R., Variengien, A., Conmy, A., Shlegeris, B., and Steinhardt, J. Interpretability in the Wild: a Circuit for Indirect Object Identification in GPT-2 Small. *International Conference on Learning Representations*, 2023. arXiv:2211.00593.

[8] Meng, K., Bau, D., Andonian, A., and Belinkov, Y. Locating and Editing Factual Associations in GPT. *Advances in Neural Information Processing Systems*, 2022. arXiv:2202.05262.

[9] OpenAI. Dreaming: Better memory for a more helpful ChatGPT. 2026. openai.com.

[10] Anthropic. Memory for Claude Managed Agents: agents that learn across sessions. 2026. claude.com.

[11] Anthropic. Bringing memory to Claude. 2025. claude.com.

[12] Anthropic. A global workspace in language models. 2026. anthropic.com.

[13] Kimi Team. Kimi K2: Open Agentic Intelligence. 2025. arXiv:2507.20534.