REMEMBER WHAT CANNOT BE RECOVERED: MEMORY ALLOCATION FROM THEORY TO FRONTIER AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Language-model agents sustain extended interactions through context compaction, persistent memory, and retrieval, choosing what to remember and what to recover later under uncertainty about future needs and recovery resources. An agent may omit a dependency from memory if alternative paths preserve the answers it needs, but that dependency becomes necessary when unavailable components block those alternatives. For ordered random workflows, we quantify the average number of edges and bits needed to preserve every answer. In the small-average-error regime we analyze, we construct an asymptotically optimal memory by arranging retained connections so that surviving routes branch and reconnect, providing alternative paths between many source-target pairs using the same stored edges. We then test whether frontier language models can choose what to remember. On random and real citation graphs, models asked to write the memory fall far short: DeepSeek V4-Pro, GLM 5.2, and Muse Glimmer 30B do no better than randomly chosen edges, and on citation graphs most of their errors are avoidable isolation errors, in which every stored neighbor of an endpoint fails, an event that an even allocation of the budget makes much less likely. Telling models how our theory spreads the budget reduces these errors for a model that follows the advice, but its memories still do no better than random ones: avoiding isolation is necessary, yet good memories also depend on which routes are kept, and the models do not work this out from the graph. Our results show that what an agent should remember depends on how its information can be recovered rather than on how important it looks, and that current models do not yet make this choice on their own.

1 INTRODUCTION

Recent language-model agents carry out extended workflows by calling tools, inspecting outputs, and revising plans across many interactions. Continuing such work requires information to survive beyond the model's current context. Deployed systems now combine context compaction with persistent notes that can be consulted in later sessions (Anthropic, 2025). Durable session stores further separate the history of an interaction from the portion currently supplied to the model, allowing earlier events to be retrieved after a context reset or execution failure (Anthropic, 2026). Agents must therefore decide what to carry into each new context.

Memory management is also becoming a learned component of the agent's policy. AgeMem, for example, exposes operations for storing, retrieving, updating, summarizing, and discarding information, and trains their use through reinforcement learning (Yu et al., 2026). Such methods manage persistent memory and working context together, yet information omitted now may still be needed for a later decision. Even when an external store retains the full history, the agent must choose what to retrieve and how to use the selected representation.

Whether information can be recovered later affects what the agent should store now. An agent might store a fact directly or retain a reference to a resource from which it can be obtained. A compact reference can save space, but its usefulness depends on the required recovery route remaining available and usable. We therefore ask: how should an agent allocate and use limited memory when information may become unavailable?

Memory problems of this kind differ along two axes. The first is how an outcome combines its prerequisites: one route may suffice, every prerequisite may be required, or contributions may add up. The second is where the uncertainty about a later request comes from: components may be

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.

unavailable, routes may fail at random, or the agent's data may leave part of the structure unresolved. Four agentic workflows illustrate these types, and Appendix H.5 places further structures on the same axes.

Example 1 (Customer support under outages)**.** A support agent keeps notes across sessions on how to look things up. From an order number it can reach the shipping address through the shipping service, the invoice in the billing system, or the customer record in the CRM. A later ticket asks whether the address can still be retrieved during a maintenance window in which some of these systems are offline. The uncertainty is which systems will be down.

Example 2 (Refunds with required approvals)**.** An operations agent issues refunds. A refund needs the order record, a payment confirmation, and a return label, each with its own upstream services. The agent asks whether a refund can still be issued while some services are offline. The uncertainty is again which systems are down, but every prerequisite is now required.

Example 3 (Package setup with unreliable downloads)**.** A coding agent sets up build environments in separate sessions. Each package can be fetched from the public index, a company mirror, or a vendored copy, and some become reachable only after others are installed. When a build needs a package, the agent must judge whether it can be obtained. No system is known to be down, yet any download may time out, so every route succeeds only with some probability.

Example 4 (Research on gene regulation)**.** A research agent analyzes perturbation screens over several weeks. Gene expression mixes the activity of latent gene programs, and each experiment weakens one program. The data show which programs act upstream of which, yet cannot tell whether an effect is direct. Before the next experiment the agent asks whether perturbing one program still changes another in a cell line where some programs are knocked out. The uncertainty lies in the knockouts and in what the data can identify.

In each example a request asks whether an outcome can still be reached along directed dependencies, possibly after some of them are lost. We represent the dependencies by a directed acyclic graph and study four models on it (Section 2): Boolean OR workflows (Example 1), AND workflows (Example 2), noisy-OR networks (Example 3), and linear causal models learned from soft interventions (Example 4). We also treat requests for the size of linear effects, where contributions add up. Without failures, OR memory is transitive reduction (Aho et al., 1972), and with failures it becomes fault-tolerant reachability (Chakraborty et al., 2022). We take the graph as known and assume a reader that answers exactly from the stored edges, which isolates the information needed to answer correctly. To quantify how much memory these workflows require on average, we use a classical ordered random-graph prior (Albert & Frieze, 1989).

For OR workflows, we identify the dependencies that must be retained to answer every request under a given failure limit. These edges form a unique minimum sufficient graph, and we characterize the failure threshold at which each edge becomes necessary. For workflows with n vertices and limiting failure fraction ρ, the optimal number of retained original edges has leading term $n \ln(1/\epsilon)/\ln(1/\rho)$ for small target average error ϵ, in a stated range of density and error. The causal model has the same budget, noisy-OR networks follow the same law with the link failure probability in place of ρ, AND workflows need only the transitive reduction, and linear effects need nearly every edge. Redundant paths save memory when they serve as backups, and save nothing when each path adds its own share to the answer.

Motivated by the theory, we ask how language-model agents choose what to remember. Recent benchmarks evaluate retrieval, test-time learning, selective forgetting, and memory of workflows in incremental interactions (Hu et al., 2026; Wu et al., 2026). We instead measure a model's choice against the best memory for the same budget, under the request distribution of the theory, and separate the choice of what to store from the reading of what was stored.

The theory holds up on graphs of moderate size: the loss of memories built by algorithms falls at the rate set by the failure fraction, backups make OR memory compressible while linear effects are not, and real citation graphs show the same dependence on failures. The models do much worse. When they choose what to store, they do no better than random edges, and most of their errors are avoidable isolation errors: an endpoint whose stored neighbors all fail, an event that an even allocation of the budget makes much less likely. Telling them how the theory spreads the budget helps a model that follows it avoid these errors, but not to link components that are close in the vertex order, which good memories also need. When answering from a given memory, models without reasoning often misread large memories and the full graph.

2 Dependency models and memory

We represent the dependencies by a finite simple directed acyclic graph $G = (V, E)$, with labeled vertices $V = [n]$. A request $q = (u, v, F)$ specifies a source u, a target v, and a set of unavailable components $F \subseteq V \setminus \{u, v\}$. Both endpoints are protected from failure. A dependency model attaches mechanisms to the edges and assigns each request an answer. A graph memory retains a subgraph $H \subseteq G$ with $|E(H)| \leq B$, and an ideal reader answers with the same model on $H - F$ (Figure 1). Coded memories are treated in Appendix C. Below we define five dependency models; the last, linear effects, serves as a contrast.

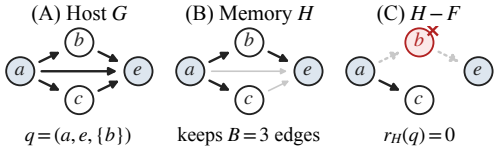

Figure 1: A request on a graph memory. (A) The request $q = (a, e, \{b\})$ asks whether e can be reached from a when b is unavailable; in the host, $r_G(q) = 1$. (B) A memory that keeps three of the five edges (dropped edges in gray). (C) In $H - F$ no route leads from a to e, so $r_H(q) = 0$.

OR workflows (Example 1). A vertex is active if any parent is active. Failures act as interventions (Pearl, 2009), and activating only u gives the answer $r_G(q) = \mathbf{1}\{u \rightsquigarrow v \text{ in } G - F\}$. The loss is $\mathbf{1}\{r_H(q) \neq r_G(q)\}$.

AND workflows (Example 2). Every vertex needs all of its parents. A request asks whether v can still be produced when F is unavailable, so the answer is $\mathbf{1}\{F \cap \mathrm{An}_G(v) = \varnothing\}$, where $\mathrm{An}_G(v)$ is the ancestor set of v, and the loss is 0–1. OR and AND are the extreme Boolean threshold rules (Rosenkrantz et al., 2022).

Noisy-OR networks (Example 3). Noisy-OR networks are a standard model of unreliable causes (Pearl, 1988). Each edge transmits activation independently with probability $\gamma \in (0, 1)$. The answer is the probability $R_G(u, v)$ that u activates v, with $F = \varnothing$, and the loss is $R_G(u, v) - R_H(u, v)$ averaged over pairs $u < v$.

Linear causal models (Example 4). Latent variables follow a Gaussian linear structural equation model $Z = B^\top Z + \varepsilon$ with $\mathrm{supp}\, B \subseteq E(G)$, observed through an unknown invertible mixing $X = MZ$. The data come from an observational environment and one soft intervention per node, which changes the node's incoming weights and scales its noise variance by $\kappa \neq 1$. Such data identify the latent graph only up to its transitive closure (Squires et al., 2023; Zhang et al., 2023), so models on the closures of $G - F$ and $H - F$ serve as truth and memory. A request asks for the distributions of X under $do(Z_F = 0)$, and the loss $D_k^{\mathrm{lin}}(H; G)$ is the smallest summed KL divergence, averaged over i.i.d. $N(0, s^2)$ weights and uniform k-subsets F.

Linear effects. The linear model is observed directly with $F = \varnothing$, and the answer is the total effect $\tau_G(u, v) = [(I - B)^{-1}]_{uv}$. The memory stores H with any weights, and $\mathcal{L}(H; G)$ is the squared error with true weights i.i.d. $N(0, s^2)$.

3 How much memory each model needs

3.1 Exact memory for OR workflows

OR workflows admit a combinatorial description of exact memory; the other models follow at the end of this subsection. Figure 2 abstracts Example 1: two backup paths make the direct connection unnecessary when at most one mediator can fail, and if both can fail, the direct connection must be retained.

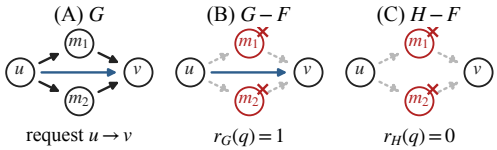

Figure 2: Without (u, v), the memory H loses both backup paths when $F = \{m_1, m_2\}$ fails (crosses; failed paths dotted). The edge is optional at $k = 1$ and necessary at $k = 2$.

We compare workflows by asking whether they give the same answer to every request when at most k components can fail, where $0 \leq k \leq n - 2$. Let $\mathscr{F}_k$ contain all vertex sets of size at most k, and let $\mathcal{Q}_k$ contain all requests (u, v, F) with distinct endpoints and $F \in \mathscr{F}_k$ excluding both endpoints. The response map $S_k(G) = (r_G(q) : q \in \mathcal{Q}_k)$ collects these answers. We call two graphs deployment-equivalent, written $G \equiv_k J$, when their response maps agree.

To answer every request correctly, a graph memory must preserve reachability after each allowed failure. For a fixed failure set F, the transitive reduction $\mathrm{TR}(G - F)$ is the smallest subgraph with the same reachability as $G - F$. We form the core by taking the union of these necessary edges over

all allowed failures:

$$E(\mathrm{Core}_k(G)) = \bigcup_{F \in \mathscr{F}_k} E(\mathrm{TR}(G - F)). \tag{3.1}$$

We call an additional edge optional if adding it to the core changes no answer. This holds when a path in the core connects its endpoints after every allowed failure. The saturation $\mathrm{Sat}_k(C)$ contains C together with all such optional edges:

$$E(\mathrm{Sat}_k(C)) = \{(u, v) : u \neq v,\ u \rightsquigarrow v \text{ in } C - F \text{ for every allowed } F\}. \tag{3.2}$$

When writing inclusions or cardinalities of graphs, we refer to their edge sets.

Theorem 3.1 (Minimum compatible graph)**.** *For every labeled DAG and $0 \leq k \leq n-2$, $\mathrm{Core}_k(G)$ has response map $S_k(G)$, is reconstructed from that map, and is contained in every compatible graph. Thus*

$$G \equiv_k J \iff \mathrm{Core}_k(G) = \mathrm{Core}_k(J). \tag{3.3}$$

Hence $|E(\mathrm{Core}_k(G))|$ is the exact edge budget for graph memory. The answers leave optional edges unresolved: the graphs with a given response map are exactly those between the core and its saturation (Theorem B.1 in Appendix B).

An edge becomes necessary when the allowed failures can block every alternative path between its endpoints. After removing $e = (u, v)$, let $\nu_G(e)$ be the maximum number of alternative u–v paths that share no internal vertices. By the directed version of Menger's theorem, $\nu_G(e)$ is also the minimum number of internal vertices whose failure blocks all alternative paths.

Corollary 3.2 (Failure threshold of an edge)**.**

$$e \in \mathrm{Core}_k(G) \iff \nu_G(e) \leq k, \qquad \mathrm{Core}_0(G) \subseteq \cdots \subseteq \mathrm{Core}_{n-2}(G) = G. \tag{3.4}$$

With r backup paths, the direct edge in Figure 2 enters the core precisely at $k = r$. Complete structural proofs are in Appendix B, which also shows that requests from one fixed source need not determine a unique minimum (Appendix B.1).

Exact memory in the other models. To leading order in the weight scale, the causal memory needs exactly $\mathrm{Core}_k(G)$ (Theorem 3.7), because its data certify reachability and nothing finer. AND workflows need even less.

Proposition 3.3 (AND memory)**.** *For every DAG and $1 \leq k \leq n-1$, a subgraph $H \subseteq G$ answers every AND request with $|F| \leq k$ correctly if and only if $\mathrm{TR}(G) \subseteq H$.*

An AND target fails exactly when some unavailable component is one of its ancestors. Every answer therefore depends only on which vertices are ancestors of which, and the transitive reduction already records this. Single failures reveal it, so allowing more failures adds nothing (Appendix H.1).

The two remaining models sit at the other extreme. In a noisy-OR network every edge raises the probability that one endpoint activates the other, and in a linear model every weight enters some total effect; both need all of G. An edge can thus be dropped for one of two reasons: parallel paths back it up, as in the OR and causal models, or a longer path already implies it, as in AND workflows.

3.2 Memory cost in ordered random workflows

Under the ordered random-graph model $G = G_{n,p}$ (Simon, 1988; Simon et al., 1993), each forward edge $i \to j$, $i < j$, is present independently with probability $0 < p \leq 1/2$. Write $N = \binom{n}{2}$ and $\Sigma_k = S_k(G)$. An optimal coded memory needs $H(\Sigma_k) = h_2(p)\mathbb{E}|\,\mathrm{Core}_k\,|/p$ bits up to one bit (Appendix C).

Theorem 3.4 (Uniform core law)**.** *With $L_p = \ln(e/p)$, uniformly in $n \geq 2$, $0 < p \leq 1/2$ and $0 \leq k \leq n-2$,*

$$\mathbb{E}|\,\mathrm{Core}_k\,| = \Theta(\min\{n^2 p, n(k + 1 + L_p)\}). \tag{3.5}$$

A present edge is in the core when at most k vertices block its alternatives, which is unlikely beyond spans of order $(L_p + k)/p$ (Appendices D–E). Without failures the core has only $\Theta(nL_p)$ edges; Table 1 gives the cost in the other models.

3.3 Loss when fewer edges are stored

OR workflows. Below the core budget, some requests must be answered incorrectly. To measure average loss, fix $\mu = \mu_{n,k}$ independently of G: choose $u < v$ uniformly, then a uniform k-subset of $V \setminus \{u, v\}$. This distribution samples allowed requests with protected endpoints.

For an integer edge budget $B \geq 0$, write $D_\mu(H; G) = \mathbb{E}_\mu \mathbf{1}\{r_H(q) \neq r_G(q)\}$. The smallest achievable loss for a given graph, and its average over graphs, are

$$D^*_\mu(B, G) = \min_{\substack{H \subseteq G \\ |E(H)| \leq B}} D_\mu(H; G), \qquad \overline{D}(B) = \mathbb{E}_G D^*_\mu(B, G). \tag{3.6}$$

Removing edges can destroy paths but cannot create them, so every error is a false negative. To reach a target loss, let $B_\epsilon = \min\{B : \overline{D}(B) \le \epsilon\}$ and $b_\epsilon(G) = \min\{B : D^*_\mu(B, G) \le \epsilon\}$. The first averages over graphs and the second holds on the observed graph.

Proposition 3.5 (The same zero-loss boundary)**.** *For every DAG whose edges respect the given order and $0 \le k \le n-2$, $D^*_\mu(B, G) = 0$ if and only if $B \ge |\operatorname{Core}_k(G)|$.*

Theorem 3.6 (Optimal budget for small average loss)**.** *Let $G = G_{n,p_n}$, $0 < p_n \le 1/2$, $\rho_n = k_n/(n-2) \to \rho \in (0,1)$, and $0 < \epsilon_n \to 0$. Write $t_n = \ln(1/\epsilon_n)$. If*

$$\frac{t_n \ln^2(n/\epsilon_n)}{np_n\epsilon_n \ln(t_n + e)} \longrightarrow 0, \tag{3.7}$$

then

$$B_{\epsilon_n} = (1 + o(1))\frac{nt_n}{\ln(1/\rho)}, \qquad \frac{b_{\epsilon_n}(G)}{nt_n/\ln(1/\rho)} \xrightarrow{\mathbb{P}} 1. \tag{3.8}$$

The optimization ranges over all subgraphs of original edges chosen after observing the host graph.

In the stated regime, the expected per-graph budget for preserving every answer is $\mathbb{E}b_0(G) = \Theta(n^2p)$, while vanishing average error is achievable with $o(n^2p)$ stored edges. Section 3.4 sketches the proof. We now ask how this law changes in the other models.

Causal models. The causal data identify the latent graph only up to its transitive closure, because mixing a latent variable with its ancestors leaves the data unchanged. A missing edge therefore costs nothing while its endpoints stay connected, and the causal loss should count exactly the requests that the OR memory answers wrongly. For small weights it does, up to a factor that depends only on n, k, and κ. Write $U(G) = \mathbb{P}_\mu(r_G(q) = 0)$ for the probability that the host itself disconnects a request.

Theorem 3.7 (Causal memory reduces to OR memory)**.** *Fix G, $H \subseteq G$, $\kappa \ne 1$, and $0 \le k \le n-2$. As $s \to 0$,*

$$D^{\mathrm{lin}}_k(H; G) = s^2\, c_{n-k}(\kappa)\, \lambda_{n,k}\, N\, D_\mu(H; G) + o(s^2), \tag{3.9}$$

where $\lambda_{n,k} = \binom{n-2}{k}/\binom{n}{k}$ and $c_d(\kappa) = (d-1)/[(\kappa+1)^2 + 2\kappa(d-1)]$. Hence $D^{\mathrm{lin}}_k(H; G)/D^{\mathrm{lin}}_k(\varnothing; G) \to D_\mu(H; G)/(1 - U(G))$, and the leading term vanishes exactly when $\operatorname{Core}_k(G) \subseteq H$.

In words, the normalized causal loss is the OR loss divided by $1 - U(G)$, the share of requests that the host itself connects. On the random hosts of Theorem 3.6 almost every request is connected, so $U(G) \to 0$ and the causal model needs the same budget, $(1 + o_{\mathbb{P}}(1))\, nt_n/\ln(1/\rho)$ (Appendix H.2).

Noisy-OR networks. In noisy-OR networks no component is unavailable; the uncertainty lies in the links instead. Each stored edge fails to transmit with probability $1 - \gamma$, just as a mediator fails with probability about ρ in the OR model. If all d stored out-edges of a source fail, it activates nothing, so the lower-bound argument carries over; a hub construction gives the upper bound, which is larger by a factor of at most two. Let $U_\gamma(G)$ be the average probability that u fails to activate v in G, and $L_\gamma = \ln(1/(1-\gamma))$.

Theorem 3.8 (Noisy-OR memory)**.** *If G is a DAG whose edges respect the vertex order and $H \subseteq G$ has average loss at most ϵ, then $|E(H)| \ge \{(n-1)[\ln(1/(\epsilon + U_\gamma(G))) - 1 + \ln 2] - O(\log n)\}/L_\gamma$. For $G = G_{n,p}$ with fixed p, γ, and $\epsilon_n \to 0$ with $n\epsilon_n/\log^4 n \to \infty$, the smallest such budget lies between $(1 - o_{\mathbb{P}}(1))\, nt_n/L_\gamma$ and $(2 + o_{\mathbb{P}}(1))\, nt_n/L_\gamma$.*

Linear effects. Up to now every request has asked whether v can be reached, and for that question one surviving path is as good as another. A linear-effect request asks instead how large the effect of u on v is. The total effect sums, over all u–v paths, the product of the weights along each path, so a parallel path adds its own share rather than serving as a backup. Refitting the remaining weights cannot make up for a lost path, because the effects T determine the weights through $B = I - T^{-1}$. Each omitted edge therefore costs the same amount, no matter which edges are kept.

Theorem 3.9 (Linear effects)**.** *Fix G and a DAG H with edges in the same vertex order. As $s \to 0$, the best weights on H give $\mathbb{E}\,\mathcal{L}(H; G) = s^2|E(G) \setminus E(H)| + o(s^2)$. The smallest budget with normalized loss at most ϵ is therefore $\lceil (1-\epsilon)|E(G)| \rceil$.*

Table 1 puts the five models side by side. In the OR, causal, and noisy-OR models, a few edges per vertex suffice, because the chance that all backups of a vertex fail together falls exponentially in their number. Linear effects offer no backups, and their budget stays close to the full edge set.

Table 1: Memory needed by each model on ordered random hosts, to leading order in the regimes of Theorems 3.6–3.9; $t = \ln(1/\epsilon)$ and $L_p = \ln(e/p)$.

Model	Exact memory	Its expected size	Budget for average loss ϵ
OR	$\mathrm{Core}_k(G)$	$\Theta(\min\{n^2p,\ n(k+1+L_p)\})$	$nt/\ln(1/\rho)$
Causal	$\mathrm{Core}_k(G)$	same as OR	$nt/\ln(1/\rho)$
Noisy-OR	G	$n^2p/2$	between nt/L_γ and $2nt/L_\gamma$
AND	$\mathrm{TR}(G)$	$\Theta(nL_p)$	—
Linear effects	G	$n^2p/2$	$(1-\epsilon)\lvert E(G)\rvert$

3.4 Proof sketch for the OR budget

Lower bound. If all d stored outneighbors of a source fail, the memory answers zero for every other target (Figure 3A). This happens with probability about ρ^d, so $d \approx \ln(1/\epsilon_n)/\ln(1/\rho)$, and a weighted degree argument handles uneven allocations.

Upper bound. We cut the vertex order into equal blocks and retain d random matchings of host edges between adjacent blocks (Figure 3B), so every vertex has d incoming and d outgoing edges within a budget of nd. Averaged over hosts, the matchings behave like independent random permutations. A request then fails if an endpoint keeps too few live neighbors, which has probability about ρ^d, or if the growing searches from the two endpoints die out or miss each other, which is exponentially less likely. Markov's inequality carries the guarantee to typical hosts, and condition (3.7) keeps the remaining errors negligible (Appendix G).

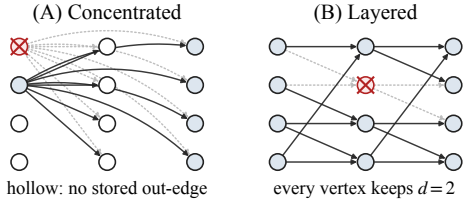

Figure 3: Two memories with the same budget. (A) Edges concentrated on two hubs leave most vertices without a stored out-edge, and one failed hub cuts off many requests. (B) The construction behind Theorem 3.6: each vertex keeps d edges to the next group, so routes branch and rejoin around a failure.

4 Empirical studies

Section 3 describes the best memory for an ideal reader on large random graphs. We first check whether its predictions already hold on graphs of 24 to 64 vertices, including real citation graphs, using memories built by algorithms (Section 4.2). We then give the same task to three language models, first on its own and then with a summary of the theory (Sections 4.3 and 4.4), and test whether they can read a memory correctly (Section 4.5).

4.1 Protocol

Each experiment separates writing a memory from reading it. The *writer*, an algorithm or a language model, sees the dependency graph, the request distribution, and a budget B. It returns a ranked list of edges, and the memory H consists of the first B entries; an entry that is not an edge of the host is wasted but still counts toward B. The *reader* answers each request by exact reachability in $H - F$, as in Section 3.3, so the loss depends only on which edges the writer kept (Figure 1).[1] We report the *normalized loss* $D_\mu(H;G)/D_\mu(\varnothing;G)$: it is one for an empty memory and zero for a memory that answers every request correctly.

Hosts and requests. We use six hosts of each of three kinds. The main random hosts are ordered random graphs $G_{64,0.3}$ with about 600 edges; budgets range from one to four edges per vertex ($d = B/n$), so even the largest keeps at most 42% of the edges. Smaller random graphs $G_{24,0.5}$ are used to compare request types and to test readers. The real hosts are citation graphs from ogbn-arxiv (Hu et al., 2020): for each of six computer science categories, we take the 64 most densely connected papers and draw an edge from each paper to every paper that cites it (Table 5). On every host and for every failure count k, we draw 2,000 test requests from $\mu_{n,k}$, and write $\rho = k/(n-2)$.

Reference memories. We compare the models with three memories built without a model. *Greedy* is the best memory we can compute: it adds, one at a time, the host edge that corrects the most of 2,000 separate training requests. Since it is a feasible memory, its loss is an upper bound on $D^*_\mu(B,G)$. *Balanced* gives each vertex its d nearest successors in the order, in the spirit of the layered

[1] Labels are arbitrary strings in the experiments; Appendix I.2 gives the full prompt.

construction behind Theorem 3.6. *Uniform random* is the chance baseline: it keeps B host edges drawn at random, and we average its loss over 20 draws.

Models and statistics. The writers are DeepSeek V4 Pro, GLM 5.2, and Muse Glimmer 30B, run at temperature zero with the reasoning settings of Appendix K. Each model writes one memory per host, budget, and condition. A reply that cannot be parsed after one retry counts as an empty memory, which happened for 4% of writes. We report means over the six hosts with 95% bootstrap intervals. We fixed every comparison and its pass or fail rule before calling any model. Appendix I.4 lists all of them, including those that fail; we do not correct for multiple comparisons.

4.2 The theory on graphs of moderate size

We first check the theory with memories built by algorithms (Figure 4). On 384 hosts small enough to enumerate, the smallest budget with zero loss always equals $|\,\mathrm{Core}_k(G)|$, as Proposition 3.5 requires (Appendix J.1). On larger hosts, greedy memories stand in for the optimum.

Failures set the rate. Theorem 3.6 predicts that each additional edge per vertex multiplies the loss by about ρ. In Panel A, a smaller ρ gives faster decay, and after the first step the decay rate is close to $\ln(1/\rho)$ (Table 7). The constants differ, as expected at $n = 64$, which lies outside the theorem's regime.

Backups can be compressed; independent contributions cannot. Panel B compares request types on the same hosts. AND memories reach zero loss once they contain the transitive reduction, and OR memories exploit backups. For linear effects, every reference memory we tried has loss $1 - B/|E|$, whatever its shape, as Theorem 3.9 predicts.

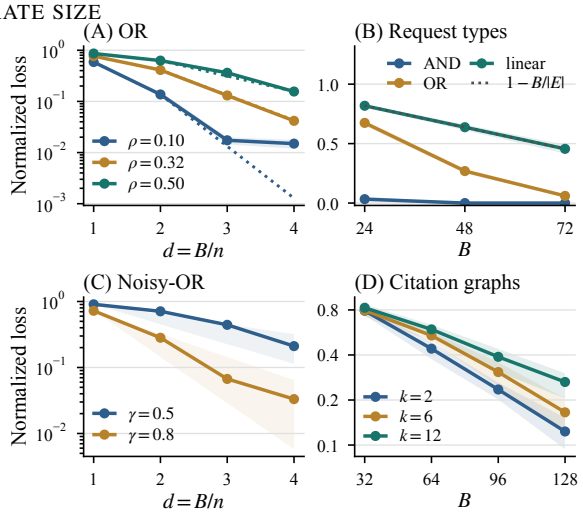

Figure 4: Reference memories follow the theory. (A) OR memories on $G_{64,0.3}$; dotted lines show the predicted rate $\ln(1/\rho)$. (B) Request types on $G_{24,0.5}$. (C) Noisy-OR networks; shaded regions show the predicted rates. (D) Citation graphs with k failures. Bands are 95% bootstrap intervals over hosts.

Reliable links need fewer edges. In noisy-OR networks (Panel C), the loss decays at a rate consistent with Theorem 3.8, and faster when links are more reliable.

Real graphs need backups. The average-case theory assumes many alternative routes. Flight connections from TravelPlanner (Xie et al., 2024) and the Gene Ontology (Ashburner et al., 2000) have few, so nearly every edge is needed even without failures (Appendix I.6). Citation graphs have many, because authors often cite a paper together with the work it builds on. There, as predicted, the loss falls more slowly as more papers fail (Panel D). We now ask whether language models find such memories.

4.3 How language models allocate memory

Models do no better than random edges. On random graphs, all three models stay far from greedy (Figure 5A). Their loss decays several times more slowly than greedy's, and none beats uniform random edges. GLM is even worse than random at $d = 2$, for every failure count. Citation graphs show the same pattern (Figure 5B). Averaged over failure counts and budgets, the models' normalized loss is about 0.89, against 0.84 for random edges and 0.46 for greedy.

Their errors are avoidable isolation errors. Lemma G.2 bounds the loss of any memory by the chance that an endpoint is *isolated*, meaning that every stored out-neighbor of the source, or every stored in-neighbor of the target, is unavailable. This chance is smallest when the budget is spread evenly over the vertices, and that minimum is the unavoidable loss. All other errors occur when routes survive from both ends but do not meet. Figure 5C separates the two kinds. The models tend to put most of their budget on a few vertices, as in Figure 3A, so isolation becomes their main source of error. On random graphs this is clearest for GLM, which stores outgoing edges for only about a

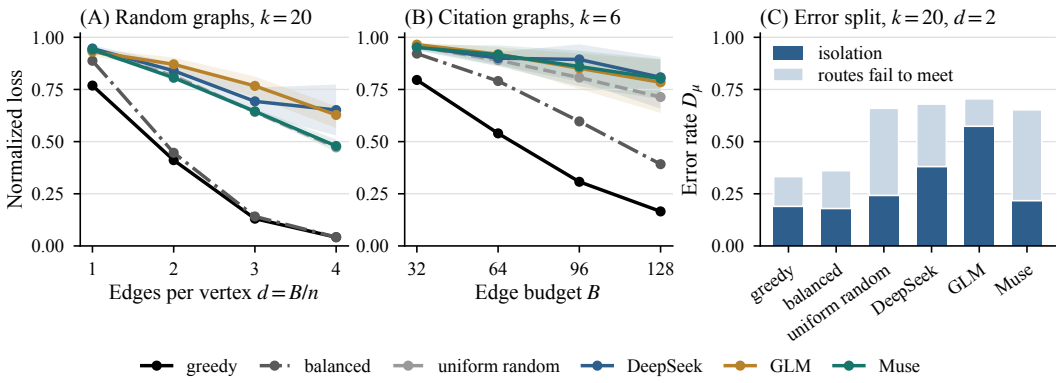

Figure 5: Model-written memories. (A, B) Losses stay far from greedy and no lower than those of uniform random edges (bands: 95% intervals over hosts). (C) Errors split into isolation errors (Lemma G.2) and failures of the surviving routes to meet.

third of the vertices that have any; most of its errors are isolation errors. On citation graphs, isolation accounts for 80% to 97% of the errors of all three models, compared with about half for random edges (Table 8). In the language-processing graph, for example, 117 of GLM's 128 links leave just three well-known papers, one of them the SQuAD dataset paper. The models seem to keep what looks important, and the lower bound shows that such an uneven allocation is costly. Random edges are spread out by chance, so they rarely isolate an endpoint, although more of their routes fail to meet. Showing paper titles and years instead of arbitrary labels changed the loss by at most 0.02.

A positional shortcut. The edge list in every prompt is shuffled, so its first B lines form a uniformly random memory. On random graphs, almost all of Muse's stored edges come from these lines, and DeepSeek often does the same (Table 11). Muse matches random edges because it keeps whichever edges the list shows first.

4.4 Can the theory guide the writer?

If models fail because they concentrate their budget, telling them to spread it should help. We test this on random graphs (Table 2). The *guidance* prompt summarizes the theory: errors come from isolated endpoints, so keep about d outgoing and d incoming edges for every vertex. It also tells the model not to copy the start of the list. The *warning* prompt contains only this last instruction, so comparing the two prompts isolates the effect of the guidance.

Guidance removes isolation, but not the gap to random. Compared with the warning alone, the guidance lowered isolation errors and loss for DeepSeek and GLM; the small gain for Muse disappears once unusable replies are set aside (Appendix I.5). GLM followed the guidance closely. Relative to its original prompt, it more than doubled its coverage, made its out-degrees more even than those of random memories (Table 9), and sharply reduced its isolation errors. Yet its loss only reached that of random edges, and no model did better.

The missing ingredient is locality. The construction behind Theorem 3.6 does more than balance degrees: it links vertices that are close in the order (Figure 3B), so that short hops chain into long routes. Balance alone helps little. At $k = 20$ and $d = 2$, giving each vertex d random successors improves only slightly on uniform random edges, whereas giving it its d nearest successors nearly matches greedy (0.45 against 0.41). Under guidance, the models' edges spanned about as many positions of the order as random edges, roughly four times as many as balanced edges (Table 8). The prompt does not show the order, and no model recovered it, so the guidance could improve only the balance.

On real graphs, general advice is not enough. In citation graphs the order is visible through publication years, so we added a hint to prefer links between papers published close in time. The models kept more local links, but only Muse lowered its loss, and only slightly; it still matched random edges. The best memories also look different here: the balanced construction falls far behind greedy (0.68 against 0.46), and greedy memories do not have even degrees. The zero-loss boundary and the isolation bound hold on every graph, but the layered construction is optimal only on random hosts. A good memory for a real graph has to be found by optimizing over that graph, and general

Writer	Prompt	Loss	Isolation	Coverage	Copied
Uniform random	–	0.71	0.23	0.80	0.27
DeepSeek	original	0.77	0.39	0.62	0.70
	warning	0.90	0.65	0.35	0.38
	guidance	0.77	0.42	0.63	0.57
GLM	original	0.80	0.54	0.35	0.39
	warning	0.82	0.57	0.28	0.38
	guidance	0.71	0.31	0.80	0.68
Muse	original	0.71	0.24	0.76	0.94
	warning	0.72	0.29	0.67	0.80
	guidance	0.71	0.23	0.81	0.96

Table 2: Guidance spreads the budget but does not beat uniform random edges ($G_{64,0.3}$, averaged over k and d). Isolation is the part of the error rate D_μ due to isolated endpoints; coverage is the share of vertices with outgoing edges that keep one; copied is the share of stored edges taken from the first B lines of the list. Table 9 adds connectivity errors and the spread of out-degrees.

advice cannot do this. An earlier study suggests that feedback on a specific instance helps: when DeepSeek saw a failure witness for a specific fact, its accuracy in choosing which fact to protect rose from 50% to 77.5% (Appendix J.3).

4.5 Reading the memory

The analysis assumes an ideal reader. To test this assumption on $G_{24,0.5}$, each model answered requests on its own memories, on greedy memories of two sizes, and on the full graph (Table 10). *Fidelity* is the share of answers that agree with the ideal reader on the same memory. Muse, run with low reasoning effort, is nearly ideal, with fidelity of at least 0.97. DeepSeek and GLM, run without reasoning, agree with the ideal reader on only about 60% of requests given the full graph or a large greedy memory, and more often on their own memories. Because the reasoning setting differs between models, we cannot tell whether the model or the setting drives the gap. In either case, the theory's budgets help only a reader that can follow the stored routes, as the earlier studies in Appendix J.2 also found.

5 Related work and scope

Fault-tolerant reachability preservers address selected source–target pairs (Baswana et al., 2018; Chakraborty et al., 2022), and robust transitive closure describes a related maximum (Mazaheri et al., 2026). Interventional Markov equivalence compares distributions under intervention targets (Hauser & Bühlmann, 2012). Our noisy-OR bounds are a directed analogue of reliable backbones (Chechik et al., 2012), and our layered memory follows multistage fault-tolerant routing (Leighton & Maggs, 1992; Leighton et al., 1998).

Our memories can also be read as partial world models, which let an agent predict the consequences of actions without taking them (Ha & Schmidhuber, 2018). Language models have served as world models for planning (Hao et al., 2023) and have been paired with learned causal world models that they query in text (Gkountouras et al., 2025). Agents that stay robust under distribution shift must learn a causal model (Richens & Everitt, 2024), and agents that generalize across multi-step goals must contain a predictive one (Richens et al., 2025). We ask how much of such a model must be kept under a memory budget when parts of the world may fail; for interventions $do(Z_F = 0)$ it is the fault-tolerant core. Learning the graph, which we take as given, is the task of causal representation learning (Schölkopf et al., 2021). Soft interventions identify it only up to transitive closure (Squires et al., 2023; Zhang et al., 2023), which is why our causal memory reduces to OR memory; general environments identify the graph but the latent variables only up to a surrounded-node ambiguity (Jin & Syrgkanis, 2024).

Theorem 3.6 assumes a failure fraction strictly between 0 and 1 and a stated range of density and error, and the causal and linear-effect results are leading order in the weight scale. Models may take shortcuts (Geirhos et al., 2020): because they copy from the top of the list, the edge order in our prompts is shuffled, and a fixed order would bias such comparisons. In the allocation tasks of Appendix P, we compare the value of information (Howard, 1966) with the reward the models obtain, and our conclusions concern the models, prompts, and graphs studied.

6 Conclusion

What an agent must store is set by what it can recover. For known OR workflows, exact memory is the fault-tolerant core, and small average error needs about $n \ln(1/\epsilon)/\ln(1/\rho)$ edges, with similar laws for the causal and noisy-OR models. Language models do not find such memories: they isolate endpoints and do no better than random edges, even when told how to spread the budget. Letting the agent simulate failures, as the greedy writer does, may close the gap.

Reproducibility Statement

The release contains complete proofs, the exhaustive loss calculations on small hosts, and the code that builds every host graph and request set from fixed seeds, computes the reference memories and all reported metrics, and calls the models. It also contains every model response from the studies in Section 4 and the plans we fixed before calling the models. Appendix I reproduces the prompts and lists every change to those plans. For the earlier studies in Appendix J, the available records support different checks, as explained in Appendix N.

AI Use Statement

Generative AI assisted with literature search, mathematical derivations, proof checking, synthetic-workflow implementation, analysis, figures, LaTeX, and editing. The source archive records the executable checks and complete proofs. The authors are responsible for every claim and artifact.

REFERENCES

Alfred V. Aho, Michael R. Garey, and Jeffrey D. Ullman. The transitive reduction of a directed graph. *SIAM Journal on Computing*, 1(2):131–137, 1972. doi: 10.1137/0201008.

Michael H. Albert and Alan M. Frieze. Random graph orders. *Order*, 6(1):19–30, 1989. doi: 10.1007/BF00341633.

Anthropic. Effective context engineering for AI agents. Anthropic Engineering, September 2025. URL https://www.anthropic.com/engineering/effective-context-engineering-for-ai-agents. Published September 29, 2025.

Anthropic. Scaling managed agents: Decoupling the brain from the hands. Anthropic Engineering, April 2026. URL https://www.anthropic.com/engineering/managed-agents. Published April 8, 2026.

Michael Ashburner, Catherine A. Ball, Judith A. Blake, David Botstein, Heather Butler, J. Michael Cherry, Allan P. Davis, Kara Dolinski, Selina S. Dwight, Janan T. Eppig, et al. Gene ontology: Tool for the unification of biology. *Nature Genetics*, 25(1):25–29, 2000. doi: 10.1038/75556.

Surender Baswana, Keerti Choudhary, and Liam Roditty. Fault-tolerant subgraph for single-source reachability: General and optimal. *SIAM Journal on Computing*, 47(1):80–95, 2018. doi: 10.1137/16M1087643.

Diptarka Chakraborty, Kushagra Chatterjee, and Keerti Choudhary. Pairwise reachability oracles and preservers under failures. In *49th International Colloquium on Automata, Languages, and Programming*, volume 229 of *Leibniz International Proceedings in Informatics*, pp. 35:1–35:16, 2022. doi: 10.4230/LIPIcs.ICALP.2022.35.

Shiri Chechik, Yuval Emek, Boaz Patt-Shamir, and David Peleg. Sparse reliable graph backbones. *Information and Computation*, 210:31–39, 2012. doi: 10.1016/j.ic.2011.10.007.

Thomas M. Cover and Joy A. Thomas. *Elements of Information Theory*. Wiley, 2 edition, 2006.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020. doi: 10.1038/s42256-020-00257-z.

John Gkountouras, Matthias Lindemann, Phillip Lippe, Efstratios Gavves, and Ivan Titov. Language agents meet causality – bridging LLMs and causal world models. In *International Conference on Learning Representations*, 2025.

David Ha and Jürgen Schmidhuber. Recurrent world models facilitate policy evolution. In *Advances in Neural Information Processing Systems*, volume 31, 2018.

Shibo Hao, Yi Gu, Haodi Ma, Joshua Hong, Zhen Wang, Daisy Wang, and Zhiting Hu. Reasoning with language model is planning with world model. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 8154–8173. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.emnlp-main.507.

Alain Hauser and Peter Bühlmann. Characterization and greedy learning of interventional markov equivalence classes of directed acyclic graphs. *Journal of Machine Learning Research*, 13:2409–2464, 2012.

Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963. doi: 10.1080/01621459.1963.10500830.

Ronald A. Howard. Information value theory. *IEEE Transactions on Systems Science and Cybernetics*, 2(1):22–26, 1966. doi: 10.1109/TSSC.1966.300074.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, volume 33, 2020.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=DT7JyQC3MR.

Jikai Jin and Vasilis Syrgkanis. Learning linear causal representations from general environments: Identifiability and intrinsic ambiguity. In *Advances in Neural Information Processing Systems*, volume 37, 2024.

F. Thomson Leighton and Bruce M. Maggs. Fast algorithms for routing around faults in multibutterflies and randomly-wired splitter networks. *IEEE Transactions on Computers*, 41(5):578–587, 1992. doi: 10.1109/12.142684.

F. Thomson Leighton, Bruce M. Maggs, and Ramesh K. Sitaraman. On the fault tolerance of some popular bounded-degree networks. *SIAM Journal on Computing*, 27(5):1303–1333, 1998. doi: 10.1137/S0097539793255163.

Bijan Mazaheri, Jiaqi Zhang, and Caroline Uhler. Relaxing faithfulness with intervention-only causal discovery. In *Proceedings of the Conference on Uncertainty in Artificial Intelligence*, 2026. To appear; arXiv:2607.11816.

Merav Parter and David Peleg. Sparse fault-tolerant BFS trees. In *Algorithms – ESA 2013*, volume 8125 of *Lecture Notes in Computer Science*, pp. 779–790. Springer, 2013. doi: 10.1007/978-3-642-40450-4_66.

Judea Pearl. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference*. Morgan Kaufmann, 1988.

Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2 edition, 2009.

Jonathan Richens and Tom Everitt. Robust agents learn causal world models. In *International Conference on Learning Representations*, 2024. URL https://arxiv.org/abs/2402.10877.

Jonathan Richens, Tom Everitt, and David Abel. General agents need world models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 51659–51687, 2025.

Daniel J. Rosenkrantz, Abhijin Adiga, Madhav Marathe, Zirou Qiu, S. S. Ravi, Richard Stearns, and Anil Vullikanti. Efficiently learning the topology and behavior of a networked dynamical system via active queries. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pp. 18796–18808, 2022.

Bernhard Schölkopf, Francesco Locatello, Stefan Bauer, Nan Rosemary Ke, Nal Kalchbrenner, Anirudh Goyal, and Yoshua Bengio. Toward causal representation learning. *Proceedings of the IEEE*, 109(5):612–634, 2021. doi: 10.1109/JPROC.2021.3058954.

Klaus Simon. An improved algorithm for transitive closure on acyclic digraphs. *Theoretical Computer Science*, 58(1–3):325–346, 1988. doi: 10.1016/0304-3975(88)90032-1.

Klaus Simon, Davide Crippa, and Fabian Collenberg. On the distribution of the transitive closure in a random acyclic digraph. In *Algorithms—ESA '93*, volume 726 of *Lecture Notes in Computer Science*, pp. 345–356. Springer, 1993. doi: 10.1007/3-540-57273-2_69.

Chandler Squires, Anna Seigal, Salil Bhate, and Caroline Uhler. Linear causal disentanglement via interventions. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 32540–32560, 2023.

Di Wu, Zixiang Ji, Asmi Kawatkar, Bryan Kwan, Jia-Chen Gu, Nanyun Peng, and Kai-Wei Chang. LongMemEval-V2: Evaluating long-term agent memory toward experienced colleagues. *arXiv preprint arXiv:2605.12493*, 2026. doi: 10.48550/arXiv.2605.12493. URL https://arxiv.org/abs/2605.12493. Work in progress.

Jian Xie, Kai Zhang, Jiangjie Chen, Tinghui Zhu, Renze Lou, Yuandong Tian, Yanghua Xiao, and Yu Su. TravelPlanner: A benchmark for real-world planning with language agents. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 54590–54613, 2024.

Yi Yu, Liuyi Yao, Yuexiang Xie, Qingquan Tan, Jiaqi Feng, Yaliang Li, and Libing Wu. Agentic memory: Learning unified long-term and short-term memory management for large language model agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 21457–21483. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.981.

Jiaqi Zhang, Chandler Squires, Kristjan Greenewald, Akash Srivastava, Karthikeyan Shanmugam, and Caroline Uhler. Identifiability guarantees for causal disentanglement from soft interventions. In *Advances in Neural Information Processing Systems*, volume 36, 2023.

A READER GUIDE TO THE SUPPLEMENT

The proofs establish how much memory an ideal reader needs to preserve its answers. The experimental appendices give the protocols, complete results, and supporting records for the model comparisons. The table below locates these materials and indicates what readers can check from the release.

Table 3: Theoretical claims and empirical studies, with locations and available evidence.

Claim or study	Where to audit	Release status
Exact memory (Theorem 3.1)	Appendix B; examples in Appendix L	Analytic proof
Loss under edge budgets	Appendix G	Analytic proof and exhaustive small-host calculations
Response entropy and coded memory	Appendices C–F	Analytic proofs
Causal, noisy-OR, and linear-effect budgets	Appendix H	Analytic proofs and numerical checks
Theory on graphs of moderate size, model writers, guidance, and readers	Appendix I	Code, prompts, and all responses
Reader accuracy with full, exact-core, and learned memories	Appendices J–M	Reported aggregates; response rows unavailable
Effect of recoverability evidence and answer labels on which fact a model protects	Appendices O.1–O.3	Released scored rows
Value of revealing the recovery state before allocation	Appendix P	One selected response per call; repeated deliveries unavailable

The proofs are ordered by their dependencies. Appendix I supports the studies of Section 4, and Appendix J presents earlier studies of exact memories and of recoverability evidence; their protocols and complete results follow in Appendices K–P.

The accompanying reproducibility documentation describes the software environment and verification procedures. Appendix N identifies which claims can be checked using the released materials.

B PROOFS OF THE DETERMINISTIC EQUIVALENCE-CLASS RESULTS

We use strict reachability: $u \rightsquigarrow v$ requires a nonempty directed path. Failure sets always exclude the queried endpoints. Theorem 3.1 is stated in the main text. We first record a companion result: the saturation describes every compatible graph.

Theorem B.1 (All compatible graphs)**.** *If C is reconstructed from a realizable response map σ, then*

$$\{G : S_k(G) = \sigma\} = \{G : C \subseteq G \subseteq \mathrm{Sat}_k(C)\}. \tag{B.1}$$

The $w = |\mathrm{Sat}_k(C) \setminus C|$ optional edges can be toggled independently, so this class contains 2^w graphs.

The answers determine the core but leave the presence of optional edges unresolved. A graph memory can omit all of these edges while preserving every allowed answer.

Lemma B.2 (Cover edges are forced)**.** *Let R be the reachability relation of a finite DAG on W. If uRv and there is no $z \in W \setminus \{u, v\}$ with uRz and zRv, then every directed graph on W with reachability relation R contains the edge (u, v).*

Proof. If the edge were absent, any path from u to v would have length at least two. Its first internal vertex z would satisfy uRz and zRv, a contradiction. □

Proof of Theorem 3.1. Fix F with $|F| \leq k$. Every edge of $\mathrm{TR}(G - F)$ belongs to the union $\mathrm{Core}_k(G)$ and has endpoints outside F, so

$$\mathrm{TR}(G - F) \subseteq \mathrm{Core}_k(G) - F \subseteq G - F.$$

The leftmost and rightmost graphs have equal reachability by definition of transitive reduction. Reachability is monotone under edge addition, so the middle graph has the same reachability relation. Hence $S_k(\mathrm{Core}_k(G)) = S_k(G)$.

For each failure set F, the corresponding entries of $S_k(G)$ determine the complete reachability relation of $G - F$ and hence its cover graph, $\mathrm{TR}(G - F)$. We can therefore reconstruct the core by taking the union over F.

To show necessity, let a DAG J on V have $S_k(J) = S_k(G)$. For each allowed F, $J - F$ realizes the same partial order as $G - F$, so Lemma B.2 gives $\mathrm{TR}(G - F) \subseteq J - F$. Taking the union over F gives $\mathrm{Core}_k(G) \subseteq J$.

The core is therefore equivalent to G and contained in every graph equivalent to G. Consequently, it is inclusion-minimal and has minimum cardinality among all graphs equivalent to G. If J is equivalent to G, then $\mathrm{Core}_k(J)$ is also equivalent to G because each core preserves its graph's response map. Applying the containment statement for G to $\mathrm{Core}_k(J)$ gives $\mathrm{Core}_k(G) \subseteq \mathrm{Core}_k(J)$. Interchanging G and J gives the reverse inclusion and hence equality of their cores. Conversely, equal cores have equal response maps for the same reason. □

Proof of Theorem B.1. Let C be reconstructed from the realizable response map σ.

If $S_k(G) = S_k(C)$, Theorem 3.1 gives $C \subseteq G$. Take any $(u, v) \in E(G) \setminus E(C)$ and any allowed failure set F excluding u, v. The direct edge survives in $G - F$, so $u \rightsquigarrow v$ there. Equality of response maps implies $u \rightsquigarrow v$ in $C - F$. Since this holds for every F, $(u, v) \in \mathrm{Sat}_k(C)$ and hence $G \subseteq \mathrm{Sat}_k(C)$.

Conversely, let $C \subseteq G \subseteq \mathrm{Sat}_k(C)$ and fix an allowed failure set F. Reachability in $C - F$ implies reachability in $G - F$. For the reverse direction, take a path in $G - F$. If it uses an added edge (x, y), its endpoints are outside F and saturation supplies an x–y path in $C - F$. Replace each added edge by such a path. The concatenation is a directed walk in the acyclic graph $C - F$, hence contains a directed path with the same endpoints. Therefore $C - F$ and $G - F$ have identical reachability for every F.

All saturation edges follow the failure-free partial order of C, so every subset between C and $\mathrm{Sat}_k(C)$ remains acyclic. There are exactly $2^{|\,\mathrm{Sat}_k(C) \setminus C|}$ such subsets. □

Proof of Corollary 3.2. Fix $e = (u, v) \in E(G)$. In $G - F$, the surviving direct edge is a cover relation exactly when $G - e - F$ has no u–v path. Thus e belongs to one of the transitive reductions in Equation (3.1) if and only if some set of at most k internal vertices meets every alternative u–v path. The directed vertex version of Menger's theorem equates the minimum cardinality of such a set with the maximum number $\nu_G(e)$ of internally vertex-disjoint alternative paths. This proves Equation (3.4). The separator F is also a distinguishing request: $r_G(u, v, F) = 1$ whereas $r_{G-e}(u, v, F) = 0$. Conversely, if no such F exists, then under every allowed failure set each use of e in a path of $G - F$ can be replaced by an alternative u–v path in $G - e - F$. Hence the two graphs agree on every request. Monotonicity in k is immediate. Every alternative path has internal vertices in $V \setminus \{u, v\}$, so deleting all at most $n - 2$ possible internal vertices distinguishes every present edge. □

B.1 A FIXED-SOURCE COUNTEREXAMPLE

With a single source, preserving every answer may require a choice among edges even when none of those edges is individually necessary. Let G have $k + 2$ parallel paths $s \to a_i \to v$. Under requests that always activate only s, no single final edge $a_i \to v$ is forced by the response map: after deleting it and at most k mediators, another path survives. Nevertheless every k-failure single-source reachability preserver must retain at least $k + 1$ final edges. Indeed, if such a preserver retained at most k final edges, deleting their corresponding mediators would eliminate all of its s–v paths, while an s–v path would still survive in G. Conversely, retaining every edge $s \to a_i$ and any $k + 1$ final edges is a preserver: after deletion of at most k mediators, at least one retained s–v path survives. Each edge $s \to a_i$ is forced by the request $(s, a_i, \varnothing)$. These preservers are therefore minimum, yet each final edge is omitted by at least one of them. Their intersection consequently contains no final edge and fails to preserve reachability. Under our all-pairs requests, however, $(a_i, v, \varnothing)$ distinguishes

each final edge, so every such edge is forced. Allowing these requests is what makes the unique minimum in Theorem 3.1 possible.

C CODED MEMORY AND RESPONSE ENTROPY

This appendix collects the exact results on coded memory, which rest on source coding (Cover & Thomas, 2006). Throughout, $G = G_{n,p}$ is the ordered random graph of Section 3 and $\Sigma_k = S_k(G)$ is its random response map.

A coded memory can represent every graph with the same response map by the same code. These graphs share their required present and absent edges and differ only in optional edges.

Theorem C.1 (Posterior and response entropy)**.** *For a realizable map σ with core C, the optional coordinates remain independent Bernoulli-p, while C is present and edges outside $\mathrm{Sat}_k(C)$ are absent. With $w_k(C) = |\mathrm{Sat}_k(C) \setminus C|$,*

$$\mathbb{P}(\Sigma_k = \sigma) = p^{|C|}(1-p)^{N-|\mathrm{Sat}_k(C)|}, \tag{C.1}$$

$$H(\Sigma_k) = h_2(p)\mathbb{E}[N - w_k(\mathrm{Core}_k(G))]. \tag{C.2}$$

To relate code length to core size, we ask when the answers reveal whether a given pair (u, v) carries an edge. Call the pair *pivotal* if, with the direct edge removed, at most k internal vertices can block every alternative u–v path; whether a pair is pivotal depends only on the other edges. Let $N_k^{\mathrm{fix}} = N - w_k(\mathrm{Core}_k(G))$ be the number of pivotal pairs.

Theorem C.2 (Pivotal coordinates)**.** $\mathbb{E}|\,\mathrm{Core}_k\,| = p\mathbb{E}N_k^{\mathrm{fix}}$, $H(\Sigma_k) = h_2(p)\mathbb{E}N_k^{\mathrm{fix}} = \frac{h_2(p)}{p}\mathbb{E}|\,\mathrm{Core}_k\,|$.

To count necessary edges, we group endpoint pairs by the number s of vertices between them. After suppressing the direct edge, let K_s be the maximum number of alternative paths that share no internal vertices. A present edge belongs to the core exactly when $K_s \le k$, because at most k failures can then block all its alternatives. The direct edge is present with probability p, independently of those alternatives, and there are $n - s - 1$ endpoint pairs with this separation. Hence

$$\mathbb{E}|\,\mathrm{Core}_k\,| = p\sum_{s=0}^{n-2}(n-s-1)\mathbb{P}(K_s < k+1). \tag{C.3}$$

Combining Theorem C.2 with Theorem 3.4 gives the entropy law. Uniformly in $n \ge 2$, $0 < p \le 1/2$ and $0 \le k \le n-2$,

$$H(\Sigma_k) = \Theta\big(\min\{n^2 h_2(p), n(k+1+L_p)h_2(p)/p\}\big), \tag{C.4}$$

$$H(\Sigma_k)/H(G) = \Theta(\min\{1, (k+1+L_p)/(np)\}). \tag{C.5}$$

For constant p and $k = o(n)$, response entropy is smaller than graph entropy by a factor of order $n/(k+1)$. When $k + 1 + L_p = \Omega(np)$, the two entropies have the same order.

For coded memory, we ask how many bits are needed to distinguish the possible response maps. An encoder stores a prefix-free program $P(G)$, and a public decoder uses it to answer requests. For a stored program z, let $\pi_z(q)$ be the probability that the decoder answers one to request q. We allow private randomness but require an error bound for every graph and request:

$$\sup_{G \text{ in support}} \sup_{q \in \mathcal{Q}_k} |\pi_{P(G)}(q) - r_G(q)| \le \eta. \tag{C.6}$$

If $\eta < 1/2$, two graphs requiring different answers cannot share the same program: the decoder would have to answer the same request both ways with probability greater than one half. The program therefore determines the entire response map, even though individual randomized answers may be wrong. This is why response entropy governs the memory requirement.

Theorem C.3 (Optimal prefix length)**.** *For any distribution on labeled DAGs and $0 \le \eta < 1/2$, the minimum expected program length satisfies $H(\Sigma_k) \le L_\eta^* < H(\Sigma_k) + 1$.* (C.7) *For a finite family $\mathfrak{G}$, optimal worst-case length is $\lceil \log_2 |\{S_k(G) : G \in \mathfrak{G}\}| \rceil$. For $\eta \ge 1/2$, a graph-independent constant-length program suffices.*

A prefix code attains the upper bound, and thresholding combined with source coding gives the lower bound (Appendix F).

Theorems C.1 and C.2 are proved in Appendix D.

D PROOFS FOR THE RANDOM-GRAPH BOUNDS

All entropies and logarithms in this appendix are base two unless $\ln$ is written explicitly. As in the main text, $S_k(g)$ is the deterministic map of a realized graph and $\Sigma_k = S_k(G_{n,p})$ is the corresponding random variable. We compute $H(\Sigma_k)$ by determining which edge coordinates the response map fixes. Whether a coordinate is fixed depends on the number of disjoint alternative paths between its endpoints. Grouping edges by span then expresses the expected core size as a sum, whose short-span terms give the lower bound used in Appendix E.

Proof of Theorem C.1. Theorem B.1 says that the equivalence class at σ consists exactly of graphs $C \cup A$, where A is an arbitrary subset of $W := \mathrm{Sat}_k(C) \setminus C$. Under the independent-edge law, the probability of this equivalence class is

$$p^{|C|}(1-p)^{N-|\,\mathrm{Sat}_k(C)|} \sum_{A \subseteq W} p^{|A|}(1-p)^{|W|-|A|} = p^{|C|}(1-p)^{N-|\,\mathrm{Sat}_k(C)|}.$$

Dividing by this probability gives conditional mass $p^{|A|}(1-p)^{|W|-|A|}$ for A, so the optional coordinates remain iid Bernoulli-p with conditional entropy $|W|h_2(p)$. Since Σ_k is a deterministic function of G,

$$Nh_2(p) = H(G) = H(\Sigma_k) + H(G \mid \Sigma_k).$$

Averaging the conditional identity proves Equation (C.2). □

D.1 PIVOTAL COORDINATES AND ARBITRARY-LENGTH PATH PACKING

Proof of Theorem C.2. For each possible forward edge $e = (i, j)$, suppress the edge and let K_e be the maximum number of internally vertex-disjoint i–j paths in the remaining graph. Put $B_e = \mathbf{1}\{K_e \le k\}$. This indicator is independent of the direct Bernoulli bit X_e.

The event $B_e = 1$ means that some set F of at most k internal vertices meets every alternative path. When $X_e = 1$, the edge e is a cover in $G - F$ and hence belongs to Core_k. When $X_e = 0$, $G - F$ disconnects its endpoints. Because the core preserves the response map, $\mathrm{Core}_k - F$ also disconnects them, so $e \notin \mathrm{Sat}_k(\mathrm{Core}_k)$. Thus the direct coordinate is fixed in either case. If $B_e = 0$, an alternative path survives every allowed failure. When $X_e = 1$, this path shows that $e \notin \mathrm{TR}(G - F)$ for every allowed F, and hence $e \notin \mathrm{Core}_k$. For either value of the edge indicator, the endpoints remain connected after every allowed failure. Because the core preserves these responses, $e \in \mathrm{Sat}_k(\mathrm{Core}_k)$, so the coordinate is optional. Consequently, pointwise,

$$N_k^{\mathrm{fix}} = N - w_k(\mathrm{Core}_k) = \sum_{e \in \mathcal{E}_n} B_e, \qquad |\,\mathrm{Core}_k\,| = \sum_{e \in \mathcal{E}_n} X_e B_e.$$

Independence gives $\mathbb{E}|\,\mathrm{Core}_k\,| = p\mathbb{E}N_k^{\mathrm{fix}}$. Equation (C.2) gives $H(\Sigma_k) = h_2(p)\mathbb{E}N_k^{\mathrm{fix}}$, proving both claims. □

To bound the contribution of alternative paths of every length, we group endpoint pairs by their span $s = j - i - 1$. Suppress the direct edge and let K_s have the distribution of the maximum number of internally vertex-disjoint source–sink paths through the s internal labels. We denote the probability of having too few such paths by

$$b_s^{(k)} := \mathbb{P}\{K_s \le k\}, \qquad a_s := b_s^{(0)}.$$

The direct edge is independent of the alternative graph, so

$$\mathbb{E}|\,\mathrm{Core}_k\,| = p \sum_{s=0}^{n-2} (n - s - 1) b_s^{(k)}.$$

Lemma D.1 (Occupation-time identity for failure-free reachability). *For $\bar{p} = 1 - p$,*

$$\sum_{s\geq 0} a_s = A(p) := \sum_{r\geq 1} \frac{\bar{p}^{r-1}}{1-\bar{p}^r} \leq \frac{C_0}{p} \ln \frac{e}{p}, \qquad 0 < p \leq \frac{1}{2}.$$

There is a universal $c_0 > 0$ such that $a_s \geq c_0$ whenever $0 \leq s \leq \ln(e/p)/(4p)$.

Proof. Expose the s internal labels in order and let Y_t count those reachable from the source after t exposures. The next label is reachable precisely when at least one of the $Y_t + 1$ fresh edges from the source or the reachable labels enters it. The final sink test likewise uses fresh edges from the reachable labels while excluding the suppressed direct source–sink edge. Thus, conditional on $Y_t = y$,

$$Y_{t+1} = Y_t + \text{Bernoulli}(1 - \bar{p}^{y+1}), \qquad a_s = \mathbb{E}\bar{p}^{Y_s}.$$

At state y, the residence time of this pure-birth chain is geometric with mean $(1 - \bar{p}^{y+1})^{-1}$. Each time index spent at that state contributes $\bar{p}^y$ to $\sum_{s\geq 0} \mathbb{E}\bar{p}^{Y_s}$. Tonelli's theorem therefore gives

$$\sum_{s\geq 0} a_s = \sum_{y\geq 0} \frac{\bar{p}^y}{1-\bar{p}^{y+1}} = \sum_{r\geq 1} \frac{\bar{p}^{r-1}}{1-\bar{p}^r} = A(p).$$

Let $\alpha = -\ln \bar{p}$. Since

$$\bar{p}A(p) = \sum_{r\geq 1} \frac{1}{e^{\alpha r} - 1},$$

we split the sum at $r = \lceil 1/\alpha \rceil$, using $e^{\alpha r} - 1 \geq \alpha r$ for the initial range and $(e^{\alpha r} - 1)^{-1} \leq (1 - e^{-1})^{-1} e^{-\alpha r}$ for the tail. Because $p \leq \alpha \leq 2p$ and $\bar{p}^{-1} \leq 2$, this proves the stated upper bound.

For the finite-window lower bound, $1 - \bar{p}^{y+1} \leq p(y + 1)$ implies

$$\mathbb{E}(Y_s + 1) \leq (1 + p)^s \leq e^{ps}.$$

The function $y \mapsto \bar{p}^y$ is convex and decreasing, so Jensen's inequality gives $a_s \geq \bar{p}^{\mathbb{E}Y_s} \geq \bar{p}^{e^{ps}}$. In the stated range, $-\ln(\bar{p})e^{ps} \leq 2e^{1/4}p^{3/4}$, which is uniformly bounded for $p \leq 1/2$. A universal positive c_0 follows. □

Lemma D.2 (Matching finite-window lower bound). *There is a universal $c > 0$ such that $\mathbb{E}|\operatorname{Core}_k(G_{n,p})| \geq c \min\{n^2 p, n(k + \ln(e/p))\}$.*

Proof. We first bound the contribution that is already present without failures, using $\operatorname{Core}_0 \subseteq \operatorname{Core}_k$. In the span sum, retain only the spans $0 \leq s \leq m_0$, where

$$m_0 = \min\left\{ \left\lfloor \frac{n-2}{2} \right\rfloor, \left\lfloor \frac{L_p}{4p} \right\rfloor \right\}.$$

Every retained span has $n - s - 1 \geq n/2$ endpoint pairs, and Lemma D.1 gives $a_s \geq c_0$. Furthermore, $\lfloor (n-2)/2 \rfloor + 1 \geq n/3$ for $n \geq 2$ and $\lfloor L_p/(4p) \rfloor + 1 > L_p/(4p)$, so

$$m_0 + 1 \geq c_0' \min\{n, L_p/p\}$$

for a universal $c_0' > 0$. The span decomposition (D.1) therefore yields, for every $n \geq 2$,

$$\mathbb{E}|\operatorname{Core}_k| \geq \mathbb{E}|\operatorname{Core}_0| \geq pc_0 \sum_{s=0}^{m_0} (n - s - 1) \geq c_1 \min\{n^2 p, nL_p\}. \tag{D.1}$$

To obtain the lower-bound contribution from the failure limit k, suppose $k \geq 1$ and let $U_s \sim \text{Bin}(s, p)$ count the internal out-neighbors of the source. Internally vertex-disjoint alternatives require distinct first internal vertices, so $K_s \leq U_s$. If $s \leq k/(2p)$, then

$$\mathbb{P}\{K_s \leq k\} \geq \mathbb{P}\{U_s \leq k\} \geq 1 - \frac{\mathbb{E}U_s}{k} \geq \frac{1}{2}.$$

Set

$$m_1 := \min\left\{ \left\lfloor \frac{n-2}{2} \right\rfloor, \left\lfloor \frac{k}{2p} \right\rfloor \right\}.$$

For $0 \le s \le m_1$, we have $n - s - 1 \ge n/2$. The inequalities $\lfloor (n-2)/2 \rfloor + 1 \ge n/3$ for $n \ge 2$ and $\lfloor k/(2p) \rfloor + 1 > k/(2p)$ also imply

$$m_1 + 1 \ge c_2 \min\{n, k/p\}$$

for a universal $c_2 > 0$. Summing these spans in (D.1) gives

$$\mathbb{E}|\operatorname{Core}_k| \ge \frac{p}{2} \sum_{s=0}^{m_1} (n - s - 1) \ge c_3 \min\{n^2 p, nk\}. \tag{D.2}$$

For $k = 0$, the right side of Equation (D.2) is zero.

To combine the contributions from short spans and the failure limit, set $U = n^2 p$, $V = nL_p$, and $W = nk$, and choose a universal $c > 0$ with $c \le \min\{c_1, c_3\}$. Combining Equations (D.1)–(D.2) then gives

$$\begin{aligned} \mathbb{E}|\operatorname{Core}_k| &\ge c \max\{\min(U, V), \min(U, W)\} \\ &= c \min\{U, \max(V, W)\} \ge \frac{c}{2} \min\{U, V + W\}, \end{aligned}$$

because $\max(V, W) \ge (V + W)/2$. This proves the lemma. □

E Proof of the uniform core-size bound

To bound the span sum in Equation (C.3), we need to control how often an endpoint pair has too few disjoint paths. We first show that the probability of having no path decays exponentially as the reachable set grows. For a graph with a small separator, we can bound from below the probability that independent deletion removes the whole separator, so the no-path estimate also bounds the event $K_s < r$. Summing over spans gives the upper bound. Together with Lemma D.2, this proves the core-size law, from which Theorem C.2 gives the entropy estimate.

Write $\bar{p} = 1 - p$. With the direct source–sink edge suppressed, let

$$a_s = \mathbb{P}\{K_s = 0\}$$

be the failure-free no-path probability. The proof below deletes internal vertices at random; given that m labels survive, the no-path probability is a_m, so we need a bound on a_m that holds for every m. Lemma D.1 controls only the sum $\sum_{s \ge 0} a_s$; the next lemma bounds each term exponentially.

E.1 A uniform no-path bound

Lemma E.1 (Uniform no-alternative-path bound)**.** *There are universal $c, C > 0$ such that*

$$\mathbb{P}\{K_s = 0\} \le \min\{1, \exp(CL_p - cps)\}. \tag{E.1}$$

Proof. Expose the s internal labels in increasing order and let Y_t be the number among the first t labels reachable from the source. As in the failure-free residence proof,

$$Y_0 = 0, \qquad Y_{t+1} = Y_t + \text{Bernoulli}(1 - \bar{p}^{Y_t + 1}), \qquad a_s = \mathbb{E}\bar{p}^{Y_s}.$$

The final identity uses only fresh edges from reachable internal vertices to the sink, with the direct source–sink edge still suppressed.

Set

$$R = \lceil 1/p \rceil, \qquad T_R = \inf\{t : Y_t \ge R\}.$$

At state $j - 1$, the holding time before the next birth is a geometric random variable G_j on $\{1, 2, \ldots\}$ with parameter $\lambda_j = 1 - \bar{p}^j$. Each newly inspected vertex exposes a fresh set of incoming edges, disjoint from those inspected earlier. The holding times are therefore independent, and

$$T_R = \sum_{j=1}^{R} G_j.$$

Put $\xi = p/8$. Since $\lambda_j \geq p$, we have $(1 - e^{-\xi})/\lambda_j \leq 1/8$. For a geometric variable of parameter λ,

$$\mathbb{E}e^{\xi G} = \frac{\lambda}{\lambda - (1 - e^{-\xi})},$$

and therefore

$$\ln \mathbb{E}e^{\xi G_j} \leq 2\frac{\xi}{\lambda_j}.$$

For $1 \leq j \leq R$, $pj \leq 1 + p \leq 3/2$, so

$$\lambda_j = 1 - (1 - p)^j \geq 1 - e^{-pj} \geq c_1 pj$$

for a universal $c_1 > 0$. Hence

$$\ln \mathbb{E}e^{\xi T_R} \leq 2\xi \sum_{j=1}^{R} \lambda_j^{-1} \leq Cp \sum_{j=1}^{R} \frac{1}{pj} \leq C_1 L_p.$$

Markov's inequality gives the hitting-time tail

$$\mathbb{P}\{T_R > s/2\} \leq \exp(C_1 L_p - ps/16). \tag{E.2}$$

For $s \geq 2$, on $\{T_R \leq s/2\}$ at least $\lfloor s/2 \rfloor \geq s/4$ inspections remain after the process has reached R. At every later state its birth probability is at least

$$\beta := 1 - \bar{p}^R \geq 1 - e^{-pR} \geq 1 - e^{-1} > 0.$$

Using fresh uniform random variables at each step, we can bound the number of later births from below by $B \sim \mathrm{Bin}(\lfloor s/2 \rfloor, \beta)$. Since $\bar{p} < 1$,

$$\mathbb{E}\big[\bar{p}^{Y_s} \mathbf{1}\{T_R \leq s/2\}\big] \leq \bar{p}^R \mathbb{E}\bar{p}^B \tag{E.3}$$

$$= \bar{p}^R (1 - \beta p)^{\lfloor s/2 \rfloor} \leq e^{-\beta p s/4}. \tag{E.4}$$

The contribution to a_s is bounded by the probability in Equation (E.2) on $\{T_R > s/2\}$ and by Equation (E.4) on $\{T_R \leq s/2\}$. Adding the contributions proves the exponential bound for $s \geq 2$. For $s = 0, 1$, the bound follows from $a_s \leq 1$ after increasing the leading universal constant. Taking the minimum with $a_s \leq 1$ proves Equation (E.1). □

E.2 From separators to path packing

Lemma E.2 (Random-deletion path-packing bound)**.** *Independently delete each of the s internal labels with probability $\delta \in (0, 1)$. If $M_\delta \sim \mathrm{Bin}(s, 1 - \delta)$, then*

$$\mathbb{P}\{K_s < r\} \leq \delta^{-(r-1)} \mathbb{E}[\mathbb{P}\{K_{M_\delta} = 0 \mid M_\delta\}]. \tag{E.5}$$

Proof. For each realized graph with $K_s < r$, the directed vertex version of Menger's theorem gives an internal source–sink separator of size at most $r - 1$. Fix one by a deterministic tie-breaking rule and call it $F(G)$. Conditional on G, the probability that an independent thinning deletes every vertex in $F(G)$ is $\delta^{|F(G)|} \geq \delta^{r-1}$. On this event, deleting additional internal vertices cannot create a path. Averaging the conditional bound over G gives

$$\mathbb{P}\{\text{the thinned graph has no path}\} \geq \delta^{r-1} \mathbb{P}\{K_s < r\}. \tag{E.6}$$

Conditional on retaining $M_\delta = m$ labels, relabel them in their original order. The resulting graph has exactly the law of the ordered Bernoulli DAG with m internal vertices. Every forward edge among retained vertices, including edges from or to the fixed endpoints, is still an independent Bernoulli-p coordinate. The left side of Equation (E.6) is thus $\mathbb{E}a_{M_\delta}$, proving Equation (E.5). □

Lemma E.3 (Uniform path-packing tail)**.** *There are universal constants $c, C > 0$ such that, for every $0 < p \leq 1/2$ and integers $r \geq 1, s \geq 0$,*

$$t_s^{(r)} := \mathbb{P}\{K_s < r\} \leq \min\{1, \exp(C(r + L_p) - csp)\}, \tag{E.7}$$

and

$$\sum_{s \geq 0} t_s^{(r)} \leq C \frac{r + L_p}{p}. \tag{E.8}$$

Proof. Take $\delta = 1/2$ in Lemma E.2 and let $M \sim \mathrm{Bin}(s, 1/2)$. Lemma E.1 and the binomial generating function give

$$t_s^{(r)} \le 2^{r-1} e^{C_0 L_p} \mathbb{E} e^{-c_0 p M} \tag{E.9}$$

$$= 2^{r-1} e^{C_0 L_p} \left(\frac{1 + e^{-c_0 p}}{2} \right)^s \le \exp(C(r + L_p) - csp), \tag{E.10}$$

uniformly on $0 < p \le 1/2$. Taking the minimum with the trivial bound $t_s^{(r)} \le 1$ proves Equation (E.7). Choose $s_0 = \lceil C_4(r + L_p)/p \rceil$ with C_4 sufficiently large. For every $s \ge s_0$, Equation (E.7) is at most $e^{-csp/2}$. The resulting geometric tail is $O(1/p)$, which is absorbed into $C(r + L_p)/p$ because $L_p > 1$. Hence

$$\sum_{s \ge 0} t_s^{(r)} \le s_0 + \sum_{s \ge s_0} e^{-csp/2} \le C \frac{r + L_p}{p}.$$

□

E.3 Completion of the uniform bound

Proof of Theorem 3.4. Put $r = k + 1$. Equation (C.3) gives

$$\mathbb{E}|\operatorname{Core}_k(G_{n,p})| = p \sum_{s=0}^{n-2} (n - s - 1) \mathbb{P}\{K_s < r\}.$$

Summing the tail bound from Lemma E.3 as in Equation (E.8) gives the upper bound $Cn(r + L_p)$. We also have $\operatorname{Core}_k(G) \subseteq G$, which bounds the expected core size by the expected number of host edges, $\binom{n}{2} p$. Lemma D.2 gives

$$\mathbb{E}|\operatorname{Core}_k| \ge c \min\{n^2 p, n(k + L_p)\}.$$

Since $r = k + 1$ and $L_p > 1$, $k + L_p = r + L_p - 1 \ge (r + L_p)/2$. Combining the two upper bounds with this lower bound proves Equation (3.5).

Finally, Theorem C.2 gives

$$H(\Sigma_k) = \frac{h_2(p)}{p} \mathbb{E}|\operatorname{Core}_k|.$$

Multiplying Equation (3.5) by $h_2(p)/p$ proves Equation (C.4). After adjusting the universal constants to cover $n = 2$, division by $H(G_{n,p}) = \binom{n}{2} h_2(p)$ proves Equation (C.5). At $\eta = 0$, Theorem C.3 gives $H(\Sigma_k) \le L_0^* < H(\Sigma_k) + 1$. Substituting Equation (C.4) locates the optimal expected lossless prefix length within one bit of the stated entropy scale. □

F Proof of the policy-memory theorem

Proof of Theorem C.3. Let P be any program satisfying Equation (C.6) for some $\eta < 1/2$. The public decoder determines the probability of answering one for each program and request. Since the correct answer has probability greater than one half, thresholding recovers every bit of the response map:

$$\Sigma(q) = \mathbf{1}\{\pi_P(q) > 1/2\}. \tag{F.1}$$

Indeed, $\pi_P(q) \le \eta < 1/2$ when $\Sigma(q) = 0$, whereas $\pi_P(q) \ge 1 - \eta > 1/2$ when $\Sigma(q) = 1$. Hence Σ is a deterministic function of P and $H(P) \ge H(\Sigma)$. For a prefix-free binary code, Shannon's converse (or Kraft's inequality and Gibbs' inequality) gives

$$\mathbb{E}|P| \ge H(P) \ge H(\Sigma).$$

For achievability, assign the possible values σ of Σ a Shannon code with length $\lceil -\log_2 \mathbb{P}(\Sigma = \sigma) \rceil$. It is prefix-free and has expected length strictly below $H(\Sigma) + 1$. The public decoder recovers σ and outputs $\sigma(q)$ deterministically, attaining zero error and therefore every $\eta \ge 0$.

At $\eta \geq 1/2$ the requirement is vacuous: one fixed program that outputs a fair coin has error exactly $1/2$ whatever the correct answer, so it serves every graph, and thresholding as in Equation (F.1) no longer recovers the map.

For the worst-case statement under $\eta < 1/2$, let $M = |S_k(\mathfrak{G})|$. Threshold reconstruction implies that distinct response maps require distinct programs. A prefix-free set of binary strings of length at most B has cardinality at most 2^B by Kraft's inequality, so $B \geq \lceil \log_2 M \rceil$. Conversely, assign the M response maps distinct fixed-length strings of length $\lceil \log_2 M \rceil$ and let the public decoder answer from the indexed response map. The singleton case uses the empty program. This attains zero error and proves equality. □

G Decision loss under an edge budget

Retaining the core is necessary and sufficient for zero loss, as stated in Proposition 3.5. To determine how much memory can be saved when errors are allowed, we derive finite-graph lower and upper bounds and then take limits to prove Theorem 3.6. Throughout, we select the memory H after observing G and before receiving the request, using only original edges and respecting the budget for every output. Write $M = n - 2$, $N = \binom{n}{2}$, $\rho_n = k/M$, and $\sigma_n = 1 - \rho_n$. We also denote the host's own disconnection probability by

$$U(G) = \mathbb{E}_\mu \mathbf{1}\{u \not\rightsquigarrow v \text{ in } G - F\}.$$

Since $H \subseteq G$, its decision loss is exactly its disconnection probability minus $U(G)$. The upper bound uses a randomized encoder. This suffices, because for each host some deterministic output does at least as well as the encoder's average.

G.1 Zero loss and the host's own disconnection probability

Proof of Proposition 3.5. If H contains $\mathrm{Core}_k(G)$, Theorem 3.1 gives equality of every supported response. If a core edge $e = (u, v)$ is missing, Corollary 3.2 gives a separator F_0 of size at most k for all alternatives to e in G. Since $k \leq n - 2$, we can pad F_0 to a k-subset of $V \setminus \{u, v\}$. The direct edge then survives in G, while no path survives in H. The pair is forward in the ordered host and its failure set has positive μ probability. Thus a zero-loss subgraph must contain the core. □

For the lossy random-host theorem, the vertex order is already the host's topological order. All finite estimates below use $n \geq 3$. At $k = n - 2$, only the two endpoints survive, so $D^*_\mu(B, G) = (|E(G)| - B)_+/N$. This endpoint case lies outside the theorem's assumption of an interior failure fraction.

Lemma G.1 (Host disconnection bound)**.** *There are universal $C, c > 0$ such that, when $\sigma_n > 0$,*

$$\mathbb{E}_G U(G) \leq \min\{1, C \ln(e/p)/(\sigma_n p n)\}. \tag{G.1}$$

Proof. With the direct endpoint edge suppressed, write $a_s = \mathbb{P}(K_s = 0)$ for the no-path probability across s internal labels. Lemma E.1 gives

$$a_s \leq \min\{1, \exp(C_2 \ln(e/p) - c_2 p s)\}. \tag{L1}$$

A uniform fixed-size failure set leaves a random number of internal labels, so we average this bound over the number of survivors.

For this purpose, let Z count survivors among s specified positions in a population of M vertices after a uniform k-subset is removed. For $0 \leq t \leq 1$,

$$\mathbb{E} t^Z \leq (1 - \sigma_n + \sigma_n t)^s. \tag{L2}$$

This is Hoeffding's comparison of sampling without and with replacement (Hoeffding, 1963, Theorem 4): Z counts survivors in a sample of s positions drawn without replacement, and $z \mapsto t^z$ is convex, so $\mathbb{E} t^Z$ is at most its value for $Z \sim \mathrm{Bin}(s, \sigma_n)$.

Conditional on the survivors, their edges remain independent Bernoulli-p. Apply (L1) and (L2), and use $1 - e^{-c_2 p} \geq c_3 p$ for $p \leq 1/2$, to bound no-path probability for a span-s query by $\min\{1, \exp(C_2 \ln(e/p) - c_3 \sigma_n p s)\}$. This also bounds the probability with the direct edge present, since suppressing it can only make disconnection more likely. There are $n - s - 1$ pairs with this span. Splitting the sum at $\lceil 2C_2 \ln(e/p)/(c_3 \sigma_n p) \rceil$ leaves a geometric tail of at most $C_3/(\sigma_n p)$. We then bound the number of pairs for each span by n and normalize by N, which proves (G.1). □

G.2 A CONVERSE FOR EVERY ADAPTIVE EDGE MEMORY

The lower bound uses the probability that all stored neighbors of a source fail. Define $q_j = (k)_j/(M)_j$ for $0 \le j \le k$ and $q_j = 0$ for $j > k$, including $j = n-1$. To bound these probabilities uniformly in the degree, for $1 \le T \le k$ set

$$\alpha = (k - T + 1)/M, \qquad c_n = \frac{(n-1)[(n-1)!]^{1/(n-1)}}{N}.$$

Lemma G.2 (Isolation converse)**.** *For every ordered host G and budget B,*

$$D^*_\mu(B, G) \ge c_n \alpha^{B/(n-1)} - \alpha^T - B/N - U(G). \tag{G.2}$$

Moreover, $c_n \to 2/e$.

Proof. Fix a feasible H, and let d_u be its outdegree. For each target $v > u$ that is not an outneighbor, all d_u outneighbors are eligible to fail, so deleting them disconnects the query with probability q_{d_u}. Discarding neighbor targets loses at most B/N, so

$$\mathbb{P}_\mu(u \not\rightsquigarrow v \text{ in } H - F) \ge \sum_{u<n} \frac{n-u}{N} q_{d_u} - B/N.$$

For $j < T$, each factor of q_j is at least α, and for $j \ge T$, $\alpha^j - \alpha^T \le 0$. Thus $q_j \ge \alpha^j - \alpha^T$ for every j. Because the weights sum to one, the subtraction contributes only one copy of the truncation term. We combine the remaining degree terms by applying the arithmetic–geometric mean inequality to the $n-1$ terms $(n-u)\alpha^{d_u}/N$, giving the lower bound $c_n \alpha^{\sum d_u/(n-1)} \ge c_n \alpha^{B/(n-1)}$. Subtracting $U(G)$ and minimizing over H now gives the claimed loss bound. To obtain the limiting constant, bound the logarithm of the factorial from above and below by integrals of $\ln x$. This gives $\ln((n-1)!)/(n-1) = \ln(n-1) - 1 + o(1)$ and hence $c_n \to 2/e$. □

G.3 EMBEDDING INDEPENDENT PERMUTATION LAYERS INTO THE HOST

To construct a memory that attains the lower bound, we build a network of matchings between consecutive layers. Fix positive integers d, L with $L \le n$, and partition the first QL vertices into $Q = \lfloor n/L \rfloor$ layers of size L. Leave the final fewer than L vertices unused. To generate the matchings between consecutive layers, put $q = 1 - (1-p)^{1/d} \ge p/d$ and couple each Bernoulli-p host indicator to the OR of d independent Bernoulli-q indicators, one for each color. For a present host edge, sample these bits conditional on their OR being one, and for an absent edge, set them all to zero. Averaging over the host makes the indicators for all colors independent.

For each color and each block between consecutive layers, independently choose a perfect matching uniformly from those available. If any block in any color has no perfect matching, output the empty graph. Otherwise, output the union H of the chosen matchings. Every outcome satisfies $|E(H)| \le dL(Q-1) \le dn$, since repeated edges only reduce the total. For this existence proof, we define the uniform choice by enumerating the available matchings.

If a Bernoulli-q bipartite block fails Hall's condition, there are s left vertices whose neighbors all lie among $s-1$ right vertices. For a large witness, we can transpose the block and take the complementary witness, so it is enough to consider $s \le \lceil L/2 \rceil$. There are at most L^{2s} witnesses of that size, each excluding at least $Ls/2$ independent edges. If $qL \ge \max\{8 \ln L, 4 \ln 2\}$, their sum is

$$2 \sum_{s \ge 1}^{\lceil L/2 \rceil} L^{2s} e^{-qLs/2} \le 2 \sum_{s \ge 1} e^{-qLs/4} \le 4e^{-qL/4}.$$

There are at most nd color/block pairs, so failure probability is at most $4nde^{-qL/4}$.

For a fixed color and block, permuting the right labels preserves both the graph's law and the uniform selection rule. These permutations act transitively on the matchings. Conditional on existence, its selected matching is therefore a uniform permutation. Independent colors, blocks and selection randomness remain independent conditional on their separate existence events. Consequently, conditional on all successes, the *annealed* network consists of d independent uniform permutations per block. This independence holds after averaging over hosts but need not hold conditional on a fixed G. Whether the construction succeeds depends on the host and the encoder's random choices, independently of the request failures.

G.4 Comparing fixed-size and independent failures

We first analyze independent failures, which are easier to handle in the layered construction, and then use a coupling to relate them to the prescribed fixed number of failures. For a fixed request pair and fixed H, take a uniform ordering of its M eligible vertices and an independent $K \sim \mathrm{Bin}(M, r)$, where $r = \rho_n + \tau < 1$ and $\tau > 0$. The first k positions form the required uniform failure set, while membership in the first K positions is iid Bernoulli-r. On $K \geq k$, the latter set contains the former, so

$$\mathbb{P}_\mu(H - F \text{ disconnects the pair}) \leq \mathbb{P}_{\mathrm{iid,r}}(H \text{ disconnects the pair}) + e^{-2M\tau^2}. \tag{G.3}$$

The error term is Hoeffding's bound on the lower tail of a binomial variable, $\mathbb{P}(K - \mathbb{E}K \leq -x) \leq e^{-2x^2/M}$ (Hoeffding, 1963), applied with $x = M\tau$. This comparison uses the fact that adding failures cannot restore a path in H. The loss relative to the host need not be monotone in the failure set.

G.5 A finite recovery bound for permutation layers

We grow reachable sets forward from the source and backward from the target, then join them across a fresh matching block. This can fail if too few neighbors survive at an endpoint, a small set fails to grow, a large set fails to persist, or the two searches fail to meet. We bound these four events separately.

Proposition G.3 (Permutation-layer recovery)**.** *Consider Q layers of width L with d independent uniform permutations between consecutive layers. With query endpoints protected, every other vertex fails independently with probability $r \in (0, 1)$. Fix $s_0 > 0$ with $1 - r \geq s_0$, and suppose*

$$d \geq 32/s_0, \quad L \geq 32ed, \quad x_c = \lfloor L/(16ed) \rfloor \geq r_0 = \lceil \sqrt{d} \rceil.$$

Choose $d/L \leq \zeta < 1$ with $m_0 = \lfloor (1-\zeta)d \rfloor \geq r_0$, and set

$$\begin{aligned} &g = s_0 d/4, && c = \min\{\ln 4/2, s_0/16\}, \quad c_b = c/(32e), \\ &\theta = s_0/(128e), && h = \lceil \ln(L/r_0)/\ln g \rceil + 4, \\ &I = \min\{1, (ed/(\zeta L))^{\zeta d} + (r_0+1)d^{r_0} r^{m_0 - r_0}\}. \end{aligned}$$

If $cdr_0 \geq \ln 2$ and the endpoint layers differ by at least $2h + 3$, then

$$\mathbb{P}(\textit{disconnection}) \leq 2I + 8e^{-cdr_0} + 4Qe^{-c_b L} + e^{-\theta^2 L}. \tag{G.4}$$

Proof. Starting from an endpoint. The d images of one vertex are iid uniform in the next layer. During sequential exposure, repetition has conditional probability at most d/L, so the repetition count is dominated by $\mathrm{Bin}(d, d/L)$. To obtain this domination, use a fresh uniform random variable for each exposure and compare it with the conditional threshold for repetition. For a binomial with mean λ, the exponential moment bound $\mathbb{E}e^{zX} \leq \exp(\lambda(e^z - 1))$ gives $\mathbb{P}(X \geq x) \leq (e\lambda/x)^x$ for real $x \geq \lambda$ by setting $e^z = x/\lambda$. Thus at least m_0 distinct images exist except with probability $(ed/(\zeta L))^{\zeta d}$. Conditional on $m \geq m_0$ distinct images, failures are independent. The probability of fewer than r_0 survivors is at most

$$\sum_{j < r_0} \binom{m}{j} (1-r)^j r^{m-j} \leq (r_0+1)d^{r_0} r^{m_0 - r_0}.$$

Combining the repetition and failure probabilities gives I. Applying the same argument to inverse permutations gives the bound at the target.

Growing a surviving set. Suppose an active set of size $x \leq x_c$ has been selected using only previous blocks. Expose its dx images in the next block. Within a permutation, each image is uniform among at least $L - x + 1$ unexposed positions. Because at most dx images have already been seen across all colors, each collision has conditional probability at most $2dx/L$. The preceding domination and tail bound give

$$\mathbb{P}(\text{at least } dx/2 \text{ collisions}) \leq (4edx/L)^{dx/2} \leq (1/4)^{dx/2}.$$

If this collision event does not occur, there are at least $dx/2$ distinct images. Their survival events are fresh and independent, so at least $s_0 dx/4 = gx$ vertices survive except with probability $e^{-s_0 dx/16}$.

Indeed, for a binomial of mean λ, applying Markov to e^{-zX} with $z = \ln 2$ gives $\mathbb{P}(X \le \lambda/2) \le \exp(\lambda((\ln 2)/2 - 1/2)) \le e^{-\lambda/8}$. It follows that each growth step fails with conditional probability at most $2e^{-cdx}$.

Starting with r_0 vertices, successful steps multiply size by at least $g \ge 8$ until x_c is reached. Along successful prefixes the failure probabilities sum to at most

$$\sum_{j\ge 0} 2e^{-cdr_0 g^j} \le \frac{2e^{-cdr_0}}{1 - e^{-cdr_0}} \le 4e^{-cdr_0},$$

using $g^j \ge j + 1$. At most $\lceil \ln(L/r_0)/\ln g \rceil + 1$ growth steps suffice. When the reachable set first crosses this threshold, we can retain exactly x_c vertices without exposing any future blocks.

Maintaining a large set. Since $L \ge 32ed$, $x_c \ge L/(32ed)$. One more growth step from x_c gives at least $s_0 d x_c/4 \ge \theta L$ live vertices, with failure probability at most $2e^{-c_b L}$. Also $\theta L \ge x_c$ under the assumed bound on d. Selecting exactly x_c known live vertices at every later layer therefore maintains at least θL live vertices, at the same error cost per step. At each step, choose the subset by label using only the history already exposed.

Joining the two searches. Before exploration, choose two adjacent middle layers using only the endpoint layer indices. The stipulated separation leaves at least h blocks on each side, enough for starting, growth and entry into the large-set stage. Explore forward from the source and backward from the target through disjoint layers and blocks. Their middle matching block remains unexposed. Conditional on live sets A, B of size at least θL, one fresh permutation misses all edges between them with probability

$$\frac{(L - |B|)_{|A|}}{(L)_{|A|}} \le (1 - |B|/L)^{|A|} \le e^{-\theta^2 L},$$

and the probability is zero when $|A| + |B| > L$. The two endpoint failures contribute at most $2I$, and the two growth phases contribute at most $8e^{-cdr_0}$. There are at most $2Q$ steps that maintain a large set, whose total failure probability is at most $4Qe^{-c_b L}$. Adding the probability that the bridge fails proves the claim. □

G.6 Finite host bound and the optimal asymptotic budget

For any parameters satisfying Proposition G.3 with $r = \rho_n + \tau$, $L \le n$, and $qL \ge \max\{8 \ln L, 4 \ln 2\}$, the construction above gives

$$\begin{aligned}\overline{D}(dn) \le \min\{1,\, 16(h+2)L/n + 4nde^{-qL/4} + e^{-2M\tau^2} + 2I \\ + 8e^{-cdr_0} + 4ne^{-c_b L} + e^{-\theta^2 L}\}.\end{aligned} \tag{G.5}$$

We exclude requests involving leftover vertices or endpoints too close for the two searches to grow. There are at most Ln pairs of the first kind and, for each source, at most $(2h+4)L$ later targets of the second kind. Dividing their total by N gives at most $16(h+2)L/n$. If fewer than $2h+4$ full layers exist this term already exceeds one. All other pairs satisfy the routing bound. We add the unconditional probability that the matching construction fails and the error from coupling the failure distributions. Conditioning on successful construction is valid because the failures are independent of the host and encoder randomness. Finally, the loss relative to the host is at most the disconnection probability in H, which proves (G.5).

Proof of Theorem 3.6. Write $t = \ln(1/\epsilon)$, $a_n = \ln(1/\rho_n)$, and let all parameters follow any sequence in (3.7). That condition implies $t = O(\ln n)$, $np \to \infty$, $L_p = O(\ln n)$, $t/n = o(\epsilon)$ and $L_p/(np) = o(\epsilon)$. Indeed, $np\epsilon \gg t \ln^2(n/\epsilon)/\ln(t+e)$ and $np \le n/2$ force $n\epsilon \to \infty$, whence $t < \ln n$ eventually. The right-hand side dominates both t and L_p.

For the lower bound choose $T = \lfloor n^{1/3} \rfloor$, eventually at most k. Then $\alpha = \rho_n(1 - O(n^{-2/3}))$, $\alpha^T = o(\epsilon)$, and $\alpha^{B/(n-1)} = (1 + o(1))\rho_n^{B/(n-1)}$ for $B = O(nt)$. For each fixed $0 < \eta < 1$ and $B \le (1-\eta)nt/a_n$, the main term in (G.2) is bounded below by a positive constant times $\epsilon^{1-\eta}$, whereas B/N and α^T are $o(\epsilon)$. Lemma G.1 gives $\mathbb{E}U = o(\epsilon)$, proving the graph-average lower

bound. Also, $\mathbb{P}(U(G) > \epsilon) \le \mathbb{E}U/\epsilon = o(1)$, so with probability $1-o(1)$ the host satisfies $U(G) \le \epsilon$, on which the pointwise lower bound exceeds ϵ.

For the upper bound fix $\eta > 0$ and choose

$$d = \lceil (1+\eta)t/a_n \rceil, \quad L = \lceil 128 d \ln(nd/\epsilon)/p \rceil, \quad \tau = \sqrt{4\ln(n/\epsilon)/(n-2)}, \quad \zeta = 1/\sqrt{\ln(L/d)}.$$

Here $r = \rho_n + \tau \to \rho \in (0,1)$, so a fixed $s_0 > 0$ applies. We have $d = \Theta(t)$ and $L = O(t\ln(n/\epsilon)/p)$. First use $1/p \le n$ eventually to obtain $\ln L = O(\ln(n/\epsilon))$. Then

$$\frac{L(1+\ln L/\ln d)}{n} = O\left(\frac{t\ln^2(n/\epsilon)}{np\ln(t+e)}\right) = o(\epsilon).$$

This estimate implies that $L < n$ eventually without having assumed it in the derivation. Since $h = O(1+\ln L/\ln d)$, it also makes the contribution from excluded pairs negligible. The inequalities $q \ge p/d$ and $L \le n$, together with the chosen constant 128, ensure the matching conditions and give $4nde^{-qL/4} = o(\epsilon)$. All routing hypotheses hold eventually, since $d \to \infty$ and $L/d^{3/2} \ge 128\ln(nd/\epsilon)/(p\sqrt{d}) \to \infty$.

The collision contribution to I is $\exp(-\omega(d))$, since its logarithm is $-d\sqrt{w} + O(d\ln w/\sqrt{w})$ with $w = \ln(L/d)$. The contribution from too few surviving neighbors has logarithm

$$-a_n d + O\Big(\zeta d + \sqrt{d}\ln d + d|\ln(\rho_n+\tau) - \ln\rho_n| + 1\Big) = -a_n d + o(d).$$

Thus $I \le \exp(-a_n d + o(d)) = \epsilon^{1+\eta+o(1)} = o(\epsilon)$. The small-set extinction term is $\exp(-\omega(d)) = o(\epsilon)$. Since $L/\ln(n/\epsilon) \ge 128 d/p \to \infty$, both the large-set and bridge terms are $o(\epsilon)$. The error from coupling the failure distributions is $(\epsilon/n)^8$. Equation (G.5) consequently gives mean loss $o(\epsilon)$ with a hard budget $dn = (1+\eta+o(1))nt/\ln(1/\rho)$.

For typical hosts, let $Z(G)$ be the conditional mean loss over the encoder's random choices. Every output is feasible, so $D^*_\mu(dn, G) \le Z(G)$. As $\mathbb{E}Z = o(\epsilon)$, Markov gives $\mathbb{P}(D^*_\mu(dn,G) > \epsilon) = o(1)$. The fixed fractional slack η makes the mean loss $o(\epsilon)$, as required for this application of Markov's inequality. Combining the upper and lower bounds for every fixed $\eta > 0$ gives (3.8). □

Similar layered constructions appear in classical work on fault-tolerant routing in random multistage networks (Leighton & Maggs, 1992; Leighton et al., 1998). The difference here is that the memory must be built inside a given random host, from its own edges and within a fixed budget. In the regime of Theorem 3.6, the resulting budget is asymptotically optimal among subgraphs of original edges.

G.7 Exhaustive finite-host calculations

To compute the exact loss curve, we sample 64 ordered Bernoulli hosts for each $(n,p) \in \{4,5,6\} \times \{0.25, 0.5\}$, with $k = \lfloor (n-2)/2 \rfloor$ and a fixed pseudorandom seed for each replicate. For every host, we enumerate all $2^{|E(G)|}$ original-edge subgraphs and all allowed requests: every size-k failure set and every surviving forward pair, with request weight $[\binom{n}{2}\binom{n-2}{k}]^{-1}$. For each budget we maximize the number of reachable requests over subgraphs of at most that size. Subtracting this count from the host's count and dividing by $\binom{n}{2}\binom{n-2}{k}$ gives exactly $D^*_\mu(B,G)$.

For the dashed comparison, we average uniformly over subgraphs with exactly $\min\{B, |E(G)|\}$ edges and then, as with the optimal curve, over 64 hosts. The host is the sampling unit, so requests from the same host are not treated as independent samples. Across 384 hosts, we evaluate 31,863 subgraphs and verify that the first zero-loss budget equals the number of edges whose individual removal changes some response, which is the core size by Theorem 3.1. We also check that loss is nonincreasing in the budget and that the optimum never exceeds the uniform-subset mean. The released host graphs and enumeration results support reproduction of these comparisons and Figure 9.

H Proofs for the other dependency models

This appendix proves Proposition 3.3 and Theorems 3.7, 3.8, and 3.9, and then places other dependency structures in the same framework. We write $N = \binom{n}{2}$ and use TR and Core_k as in Section 3. For an edge set P closed under transitivity, $\mathrm{An}_P(j) = \{i : (i,j) \in P\}$. For a DAG J, $\mathrm{TC}(J)$ denotes its transitive closure.

H.1 AND WORKFLOWS

Proof of Proposition 3.3. In an AND workflow with all sources available, v is produced exactly when every vertex on every path into v works, that is, when $F \cap \mathrm{An}_G(v) = \varnothing$. Since $H \subseteq G$, we have $\mathrm{An}_H(v) \subseteq \mathrm{An}_G(v)$. If $\mathrm{TR}(G) \subseteq H$, then H and G have the same transitive closure, hence the same ancestor sets and the same answers. Conversely, let $(a, b) \in \mathrm{TR}(G) \setminus E(H)$. The edge is the only a–b path in G, so $a \notin \mathrm{An}_H(b)$ while $a \in \mathrm{An}_G(b)$. The request with target b and $F = \{a\}$, which is allowed because $k \geq 1$, is answered one by H and zero by G. □

The errors of an AND memory are false positives: a missing ancestor makes the memory believe the target is still producible. For $k = 0$ every answer is one and the empty memory suffices. Under $G_{n,p}$ the exact budget $|\mathrm{TR}(G)| = |\mathrm{Core}_0(G)|$ has mean $\Theta(\min\{n^2 p, n(1 + L_p)\})$ by Theorem 3.4, independently of $k \geq 1$.

H.2 LINEAR CAUSAL MODELS

The model under one failure set. Let J be a DAG on a vertex set W of size $d \geq 2$, with edges respecting the vertex order. A parameter $\theta = (B, \Omega, \{b^{(j)}, \omega^{(j)}\}_{j \in W}, M)$ has $\mathrm{supp}\, B \subseteq E(J)$, diagonal $\Omega \succ 0$, $\mathrm{supp}\, b^{(j)} \subseteq \mathrm{pa}_J(j)$, $\omega^{(j)} > 0$ and $M \in GL(d)$. Environment 0 uses (B, Ω). Environment $j \in W$ replaces column j of B by $b^{(j)}$ and Ω_{jj} by $\omega^{(j)}$. The observed covariance in environment e is $\Sigma^{(e)} = M(I - B^{(e)\top})^{-1}\Omega^{(e)}(I - B^{(e)\top})^{-\top}M^\top$. We write $\Theta(J)$ for all such parameters and $\mathcal{D}(J)$ for the set of covariance tuples they produce.

Lemma H.1 (Failures restrict the graph)**.** *Under $do(Z_F = 0)$, the law of $Z_{V \setminus F}$ in a model on G equals the law of the restricted model on $G - F$ in every environment $e \notin F$.*

Proof. For $i \notin F$ the equation of Z_i sums over parents a, and every parent in F contributes $B_{ai}Z_a = 0$. □

Thus, under failure set F, the data come from a model on $G - F$ with vertex set $V \setminus F$ and $d = n - k$ vertices. For each F we write $P_F = \mathrm{TC}(G - F)$ and $Q_F = \mathrm{TC}(H - F)$. Since $H \subseteq G$, $Q_F \subseteq P_F$. The true model on P_F has $B = sW$, $b^{(j)} = sW'_{\cdot j}$, $\Omega = I$, $\omega^{(j)} = \kappa$ and $M = I$, where the entries of W, W' on P_F are i.i.d. $N(0, 1)$. Write $\Delta = W' - W$. The loss under F is

$$\mathrm{L}_F = \mathbb{E}_{W,W'} \inf_{\theta' \in \Theta(Q_F)} \sum_e \mathrm{KL}\big(N(0, \Sigma_\theta^{(e)}) \,\|\, N(0, \Sigma_{\theta'}^{(e)})\big),$$

and $D_k^{\mathrm{lin}}(H; G) = \mathbb{E}_F \mathrm{L}_F$ with F uniform over k-subsets of V.

Canonical form. A canonical parameter on a DAG J makes the latent coordinates independent in environment 0, with variances D_{ii}. In environment j it sets $Z_j = c_j^\top Z + \varepsilon_j^{(j)}$ with $\mathrm{supp}\, c_j \subseteq \mathrm{An}_{\mathrm{TC}(J)}(j)$ and leaves the other coordinates unchanged. The mixing is any $M \in GL(d)$.

Lemma H.2 (Canonical form)**.** *Every model on a DAG J has the same data as a canonical parameter on J. If Q is transitively closed, the canonical parameters on Q produce exactly $\mathcal{D}(Q)$.*

Proof. Given $\theta \in \Theta(J)$, set $\tilde{Z} = (I - B^\top)Z$ in every environment. For $i \neq j$, the equation of Z_i is unchanged in environment j, so $\tilde{Z}_i = \varepsilon_i$. In environment j, $\tilde{Z}_j = d_j^\top Z + \varepsilon_j^{(j)}$ with $d_j = b^{(j)} - B_{\cdot j}$. Substituting $Z = (I - B^\top)^{-1}\tilde{Z}$ gives $\tilde{Z}_j = c_j^\top \tilde{Z} + \varepsilon_j^{(j)}$ with $c_j = (I - B)^{-1}d_j$. The entry $[(I - B)^{-1}]_{ia}$ vanishes unless $i = a$ or $i \rightsquigarrow a$ in J, and d_j is supported on $\mathrm{pa}_J(j)$, so c_j is supported on $\mathrm{An}_{\mathrm{TC}(J)}(j)$. The observation map becomes $\tilde{M} = M(I - B^\top)^{-1}$, common to all environments. When Q is transitively closed, $\mathrm{An}_Q(j) = \mathrm{pa}_Q(j)$, so every canonical parameter on Q is itself a model on Q with zero observational weights. □

For the truth, $d_j = s\Delta_{\cdot j}$ and $c_j = s\, t_j$ with $t_j = (I - sW)^{-1}\Delta_{\cdot j} = \Delta_{\cdot j} + O(s)$. Because W is strictly upper triangular, $(I - sW)^{-1}$ is a polynomial in s. The KL divergence is invariant when both arguments undergo the same invertible linear map, and the reader ranges over all mixings, so we may

work in canonical coordinates on both sides. With base matrices $\Lambda^{(0)} = I$ and $\Lambda^{(j)} = I+(\kappa-1)e_je_j^\top$, the truth is

$$\Sigma^{(0)} = I, \qquad \Sigma^{(j)} = \Lambda^{(j)} + s(e_jt_j^\top + t_je_j^\top) + s^2|t_j|^2e_je_j^\top.$$

A canonical reader parameter $(D', \{c_j, \omega'^{(j)}\}, M')$ gives $\Sigma'^{(e)} = M'S'^{(e)}M'^\top$ with $S'^{(0)} = D'$ and $S'^{(j)} = D' + (\omega'^{(j)} - D'_{jj} + c_j^\top D'c_j)e_je_j^\top + e_j(D'c_j)^\top + (D'c_j)e_j^\top$. Rescaling or flipping latent coordinates maps canonical parameters to canonical parameters with the same data.

KL near the identity. Let $R = \Sigma^{-1/2}\Sigma'\Sigma^{-1/2}$ with eigenvalues λ_i and $\varphi(\lambda) = \lambda^{-1} - 1 + \ln\lambda$. Then $\mathrm{KL}(N(0,\Sigma)\|N(0,\Sigma')) = \frac{1}{2}\sum_i \varphi(\lambda_i)$. Since $\varphi(1) = \varphi'(1) = 0$ and $\varphi''(1) = 1$,

$$\mathrm{KL} = \tfrac{1}{4}\|R - I\|_F^2 + O(\|R - I\|^3). \tag{H.1}$$

The function φ decreases on $(0,1)$, increases on $(1,\infty)$ and diverges at both ends, so $\varphi(\lambda) \geq c_0 \min\{(\lambda-1)^2, 1\}$ for some $c_0 > 0$. Hence $\mathrm{KL} \leq \eta \leq c_0/2$ implies $\|R - I\|_F \leq \sqrt{2d\eta/c_0}$.

Lemma H.3 (Localization). *Fix d, $\kappa \neq 1$, $Q \subseteq P$, the weights, and $K > 0$. There are $s_0, C > 0$ such that for $s < s_0$, every canonical reader parameter with $\sum_e \mathrm{KL} \leq Ks^2$ can be rescaled to have $D' = I$, $M' = I + sA$, $c_j = su_j$ and $\omega'^{(j)} = \kappa + s\tau_j$ with $\|A\|, \|u_j\|, |\tau_j| \leq C$.*

Proof. The KL bound gives $\Sigma'^{(e)} = \Lambda^{(e)} + O(s)$. Let $U = M'D'^{1/2}$. Then $UU^\top = \Sigma'^{(0)} = I + O(s)$, and $U^\top U$ has the same eigenvalues, so the polar decomposition gives $U = O + O(s)$ with O orthogonal. Let $R_j = D'^{-1/2}(S'^{(j)} - S'^{(0)})D'^{-1/2} = U^{-1}(\Sigma'^{(j)} - \Sigma'^{(0)})U^{-\top}$. Then $R_j = (\kappa-1)vv^\top + O(s)$ with $v = O^\top e_j$. By the canonical form, R_j vanishes outside the entry (j,j) and the entries $(i,j),(j,i)$ with $i \in \mathrm{An}_Q(j)$, all of which have $i < j$. The zero diagonal entries give $v_i^2 = O(s)$ for $i \neq j$, so $v_j^2 = 1 - O(s)$. The zero entries (i,j) with $i > j$ then give $v_i = O(s)$. Since $v_i = O_{ji}$, the strictly upper part of O is $O(s)$. Write $O = L + E$ with L lower triangular and $\|E\| = O(s)$. Then $O^{-1} = L^{-1} + O(s)$ is lower triangular up to $O(s)$, while $O^{-1} = O^\top$ is upper triangular up to $O(s)$. Hence O is a signed identity up to $O(s)$. After rescaling by $D'^{-1/2}$ and the signs, $D' = I$ and $M' = I + O(s)$. Reading $R_j = (\kappa-1)e_je_j^\top + O(s)$ entrywise gives $c_j = O(s)$ and $\omega'^{(j)} = \kappa + O(s)$. □

Lemma H.4 (One failure set). *With $g_d(\kappa) = (d-1)/\big(2[(\kappa+1)^2 + 2\kappa(d-1)]\big)$, for fixed weights*

$$\inf_{\theta' \in \Theta(Q)} \sum_e \mathrm{KL} = s^2 g_d(\kappa) \sum_{(i,j)\in P\setminus Q} \Delta_{ij}^2 + O(s^3),$$

and $\mathrm{L} = s^2 c_d(\kappa)|P \setminus Q| + o(s^2)$ *with* $c_d = 2g_d$.

Proof. In the local form of Lemma H.3, $\Sigma'^{(e)} - \Sigma^{(e)} = s\Gamma^{(e)} + O(s^2)$ with $\Gamma^{(0)} = A + A^\top$ and $\Gamma^{(j)} = A\Lambda^{(j)} + \Lambda^{(j)}A^\top + \tau_je_je_j^\top + e_j(u_j - t_j)^\top + (u_j - t_j)e_j^\top$. By (H.1), and since replacing $\Sigma^{(e)}$ by $\Lambda^{(e)}$ in the normalization costs $O(s^3)$,

$$\sum_e \mathrm{KL} = \frac{s^2}{4}\sum_e \sum_{a,b} \frac{(\Gamma_{ab}^{(e)})^2}{\Lambda_a^{(e)}\Lambda_b^{(e)}} + O(s^3)$$

uniformly over bounded local parameters. The diagonal entries involve only A_{aa} and τ and vanish at $A_{aa} = \tau = 0$. For $i < j$, the entry (i,j) involves only $\alpha = A_{ij}$, $\beta = A_{ji}$ and $u = u_j(i)$, which is free when $(i,j) \in Q$ and zero otherwise. Its value is $\alpha + \beta$ in environment 0 and in the $d-2$ environments outside $\{i,j\}$, $\alpha + \kappa\beta$ in environment i, and $\kappa\alpha + \beta + u - t$ in environment j, where $t = t_j(i)$. Hence the quadratic form splits over pairs with

$$f_{ij} = \tfrac{d-1}{2}(\alpha+\beta)^2 + \tfrac{1}{2\kappa}(\alpha + \kappa\beta)^2 + \tfrac{1}{2\kappa}(t - \kappa\alpha - \beta - u)^2.$$

If $(i,j) \in Q$, the choice $u = t$ gives zero. If $(i,j) \notin P$, then $t = 0$. If $(i,j) \in P \setminus Q$, write $f = \frac{1}{2}x^\top \mathbf{G}x - b^\top x + t^2/(2\kappa)$ with $x = (\alpha,\beta)$, $a = \kappa + \kappa^{-1}$, $m = d-1$, $b = t(1,\kappa^{-1})$ and $\mathbf{G} = \left(\begin{smallmatrix} m+a & m+2 \\ m+2 & m+a \end{smallmatrix}\right)$. Then $\det\mathbf{G} = (\kappa-1)^2\Phi/\kappa^2 > 0$ with $\Phi = (\kappa+1)^2 + 2\kappa m$, and a direct computation gives $b^\top \mathrm{adj}(\mathbf{G})b = t^2(\kappa-1)^2[m\kappa + (\kappa+1)^2]/\kappa^3$. So $\min f = t^2/(2\kappa) - \frac{1}{2}b^\top\mathbf{G}^{-1}b = t^2m/(2\Phi) = t^2g_d(\kappa)$.

The minimizing local parameters are bounded, which gives the upper bound $s^2 g_d \sum t_j(i)^2 + O(s^3)$. For the lower bound, take a reader parameter within s^3 of the infimum. By the upper bound its divergence is at most Ks^2, so Lemma H.3 applies and the divergence is at least s^2 times the minimum of the quadratic form, minus $O(s^3)$. Finally $t = \Delta + O(s)$. For the average, keeping the true canonical parameter and deleting its coefficients outside Q gives divergence $\frac{s^2}{2\kappa} \sum_j |t_j|_{P \setminus Q}|^2$, which is a polynomial in the Gaussian weights. Dominated convergence and $\mathbb{E}\Delta_{ij}^2 = 2$ finish the proof. □

Lemma H.5 (Averaging over failure sets)**.** *With $\lambda_{n,k} = \binom{n-2}{k} / \binom{n}{k}$ and F uniform over k-subsets of V, $\mathbb{E}_F |\mathrm{TC}(G - F) \setminus \mathrm{TC}(H - F)| = \lambda_{n,k} N D_\mu(H; G)$.*

Proof. A pair $u < v$ lies outside F with probability $\lambda_{n,k}$. Given this event, F is a uniform k-subset of $V \setminus \{u, v\}$, which is the conditional law under μ. Summing over the N pairs gives the claim, because $H \subseteq G$ makes every error a pair connected in $G - F$ and disconnected in $H - F$. □

Proof of Theorem 3.7. Lemma H.1 turns the model under F into a model on $G - F$. Lemma H.4 with $d = n - k$, $P = P_F$ and $Q = Q_F$ gives $\mathrm{L}_F = s^2 c_{n-k}(\kappa) |P_F \setminus Q_F| + o(s^2)$. Averaging over the finitely many F and applying Lemma H.5 gives (3.9). For $H = \varnothing$, $D_\mu(\varnothing; G) = 1 - U(G)$, which gives the ratio. The leading term vanishes exactly when $D_\mu(H; G) = 0$, which by Proposition 3.5 happens exactly when $\mathrm{Core}_k(G) \subseteq H$. □

Corollary H.6 (Causal budget)**.** *Let $b_\epsilon^{\mathrm{lin}}(G)$ be the smallest $|E(H)|$, $H \subseteq G$, with $\lim_{s \to 0} D_k^{\mathrm{lin}}(H; G) / D_k^{\mathrm{lin}}(\varnothing; G) \leq \epsilon$. Under the assumptions of Theorem 3.6, $b_{\epsilon_n}^{\mathrm{lin}}(G) / (n t_n / \ln(1/\rho)) \to 1$ in probability.*

Proof. By Theorem 3.7, $b_\epsilon^{\mathrm{lin}}(G) = \min\{|E(H)| : D_\mu(H; G) \leq \epsilon(1 - U(G))\}$, so $b_\epsilon(G) \leq b_\epsilon^{\mathrm{lin}}(G)$, and on $\{U(G) \leq \eta\}$ also $b_\epsilon^{\mathrm{lin}}(G) \leq b_{\epsilon(1-\eta)}(G)$. Condition (3.7) forces $np_n \geq \ln^2 n$ eventually, so Lemma G.1 gives $\mathbb{E}U(G) \to 0$. With $\eta_n = (\mathbb{E}U(G))^{1/2}$, Markov's inequality gives $\mathbb{P}(U(G) \leq \eta_n) \to 1$. The target $\epsilon_n(1 - \eta_n)$ changes t_n by $o(1)$ and the left side of (3.7) by a factor $1 + o(1)$, so (3.8) applies to both targets and the two bounds agree. □

Numerical check. For $n = 3$ with $Q = \{(1, 2), (1, 3)\}$, $\kappa = 2$, and for $n = 4$ with $Q = \{(1, 2), (3, 4)\}$, $\kappa = 1.5$, we minimized the divergence over all reader parameters, including the full mixing, by BFGS. At $s = 0.01$ the ratio of the minimum to the leading term of Lemma H.4 was between 0.992 and 1.0001 in five draws.

H.3 Noisy-OR networks

Each edge works independently with probability γ, and $r = 1 - \gamma$. For a graph K, let K_γ keep the working edges and $R_K(u, v) = \mathbb{P}(u \rightsquigarrow v \text{ in } K_\gamma)$. The loss of $H \subseteq G$ is $D_\gamma(H; G) = N^{-1} \sum_{u<v} (R_G(u, v) - R_H(u, v))$, and $U_\gamma(G) = N^{-1} \sum_{u<v} (1 - R_G(u, v))$.

Exact answers. If $e = (a, b) \in G \setminus H$, the event that H_γ has no a–b path has probability at least $r^{|E(H)|} > 0$ and is independent of e, so $R_G(a, b) - R_H(a, b) \geq \gamma r^{|E(H)|} > 0$.

Lower bound. Let d_u be the outdegree of u in H. If all d_u outgoing edges fail, u reaches nothing, so $R_H(u, v) \leq 1 - r^{d_u}$. With weights $w_u = (n - u)/N$, which sum to one,

$$\epsilon' := D_\gamma(H; G) + U_\gamma(G) \geq \sum_{u<n} w_u r^{d_u}.$$

Set $\lambda = \epsilon'/(n-1)$ and define a_u by $w_u r^{a_u} = \lambda$. Convexity of $x \mapsto r^x$ gives $r^d \geq r^a(1 - L_\gamma(d - a))$ for all real d, a. Multiplying by w_u and summing over $u < n$ gives $\epsilon' \geq \epsilon' - \lambda L_\gamma \sum_u (d_u - a_u)$, so $|E(H)| = \sum_u d_u \geq \sum_u a_u$. Since $(n-1)w_u = 2(n-u)/n$, Stirling's formula gives $\sum_{u<n} \ln((n - 1)w_u) = -(1 - \ln 2)(n - 1) + O(\log n)$, which proves the lower bound in Theorem 3.8.

Host disconnection. For a pair with s' vertices between its endpoints, the direct edge and the s' two-edge paths are independent, so $1 - R_G(u, v) \leq (1 - p\gamma)(1 - p^2\gamma^2)^{s'}$ for $G = G_{n,p}$. Summing over pairs gives $\mathbb{E}U_\gamma(G) \leq 2/(np^2\gamma^2)$.

Upper bound. Let $h = \lceil C_1 \log n \rceil$ with C_1 large, $s = \lceil h^2 \log n \rceil$ and $d = \lceil \ln(t_n/\epsilon_n)/L_\gamma \rceil$. Cut the vertex order into windows of size s and let the hub set K_i be the first h vertices of window i. The memory keeps (a) for each vertex of window i, edges to d host outneighbors in K_{i+1}; (b) for each vertex of window j, edges from d host inneighbors in K_{j-1}; and (c) every host edge between consecutive hub sets. It has at most $2dn + O(n/\log n) = (2 + o(1))nt_n/L_\gamma$ edges. With probability $1 - o(1)$, every vertex has at least $ph/2 \geq d$ host neighbors in the adjacent hub sets, and every hub has at least $ph/2$ host outneighbors in the next hub set.

Fix u in window i and v in window $j \geq i + 4$, and let A_ℓ be the hubs of K_ℓ reached from u. We average over the host and the link failures jointly. First, $A_{i+1} = \varnothing$ with probability r^d. Second, a reached hub has at least $ph/2$ edges of type (c) into K_{i+2}, so $|A_{i+2}| \geq p\gamma h/4$ except with probability e^{-ch}. The type (c) edges into K_ℓ are unexposed when $A_{\ell-1}$ is known, so given $A_{\ell-1}$ each hub of K_ℓ is missed independently with probability $(1 - p\gamma)^{|A_{\ell-1}|} \leq \eta := (1 - p\gamma)^{p\gamma h/4} = n^{-c'}$. By induction, every later hub set keeps at least $h/2$ reached hubs except with probability $o(n^{-2})$. The inneighbors of v in K_{j-1} are chosen through host edges independent of A_{j-1}, so v is missed with probability at most $(r + \gamma\eta)^d \leq r^d e^{d\gamma\eta/r} = r^d(1 + o(1))$. Each far pair therefore fails with probability at most $2r^d(1 + o(1)) + o(n^{-2}) \leq (2 + o(1))\epsilon_n/t_n$. Pairs with $j \leq i + 3$ make up a fraction $O(s/n) = o(\epsilon_n/t_n)$. Markov's inequality then bounds the loss by ϵ_n with probability $1 - O(1/t_n) - o(1)$.

Proof of Theorem 3.8. The exact and lower bounds are proved above for every DAG. For $G_{n,p}$, the host bound and Markov's inequality give $U_\gamma(G) \leq \epsilon_n/t_n$ with probability tending to one, since $n\epsilon_n/\log^4 n \to \infty$ and $t_n \leq \ln n$. Then $\ln(1/(\epsilon_n + U_\gamma)) = t_n - o(1)$, and the lower bound becomes $(1 - o(1))nt_n/L_\gamma$. The upper bound is the hub construction. □

The constant. The hub construction uses edges of type (a) and (b) that each serve one ordinary vertex. The layered matchings of Appendix G give every vertex d incoming and d outgoing edges with nd edges in total. In simulations with $n = 600$, $p = 0.5$, $\gamma \in \{0.5, 0.7\}$ and layers of 30 vertices, the error of the layered memory on distant pairs approached $2r^d$ as d grew, and it was far below the hub error at equal budget. We expect the recovery analysis of Proposition G.3, with r in place of ρ, to give the constant one.

H.4 Linear effects

Proof of Theorem 3.9. Write $T = (I - B)^{-1}$ and $T' = (I - B')^{-1}$ for the truth and for weights B' on H. Both are unit upper triangular, so the loss is $\|T - T'\|_F^2$. Keeping the true weights on $E(G) \cap E(H)$ gives $T' - T = s(W|_H - W) + O(s^2)$ and loss $s^2 \sum_{e \in G \setminus H} W_e^2 + O(s^3)$. For the lower bound, take B' with loss at most Ks^2. Since $\|T - I\|_F = O(s)$, also $\|T' - I\|_F = O(s)$. The matrix $N' = T' - I$ is nilpotent, so $B' = I - T'^{-1} = -\sum_{\ell \geq 1} (-N')^\ell$ and $\|B'\|_F \leq 2\|N'\|_F = O(s)$ for small s. Write $B' = sA'$ with A' bounded and supported on $E(H)$. Then $T' = I + sA' + O(s^2)$ and $T = I + sW + O(s^2)$, so the loss is $s^2\|W - A'\|_F^2 + O(s^3) \geq s^2 \sum_{e \in G \setminus H} W_e^2 - O(s^3)$. The upper bound is a polynomial in W, so dominated convergence gives $\mathbb{E}\mathcal{L}(H; G) = s^2|E(G) \setminus E(H)| + o(s^2)$. For $H = \varnothing$ this is $s^2|E(G)| + o(s^2)$. A budget with normalized loss at most ϵ must keep at least $(1 - \epsilon)|E(G)|$ edges of G, and keeping that many edges of G suffices. □

With failures, $do(Z_F = 0)$ restricts the effect matrix to $V \setminus F$, and the same argument gives loss proportional to $\mathbb{E}_F|(E(G) \setminus E(H)) \cap \binom{V \setminus F}{2}| = \lambda_{n,k}|E(G) \setminus E(H)|$, so the budget does not change. In a numerical check with $n = 6$, $|E(G)| = 8$ and $|E(H)| = 2$, BFGS over all weights on H from starting points of scale up to three gave ratios between 0.993 and 1.023 to the leading term for $s \in \{0.1, 0.03, 0.01\}$.

H.5 Other dependency structures

Many requests have the same shape: combine the values along each path ($\bigotimes$), then aggregate over all paths ($\bigoplus$). Table 4 places common agent dependencies in this frame. In the OR, noisy-OR, and causal models the aggregation is a union or a maximum, so a second path adds nothing while the first one survives, and parallel paths act as backups. Linear effects add over paths, so every path

contributes its own share. AND vertices do not fit this frame: they need every input, so the answer depends only on ancestry.

Table 4: Dependency structures that agents meet, with the corresponding requests and the status of our results.

Structure	Agent setting	Request	Status
OR	Alternative lookup routes (Example 1)	Reachable after failures?	Core (Theorem 3.1); budget $nt/\ln(1/\rho)$ (Theorem 3.6)
AND	Required approvals (Example 2)	Producible after failures?	Transitive reduction for every $k \geq 1$ (Proposition 3.3)
Noisy OR	Unreliable downloads (Example 3)	Success probability	Budget between 1 and 2 times nt/L_γ (Theorem 3.8)
Linear causal	Perturbation screens (Example 4)	Interventional distribution	Same as OR to leading order (Theorem 3.7)
Linear effects	Shock propagation in supply chains	Effect size	Almost no saving (Theorem 3.9)
AND–OR recipes	Tool chains with alternative sets of inputs; build systems; planning	Producible after failures?	A fact fails when F hits every recipe; the threshold $\tau(f)$ in Section J.3 generalizes $\nu_G(e)$. Budgets open
Threshold t of m	Quorums, majority votes, replicated storage	Enough inputs survive?	Interpolates between OR ($t = 1$) and AND ($t = m$). Open
Min-cost path	Cheapest or fastest recovery route	Best cost after failures	Idempotent aggregation, but the best path moves with F; single-source fault-tolerant BFS structures need $\Theta(n^{3/2})$ edges in the worst case (Parter & Peleg, 2013). Lossy budgets open
Most reliable path	Choosing one route by success rate	Best success rate	Idempotent; only the best path counts. Open
Capacity	Throughput and rate limits	Maximum flow	Additive along parallel paths but capped by cuts. Open
Stochastic delays	Plans with deadlines	Done in time?	Parallel branches combine by maximum, so extra branches never act as backups. Open
Inhibition	Circuit breakers, fallbacks triggered by failure	Any of the above	Removing edges can create answers, so the false-negative structure of Proposition 3.5 fails
Cycles, unknown graphs	Retries, feedback, learning while acting	Any of the above	Outside the known-DAG setting

I Details of the memory-allocation studies

This appendix supports Sections 4.1–4.5. It describes the hosts and reference memories, reproduces the prompts, lists every change to the plan we fixed before calling the models, reports every preregistered test, and gives results that the main text only summarizes.

I.1 Hosts, requests, and reference memories

Random hosts. All random hosts come from fixed seeds. Six hosts $G_{64,0.3}$ have 560 to 626 edges. We use $k \in \{6, 20, 31\}$, so that $\rho = k/62$ is about 0.10, 0.32, and 0.50, and budgets $B \in \{64, 128, 192, 256\}$, that is, $d = 1, \ldots, 4$. Six hosts $G_{24,0.5}$ have 115 to 150 edges and serve for request types ($k = 7$, $B \in \{24, 48, 72\}$) and for readers. Vertex labels are random strings of four capital letters, and both the edge list and the list of components are shuffled, so the prompt does not reveal the vertex order.

Citation hosts. For each of six arXiv categories in ogbn-arxiv (computation and language, computer vision, machine learning, cryptography and security, data structures and algorithms, and robotics), we take the papers of that category and the citations among them, and repeatedly remove a paper of smallest degree until 64 remain. An edge runs from a paper to each remaining paper that cites it, and vertices are ordered by year. The prompts show random labels together with titles and years taken from the dataset's title mapping, which contains a few mismatched titles that we left unchanged. Table 5 describes the six hosts. We use $k \in \{2, 6, 12\}$ and $B \in \{32, 64, 96, 128\}$.

Table 5: The six citation hosts. TR share is the fraction of edges in the transitive reduction; reachable is the share of requests with a positive answer on the host at $k = 6$.

Category	Edges	TR share	$\lvert \mathrm{Core}_2 \rvert$	$\lvert \mathrm{Core}_6 \rvert$	$\lvert \mathrm{Core}_{12} \rvert$	Reachable
Computation and language	723	0.26	364	545	676	0.62
Computer vision	1,077	0.15	375	616	840	0.78
Machine learning	748	0.22	338	535	712	0.69
Cryptography and security	437	0.36	293	413	434	0.38
Data structures and algorithms	566	0.24	288	456	551	0.65
Robotics	255	0.37	198	250	255	0.21

Requests and scoring. Each host and failure count has 2,000 test requests drawn from $\mu_{n,k}$, and greedy memories are built on a separate set of 2,000 training requests. For noisy-OR, the probabilities R_G and R_H are estimated from the same 2,000 random draws of working links, so that the comparison between host and memory is paired. For linear effects, the weights are hidden from the writer and the loss is averaged over 200 independent draws of weights with $s = 0.05$. With $s = 0.25$ the loss of every reference memory is at most 0.07 above $1 - B/|E|$.

Reference memories. Greedy adds the host edge that corrects the most training requests, breaking ties by the smallest edge in the vertex order. Because a new edge can only add paths, it corrects exactly the wrong requests in which the source reaches its tail and its head reaches the target in $H - F$, which lets us compute each step exactly and quickly. Balanced gives every vertex its nearest successors in the order, $\lfloor B/n \rfloor$ each, and hands out the remaining slots one per vertex. Uniform memories average 20 random draws of B host edges. For AND requests the reference stores the transitive reduction first, and for linear effects it is the best of the greedy, transitive-reduction-first, and uniform memories.

Models. DeepSeek V4 Pro and GLM 5.2 run without reasoning and Muse Glimmer 30B with low reasoning effort, all at temperature zero, as in Appendix K. Replies may use up to 16,384 tokens. A reply that fails, or cannot be parsed, is retried once; after that the memory is empty. Table 12 reports how often this happened.

I.2 PROMPTS

Every writer prompt has the same frame, in which the meaning of an edge and the request distribution are filled in for each request type.

```
You are preparing a persistent memory file for a future session of yourself.
In that session you will NOT see the dependency list below. You will see only
the memory file you write now. Your memory is limited to at most {B} lines,
each copying exactly one dependency from the list below.

{meaning of an edge and of the requests}

Components ({n}): {labels, shuffled}

Dependencies ({m}), one per line:
{edges, shuffled}

Choose at most {B} dependencies to keep, ordered from most to least important.
If you list more than {B}, only the first {B} are kept. Return only a JSON
object of the form {"edges": ["XXXX -> YYYY", ...]} with no other text.
```

For OR requests the meaning reads as follows; the AND and noisy-OR versions change only the rule for when a component is available and what a request asks.

```
Meaning: "X -> Y" means Y can be obtained from X. A component can be obtained
if at least one component pointing to it can be obtained. In the future
session, each request gives a source component (which you have), a target
component, and a set of exactly {k} unavailable components (never the source
or the target). The answer is "yes" if some chain of dependencies leads from
the source to the target without passing through an unavailable component.
Requests pick a source and a later target uniformly at random and pick the
{k} unavailable components uniformly at random. You will answer using only
the dependencies in your memory. Choose them to minimize the fraction of
requests you answer incorrectly.
```

For linear effects the weights are not shown:

```
Meaning: "X -> Y" means X has a direct linear effect on Y. Each dependency
has its own coefficient, which you are not shown: the coefficients are drawn
independently at random, with mean zero and the same small spread. The total
effect of X on Y is the sum, over all chains of dependencies from X to Y, of
the product of the coefficients along the chain. In the future session you
must report the total effect for every pair of components, computed from the
dependencies in your memory with their true coefficients. Choose them to
minimize the expected sum of squared errors over all pairs.
```

In the noisy-OR study, one version states the probability γ that each link works, and the other says only that links are unreliable. In Section 4.4, the warning and the guidance are added after the meaning, with $d = B/n$:

```
The dependency list below is in random order. Do not simply copy its first
lines; choose each dependency you keep deliberately.

Guidance from an analysis of this task: the main source of errors is a
request in which every kept dependency leaving the source, or every kept
dependency entering the target, passes through an unavailable component.
So spread your budget evenly: keep about {d} outgoing and about {d} incoming
dependencies for every component, as far as the list allows, instead of many
dependencies of a few components. Do not leave any component with no kept
outgoing dependency, or no kept incoming dependency, if the list has one.
```

The citation prompts use the same frame, describe a research assistant who must decide whether an idea can pass from one paper to a later one when some papers are unavailable, and list each paper as a label with its year and title. Their guidance adds one sentence: “Prefer links between papers that are close in publication time, so that the kept links form short steps that can be chained.” The reader prompt lists the memory, explains the OR rule, and asks for yes or no answers to 20 requests at a time.

I.3 CHANGES TO THE PLAN

We fixed a first plan before any model call and revised it after inspecting whether each test measured what the theory predicts. The revision changed four items, listed below, and we report only the revised versions.

- *Budget law.* The first plan used $G_{24,0.5}$, where budgets of three or four edges per vertex already keep 53% to 71% of all edges, which distorts the decay. We moved this study to $G_{64,0.3}$ and chose k to keep the same failure fractions.
- *Linear effects.* The first plan showed the weights to the writer. Theorem 3.9 averages over weights for a memory chosen from the graph alone, so that condition lies outside the theorem: keeping the edges with the largest weights loses only 0.19 at $B = 48$, far below $1 - B/|E| = 0.64$. We replaced it with the hidden-weight version.
- *Noisy-OR.* The first plan used two budgets and tested whether stating γ makes models keep more redundant edges. The construction in Theorem 3.8 does not depend on γ, so the theorem makes no such prediction. We moved the study to $G_{64,0.3}$ with four budgets and test the decay rate instead; the stated-γ comparison is descriptive.
- *Core alignment.* On $G_{24,0.5}$, Core_7 contains 98% of all edges, so alignment with the core carries little information. We report it only for the smallest failure count.

Three further changes were made during the study. Replies at $B = 256$ from GLM were cut off at 4,096 tokens in the smoke test, so we raised the limit to 16,384. The size cap for real hosts was raised from 600 to 1,100 edges after the scan of candidate tasks (Section I.6), because three citation hosts exceeded it; the cap served only to bound prompt length. A billing interruption at the provider stopped the citation run partway; the failed calls were discarded and run again.

I.4 ALL PREREGISTERED TESTS

Table 6 lists every comparison we fixed in advance, with its rule and outcome. A check mark means the rule is met. Rows marked *descriptive* had no rule.

I.5 ADDITIONAL RESULTS

Decay rates. Table 7 gives the local decay rate of greedy memories, the fall in log loss per unit of d, next to the rate $\ln(1/\rho)$ predicted by Theorem 3.6. Rates fitted over all budgets with loss at least 0.05 are 0.63, 0.78, and 0.82 of $\ln(1/\rho)$ for greedy, and 0.06 to 0.26 for the three models.

Table 7: Local decay rates of greedy memories on $G_{64,0.3}$. The rate for $\rho = 0.10$ drops after $d = 3$ because the loss levels off near 0.015.

k	ρ	$\ln(1/\rho)$	$d = 1 \to 2$	$d = 2 \to 3$	$d = 3 \to 4$
6	0.10	2.34	1.47	2.08	0.16
20	0.32	1.13	0.63	1.15	1.14
31	0.50	0.69	0.32	0.55	0.85

All failure counts and noisy-OR. Figure 6 shows model-written OR memories for every failure count, and Figure 7 the noisy-OR study with the models. Whether or not the prompt states γ, the models' losses decay far more slowly than the reference, and stating γ changes the loss by at most 0.02.

Table 6: Preregistered comparisons. Entries are means over hosts with 95% bootstrap intervals where they are informative. Loss is normalized loss; slopes are per unit of d on random hosts and per 32 edges on citation hosts.

Comparison	Rule	DeepSeek	GLM	Muse	Reference memories
Budget law, $G_{64,0.3}$					
Slope at $k = 6$ minus slope at $k = 31$	< 0	-0.09 ✓	-0.02 ✓	-0.11 ✓	greedy -0.90 $[-0.97, -0.80]$ ✓
Normalized rate, $k = 6$ minus $k = 31$	$= 0$	descriptive	descriptive	descriptive	greedy -0.19 $[-0.23, -0.16]$ fails
Normalized rate, $k = 20$ minus $k = 31$	$= 0$	descriptive	descriptive	descriptive	greedy -0.04 $[-0.06, -0.01]$ fails
Loss minus greedy, $d = 2$, each k	> 0	✓ (0.22–0.68)	✓ (0.25–0.70)	✓ (0.21–0.63)	
Loss minus uniform, $d = 2$, $k = 6$	< 0	$+0.05$ fails	$+0.06$ fails	-0.01 fails	
Loss minus uniform, $d = 2$, $k = 20$	< 0	$+0.02$ fails	$+0.06$ fails	-0.01 ✓	
Loss minus uniform, $d = 2$, $k = 31$	< 0	$+0.01$ fails	$+0.04$ fails	0.00 fails	
Isolation errors minus uniform, $d = 2$	> 0	$k = 31$ only	all k	none	
Core_6 alignment minus uniform, $d = 2$	descriptive	-0.10	-0.02	$+0.03$	uniform 0.77, greedy 0.98
Request types, $G_{24,0.5}$, $B = 48$					
OR minus AND; linear minus OR	both > 0				0.27 $[0.25, 0.29]$; 0.37 $[0.33, 0.39]$ ✓
AND memory: TR share minus $\lvert \mathrm{TR} \rvert / \lvert E \rvert$	> 0	0.10 ✓	0.17 ✓	0.14 ✓	
Overlap of OR and AND memories	< 0.8	0.37 ✓	0.33 ✓	0.64 ✓	
Linear loss minus $1 - B/\lvert E \rvert$	descriptive	$+0.022$	$+0.003$	$+0.001$	best $+0.001$
Noisy-OR, $G_{64,0.3}$					
Rate at $\gamma = 0.8$ over rate at $\gamma = 0.5$	> 1	descriptive	descriptive	descriptive	greedy 2.46 ✓, balanced 2.00 ✓
Same ratio contains $L_{0.8}/L_{0.5} = 2.32$	contains	descriptive	descriptive	descriptive	greedy $[2.38, 2.54]$, balanced $[1.92, 2.09]$: both fail
Stated minus unstated γ, loss	descriptive	$-0.01/+0.02$	$+0.02/-0.01$	$0.00/0.00$	
Guidance, $G_{64,0.3}$, averaged over k and d					
Isolation errors, guidance minus warning	< 0	-0.24 ✓	-0.26 ✓	-0.06 ✓	
Loss, guidance minus warning	< 0	-0.13 ✓	-0.11 ✓	-0.01 ✓	
Loss, warning minus original	descriptive	$+0.14$	$+0.02$	$+0.01$	
Loss, guidance minus uniform	< 0	$+0.06$ fails	-0.01 fails	0.00 fails	
Loss, guidance minus balanced	descriptive	$+0.39$	$+0.33$	$+0.33$	
Citation graphs, averaged over k and B					
Loss, titles minus labels	< 0	$+0.01$ fails	-0.01 fails	$+0.02$ fails	
Loss, guidance minus titles	< 0	-0.01 fails	0.00 fails	-0.04 ✓	
Isolation errors, guidance minus titles	< 0	-0.04 fails	-0.01 fails	-0.12 ✓	
Edge span, guidance minus titles	< 0	-3.9 ✓	-3.4 ✓	-5.0 ✓	
Loss, guidance minus uniform	< 0	$+0.04$ fails	$+0.04$ fails	0.00 fails	
Loss, guidance minus greedy	descriptive	$+0.43$	$+0.43$	$+0.39$	
Greedy slope, $k = 2$ minus $k = 12$	< 0				-0.23 $[-0.29, -0.18]$ ✓
Readers, $G_{24,0.5}$					
Reading gap, full minus greedy at $B = 48$	> 0	$+0.02$ fails	$+0.08$ fails	$+0.01$ fails	
Fidelity, own minus greedy memories	descriptive	$+0.19$	$+0.19$	$+0.01$	

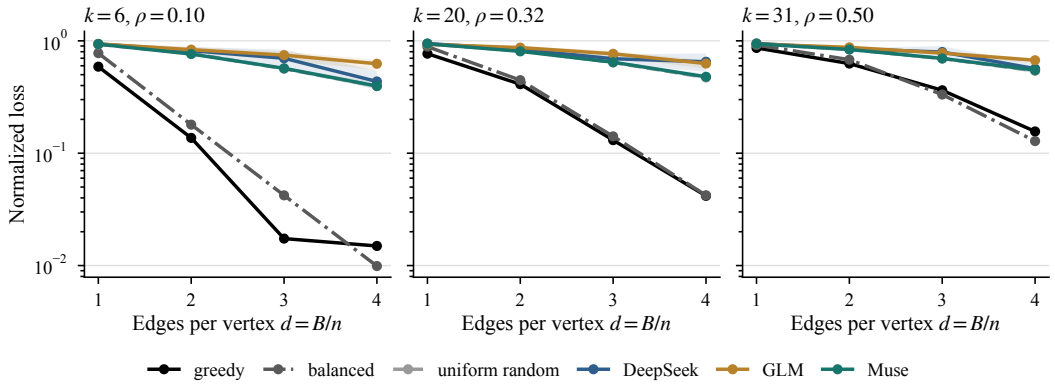

Figure 6: Model-written OR memories on $G_{64,0.3}$ for each failure count.

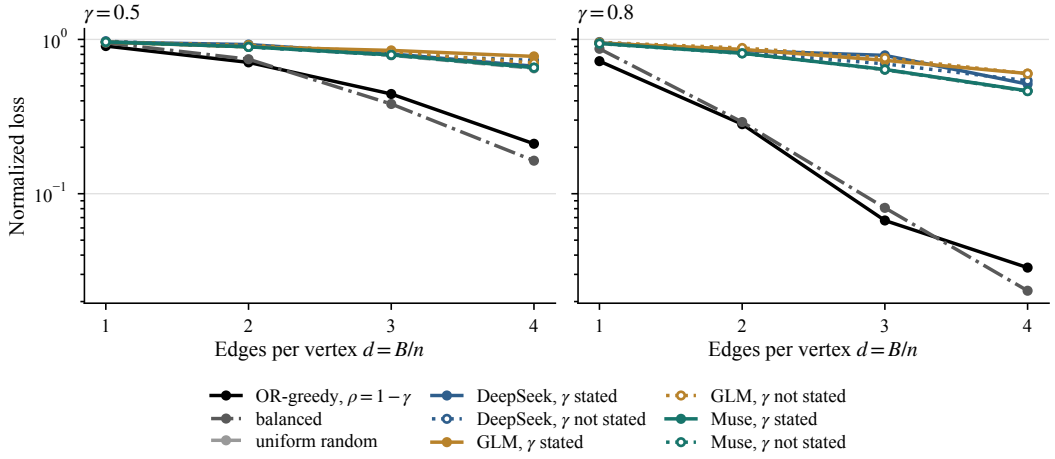

Figure 7: Noisy-OR memories on $G_{64,0.3}$. Solid model lines state γ in the prompt; dotted lines say only that links are unreliable.

Citation graphs. Figure 8 shows all three prompts on citation graphs. The balanced construction loses 0.68 on average and greedy 0.46; greedy memories give outgoing edges to 65% of papers, with a coefficient of variation of out-degrees of 1.35, so the best memories we can compute do not spread their budget evenly on these graphs.

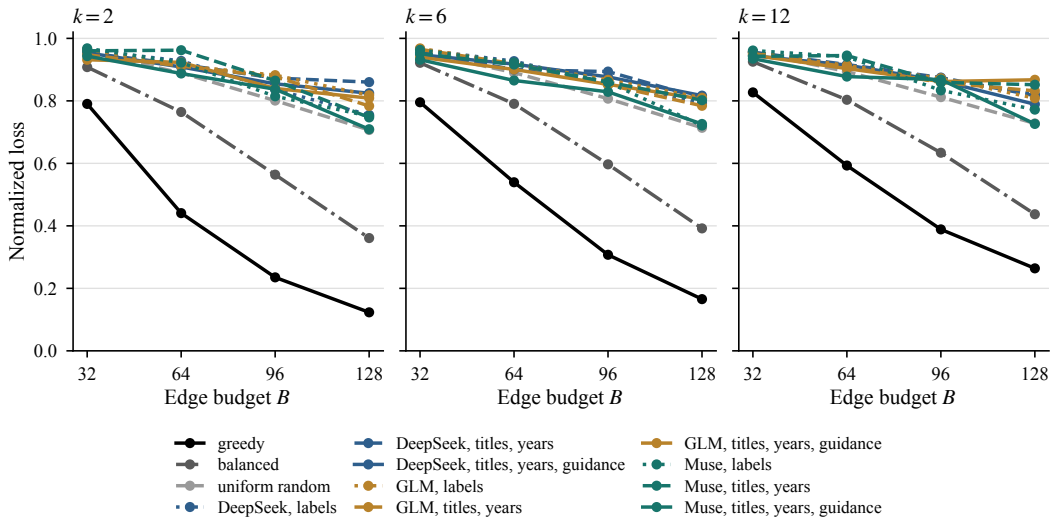

Figure 8: Model-written memories on citation graphs for the three prompts: arbitrary labels only, titles and years, and titles and years with guidance.

Further measurements. Table 8 collects measurements that Section 4 summarizes in words.

Table 8: Measurements behind the summaries in Section 4. Ranges run over hosts, failure counts, or models as stated.

Measurement	Value
Noisy-OR decay rate of greedy OR memories, $\gamma = 0.5$ (allowed range)	0.49 $[0.35, 0.69]$
Noisy-OR decay rate of greedy OR memories, $\gamma = 0.8$ (allowed range)	1.19 $[0.80, 1.61]$
Linear effects: largest deviation of a reference memory from $1 - B/\lvert E\rvert$	0.002
Share of edges in the transitive reduction: flights and Gene Ontology; citation graphs	92% to 100%; 27%
Greedy slope per 32 edges on citation graphs, $k = 2$, 6, 12	-0.63, -0.55, -0.40
Fitted decay rate as a share of $\ln(1/\rho)$: greedy; models	63% to 82%; 6% to 26%
GLM loss minus uniform random, $d = 2$, over k	$+0.035$ to $+0.062$
Citation graphs, loss averaged over k and B: models; uniform random; greedy	0.88 to 0.89; 0.84; 0.46
GLM, $d = 2$: share of vertices with outgoing edges that keep one	28% to 34%
Isolation share of errors, random graphs, $d = 2$: GLM; uniform random	80% to 85%; 26% to 49%
Isolation share of errors, citation graphs: models; uniform random	80% to 97%; 51%
Muse edges taken from the first B lines (chance)	93% to 100% (27%)
Loss at $k = 20$, $d = 2$: d random successors; uniform random; d nearest successors; greedy	0.75; 0.82; 0.45; 0.41
Mean edge span in positions: models under guidance; uniform random; balanced	17 to 21; 22; 5
Citation guidance: span reduction; Muse loss reduction	3.4 to 5.0; 0.04
Readers: fidelity of Muse; agreement of DeepSeek and GLM on the full graph or $B = 48$ greedy memory; fidelity gain on own memories	0.97 to 1.00; 58% to 63%; 0.19

Guidance, all measures. Table 9 extends Table 2 with connectivity errors, the part of D_μ in which routes survive from both endpoints but fail to meet, and spread, the coefficient of variation of stored out-degrees.

Table 9: Guidance study on $G_{64,0.3}$, averaged over k and d, with all measures.

Writer	Prompt	Loss	Isolation	Connectivity	Coverage	Spread	Copied
Uniform random	–	0.71	0.23	0.34	0.80	0.84	0.27
DeepSeek	original	0.77	0.39	0.23	0.62	1.38	0.70
	warning	0.90	0.65	0.07	0.35	2.15	0.38
	guidance	0.77	0.42	0.20	0.63	1.13	0.57
GLM	original	0.80	0.54	0.11	0.35	2.04	0.39
	warning	0.82	0.57	0.09	0.28	2.23	0.38
	guidance	0.71	0.31	0.26	0.80	0.49	0.68
Muse	original	0.71	0.24	0.34	0.76	1.01	0.94
	warning	0.72	0.29	0.29	0.67	1.35	0.80
	guidance	0.71	0.23	0.35	0.81	0.76	0.96

Readers. Table 10 gives the accuracy and fidelity of each model reader on each memory.

Table 10: Accuracy / fidelity of model readers on $G_{24,0.5}$ with $k = 7$. Fidelity is agreement with the ideal reader on the same memory. The last row gives the accuracy of the ideal reader. Own memories are those the model wrote itself.

Reader	Own, $B = 24$	Own, $B = 48$	Greedy, $B = 24$	Greedy, $B = 48$	Full graph
DeepSeek	0.30 / 0.94	0.40 / 0.87	0.30 / 0.83	0.41 / 0.61	0.61 / 0.61
GLM	0.30 / 0.96	0.31 / 0.91	0.34 / 0.86	0.43 / 0.63	0.58 / 0.58
Muse	0.31 / 1.00	0.50 / 0.99	0.47 / 1.00	0.76 / 0.97	0.98 / 0.98
Ideal reader	0.34	0.53	0.47	0.78	1.00

Copying the start of the list. Table 11 reports the share of stored edges that come from the first B lines of the shuffled list. By chance it would equal $B/|E|$ on average, which is 0.27 on $G_{64,0.3}$, 0.36 on $G_{24,0.5}$, and about 0.15 on citation graphs.

Table 11: Share of stored edges copied from the start of the list.

Condition	Chance	DeepSeek	GLM	Muse
OR, $G_{64,0.3}$	0.27	0.65–0.74	0.36–0.40	0.93–0.97
Noisy-OR, $G_{64,0.3}$	0.27	0.72–0.79	0.41–0.47	0.96–1.00
Request types, $G_{24,0.5}$	0.36	0.59–0.65	0.38–0.41	0.55–0.79
Guidance: orig./warn./guid.	0.27	0.70 / 0.38 / 0.57	0.39 / 0.38 / 0.68	0.94 / 0.80 / 0.96
Citation: labels/titles/guid.	0.15	0.53 / 0.20 / 0.34	0.19 / 0.18 / 0.19	0.53 / 0.41 / 0.70

Unusable replies. Table 12 counts writes that remained unusable after the retry; all reader batches were usable. Most unusable DeepSeek replies repeat a chain of edges until the token limit. Recomputing the comparisons with usable writes only changes one conclusion. Muse's gain from guidance over the warning disappears ($+0.003$ $[-0.007, 0.012]$), because it came from four unusable replies under the warning. The gains for DeepSeek and GLM remain (-0.12 and -0.12), no model beats uniform random edges with or without guidance, and the citation comparison of titles with labels stays at $+0.01$, -0.01, and $+0.01$.

Table 12: Unusable writes out of all writes in each study.

Study	DeepSeek	GLM	Muse
Budget law and noisy-OR ($G_{64,0.3}$)	3 / 168	1 / 168	0 / 168
Request types ($G_{24,0.5}$)	0 / 54	0 / 54	0 / 54
Guidance: warning / guidance	8 / 72, 0 / 72	2 / 72, 2 / 72	4 / 72, 1 / 72
Citation: labels / titles / guidance	0, 0, 7 of 72 each	0, 0, 4 of 72 each	5, 11, 3 of 72 each

I.6 Choosing a real task

We wanted a real task that matches the OR model: a natural order, failures of components, and answers that hold when any route survives. Before calling any model we required that, averaged over six hosts of 64 vertices, at most 60% of edges lie in the transitive reduction, that the greedy loss at $B = 128$ grows by at least 0.1 from $k = 2$ to $k = 12$, that at least 20% of requests have a positive answer, and that the greedy loss at $B = 64$ lie between 0.3 and 0.9. The first two conditions ask for real backups and for loss that depends on failures.

Flight connections. From the TravelPlanner flight database we built six hosts from one day of flights among three hub cities, with an edge from one flight to another when the second leaves the arrival city of the first between 30 minutes and 5 hours later. The hosts have 276 to 322 edges, but 95% to 100% of them lie in the transitive reduction, far above the 60% limit: a direct connection can only be replaced by a round trip that returns in time. The greedy loss at $B = 128$ changes by at most 0.12 between $k = 2$ and $k = 12$. *Gene Ontology.* Six subgraphs of 64 terms below broad biological processes have 90 to 131 edges, of which 92% to 100% lie in the transitive reduction, and only 12% of requests have a positive answer. *Citation graphs* meet every condition except the size cap discussed in Section I.3: on average 27% of edges lie in the transitive reduction, the greedy loss grows by 0.14 from $k = 2$ to $k = 12$, 56% of requests have a positive answer, and the greedy loss at $B = 64$ is 0.54.

J Earlier studies of exact memories and recovery evidence

This appendix collects three earlier studies. The first computes exact losses on hosts small enough to enumerate. The second asks whether readers use a dependency that failures make necessary. The third asks whether evidence about recoverability helps a model decide which fact to protect. Their protocols and complete results follow in Appendices K–P.

J.1 Exact loss on small graphs

To see how much edge selection matters at a fixed memory budget, we compare the smallest achievable loss with the average loss from uniformly selected edge subsets (Figure 9). We sample 64 hosts for each $n \in \{4, 5, 6\}$ and $p \in \{0.25, 0.5\}$, with $k = \lfloor (n-2)/2 \rfloor$, for 384 hosts in total, and enumerate every original-edge subgraph and every request under $\mu_{n,k}$. At each integer budget B, we compute $D^*_\mu(B, G)$ exactly by minimizing over subgraphs with at most B edges, evaluating 31,863 subgraphs in total.

Uniform subsets use B host edges, or all edges when fewer than B exist. In all 384 hosts, the first zero-loss budget equals the core size, as required by the characterization.

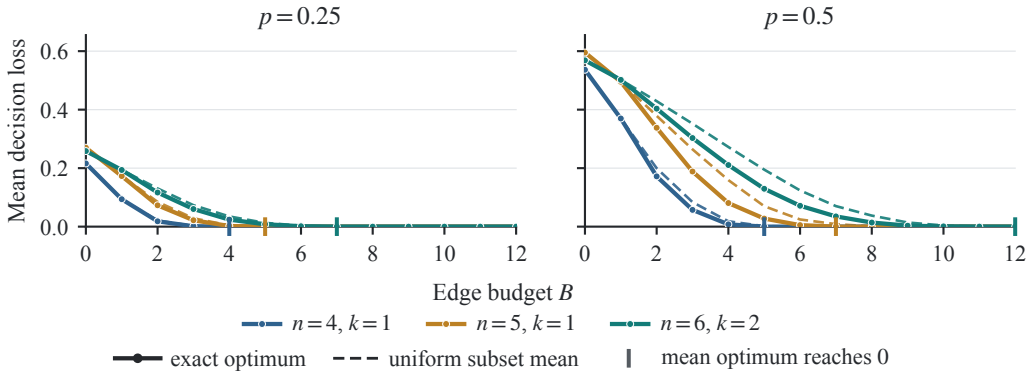

Figure 9: Exact selection improves on uniform subsets before both reach zero. Curves average 64 hosts per cell: solid lines are exact optima and dashed lines average uniform edge subsets. Ticks mark the smallest budget at which the sampled mean optimal loss is zero.

J.2 READER USE OF NECESSARY DEPENDENCIES

The exact core preserves every answer, but a model that reads it may still fail to use the information it contains (Figure 10). We use six 24-node graphs, each checked to have the intended cores, and three models as writers and readers to test whether replacing the full graph with its exact core preserves accuracy. We also examine whether readers use a dependency when failures make it necessary, and how performance changes when the memory limit becomes binding.

The six graphs form three pairs. The graphs in Pair A differ by an optional edge that changes no correct answer. In Pair B, the changed edge becomes necessary when two mediators fail. Pair C varies the recovery paths while keeping graph size and request counts fixed. We assign arbitrary node identifiers and choose the requests in advance, including those that distinguish the paired graphs.

We supply full graphs and exact cores directly to the reader as text. To produce a learned memory, a writer receives the complete graph and proposes a ranked edge list. A script, the compiler, checks the syntax of the proposal and keeps its records in rank order, up to 34 at 512 bytes or 68 at 1024 bytes, without verifying that the proposed edges exist in the graph. The full-graph control has no byte limit. Readers answer batches of 12 prespecified hidden requests and propose routes. We count invalid batches as errors overall and include a comparison in the paired-valid results only when both batches pass the validity checks.

Exact-core memories average 403.3 bytes under the ASCII encoding, compared with 1150 bytes for full graphs. Among paired-valid outputs, exact-core accuracy is 1.4 percentage points lower for DeepSeek, 1.0 point lower for GLM, and 0.9 points higher for Muse than full-graph accuracy.

We compare Core_k with Core_{k-1} on requests whose answers depend on the extra edges in Core_k. Among paired-valid outputs, accuracy with Core_k is 22.9 percentage points higher for DeepSeek (48 decisions), unchanged for GLM (48 decisions), and 100 points higher for Muse (20 decisions). These decisions come from six workflow batches and are dependent. Individual responses are unavailable, so we cannot independently check which comparisons passed the validity rules.

Pair C tests whether a structural difference in memory requirements also appears in learned memories. At $k = 2$, the exact cores occupy 510 bytes for `REDUNDANT` and 645 bytes for `SPARSE`, so only the first fits within 512 bytes, while both fit within 1024 bytes. We ask whether reducing the learned-memory budget harms `SPARSE` more when writers receive the deployment failure limit k. Let $E_{w,b}$ be end-to-end request error for workflow w and byte budget b, counting invalid batches as errors. The

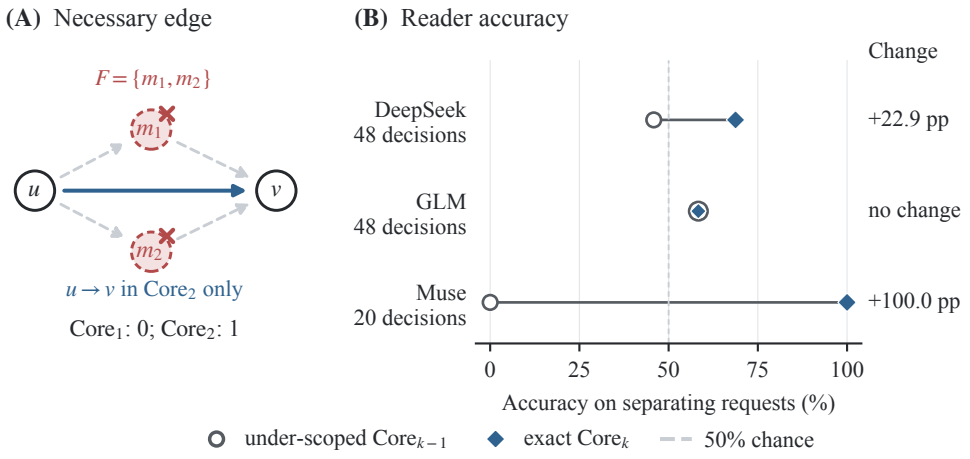

Figure 10: Readers differ in their use of necessary dependencies. (A) Two failed mediators expose the edge omitted by Core_1 and retained by Core_2. (B) Paired-valid accuracy on separating requests, where the stored graphs disagree on reachability. Labels give accuracy differences and retained decision counts.

difference in differences measures how much larger the budget penalty is for SPARSE:

$$\text{DID} = (E_{\text{SPARSE},512} - E_{\text{SPARSE},1024}) - (E_{\text{REDUNDANT},512} - E_{\text{REDUNDANT},1024}).$$

Table 13: Pair C error DID, in percentage points, with invalid batches counted as errors. Positive values mean the smaller budget harms SPARSE more.

Reader	DeepSeek	GLM	Muse
DID	+8.3	−16.7	+58.3

The smaller budget harms SPARSE more for DeepSeek and Muse, but the difference reverses for GLM (Table 13). Because each model both writes and reads its memory, these results reflect selection, serialization, and readout together. Conditions, invalidity rates, and route diagnostics are in Appendices K–M.

J.3 Recoverability evidence and memory selection

When two equally valued facts compete for one protected slot, we test whether recovery evidence helps the model protect the fact that can become unrecoverable under the allowed failures. Recovering a requires either both x and y or both x and z, whereas recovering b requires either both x and y or both z and w. Both methods for a depend on x, so losing that resource makes a unrecoverable. Let $\mathcal{P}(f)$ list the recovery-resource sets for fact f, and let $\tau(f)$ count the fewest failures that hit every method:

$$\tau(f) = \min\{|F| : F \cap P \neq \varnothing \text{ for every } P \in \mathcal{P}(f)\}. \tag{J.1}$$

Each case has exactly one fact with $\tau(f) \leq k$, and protecting it is scored as correct. The connection to graph separators is that a set of failed resources can block every recovery method, just as failed vertices can block every path. Figure 11 shows the resource sets in this example.

We separate the effects of recovery evidence and action labels by giving DeepSeek-V4 Pro 0813 four forms of feedback during in-context adaptation at $k = 1$: neither failure evidence nor an action label (*Current*); a failure witness for the vulnerable fact and a certificate that the other fact survives every

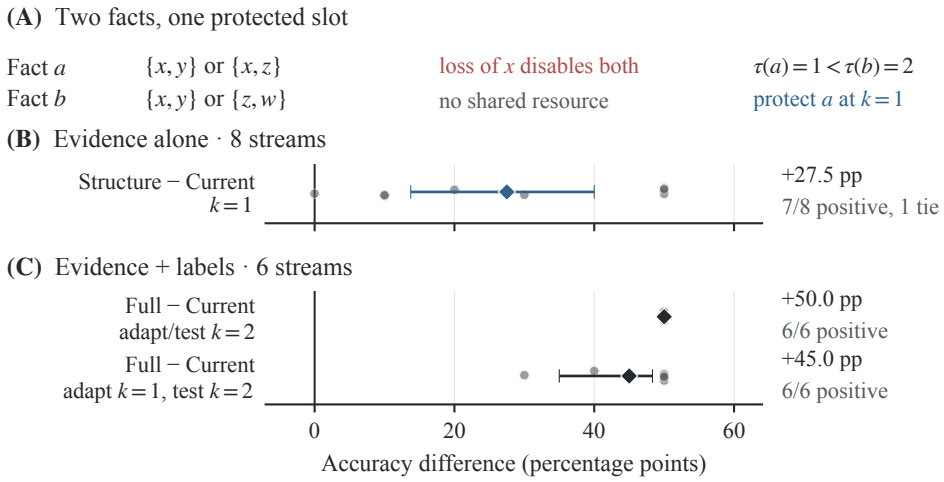

Figure 11: Recoverability evidence improves selection. (A) Each resource set is a recovery method requiring every listed resource. (B) Structure versus Current at $k = 1$, eight streams. (C) Full versus Current at $k = 2$, six streams, with Full supplying action labels as well as structural evidence. Dots are paired-stream differences, diamonds are means, and counts give positive differences and ties. Whiskers are descriptive 95% bias-corrected and accelerated (BCa) intervals, resampling whole streams within each fixed panel.

allowed failure set (*Structure*); the correct action alone (*Labels*); or both forms of feedback (*Full*). We run eight independent streams. In each, the model first makes six choices with feedback, which form its history; ten held-out choices are then each made separately, starting from that history. Target position and evidence order are balanced.

Structure achieves 77.5% held-out accuracy compared with 50.0% for Current. The paired-stream difference averages 27.5 points, with a positive difference in seven streams and a tie in one, and Structure accuracy is 77.5% in both target positions. DeepSeek selects the vulnerable fact more accurately when given recovery evidence alone.

In a separate six-stream comparison at $k = 2$, we test Full against Current after adaptation at either $k = 2$ or $k = 1$, using the latter to examine transfer. Here Full supplies both structural evidence and action labels. The accuracy difference is $+50.0$ points after adaptation at $k = 2$ and $+45.0$ after adaptation at $k = 1$ followed by transfer to $k = 2$. The component and transfer records are in Appendix O.

J.4 Recovery-state information and reward

A recovery state specifies which facts can be recovered, and at what cost, in each possible future. To measure the benefit of knowing this state before choosing what to store, we compare choices made before the active state is named (Hidden) with choices made afterward (Revealed), and compare each with the oracle allocation for that condition. Consider two equally weighted facts and one slot: in each of two states, one fact is perfectly recoverable and the other unavailable, and their roles reverse across states. A fixed allocation has worst-state value $1/2$, whereas storing the unavailable fact after observing the state attains one (Figure 12).

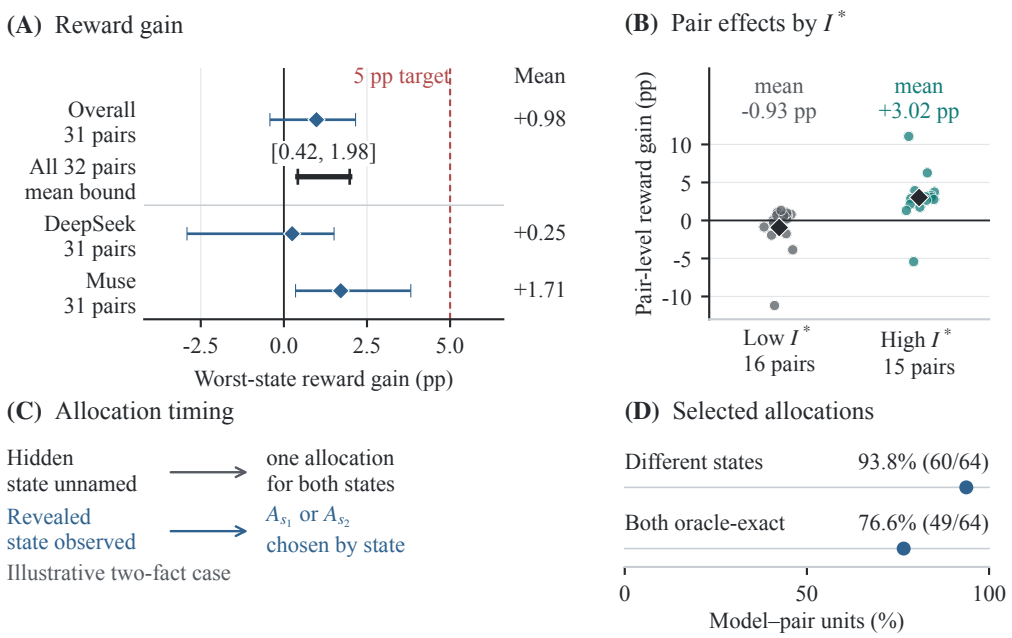

Figure 12: Frequent allocation changes accompany modest reward gains. (A) Means and descriptive BCa intervals for 31 complete pairs, with the planned five-point target. The separate 32-pair bound ranges the missing Hidden score over $[0, 1]$ and is not a confidence interval. (B) Pair effects in designed low/high-information strata, averaged over both models. (C) Hidden/Revealed timing for two facts. (D) Selected allocations across all 64 model–pair units, including the incomplete pair.

For each of 32 allocation cases, DeepSeek and Muse receive the same facts, question weights, memory capacity, and two recovery-cost tables. Each table represents one recovery state and lists the cost of recovering each fact in each possible future scenario, or world. We show both tables in both conditions and reveal the active-state identifier only in Revealed. The cases vary memory load, binary versus graded costs, the distribution of question weights, and the reward an optimal allocator can

gain from knowing the state. Across states, we rearrange the costs assigned to facts within worlds, preserving both each fact's collection of costs across worlds and each world's collection of costs across facts.

For allocation A, $u_{fsw}(A)$ scores fact f in state s and world w. A stored fact earns one, recovering an unstored fact incurs a penalty of $c_{fsw}/15$ from that unit reward, and an unavailable unstored fact earns zero.

For each recovery state, we evaluate an allocation by its lowest weighted reward across that state's possible worlds. We then compare the two conditions using the lowest reward each achieves across the recovery states. With positive future-question weights a_f, define $V_s(A) = \min_w \frac{\sum_f a_f u_{fsw}(A)}{\sum_f a_f}$, $\Delta V = \min_s V_s(A_s^{\mathrm{rev}}) - \min_s V_s(A^{\mathrm{hid}})$. With b_{slot} slots, the Hidden oracle chooses one allocation for both states, while the Revealed oracle chooses an allocation separately for each state: $V^*_{\mathrm{hid}} = \max_{|A|=b_{\mathrm{slot}}} \min_s V_s(A)$, $V^*_{\mathrm{rev}} = \min_s \max_{|A_s|=b_{\mathrm{slot}}} V_s(A_s)$. The difference $I^* = V^*_{\mathrm{rev}} - V^*_{\mathrm{hid}}$ measures how much state information improves the best achievable worst-state reward. In each condition, regret is the oracle reward minus the model's reward. We can then express the realized gain as the oracle opportunity minus the change in regret: $\Delta V = I^* - (\mathrm{Reg}^{\mathrm{rev}} - \mathrm{Reg}^{\mathrm{hid}})$. We average ΔV over the two models within each pair and then over pairs.

Some calls were delivered more than once; we keep the earliest structured response (Appendix P). With this selection, the two Revealed allocations differ between states in 60 of 64 model–case units, each representing one model evaluated on one case. In 49 units, both state-specific allocations match their respective oracles. Across the 31 cases with complete Hidden and Revealed outputs for both models, the mean reward gain is $+0.98$ percentage points. Letting the unresolved Hidden reward vary over $[0, 1]$ bounds the mean across all 32 cases to $[0.42, 1.98]$ percentage points, entirely below the prespecified five-point criterion.

Across the same 31 complete cases, optimal allocation benefits by 1.61 percentage points from state information. The average increase in regret from Hidden to Revealed is 0.63 points, leaving the observed gain of 0.98 points. Cases designed to have low and high oracle gains I^* yield mean model gains of -0.93 and $+3.02$ percentage points, respectively. Repeated deliveries of the same call sometimes disagree, and the discarded responses are not in the release. The selected outputs, missingness bound, and full reward definitions are reported in Appendix P.

K Experimental protocol

K.1 Task and what each role sees

Each workflow is a deterministic DAG on 24 nodes with opaque labels. Given a source u, a target v, and a declared failed-node set F with $|F| \le k$, the reader should COMMIT when $u \rightsquigarrow v$ in $G - F$ and ESCALATE otherwise. A committed answer must also propose a route from u to v using true edges and avoiding failed or repeated nodes. The 24-node workflow therefore permits routes of at most 24 nodes. We score route validity separately from the binary action.

Before writing the memory, the writer sees the whole workflow: every edge, the rule that decides between COMMIT and ESCALATE, and the failure limit k. It writes a memory within a byte budget, and a fixed reader later answers requests from that memory alone.

K.2 Matched pairs of workflows

Pair A: equivalence.
The two graphs differ by one edge with three internally vertex-disjoint alternatives. Their complete response maps agree for $k \le 2$.

Pair B: pivotality.
The two graphs differ by one edge with two alternatives. The edge changes no answer at $k = 1$, but at $k = 2$ a prespecified pair of failures (a separator) makes it necessary.

Pair C: compressibility.
The Redundant and Sparse workflows have equal node count, edge count, source bytes,

query count, and outage-count distribution. Their path organization produces different exact-core sizes at $k = 2$.

Pair members use the same node permutation, dictionary, source order, and query order. Each workflow is written out both as structured relation records and in controlled natural language. Numeric node identifiers are permuted after graph construction and contain no topological ordering information.

K.3 How the experimental contrasts relate to the theory

Distinction	Controlled contrast	Deployment question
$G \equiv_k J$	Pair A plus/minus	Invariance to an invisible graph change.
Separator pivotal at k	Pair B separator contrast	Retention and use of a newly exposed distinction.
Core_k is sufficient	Full vs. exact core	Readout after irrelevant structure is removed.
$\mathrm{Core}_k \neq \mathrm{Core}_{k-1}$	Exact core-scope contrast	Do decisions change on requests the two cores answer differently?
Failure limit told to the writer	Learned memory written for k vs. $k-1$	Do predictions change across complete 12-request batches?
Core size varies with redundancy	Pair C at 512/1024 bytes	Is the 512-byte error penalty larger for Sparse than for Redundant?

K.4 Writer, compiler, and memory conditions

The primary outcome is accuracy on each binary request, with invalid batches counted as failures. The comparisons across failure limits use different subsets of requests for exact and learned memories. For exact memories, we include only comparisons in which both batches are valid and score only requests certified to distinguish the two memories. For learned memories, we average all 12 requests in each cell whose writer and reader outputs satisfy the task requirements. With exact memories, we can therefore check whether a valid reader uses information needed at the larger failure limit. With learned memories, we check whether the failure limit given to the writer changes predictions across the complete batch.

The learned writer returns a ranked list of edges in a fixed format, and a deterministic script, the compiler, checks the whole list. A malformed record, extra prose, an unknown node, or any duplicate edge invalidates the proposal. For a valid proposal, the compiler takes up to the first 68 ranked records at 1024 bytes (34 at 512 bytes) and canonically serializes the selected edge set. A cyclic selected prefix invalidates the compiled memory. The compiler does not rerank the list, remove false edges, deduplicate, or repair it. If the proposal or compiled memory is invalid, the reader receives the empty memory, and the outcome is counted as a writer failure in the end-to-end evaluation. Because the full-graph control supplies the complete graph without the 1024-byte cap, comparing it with a bounded condition changes both graph content and memory length. The compiler never sees the correct answers.

In the main comparisons, the writer and reader use the same model. Four prespecified cross-reader diagnostics use a DeepSeek-written structured priority memory with GLM or Muse as the reader. DeepSeek and GLM use provider reasoning effort *none*, while Muse uses *low*. No response is followed by interactive feedback, and semantic invalidity does not trigger another attempt.

K.5 Prespecified requests and estimands

Each reader condition receives an ordered batch of 12 hidden requests. The batch includes ordinary reachable and unreachable cases, failure-sensitive cases, and prespecified pair-specific witnesses. The Pair B separator appears in the shared plus/minus batch. Pair C uses the same requests at 512 and 1024 bytes.

Binary decision correctness is the primary row-level outcome. We retain unsafe commitments, unnecessary escalations, and invalid committed routes as diagnostics. A reader response is valid only if it satisfies the prespecified output schema for the full batch. End-to-end accuracy gives no credit to

Table 14: Memory conditions supplied to the fixed reader. The exact, padded, and learned variants differ in how graph information is selected or rendered.

Condition	Construction
Full graph	Complete graph serialization.
Exact core	Exact $\mathrm{Core}_k(G)$ serialization computed symbolically.
Core for $k-1$	Exact $\mathrm{Core}_{k-1}(G)$ used at failure limit k; $k=0$ aliases are excluded from comparisons across failure limits.
Padded core	Exact Core_k followed by true optional edges until the main byte budget is filled.
Learned true scope	Learned ranking written for the failure limit used at test time.
Learned under scope	Same writer instructions, but stating the failure limit $k-1$.
Generic prose	Task-generic prose memory, separately validated for ASCII and byte budget but not for symbolic graph sufficiency.
Empty memory	No graph-dependent memory.
Truncated core	Pair C boundary control at 512 bytes.

an invalid response. For the paired-valid diagnostic, we first check that both writer memories and both reader responses in the comparison satisfy the task requirements. We average request accuracy only over comparisons that pass these checks.

For reader r, writer condition s (true or under-scoped), Pair C world w, and byte budget b, let $E^{r,s}_{w,b}$ be the mean binary request error over all Pair C requests assigned to that cell, counting every request in an invalid batch as an error. For each reader, the prespecified true-scope contrast is

$$\mathrm{DID}^{r,\mathrm{true}} = \left(E^{r,\mathrm{true}}_{\mathrm{SPARSE},512} - E^{r,\mathrm{true}}_{\mathrm{SPARSE},1024}\right) - \left(E^{r,\mathrm{true}}_{\mathrm{REDUNDANT},512} - E^{r,\mathrm{true}}_{\mathrm{REDUNDANT},1024}\right).$$

We report the same four-cell calculation descriptively for the under-scoped condition. A positive value means that the 512-byte budget harms the sparse world more, as predicted from the difference in exact-core sizes. This lets us check whether that difference also appears in learned-memory performance.

L Structural verification and identification checks

Before collecting model responses, we verify the graph properties exhaustively. We compute the complete S_k table for every ordered surviving endpoint pair and every F with $|F| \le k$, reconstruct Core_k, and check the intended differences within each matched pair.

L.1 Checks for each pair

For Pair A, the perturbed edge has three internally vertex-disjoint alternatives after it is removed. Exhaustive symbolic enumeration confirms that the plus and minus graphs have identical complete response maps for $k = 0, 1, 2$ and hence identical exact-core serializations at each deployed scope. By comparing the model's predictions on the finite hidden batch, we check whether its answers are also unchanged.

For Pair B, the perturbed edge has exactly two disjoint alternatives. At $k = 2$, the prespecified separator disables one internal vertex on each alternative. The plus graph remains reachable through the direct edge, whereas the minus graph does not. Both sides include this request in the same ordered 12-query batch. Every query whose correct answer is unchanged serves as a control for effects unrelated to the perturbed edge.

For Pair C, both raw graph serializations are 1035 bytes. At $k = 2$, the exact-core serializations are 510 bytes for Redundant and 645 bytes for Sparse. Hence the first exact core fits the 512-byte boundary and the second does not, while both fit at 1024 bytes. These size differences are established from the full response map and the compiler grammar before model performance is measured.

L.2 Collection integrity and leakage checks

During collection, we hold fixed the node relabelings, source records, tool schemas, query batches, budgets, model inputs, and symbolic certificates. The detailed collection record and the individual responses are not in the archive, so these choices cannot be checked independently. The retained aggregates do support a check for leakage: rules that choose actions without graph information achieve near-chance accuracy on the collected labels.

Rule	Accuracy (%)
Always COMMIT	50.46
Always ESCALATE	49.54
Commit on the first six batch positions	47.69
Commit by query-ID parity	49.54

The labels, pair membership, intended witness, and graph topology are not available to the reader beyond what its deployed memory and request encode.

M Complete empirical results and diagnostics

We report every aggregate end-to-end result cited in the main text. The archive contains these aggregates but omits the individual responses needed to recompute them and the corresponding figure independently. The separator-feedback results can be checked against the released scored responses. For the recovery-state study, the release includes the selected response for each call but not the discarded repeated deliveries, so results can be recomputed only under the fixed selection rule.

M.1 End-to-end condition summary

Table 15: End-to-end main-condition accuracy / valid-reader rate (%), pooled over the fixed workflows, failure limits, source renderings, and requests, with invalid outputs counted as errors.

Condition	DeepSeek	GLM	Muse
Full information	77.8 / 100.0	81.5 / 94.4	66.2 / 66.7
Exact Core_k	76.4 / 100.0	83.8 / 100.0	50.0 / 50.0
Exact Core_{k-1}	74.5 / 100.0	78.7 / 100.0	66.2 / 88.9
Padded core	78.7 / 100.0	85.2 / 94.4	68.5 / 72.2
Learned true scope	75.0 / 100.0	76.9 / 100.0	60.0 / 77.8
Learned under scope	75.2 / 100.0	75.7 / 100.0	59.0 / 77.8
Generic prose	73.8 / 97.2	82.6 / 100.0	69.9 / 83.3
Empty memory	47.7 / 100.0	49.5 / 100.0	49.5 / 100.0

For Muse, exact-core end-to-end accuracy and the valid-reader rate are both 50.0%. Invalid responses count as failures in this table. The paired-valid comparison below instead restricts accuracy to responses that satisfy both output formats.

M.2 Exact core versus full information

Removing edges outside the exact core leaves every correct answer unchanged. We compare exact-core and full-graph memory on the same request batches to check whether the reader maintains its accuracy, restricting this comparison to paired-valid outputs.

Table 16: Paired-valid exact-core comparison. n_q counts paired query decisions.

Reader	n_q	Full (%)	Exact core (%)	Core–full (pp)
DeepSeek	216	77.8	76.4	-1.4
GLM	204	86.3	85.3	-1.0
Muse	108	99.1	100.0	$+0.9$

In the fixed ASCII representation, replacing the full graph with its exact core reduces mean length from 1150 to 403.3 bytes. Across paired-valid rows, the corresponding exact-core minus full-information accuracy differences range from -1.4 to $+0.9$ percentage points across readers, with graph content and memory length changing together.

Pair A lets us check whether readers give the same predictions when an optional edge changes. The exact-core serialization is identical in the two worlds, and every reader gives identical predictions from it. With the full graph, agreement is 77.8% for DeepSeek and 100% for GLM and Muse.

The separating request and paired-valid witness results are shown in Figure 10, with the reported aggregates given here.

M.3 Comparisons across failure limits

The released learned-memory results allow us to compare predictions across complete request batches. Most of the 12 requests in each batch do not separate Core_k from Core_{k-1}, and without the individual responses we cannot restrict the learned-memory comparison to separating requests. Across those complete batches, 723 of 744 query predictions agree between true- and under-scoped memory. These predictions come from the 62 comparisons in which every writer and reader output satisfies the task requirements. Table 17 breaks the result down by reader:

Table 17: Learned true-scope versus under-scoped memory among cells in which both writer and reader outputs satisfy the output format. Each eligible cell contains 12 dependent query rows.

Reader	Eligible cells	True (%)	Under (%)	Identical predictions (%)
DeepSeek	25	69.3	70.3	97.0
GLM	18	84.3	85.2	96.3
Muse	19	72.4	70.6	98.2
All	62	74.6	74.7	97.2

The 97.2% pooled agreement describes full request batches, leaving the learned writers' behavior on separating requests unresolved. To check whether readers use information needed at the larger failure limit, we turn to exact memories and the prespecified larger-budget requests for which both responses are valid. Accuracy on these requests is 69.8% with exact Core_k and 43.1% with exact Core_{k-1}. The corresponding accuracies are 96.0% and 98.7% on ordinary reachable requests, and 89.4% and 87.7% on ordinary unreachable requests. The accuracy difference is therefore concentrated on requests constructed to require the larger failure limit.

DeepSeek contributes $N = 48$ paired rows and scores 68.8% versus 45.8%. GLM also contributes $N = 48$ and scores 58.3% under both memories. Muse contributes $N = 20$ and scores 100% versus 0%. Because these requests distinguish reachability in Core_k and Core_{k-1}, correct answers require the reader to use information needed at the deployment failure limit.

M.4 Pair B separator case

We count a Pair B case as successful only when both the plus and minus responses are valid and correct and the predicted action flips between them. Each unit below is one matched plus/minus response pair. The first three rows contain two renderings for each of three readers. The exact-core and full rows each contain one pair per reader, and the cross-reader row contains two pairs.

Memory	Successful pairs	Matched pairs
Generic prose, own-model × rendering	5	6
Learned true-scope priority	0	6
Learned under-scoped priority	0	6
Exact Core_2	1	3
Full information	0	3
Cross-reader priority	0	2

DeepSeek predicts COMMIT on both sides from the exact core. GLM correctly flips from the exact core but not from its learned priority memory. Muse produces an invalid minus-side response in the exact-core and several other conditions. Here, success requires correct answers on both graphs and an action flip, whereas Section J.2 compares Core_k and Core_{k-1} on certified witnesses in the same graph.

M.5 Representation form and learned capacity

We vary both the form of the information and the space available to retain it. To examine the reader's use of additional information, we add optional edges to the exact core and measure accuracy among paired-valid outputs. For learned-priority writers, we compare structured relation records with controlled natural language while keeping the compiler fixed. Pair C then lets us compare the error penalties at 512 and 1024 bytes.

Among comparisons with two valid outputs, adding truthful optional edges to the exact core changes accuracy by $+2.3$ percentage points for DeepSeek, $+5.9$ points for GLM, and -2.1 points for Muse. When the writer receives structured relation records in place of controlled natural language, paired-valid accuracy with learned-priority memory changes by $+4.6$ points for DeepSeek, $+15.7$ points for GLM, and -0.5 points for Muse. In both comparisons, memory length changes together with content or form.

At the Pair C boundary, the prespecified true-scope error DID is $+8.3$ percentage points for DeepSeek, -16.7 points for GLM, and $+58.3$ points for Muse. The corresponding under-scoped values are $+8.3$, 0.0, and $+58.3$ points. That the Sparse core is larger than the Redundant one is a property of Pair C, checked exhaustively. The learned writer produces a ranked edge list, which need not reproduce that core, and the compiler keeps only the prefix that fits the budget. The mixed DID signs show that the exact-core size ordering does not consistently predict the learned-memory budget penalty across readers in these comparisons.

M.6 Invalidity and route diagnostics

Table 18: Invalid logical outputs by model. Writer counts combine invalid priority proposals and prose memories.

Stage	DeepSeek	GLM	Muse
Writer output	3	22	9
Reader response	1	2	44
Compiled memory	0 across all models		

Executable success, which also requires a valid route, can be lower than binary decision accuracy. In the main learned true-scope condition, DeepSeek achieves 75.0% binary success and 72.0% executable success. GLM achieves 76.9% and 75.5%, respectively. With exact under-scoped memory, the corresponding DeepSeek rates are 74.5% and 62.5%, while the GLM rates are 78.7% and 68.5%. The theory covers only the binary reachability answer, not the route.

N Experimental reproducibility

The end-to-end study uses fixed pseudorandom draws and a maximum persistent-memory budget of 1024 bytes. It also includes a full-information control with no byte cap and a Pair C comparison at the 512-byte boundary. Provider reasoning effort is set to *none* for DeepSeek-V4-Pro-0813 and GLM-5.2 and to *low* for Muse Glimmer 30B. Only rate limiting, temporary service unavailability, or a timeout triggers a retry, which always uses an identical request. Semantically invalid responses are retained without another attempt.

Every writer and reader response required by the reported comparisons was returned. Retries and invalid outputs remain represented in the stated denominators.

The available records determine which results can be independently checked. For the end-to-end study, the release supports checks of aggregate arithmetic and reconstruction of Figure 10. Individual

scored responses and stream-level summaries are available for the feedback component and transfer studies, while Figure 11 uses the archived interval and sign-symmetry summaries. For the recovery-state study, readers can recompute the statistics from the 192 responses, one per call, selected by the fixed rule for repeated deliveries and reconstruct Figure 12. The accompanying reproducibility documentation specifies the software environment and released materials. Collection code is not included.

O SEPARATOR-FEEDBACK EXPERIMENTS

Two equally valued facts compete for one protected slot in each allocation task. Section J.3 gives the concrete instance, with recovery-method families $\{x, y\}, \{x, z\}$ and $\{x, y\}, \{z, w\}$. Each fact has several recovery methods, each method survives only when all of its resources survive, and $\tau(f)$ is the fewest resource failures that disable every method. With a failure limit of k, the target is the fact with $\tau(f) \leq k$, because protection preserves the fact that can otherwise become unrecoverable. We compare conditions by taking the difference in held-out accuracy within each stream and averaging those paired differences over the fixed panel. BCa intervals describe resampling over these constructed streams, and the sign-flip calculations assume exchangeable effect signs. These summaries apply to the fixed panels, without a randomized treatment-assignment or broader population interpretation.

O.1 PRIMARY FEEDBACK-COMPONENT STUDY

To separate the value of recoverability evidence from that of a supplied answer, we cross two feedback factors: evidence about each fact and labels naming the correct action. The four conditions provide neither field (Current), the identities of the facts to store and drop (Labels), evidence alone (Structure), or both fields (Full). Structure supplies a minimum failure witness for the fact to store and a certificate that no allowed failure witness exists for the fact to drop. It therefore gives evidence for the answer without naming the action. We balance the order of the evidence records independently of the order of the candidate facts.

We evaluate DeepSeek V4 Pro 0813 at temperature zero and low reasoning effort on eight independent streams. In each condition and stream, the model first makes six choices with feedback; each of ten held-out choices then starts separately from the resulting history. A balanced Latin-square schedule assigns four renamed and reordered copies of each base case to the conditions. Preliminary checks and reserve streams are excluded from inference.

Table 19: Feedback-component results at $k = 1$. Condition intervals resample the eight stream accuracies. Selection-position rates use all 80 held-out calls per condition, including unstructured outputs.

Condition	Correct	Accuracy	BCa interval	Structured	Select first	Streams
Current	40/80	50.0%	[50.0, 50.0]	100.0%	87.5%	8
Structure	62/80	77.5%	[63.8, 90.0]	100.0%	50.0%	8
Labels	76/80	95.0%	[87.5, 97.5]	98.8%	51.3%	8
Full	80/80	100.0%	[100.0, 100.0]	100.0%	50.0%	8

Table 20: Feedback-component accuracy by target position at $k = 1$. Each cell pools the held-out choices at that position across eight independent streams.

Condition	Target first	Target second	Difference
Current	87.5%	12.5%	75.0 pp
Structure	77.5%	77.5%	0.0 pp
Labels	97.5%	92.5%	5.0 pp
Full	100.0%	100.0%	0.0 pp

In the planned comparison, Structure improves on Current in seven streams and ties it in one, giving a mean accuracy difference of 0.275. The descriptive BCa interval is $[0.1375, 0.4000]$, and the sign-symmetry value is $p = 0.015625$. The planned exploratory comparisons give Labels–Current

(0.450, $[0.375, 0.475]$, $p = 0.0078125$), Full–Labels (0.050, $[0, 0.100]$, $p = 0.25$), and Full–Current (0.500, $[0.500, 0.500]$, $p = 0.0078125$). The factorial interaction compares the action-label gain with Structure evidence to the gain without it: (Full − Structure) − (Labels − Current) $= -0.225$ ($[-0.400, -0.075]$). The gains from the two feedback types are less than additive on this accuracy scale, with Labels and Full already close to 100%. The Structure comparison measures the effect of the certificate fields supplied together.

O.2 Transfer with up to two failures

The $k = 2$ cases pair one fact with $\tau = 2$ and one with $\tau = 3$. The two facts have the same number of recovery methods, method-size multiset, resource count, degree profile, and pairwise-overlap signature. To test transfer, we vary only the six adaptation cases: Full and Current adapt at $k = 2$, while Full-transfer and Current-transfer adapt at $k = 1$. All conditions then receive the same kind of $k = 2$ held-out cases. The Direct control states the rule explicitly (protect the fact whose methods can all be disabled by k failures) and supplies a correct six-case history. We use six independent confirmation streams, with ten held-out choices per condition and balanced target positions.

We compare Full with Current when adaptation and testing both use $k = 2$, and Full-transfer with Current-transfer after $k = 1$ adaptation and $k = 2$ testing. Comparing Full at $k = 2$ with Full-transfer measures the accuracy difference between adaptation regimes, while Direct checks whether the model can apply a supplied rule.

Table 21: Learning, transfer, and Direct control at $k = 2$. The Direct structured rate equals its accuracy because every structured Direct response is correct. Selection-position rates use all 60 held-out calls per condition.

Condition	Correct	Accuracy	BCa interval	Structured	Select first	Streams
Full ($k = 2$)	60/60	100.0%	[100.0, 100.0]	100.0%	50.0%	6
Current ($k = 2$)	30/60	50.0%	[50.0, 50.0]	100.0%	100.0%	6
Full transfer ($k = 1 \to 2$)	58/60	96.7%	[90.0, 98.3]	100.0%	50.0%	6
Current transfer ($k = 1 \to 2$)	31/60	51.7%	[50.0, 53.3]	100.0%	98.3%	6
Direct ($k = 2$)	53/60	88.3%	[81.7, 91.7]	88.3%	41.7%	6

In the planned comparison at $k = 2$, Full exceeds Current by 0.500 in every stream. The descriptive BCa interval is $[0.500, 0.500]$, and the sign-symmetry value is $p = 0.03125$. The exploratory comparison of Full-transfer with Current-transfer gives a difference of 0.450 ($[0.350, 0.483]$; $p = 0.03125$), positive in all six streams. The observed gap between Full at $k = 2$ and Full-transfer is 0.033 ($[0, 0.067]$; $p = 0.50$), which does not establish equivalence between the regimes.

Accuracy also varies with the target's position. Listing accuracy with the target first and second, respectively, gives 100.0% versus 0.0% for Current at $k = 2$, 100.0% versus 100.0% for Full at $k = 2$, 100.0% versus 3.3% for Current-transfer, and 96.7% versus 96.7% for Full-transfer.

Figure 11 displays the recovery-method example and the three paired-stream contrasts. Tables 19–21 retain the full condition and position results.

O.3 Component-study validity and release

The archived analysis plan required each primary contrast to exceed 0.15, have a BCa lower bound above zero, and have a sign-symmetry value below 0.05. Both primary comparisons met all three criteria. The proportion of held-out responses with the required structure is 99.7% in the component study and 97.7% in the transfer study. Direct accuracy at $k = 2$ is 88.3%, exceeding the specified capability floor of 85%. All 956 confirmation calls returned successfully. Ten length-limited responses remain in their original denominators: three Labels calls and seven Direct held-out calls. Neither primary contrast contains an unstructured response.

The held-out topologies do not overlap calibration or prior-study topologies. The accompanying documentation records the protocol and any departures from it. Readers can check the scores and denominators against the individual responses, while the stream summaries and archived analyses contain the reported differences between paired streams, intervals, and sign-symmetry calculations.

Figure 11 uses the released scored rows and the archived interval and sign-symmetry summaries (Appendix N).

P Recovery-state revelation experiment

P.1 Question and design

We ask whether knowing the recovery state before choosing what to store improves the value of a bounded memory allocation. Consider two facts with equal future-question weights, one memory slot, and one future world per state. Let cost 0 mean perfect recovery and U mean that an unstored fact is unavailable:

State	Fact 1 cost	Fact 2 cost	Revealed choice
s_1	U	0	store Fact 1
s_2	0	U	store Fact 2

A Hidden policy must choose one allocation before nature reveals s, and either choice has worst-state value $1/2$. A Revealed policy selects A_s after observing s and obtains value one by storing the unavailable fact.

A matched case has the same facts, values, future-query weights, capacity, and two possible recovery states in both conditions. Each state is represented by a matrix with facts as rows and worlds as columns, whose entries give a recovery cost c_{fsw} or mark recovery as unavailable (U). The Hidden condition sees both matrices but no realized state identifier and returns one memory. The Revealed condition additionally sees the realized identifier and returns one memory for each state-specific call.

Across states, each fact keeps the same collection of costs over worlds, and each world keeps the same collection of costs over facts. Only the assignment of costs to facts within each world changes. The conditions therefore differ in whether the model knows which assignment is active. To judge how well it uses this information, we compare its realized reward difference, ΔV_θ, with the gain available to an oracle, I^*. Their difference, $I^* - \Delta V_\theta$, is the additional regret in the Revealed condition, as the identity below shows. Each pair has a unique Hidden oracle and unique state-specific oracles, and those optima differ.

Low-load cases contain six facts, two memory slots, and four worlds per state, compared with nine facts, four slots, and six worlds per state in high-load cases. The fixed panel includes every combination of four binary factors: load, binary versus graded recovery costs, near-equal versus heavy-tailed future weights, and low versus high designed oracle gain from state revelation (I^*, defined below). Each of the 16 combinations contains two pairs, giving 32 pairs in total. Fact order, state order, and opaque identifiers are balanced or randomized. We fixed this set of cases before running either model.

P.2 Reward, oracles, and estimand

For fact f, allocation A, state s, and world w, define

$$u_{fsw}(A) = \begin{cases} 1, & f \in A, \\ 1 - c_{fsw}/15, & f \notin A \text{ and recovery costs } c_{fsw}, \\ 0, & f \notin A \text{ and recovery is unavailable.} \end{cases}$$

Using a_f as the future-question weight, we average the fact scores and take the worst value across worlds to obtain the state value:

$$V_s(A) = \min_w \frac{\sum_f a_f u_{fsw}(A)}{\sum_f a_f}.$$

For a capacity of b_{slot} facts, an oracle chooses the best allocation allowed by the information available in each condition. Its values are

$$V^*_{\text{hid}} = \max_{|A|=b_{\text{slot}}} \min_s V_s(A), \qquad V^*_{\text{rev}} = \min_s \max_{|A_s|=b_{\text{slot}}} V_s(A_s),$$

and the gain from revealing the state is $I^* = V^*_{\text{rev}} - V^*_{\text{hid}}$. Applying the same worst-state criterion to the model's allocations A^{hid}_θ and $A^{\text{rev}}_{\theta,s}$ gives the realized values and their difference:

$$V^{\text{hid}}_\theta = \min_s V_s(A^{\text{hid}}_\theta), \qquad V^{\text{rev}}_\theta = \min_s V_s(A^{\text{rev}}_{\theta,s}), \qquad \Delta V_\theta = V^{\text{rev}}_\theta - V^{\text{hid}}_\theta.$$

The regret in each condition is the difference between the oracle value and the model's realized value. Writing $\text{Reg}^{\text{hid}}_\theta = V^*_{\text{hid}} - V^{\text{hid}}_\theta$ and $\text{Reg}^{\text{rev}}_\theta = V^*_{\text{rev}} - V^{\text{rev}}_\theta$, we obtain the identity

$$\Delta V_\theta = I^* - (\text{Reg}^{\text{rev}}_\theta - \text{Reg}^{\text{hid}}_\theta).$$

For each pair, we first average ΔV_θ over the two models. The planned estimand is the mean of those pair effects across the fixed 32-pair panel. The analysis plan prespecifies 0.05 as the minimum meaningful mean effect. With one unresolved Hidden call, we report worst-case bounds for that target and, separately, a complete-case mean over 31 pairs. For descriptive comparison, a BCa procedure resamples those constructed pairs 20,000 times, and a two-sided sign-symmetry calculation uses one million within-pair sign flips under an exchangeable-sign null. These summaries treat the constructed pairs as an empirical distribution, and their uncertainty interpretation is limited to that distribution. To bound the mean for the fixed panel of 32 pairs, we allow the missing Hidden score to range over $[0, 1]$.

P.3 Models and results

DeepSeek V4 Pro 0813 and Muse Glimmer 30B receive identical public information at temperature zero and low reasoning effort. Each model makes one Hidden and two Revealed calls per pair, for 192 planned calls. A separate control gives the model the target allocation and asks it to copy it. All eight responses in this control are exact, with four responses per model.

Across the 31 complete pairs, 22 reward effects are positive, one is zero, and eight are negative (Table 22). Both Revealed calls are available for every model and pair, so the allocation-change and oracle-match columns include cases with the missing Hidden call.

Figure 12 shows how the selected allocations change and how much reward is gained on complete pairs. It also shows the bound obtained by varying the missing score and the effects within the designed strata.

Letting the missing Hidden score vary over $[0, 1]$ bounds the fixed 32-pair mean to $[0.42, 1.98]$ pp. The entire interval is below the analysis plan's 5 pp criterion. The complete-pair results below describe the 31 pairs with all required calls.

Across the 31 complete pairs, the 1.61 pp mean oracle opportunity is reduced by 0.63 pp of additional Revealed regret, leaving the observed gain of 0.98 pp. DeepSeek matches the Hidden oracle in all 31 complete pairs, and its two allocations after revelation differ between states in all 31. It fails to match the Revealed oracle in two pairs, whose effects of -22.38 and -14.76 pp reduce its mean to $+0.25$ pp, compared with $+1.71$ pp for Muse. These two DeepSeek misses account for most of the additional Revealed regret.

Table 22: Recovery-state revelation results. Hidden and Revealed rewards are reported as percentages, with differences and their descriptive pair-level BCa intervals in percentage points. Reward columns use 31 complete pairs. Revealed-only diagnostics use all 32 pairs per model: "Selected differ" compares the two allocations, and "Both exact" requires both state-specific oracles.

Analysis	Pairs	Hidden	Revealed	ΔV [BCa interval]	Selected differ	Both exact
Pair-cluster mean	31	66.02	67.00	$+0.98$ $[-0.42, 2.16]$	60/64	49/64
DeepSeek	31	66.76	67.01	$+0.25$ $[-2.91, 1.51]$	32/32	30/32
Muse	31	65.28	67.00	$+1.71$ $[0.35, 3.82]$	28/32	19/32

Table 23: Reward gains by designed oracle information value. The rows use the 31 pairs with complete model effects: 16 low-I^* and 15 high-I^* pairs. The unresolved pair belongs to the high-I^* stratum. The high–low row subtracts the low-I^* group's mean reward gain from the high-I^* group's mean. This difference is not an estimated treatment effect. The bound accounting for the missing score across all 32 pairs is reported separately.

Stratum	Pairs	Mean ΔV (pp)	Sign count $(+/=/-)$
Low I^*	16	-0.93	$8/1/7$
High I^*	15	$+3.02$	$14/0/1$
High minus low	31	$+3.95$	—

P.4 Delivery and analysis audit

The 192 planned calls produced 306 response records, including 114 duplicate deliveries. The fixed selection rule takes the earliest structured original response or, if none exists, the earliest structured repair. It yields structured allocations for 191 calls, leaving one DeepSeek Hidden call unresolved. The complete-case estimate uses the 31 pairs with both models and information conditions complete.

The release includes one selected record for each of the 192 calls and an audit of disagreements, sufficient to recompute the complete-case statistics under the fixed rule. Alternative reward effects, oracle counts, or sharp bounds across selections require the unreleased duplicate responses. After four of the first 24 calls failed to return valid memory, we amended only the token limits and timeout settings, recording the change before inspecting any reward effect or interim statistic. The corpus, prompts, conditions, models, scoring rule, factor panel, and estimand remained fixed.

The unplanned duplicate deliveries show variation within conditions: 108 logical calls have multiple structured responses, with different choices in 27 calls (26 involve Muse). Thus "Selected differ" in Table 22 records whether the two state-specific calls selected by the fixed rule produce different allocations. Given the disagreements among repeated calls, this measure alone cannot establish a stable allocation preference within either state.

The missing unit is a DeepSeek Hidden call in the cell with high load, binary costs, nearly equal weights, and high I^*. Because each pair averages two model effects and each effect lies in an interval of width one, this call contributes at most $1/(2 \cdot 32) = 1/64$, or 1.5625 pp, to the width of the panel-mean bound. The $+0.42$ pp lower endpoint assigns the unresolved call a Hidden value of one, while the $+1.98$ pp upper endpoint assigns it zero. If the unresolved Hidden score is imputed as zero, the 32-pair mean is 1.98 pp. Resampling that completed panel gives a BCa interval of $[0.36, 5.53]$ pp and sign-symmetry value $p = .0787$. Those summaries assume that the missing score is zero. The bound of $[0.42, 1.98]$ pp above allows the missing score to take any value in $[0, 1]$ while holding all observed calls fixed.