FROM EXPERIENCE TO KNOWLEDGE: RECURSIVE PREDICTIVE COMPRESSION FOR AGENT MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

With limited resources, agents must turn experience into reusable knowledge while retaining key details. We introduce Recursive Predictive Compression (RPC), which unifies experience internalization and generative recall through one recursive predictive representation. RPC repeatedly decomposes an experience's contribution to prediction into shared structure and local residuals using the same rule across scales. Measured predictive transfer proposes candidate scopes; later task and recall outcomes select the depth and scope of reuse and the details to retain. Compatible contributions enter a bounded world state, allowing separate records to be released while their predictive influence survives. Recall draws on internalized world knowledge, supplemented by retained key details. RPC thus connects compression, transfer, and internalization while giving each operation a distinct role. Under a fixed linear predictive interface and the stated compatibility conditions, decomposition is exact before lossy release, and absorption preserves the full predictive distribution. Evaluations on controlled decisions, hidden-rule inference, long-horizon agent memory, and visual prediction show gains in prediction and reconstruction, alongside improvements in downstream performance.

1 INTRODUCTION

Agents continually acquire experience that should improve how they predict, adapt, and act. An agent learning how a tool behaves should retain a reusable rule together with details of consequential exceptions. Under limited resources, memory must turn such experience into lasting knowledge without requiring every interaction to remain as a separate record. This goal connects rapid learning from individual events with the gradual integration of shared patterns (McClelland et al., 1995).

Retaining too much detail increases storage cost, while extending shared information to unsupported contexts can spread errors. The challenge is to decide which event-specific details to retain, where shared knowledge should be reused, and when an explicit record can be released without losing its predictive influence. These decisions jointly determine how experience becomes reusable knowledge while remaining available for specific recall.

Existing systems address important parts of this problem. Retrieval-based memories make stored information available to later queries (Lewis et al., 2020; Gutiérrez et al., 2024). Hierarchical methods organize memories at multiple levels (Rezazadeh et al., 2025; Xu et al., 2026), while test-time memories update recurrent states or learned modules (Sun et al., 2024; Behrouz et al., 2026). Generative models connect shared knowledge with event reconstruction (Spens & Burgess, 2024), and RGMem consolidates user information across scales (Tian et al., 2025). We apply compression, transfer, and internalization to the same quantity: each experience's contribution to prediction. Appendix A provides a detailed comparison.

We introduce Recursive Predictive Compression (RPC), which unifies experience internalization and generative recall through one recursive predictive representation. Inspired by the role of surprise in learning (Stahl & Feigenson, 2015), RPC uses unexpected outcomes to propose corrections to the current predictor. The same rule recursively separates each correction into a candidate shared component and local residuals across scales. Measured predictive transfer proposes nested candidate scopes, while later task and recall outcomes select the depth and scope of reuse and the details to retain. The outcome used to form a correction is excluded from the evidence for accepting it. Different experiences can therefore reach different depths of reuse within the same memory process.

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.

This recursive representation connects knowledge formation with generative recall. Compatible predictive contributions are absorbed into a fixed-capacity world state, where they continue to shape future predictions after their separate records are released. Recall draws on this internalized world knowledge, supplemented by retained key details. The recall cue selects relevant residuals, which supply the details and exceptions specific to an event. Forgetting can therefore take two forms: releasing a record whose predictive influence has been internalized, or discarding residual detail when later evidence supports doing so.

RPC thus connects compression, transfer, and internalization while distinguishing their effects. Compression determines which details remain; scope expansion determines where a contribution is reused; absorption determines where its predictive influence is stored. Under a fixed linear predictive interface and the stated compatibility conditions, shared components and local residuals exactly reconstruct the original correction before lossy release, and absorption preserves the full predictive distribution. A joint update bound accounts for transfer gains on newly covered contexts, the risk of discarding detail, and changes in resource use, giving a sufficient condition for improving the combined loss–resource objective.

We evaluate RPC on hidden-rule inference in Symbolic Alchemy, controlled decisions and multiscale reconstruction, long-horizon memory in LongMemEval-V2, and visual prediction in Two-Episode Procgen, each under a specified memory-state cap. In Alchemy, RPC reduces future-recall NLL from 1.234 with H-MEM to 0.951 and increases closed-loop return from 64.0 with MemTree to 84.6 under a shared controller. At 8 KiB, it improves controlled decision accuracy from 58.31% with MemTree to 59.40%, alongside higher weighted utility. Answer-content reconstruction and visual prediction provide complementary evidence for retaining useful information. Paired ablations support the value of scope selection, validation using later outcomes, and residual detail. In a separate controlled absorption comparison, RPC uses 16.8% less resident memory state than the variant without absorption.

Our contributions are threefold:

- **A unified recursive predictive representation.** RPC connects experience internalization, generative recall, and selective forgetting by recursively separating predictive contributions into shared structure and local residuals.
- **Compression, scope expansion, and prediction-preserving absorption.** We establish conditional guarantees for exact decomposition and prediction-preserving absorption, derive a joint loss–resource update bound, and bound selection error among frozen candidate scopes under explicit fixed-block sampling assumptions.
- **Empirical evidence under memory budgets.** Across four task families, RPC improves prediction and reconstruction, decisions and utility in controlled tasks, and Alchemy return under a shared controller. Paired ablations support the value of scope selection, validation on later outcomes, residual detail, and absorption.

2 Recursive Predictive Compression

RPC recursively splits predictive corrections into shared structure and local detail. Compression releases detail, while scope expansion broadens reuse. Under the stated interface and compatibility conditions, absorption moves contributions into the world state without changing predictions. Later outcomes guide reuse and retention; recall combines the world state with retained detail within the memory budget (Figure 1).

2.1 Recursive predictive representation

A predictive correction represents a proposed local change to the predictor after an observed outcome. The core rule splits it into candidate shared structure and local detail, then repeats the split on the shared component. A correction $h \in \mathcal{H}_Z$ assigns d-dimensional predictive contributions to fixed context anchors $\mathcal{Z} = \{z_1, \ldots, z_N\}$, with positive averaging weights ν_i summing to one. Consider nested candidate partitions $\mathcal{P}_0 \preceq \cdots \preceq \mathcal{P}_K$, starting from singletons, with $\mathcal{G}_k = \sigma(\mathcal{P}_k)$ and $\mathcal{G}_{k+1} \subseteq \mathcal{G}_k$. Their construction from predictive transfer follows in Section 2.2.

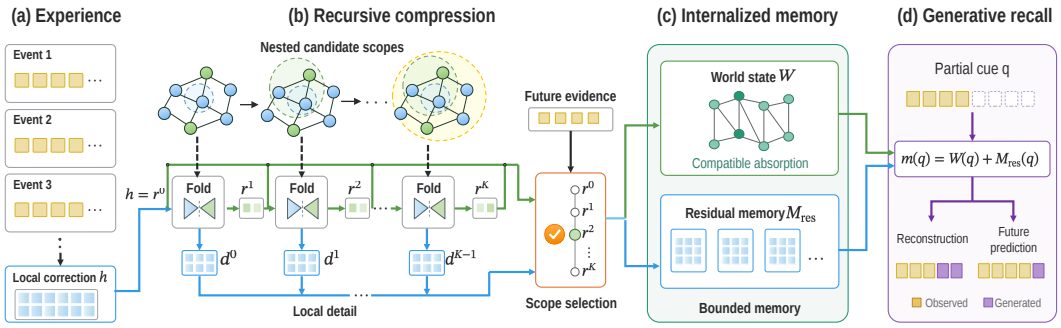

Figure 1: **RPC overview. (a)** Experience produces a local correction h. **(b)** Recursive folds separate shared structure from local detail along nested scopes proposed by diffusion, exactly reconstructing h before lossy release. Future evidence selects the coarse component (r^2 here) and retained details. **(c)** Compatible contributions enter the fixed-capacity world state W, preserving the full predictive distribution under the stated interface and compatibility conditions; local detail remains in residual memory. **(d)** The world prior and cue-relevant residuals support reconstruction and future prediction.

The fold $Q_k f = \mathbb{E}_\nu[f \mid \mathcal{G}_k]$ takes the ν-weighted average within each group. Starting with $r^0 = h$, RPC applies the same rule at successive coarser partitions:

$$r^{k+1} = Q_{k+1} r^k = Q_{k+1} h, \qquad d^k = r^k - r^{k+1} = (Q_k - Q_{k+1})h. \tag{1}$$

The remainder r^{k+1} is candidate shared structure; d^k retains the variation removed by that average. After two folds, for example, $h = d^0 + d^1 + r^2$: the coarse component and both levels of detail recover the original correction. Because the Q_k are nested orthogonal projections and $Q_0 = I$,

$$h = \sum_{k=0}^{K-1} d^k + r^K, \qquad \mathbb{E}_\nu[d^k \mid \mathcal{G}_{k+1}] = 0. \tag{2}$$

The components are mutually orthogonal. Together, they reconstruct h, and their squared norms sum to $\|h\|_\nu^2$. This identity holds at every depth before lossy release. Folding reorganizes information; compression releases selected detail, and scope expansion changes where the coarse component is used. Subsequent outcomes govern these prediction-changing operations.

Predictive interface and new experience. The same representation connects new observations to future predictions. A query $q = (a, x)$ specifies a task and context, possibly a partial recall cue, and fixed routing π assigns it to an anchor. In memory state X, a fixed-capacity world state W and explicit residual records S jointly correct the frozen base predictor. Writing M_S^{res} for the field synthesized from retained records,

$$\begin{aligned} m_X(q) &= W(z_{\pi(q)}) + M_S^{\text{res}}(z_{\pi(q)}), \\ \eta_X(q) &= \eta_0(q) + B_\theta(q) m_X(q), \\ P_X(\cdot \mid q) &= p_a(\cdot; \eta_X(q)), \qquad \widehat{y}_X(q) = g_a(\eta_X(q)). \end{aligned} \tag{3}$$

Here η_0 is the memory-free natural parameter and $B_\theta(q)$, abbreviated $B(q)$, is a shared linear readout. The task distribution p_a, decision link g_a, and base model may be nonlinear; each task declares its proper loss and normalization. Structural guarantees require aligned code coordinates, with routing, geometry, and readout fixed within each comparison epoch. These identities assume exact arithmetic. In finite precision, representable coefficients alone do not ensure exact preservation; rounding and evaluation order must also be accounted for.

RPC predicts from X_t^- before observing y_t. A fixed rule uses the outcome and cached prediction to form a bounded correction $u_t \in \mathbb{R}^d$ in shared coordinates, producing

$$h_t(z_j) = \mathbf{1}\{j = \pi(q_t)\} u_t, \qquad j = 1, \ldots, N. \tag{4}$$

The new correction is held for validation on later outcomes; compatible corrections may be combined before decomposition. The correction need not equal the raw prediction error.

2.2 SELECTING AND VALIDATING BROADER REUSE

Candidate scopes from predictive transfer. The recursive rule takes nested groups as input; predictive transfer guides their construction. RPC builds a symmetric graph whose edges require positive held-out transfer in both directions. Heat diffusion $\mathcal{T}_{s,0} = e^{-s\mathcal{L}_0}$, with self-adjoint positive-semidefinite Laplacian $\mathcal{L}_0$, smooths anchor signals to propose these candidate groups. Diffusion distances do not increase with scale s, so edges admitted at a fixed threshold $\varepsilon_{\text{geo}} > 0$ persist and connected components merge into coarser partitions. Distinct merge events define the candidate depths. Diffusion scale differs from online time t; geometry stays fixed within each epoch, and recalibration restarts affected comparisons. Appendix B.3 gives the construction and nestedness proof.

Expanding reuse while preserving local predictions. Write $A_k(i)$ for the cell containing z_i. For a mode h_m centered at anchor m, let $r_m^k = Q_k h_m$, $r_m^k(q) = r_m^k(z_{\pi(q)})$, and $\chi_k(q; m) = \mathbf{1}\{z_{\pi(q)} \in A_k(m)\}$. Keeping its current scope contributes $M_{m,k}^{\text{keep}} = \chi_k r_m^k$. To test a broader ancestor $\ell > k$, RPC retains a local compensating term and expands only the coarse remainder:

$$M_{m,k\to\ell}^{\text{jump}}(q) = \chi_k(q; m)\big(r_m^k(q) - r_m^\ell(q)\big) + \chi_\ell(q; m) r_m^\ell(q). \tag{5}$$

The compensating term preserves predictions within the original scope; predictions can change only in the added region $A_\ell(m) \setminus A_k(m)$. RPC compares distinct ancestors in the frozen family, allowing useful broad scopes even when intermediate steps fail. Each candidate endpoint contains one coarse component and its associated retained detail.

Future evidence selects deployment. The decomposition supplies candidate structure; subsequent task and recall outcomes determine whether broader reuse or detail release is useful. For an admission, scope expansion, residual reduction, or readout change $\rho : X \to X'$, RPC records both endpoints' predictions on the same subsequent queries before observing their outcomes. Formation and fitting observations are excluded: surprise creates a candidate, while later outcomes validate its use.

Let $\widehat{g}_{\rho,v}$ be the mean paired loss reduction for an applicable task or recall outcome type v, with positive values favoring the candidate. The score s_ρ combines declared gains with storage and computation costs. RPC accepts a prediction-changing rewrite only when

$$s_\rho > \tau_\rho, \qquad \widehat{g}_{\rho,v} - b_{\rho,v} \geq -\epsilon_v \quad (v \in \mathcal{V}_\rho), \tag{6}$$

The first condition checks the overall score against τ_ρ, including resource costs. The second checks each estimated task or recall gain against its tolerated degradation $\epsilon_v \geq 0$, after subtracting the uncertainty penalty $b_{\rho,v} \geq 0$. Scores and checks are fixed before evaluation; prediction and task readouts may use separate empirical gates.

2.3 INTERNALIZATION AND GENERATIVE RECALL

Compatible absorption preserves an accepted contribution while changing where it is stored. Under the conditions below, its predictive influence persists in world state after its separate record is released. A residual mode (ζ, c) contributes $\Psi_\zeta c$, where its key ζ fixes the scope, coefficient space, representation version, and linear synthesis map. World coefficients with the same key use the same map.

Aggregation and absorption. If $m \geq 2$ records share a key, no active comparison refers to them separately, and their sum fits in one coefficient block, RPC replaces them by $(\zeta, \sum_{j=1}^m c_j)$. Linearity preserves the combined field. Let $c_{\text{mode}}(\zeta) > 0$ be the words allocated to one independent record.

The fixed-capacity world state has preallocated slots, counted in resident storage, for a finite set of terminal keys $\mathcal{Z}_W$:

$$W = \sum_{\zeta \in \mathcal{Z}_W} \Psi_\zeta w_\zeta.$$

A deployed mode is world-compatible when an accepted depth rewrite has placed it at one of these keys, no active comparison refers to it separately, and $w_\zeta + c$ is representable. RPC updates $w_\zeta \leftarrow w_\zeta + c$ and releases the explicit record. The full task and recall distributions stay unchanged while the world-only prior gains the absorbed contribution. Incompatible corrections remain explicit. Storage is saved by aggregation, absorption, or evidence-supported residual release, rather than by folding alone.

Recall from world knowledge and retained detail. Recall reads this same representation using a partial cue. For $q = (a_{\mathrm{rec}}, \xi)$, compared predictions receive the same cue and, for token-level evaluation, the same fixed or teacher-forced prefix. Write $\overline{W}_X(q) = W(z_{\pi(q)})$ and $\overline{M}_X(q) = M_S^{\mathrm{res}}(z_{\pi(q)})$. The world state supplies a cue-conditioned prior, and residuals refine it:

$$P_W(e \mid \xi) = p_{a_{\mathrm{rec}}}\big(e; \eta_0(q) + B(q)\overline{W}_X(q)\big)\,, \tag{7}$$

$$P_X(e \mid \xi) = p_{a_{\mathrm{rec}}}\big(e; \eta_0(q) + B(q)[\overline{W}_X(q) + \overline{M}_X(q)]\big)\,. \tag{8}$$

With no active residual contribution, the full read reduces to P_W. Absorption changes what this prior carries without changing the combined reconstruction. The cue selects a reconstruction through the fixed routing map; it need not identify a unique historical record.

An optional calibrated readout weights retained detail by a strength in $[0, 1]$, fitted from past evidence without the query outcome. Exact rewrites additionally require unchanged readout parameters and decoded inputs; subsequent recall outcomes validate the detail's continued use (Appendix B.5).

2.4 Online updates under a memory budget

The complete state is $X = (W, S, R^{\mathrm{prob}}, \Xi)$, including probationary records and the readout parameters, comparison statistics, and reserved caches in Ξ. Let $F_X : q \mapsto P_X(\cdot \mid q)$ denote the deployed predictor and $C(X)$ the total resident word count. Every operation must satisfy $C_{\mathrm{peak}}(\mathcal{A}; X) \le B_{\max}$, counting resident state, both endpoints, and workspace, with shared allocations counted once. Exact aggregation and absorption require invariance and a positive resource benefit, including rewrite cost.

Each round predicts, scores older candidates on the observed outcome, deploys accepted changes, and forms a new atom for later evaluation; compatible modes can then be aggregated or absorbed. Comparisons restart when a rewrite changes their paired predictions or score terms. Algorithm 1 gives the full update order; Appendix B details the resource rules.

3 Theoretical Analysis

RPC separates the cost of discarding detail, the benefit of broader reuse, and the storage gain from internalization. We first characterize these effects, then combine them into a sufficient condition for an entire memory update to improve its loss–resource objective. Comparisons use fixed routing, a shared linear predictive interface, and a common query distribution; exact internalization also requires compatible synthesis maps and the stated allocation conditions. Proofs are in Appendix C.

3.1 Compression and future risk

For the true future outcome law, let $f_q(\eta) = \mathbb{E}_*[\ell_q(\eta, Y) \mid q]$ be the conditional loss and $R(X) = \mathbb{E}_q f_q(\eta_X(q))$. Assume f_q has an L-Lipschitz gradient along each compared parameter segment, $\|B(q)\|_{\mathrm{op}} \le b$, and $\Pr(\pi(q) = i) \le \gamma\nu_i$. The coverage factor γ relates future queries to the reference weights used for compression. Write $\varepsilon_X^2 = \mathbb{E}_q \|\nabla f_q(\eta_X(q))\|^2 < \infty$.

Theorem 3.1 (Compression distortion and risk)**.** *For a rewrite with actual synthesized field change $e = (W' + M_{S'}^{\mathrm{res}}) - (W + M_S^{\mathrm{res}})$ and $D = \|e\|_\nu$,*

$$R(X') - R(X) \le \varepsilon_X b\sqrt{\gamma}\, D + \frac{Lb^2\gamma}{2} D^2. \tag{9}$$

If the rewrite only removes orthogonal details $\{d^k : k \in \mathcal{D}\}$ from one recursive decomposition, then $D^2 = \sum_{k\in\mathcal{D}} \|d^k\|_\nu^2$.

Denote this upper bound by $\mathcal{B}_X(D)$. At a conditionally stationary predictor ($\varepsilon_X = 0$), the bound is quadratic in code distortion. Otherwise the first-order term matters: a large residual can reinforce a mistaken prediction. Appendix C.2 separates distribution change from model bias exactly for exponential-family log loss. For overlapping or masked records, D measures the combined field change; individual record energies need not add.

3.2 BROADER REUSE AND SUBSEQUENT EVIDENCE

Consider a single ancestor jump, with all other contributions fixed, and let $F = A_\ell(m) \setminus A_k(m)$ and $p_F = \Pr(z_{\pi(q)} \in F)$.

Theorem 3.2 (Frontier transfer gain)**.** *For any integrable prediction loss and $p_F > 0$, the gain $G = R(X^{\mathrm{keep}}) - R(X^{\mathrm{jump}})$ satisfies*

$$G = p_F \mathbb{E}_*[\ell_{\mathrm{keep}} - \ell_{\mathrm{jump}} \mid F]. \tag{10}$$

For squared error with $\widehat{y}_X = \eta_X$, set $a(q) = \mathbb{E}_*[Y \mid q] - \widehat{y}_{\mathrm{keep}}(q)$ and $v(q) = B(q) r^{\ell}_m(q)$. Then

$$G = p_F \left(2\mathbb{E}[a^\top v \mid F] - \mathbb{E}[\|v\|^2 \mid F] \right). \tag{11}$$

The gain is zero when $p_F = 0$.

The original scope stays intact; all gain comes from the added region. Under squared error, reuse helps exactly when twice its alignment with the remaining error exceeds its squared magnitude. This identifies what later outcomes must validate after geometry proposes a scope.

Selecting among candidate scopes. For a fixed-block specialization of validation, freeze M feasible candidates before n independent future observations, with paired loss differences bounded by H in absolute value. Let $\beta_n = H\sqrt{2\log(2M/\delta)/n}$. Deploy the empirical best candidate only if its mean gain exceeds β_n; otherwise retain the current state. With probability at least $1 - \delta$,

$$G_{\mathrm{deploy}} \geq \max\{0, \max_{\rho} G_\rho - 2\beta_n\}. \tag{12}$$

Thus evidence controls the gap to the best proposed scope while protecting the current predictor. Appendix C.4 states the sampling conditions and proof. This fixed-block rule is distinct from the task-specific empirical gates used in the experiments.

3.3 INTERNALIZATION AND THE COMPLETE UPDATE

Under the compatibility and allocation conditions in Section 2.3, aggregation and absorption preserve the full predictor under any future outcome law. Aggregating m compatible records releases $(m-1)c_{\mathrm{mode}}$ words; absorbing one into a preallocated world slot releases c_{mode}. An experience can therefore release its separate record while retaining its predictive influence.

Proposition 3.3 (Joint benefit of reuse, compression, and internalization)**.** *Let $X \to X^+$ be a scope expansion with gain G, $X^+ \to X'$ a compression satisfying Theorem 3.1 with distortion D, and $X' \to \bar{X}$ compatible exact internalization. For $J = R + \lambda C + \kappa \mathbb{E}\Omega$, $\lambda, \kappa \geq 0$, let $S_C = C(X) - C(\bar{X})$ and $\Delta_\Omega = \mathbb{E}\Omega(\bar{X}) - \mathbb{E}\Omega(X)$. Then*

$$J(\bar{X}) - J(X) \leq -G + \mathcal{B}_{X^+}(D) - \lambda S_C + \kappa \Delta_\Omega. \tag{13}$$

The complete update improves this objective when transfer and storage gains exceed the compression bound and added computation. Crucially, compatible absorption contributes no additional distortion: future evidence assesses broader reuse, compression controls the cost of released detail, and internalization retains the resulting predictor with less explicit state. Appendix C.5 accounts for rewrite work and peak capacity.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Evaluation settings. We evaluate reusable knowledge and retained detail under bounded memory. Symbolic Alchemy is the central test: persistent hidden chemistry links earlier interventions to later rule recall and decisions (Wang et al., 2021). Controlled synthetic tasks isolate recursive sharing, scale selection, and resource use under known structure. LongMemEval-V2 tests answer-content retention and QA over web-agent histories (Wu et al., 2026); Two-Episode Procgen tests visual prediction and control on a second attempt at the same level (Cobbe et al., 2020). We measure prediction, reconstruction, downstream performance, and online memory costs. Appendices D.1 and D.3 specify the tasks and metrics.

Comparison protocol. We reproduce the core mechanisms of RGMem (Tian et al., 2025), H-MEM (Yu et al., 2026), MemTree (Rezazadeh et al., 2025), DeMem (Zou et al., 2026), and Titans (Behrouz et al., 2026), alongside RPC and no memory. Within each environment, methods share observations, frozen task models, state caps, and evaluation streams. RPC proposes shared scopes, assesses reuse with subsequent evidence, and combines shared state with retained detail. Numerical tasks use predictive coordinates; LongMemEval-V2 uses bounded text with a shared frozen QA reader. Appendix D.2 specifies the comparison protocol.

4.2 Learning and reusing hidden rules in Alchemy

Stone–potion transformations reveal chemistry that remains useful across trials. We test prediction, recall, and decisions, followed by sustained use and internalization.

Performance across prediction, recall, and decisions. Figure 2 connects these three uses of retained chemistry. RPC reduces transition NLL from H-MEM's 1.664 to 1.615 and future-recall NLL from 1.234 to 0.951 (22.9%). With the same public-value controller for all methods, mean closed-loop return rises from MemTree's 64.0 to 84.6, a paired gain of 20.6 with a 95% interval of $[6.2, 35.0]$. RPC leads all reproduced baselines, including RGMem, on all three endpoints, linking retained knowledge to better action selection.

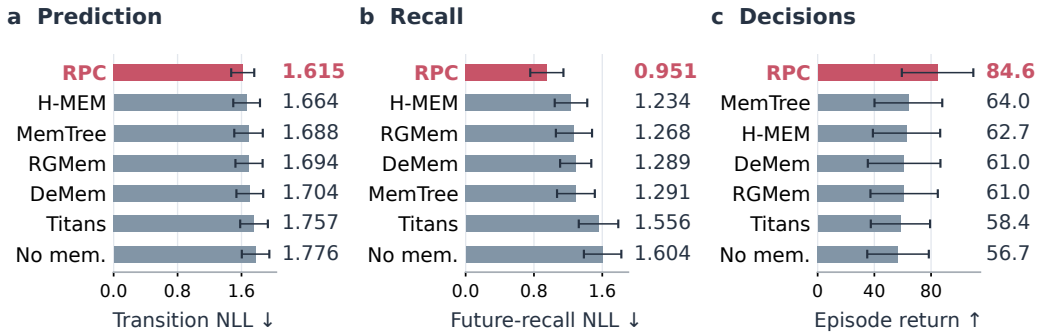

Figure 2: Prediction, recall, and decisions in Alchemy at 64 KiB. **a–b,** Transition and future-recall NLL on a shared intervention stream. **c,** Return in separate closed-loop episodes with a shared controller. Bars show means and 95% Student-t intervals over ten seeds.

Sustained use within bounded state. With a reference controller fixed across 5, 10, and 20 trials, RPC exceeds no memory by 3.00, 3.70, and 4.09 return per trial (Figure 3). Across this fourfold increase in experience, future-recall NLL decreases from 1.13 to 0.74 while resident state grows by only 11.5%, showing continued usefulness with modest storage growth.

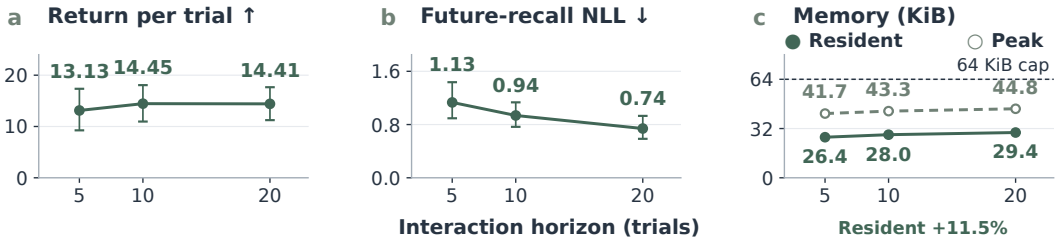

Figure 3: Longer Alchemy interaction with a fixed reference controller. **a,** Return per trial. **b,** Future-recall NLL. **c,** Resident and peak state. Error bars show 95% bootstrap intervals over ten seeds; recall queries vary across horizons.

Internalization and later reuse. Figure 4 shows how an experience's predictive contribution survives the release of its explicit record and supports a later query. The illustrated absorption releases 144 B without changing predictions. On a later unseen pairing in the same chemistry, world-only readout retains a 3.99-percentage-point probability gain over no memory, close to full RPC's 4.38-

point gain alongside retained local detail. All six compatible absorptions recorded in the ten-world evaluation logs preserve the complete correction across routing anchors (Appendix H).

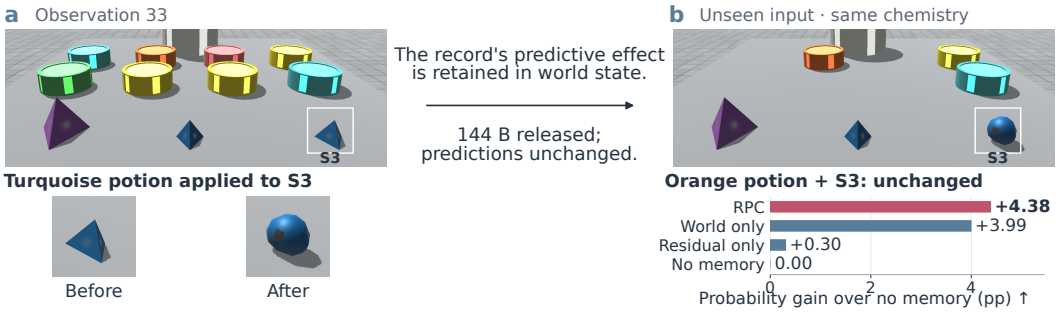

Figure 4: Internalization and later reuse in Alchemy. **a,** Absorption moves an explicit contribution into world state without changing predictions. **b,** Outcome-probability gains on a later unseen pairing over no memory (57.26%). Components are removed only at readout from the same frozen post-absorption state. Scenes render logged states in the native engine.

4.3 Controlled validation of recursive memory

Performance under a hard memory budget. The controlled suite tests recovery and reuse of known hierarchical structure across 25 conditions spanning noise, rare events, prior bias, and distribution shift. Figure 5 compares performance and peak allocation at 8 KiB. RPC leads the means on all four displayed endpoints with the lowest mean peak allocation among memory methods (5.827 KiB), and retains the four performance leads at 16 and 32 KiB. RPC reaches 59.40% decision accuracy versus MemTree's 58.31%, and utility 0.223 versus 0.196, connecting retained information to action selection. All RPC runs remain within the cap; Appendix F gives complete results.

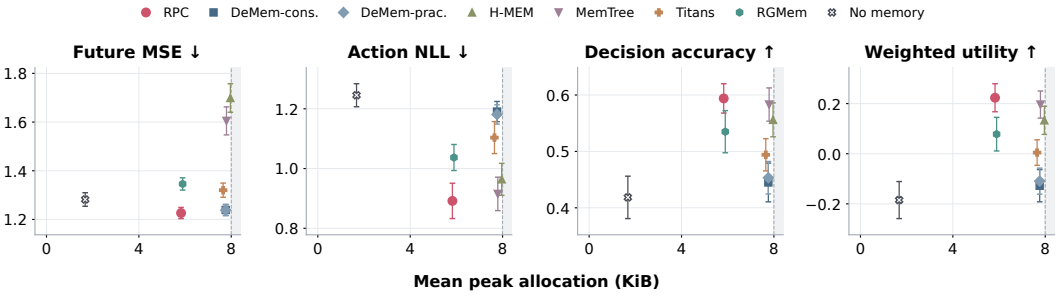

Figure 5: Controlled-suite performance against mean peak allocation at the 8 KiB cap. Points show means over ten seeds, with 95% Student-t confidence intervals.

Reconstruction at known abstraction levels. A known-structure task tests fine, intermediate, and coarse reconstruction from one 8 KiB memory. Readouts receive the same experience and perturbed cues without hierarchy labels. RPC selects the matching depth for every query, restoring residual detail for fine queries (Figure 6). Macro NMSE improves by 7.43 orders of magnitude over no memory, exceeding fixed-depth, randomized-hierarchy, and world-only controls. The same bounded representation thus recovers detail and broader regularities at the requested scale in this controlled setting.

4.4 Text retention and visual prediction

LongMemEval-V2: preserving answer-relevant content. At 128 KiB, RPC raises answer-token reconstruction from BM25's 0.323 to 0.519 and exact reconstruction from 0.257 to 0.305 on 226 phrase-answer questions. These metrics measure retained content before generation. With the shared frozen reader, RPC also achieves the highest observed QA count among the compared methods. Appendix G reports all three measures.

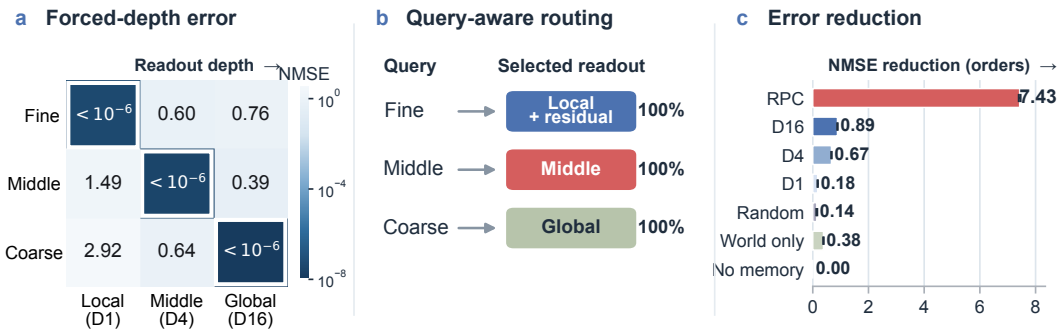

Figure 6: Recovery of known multiscale structure at the 8 KiB cap. **a,** Reconstruction error for each query scale (rows) and forced readout depth (columns); lower is better. **b,** RPC's selected memory readout for each query scale. **c,** Overall reconstruction improvement across scales relative to no memory; higher is better, and each unit represents a tenfold error reduction.

Two-Episode Procgen: using visual experience again. At 64 KiB, RPC reduces next-latent MSE from RGMem's 0.0159 to 0.0141 (11.2%). Mean attempt-two return is 0.896 versus H-MEM's 0.861, and adaptation return is 0.057 versus 0.022; success matches H-MEM. Prior experience supports better visual prediction and higher mean return under the shared controller (Appendix G).

4.5 Component ablations and resource use

Paired ablations test scope selection, evidence gating, sharing, residual detail, and absorption (Table 1). Controlled ablations use the same task-calibration procedure to compare memory components; Appendix J gives the protocol and full results.

Table 1: Component ablations. $\Delta = \mathrm{RPC} - \mathrm{ablated}$, before rounding.

Ablation (setting)	Metric	RPC	Ablated	Δ [95% CI]
Fixed depth (Synthetic)	Future MSE ↓	**1.226**	1.235	-0.009 $[-0.012, -0.006]$
Ungated (Synthetic)	Future MSE ↓	**1.226**	1.429	-0.202 $[-0.237, -0.168]$
No absorption (Synthetic)	State (KiB) ↓	**17.27**	20.75	-3.49 $[-3.96, -3.01]$
No sharing (Alchemy)	Recall NLL ↓	**0.951**	1.548	-0.597 $[-0.678, -0.516]$
World only (LongMem)	Answer-token recon. ↑	**0.519**	0.474	$+0.044$ $[+0.024, +0.068]$
World only (Procgen)	Next-latent MSE ↓	**0.0141**	0.0200	-0.0060 $[-0.0079, -0.0041]$

Caps: 8 KiB (synthetic components), 32 KiB (absorption), 64 KiB (Alchemy/Procgen), 128 KiB (LongMemEval-V2). Paired intervals use Student-t estimates over ten seeds; LongMemEval-V2 uses question bootstrap. Absorption uses 512 formation observations; other synthetic pairs use 128.

Removing evidence gating raises future MSE from 1.226 to 1.429, supporting validation beyond formation contexts; removing hierarchical sharing raises Alchemy recall NLL from 0.951 to 1.548. World-only readouts lose answer content and visual predictive detail, showing the value of retained local exceptions. The internalization path reduces resident state by 16.8% in the controlled comparison. Appendix I reports resident and peak memory, query and update time, and evidence-evaluation costs across environments.

5 Conclusion

RPC unifies experience internalization and generative recall through a recursive representation of shared structure and local residuals. Subsequent evidence guides reuse and retention under a memory budget. Under the stated interface and compatibility conditions, decomposition is exact before lossy release, and absorption preserves the complete predictive distribution after explicit records are released. Our analysis combines compression risk, transfer gains, and resource changes in a joint update bound. Across four environments, experiments show prediction and reconstruction gains, improved decisions and utility in controlled tasks, and higher Alchemy return. Together, these results provide a concrete mechanism for turning experience into reusable knowledge while retaining consequential details.

AI USE STATEMENT

In this work, we used generative AI tools to polish the manuscript by improving the clarity and readability of the writing. We also used these tools to help identify relevant literature and to support the planning and refinement of experimental designs. In addition, we used them for code debugging, including identifying potential errors and exploring possible fixes. We have reviewed all AI-assisted code, analyses, and manuscript text for accuracy and consistency. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

REFERENCES

Ali Behrouz, Peilin Zhong, and Vahab Mirrokni. Titans: Learning to memorize at test time. *Advances in Neural Information Processing Systems*, 38:113506–113543, 2026.

Karl Cobbe, Chris Hesse, Jacob Hilton, and John Schulman. Leveraging procedural generation to benchmark reinforcement learning. In *International conference on machine learning*, pp. 2048–2056. PMLR, 2020.

Ronald R Coifman and Stéphane Lafon. Diffusion maps. *Applied and computational harmonic analysis*, 21(1):5–30, 2006.

Ronald R Coifman and Mauro Maggioni. Diffusion wavelets. *Applied and computational harmonic analysis*, 21(1):53–94, 2006.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, et al. Lightmem: Lightweight and efficient memory-augmented generation. *arXiv preprint arXiv:2510.18866*, 2025.

Zafeirios Fountas, Adnan Oomerjee, Haitham Bou-Ammar, Jun Wang, and Neil Burgess. Why the brain consolidates: Predictive forgetting for optimal generalisation. *arXiv preprint arXiv:2603.04688*, 2026.

Bernal J Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. *Advances in neural information processing systems*, 37:59532–59569, 2024.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33: 9459–9474, 2020.

James L McClelland, Bruce L McNaughton, and Randall C O'Reilly. Why there are complementary learning systems in the hippocampus and neocortex: insights from the successes and failures of connectionist models of learning and memory. *Psychological review*, 102(3):419, 1995.

Charles Micchelli and Massimiliano Pontil. Kernels for multi–task learning. *Advances in neural information processing systems*, 17, 2004.

David G Nagy, Gergő Orbán, and Charley M Wu. Adaptive compression as a unifying framework for episodic and semantic memory. *Nature Reviews Psychology*, 4(7):484–498, 2025.

Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for llms. In *International Conference on Learning Representations*, volume 2025, pp. 990–1023, 2025.

Eleanor Spens and Neil Burgess. A generative model of memory construction and consolidation. *Nature human behaviour*, 8(3):526–543, 2024.

Eleanor Spens and Neil Burgess. Hippocampo-neocortical interaction as compressive retrieval-augmented generation. *Nature Communications*, 2026.

Aimee E Stahl and Lisa Feigenson. Observing the unexpected enhances infants' learning and exploration. *Science*, 348(6230):91–94, 2015.

Weinan Sun, Madhu Advani, Nelson Spruston, Andrew Saxe, and James E Fitzgerald. Organizing memories for generalization in complementary learning systems. *Nature neuroscience*, 26(8): 1438–1448, 2023.

Yu Sun, Xinhao Li, Karan Dalal, Jiarui Xu, Arjun Vikram, Genghan Zhang, Yann Dubois, Xinlei Chen, Xiaolong Wang, Sanmi Koyejo, et al. Learning to (learn at test time): Rnns with expressive hidden states. *arXiv preprint arXiv:2407.04620*, 2024.

Ao Tian, Yunfeng Lu, Xinxin Fan, Changhao Wang, Lanzhi Zhou, Yeyao Zhang, and Yanfang Liu. Rgmem: Renormalization group-based memory evolution for language agent user profile. *arXiv e-prints*, pp. arXiv–2510, 2025.

Jane X Wang, Michael King, Nicolas Porcel, Zeb Kurth-Nelson, Tina Zhu, Charlie Deck, Peter Choy, Mary Cassin, Malcolm Reynolds, Francis Song, et al. Alchemy: A benchmark and analysis toolkit for meta-reinforcement learning agents. *arXiv preprint arXiv:2102.02926*, 2021.

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.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, et al. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. In *International Conference on Learning Representations*, volume 2026, pp. 91581–91617, 2026.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z Pan, et al. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12805–12825, 2026.

Jiawei Yu, Yixiang Fang, Xilin Liu, and Yuchi Ma. H-mem: A novel memory mechanism for evolving and retrieving agent memory via a hybrid structure. *arXiv preprint arXiv:2605.15701*, 2026.

Dylan Zhang, Yanshan Lin, Zhengkun Wu, Yihang Sun, Bingxuan Li, Dianqi Li, and Hao Peng. Useful memories become faulty when continuously updated by llms. *arXiv preprint arXiv:2605.12978*, 2026.

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents. *arXiv preprint arXiv:2506.15841*, 2025.

Mingxi Zou, Zhihan Guo, Langzhang Liang, Zhuo Wang, Qifan Wang, Qingsong Wen, Irwin King, Lizhen Qu, and Zenglin Xu. Remember the decision, not the description: A rate-distortion framework for agent memory. *arXiv preprint arXiv:2605.10870*, 2026.

A Related Work

Memory internalization and generative recall. Complementary learning systems connect rapid episodic encoding with slower integration of shared regularities (McClelland et al., 1995). Generalization-optimized consolidation links memory transfer to its benefit for generalization (Sun et al., 2023). Adaptive compression treats episodic and semantic memory as a joint structure-learning process (Nagy et al., 2025). Generative models learn reusable structure through replay and reconstruct experiences from this structure and event-specific information (Spens & Burgess, 2024; 2026). RPC explicitly represents experience as predictive corrections, recursively separating reusable components from local residuals across scales. Recall combines the evolving world prior with surviving residuals, linking future predictive contributions to the information needed for reconstruction. A precise preservation criterion governs the transition from explicit memory to shared predictive state.

Hierarchical memory and adaptive abstraction. RAG augments generation with retrieved passages (Lewis et al., 2020), while HippoRAG connects evidence through a knowledge graph and personalized PageRank (Gutiérrez et al., 2024). MemTree evolves a semantic summary hierarchy (Rezazadeh et al., 2025); H-Mem combines a temporal-semantic tree with an entity graph (Yu et al., 2026). MemGAS constructs multigranularity associations and adapts retrieval granularity to the current query (Xu et al., 2026). RPC's adaptive depth governs how broadly an experience changes future predictions across contexts, extending beyond retrieval granularity. Measured predictive transfer defines candidate scopes, and subsequent outcomes determine whether a correction is deployed more broadly. Diffusion geometry (Coifman & Lafon, 2006; Coifman & Maggioni, 2006) and multi-task kernels (Micchelli & Pontil, 2004) supply tools for constructing these candidates.

Predictive compression and multiscale semantic memory. RGMem (Tian et al., 2025) evolves conversational evidence into relation-level summaries and user profiles through thresholded semantic synthesis and hierarchical updates. RPC connects compression, transfer, and internalization within one recursive predictive operator, extending beyond conversational user profiling. It compresses predictive corrections into coarse components and residual details that exactly reconstruct the original correction before lossy release. Under the stated fixed routing, linear interface, coefficient compatibility, and world-slot allocation conditions, absorption preserves the complete predictive distribution while moving explicit contributions into bounded world state, allowing separate records to be removed without losing their predictive influence.

Memory updates, consolidation, and forgetting. LightMem separates filtering, consolidation, and offline updates (Fang et al., 2025). Memory-R1 learns memory-management actions from downstream rewards (Yan et al., 2026), while MEM1 jointly learns compact state updates and reasoning (Zhou et al., 2025). Test-time training and Titans update neural memory during sequence processing (Sun et al., 2024; Behrouz et al., 2026). Predictive forgetting targets future generalization (Fountas et al., 2026), and repeated LLM consolidation can reduce memory utility (Zhang et al., 2026). DeMem refines bounded memory partitions when decision conflicts are certified (Zou et al., 2026). RPC's additive decomposition combines shared coarse structure with distinct local residuals, preserving explicit local exceptions. Subsequent task and recall outcomes govern prediction-changing rewrites. It connects three distinct operations: compression controls which detail survives, scope expansion controls where a correction is reused, and compatible absorption changes where its influence is stored while preserving predictions.

B Model and Algorithm Details

B.1 Shared interface and innovation construction

For the finite anchor bank $\mathcal{Z}$ and positive reference weights ν, the shared memory space is

$$\mathcal{H}_Z = L^2(\mathcal{Z}, \nu; \mathbb{R}^d), \qquad \langle f, g \rangle_\nu = \sum_i \nu_i f(z_i)^\top g(z_i). \tag{14}$$

The routing map $\pi : \mathcal{Q} \to \{1, \dots, N\}$ is fixed and measurable. Memory codes share aligned coordinates across tasks, with a fixed linear readout $B(q)$ into natural parameters. The task distribution and decision link may be nonlinear. For a mode key ζ, the linear synthesis map $\Psi_\zeta : \mathcal{C}_\zeta \to \mathcal{H}_Z$ is shared by all records and world coefficients carrying that key.

One concrete innovation rule is

$$u_t = \operatorname*{arg\,min}_{\|u\| \le U} \left\{ \widetilde{\ell}_{a_t}\big(g_{a_t}(\eta_t^-(q_t) + B(q_t)u), y_t\big) + \frac{1}{2\eta} \|u\|_{G_t}^2 \right\}. \tag{15}$$

Here $\eta_t^-(q_t) = \eta_{X_t^-}(q_t)$ is cached before revealing y_t, $G_t \succ 0$, $\eta > 0$, and $U < \infty$. If the loss term is convex in u, the positive-definite quadratic makes the solution unique. The main structural results require only a bounded deterministic innovation in the common coordinates. Compatible corrections can be averaged during formation; the resulting code is the object subsequently decomposed exactly.

B.2 Transfer calibration

The formal hierarchy uses a fixed graph whose edges indicate validated predictive transfer. To construct it without reusing validation outcomes, use three disjoint data splits: produce source innovations, propose candidate edges and fix their symmetric magnitudes $\widehat{\kappa}_{ij} \ge 0$, then validate transfer directions. For a source innovation u_p and target event r, define

$$\Gamma_{p\to r} = \widetilde{\ell}_{a_r}\big(g_{a_r}(\eta_r^-), y_r\big) - \widetilde{\ell}_{a_r}\big(g_{a_r}(\eta_r^- + B(q_r)u_p), y_r\big).$$

Let $\underline{\gamma}_{i\to j}$ be the simultaneous lower confidence bound for transfer from i to j. Set

$$w_{ij} = \widehat{\kappa}_{ij} \mathbf{1}\{\underline{\gamma}_{i\to j} > 0,\ \underline{\gamma}_{j\to i} > 0\}.$$

Both directions must pass, with a summable error allocation over tested directions. Anchors, assignments, readout, and source innovations are fixed before the validation outcomes are used. The graph remains fixed within an online comparison epoch. Persistent routing novelty may motivate recalibration between epochs; changes to graph or readout versions restart affected comparisons.

B.3 Heat diffusion and hierarchy construction

For scalar fields, $e_i(z_j) = \mathbf{1}\{i = j\}/\nu_i$ is the evaluation representer: $\langle e_i, f\rangle_\nu = f(z_i)$. Let $\mathcal{E} = \{\{i,j\} : w_{ij} > 0\}$ and define the weighted edge difference in the standard edge space by

$$(C_0 f)_{\{i,j\}} = \sqrt{w_{ij}}\big(f(z_i) - f(z_j)\big).$$

With the adjoint taken between the ν-weighted anchor space and that edge space, set

$$\mathcal{L}_0 = \frac{1}{2} C_0^* C_0, \qquad \mathcal{T}_{s,0} = e^{-s\mathcal{L}_0}. \tag{16}$$

Thus $\mathcal{L}_0$ is self-adjoint and positive semidefinite. The graph signal $f_s = \mathcal{T}_{s,0} f$ follows the deterministic heat equation

$$\frac{\partial f_s}{\partial s} = -\mathcal{L}_0 f_s, \qquad f_0 = f.$$

The anchor domain is discrete, while $s \ge 0$ is a continuous scale; the construction introduces no stochastic noise. Define the diffusion distance by

$$D_s(i,j) = \|\mathcal{T}_{s,0}(e_i - e_j)\|_\nu, \tag{17}$$

For a ν-orthonormal eigensystem (μ_ℓ, φ_ℓ) of $\mathcal{L}_0$, it satisfies

$$D_s^2(i,j) = \sum_\ell e^{-2s\mu_\ell}\big(\varphi_\ell(z_i) - \varphi_\ell(z_j)\big)^2. \tag{18}$$

For every calibrated edge, its first threshold scale is

$$\tau_{ij} = \inf\{s \ge 0 : D_s(i,j) \le \varepsilon_{\mathrm{geo}}\},$$

with $\inf \varnothing = \infty$. Starting from singleton cells, add edges at each distinct finite τ_{ij}, including zero, and add ties together. Retain a new partition only when connected components merge. These merge events discretize heat flow into the levels $k = 0, \dots, K$, with

$$\mathcal{P}_0 \preceq \mathcal{P}_1 \preceq \cdots \preceq \mathcal{P}_K, \qquad \mathcal{G}_k = \sigma(\mathcal{P}_k), \quad \mathcal{G}_{k+1} \subseteq \mathcal{G}_k. \tag{19}$$

The hierarchy is configured at the beginning of an epoch, rather than recomputed for each incoming atom.

Proposition B.1 (Diffusion-induced hierarchy)**.** *For every anchor pair, $D_s(i,j)$ is nonincreasing in s. Threshold edges therefore persist, and the resulting component partitions are nested.*

Proof. For $v_{ij} = e_i - e_j$, self-adjointness gives

$$D_s^2(i,j) = \left\langle v_{ij}, e^{-2s\mathcal{L}_0} v_{ij} \right\rangle_\nu .$$

Differentiating,

$$\frac{d}{ds} D_s^2(i,j) = -2 \left\| \mathcal{L}_0^{1/2} e^{-s\mathcal{L}_0} v_{ij} \right\|_\nu^2 \le 0.$$

A threshold edge cannot disappear as s increases. Adding edges can merge components but cannot split them, proving the nested partition claim. □

B.4 Paired evidence and online updates

We compare states through

$$J(X) = \mathbb{E}[\widetilde{\ell}(F_X)] + \lambda C(X) + \kappa \mathbb{E}[\Omega(X)], \tag{20}$$

where $\widetilde{\ell}$ is the declared task-normalized loss, $\lambda, \kappa \ge 0$ price resources, and Ω includes deployed and candidate inference and online decision work.

A proposed rewrite $\rho : X \to X'$ is compared with the deployed state on the same subsequent queries. Both predictions are recorded before each outcome. For outcome type v and its nonempty evidence set $E_{\rho,v}$, the paired gain is

$$\widehat{g}_{\rho,v} = \frac{1}{|E_{\rho,v}|} \sum_{t \in E_{\rho,v}} \left[\widetilde{\ell}_t^v(F_X) - \widetilde{\ell}_t^v(F_{X'}) \right]. \tag{21}$$

Formation and fitting outcomes are excluded: a surprising observation creates a candidate, while later outcomes validate its use.

Scores can use mean gains directly or subtract a fixed multiple of their standard error; all score definitions and thresholds are fixed before evaluation. Algorithm 1 applies ranked admissions, scope changes, residual reductions, and readout changes within epochs of fixed routing, geometry, readout, and world slots.

For recall, both endpoints receive the same partial cue and, for token prediction, the same observed prefix. Each candidate stores only the counts and gain statistics required by its declared score and uncertainty penalty.

A change to the candidate, deployed comparator, routing, geometry, or readout starts a fresh comparison unless all paired predictions and score terms remain unchanged. This check also applies between successive accepted rewrites within a round, so a later rewrite cannot rely on evidence for an invalidated comparator.

B.5 Readout compatibility and exact rewrites

The optional calibrated decoder controls how much retained detail is unfolded:

$$\eta_{X,\phi}(q) = \eta_{X,\phi}^{\mathrm{base}}(q) + \beta_{X,\phi}(q)\, \Delta\eta_X^{\mathrm{detail}}(q), \qquad 0 \le \beta_{X,\phi}(q) \le 1.$$

The base read combines the world prior with the selected correction; the refinement uses only detail still present in X. The parameters ϕ and addressing statistics belong to Ξ, and F_X includes the calibrated distribution $p_a(\cdot; \eta_{X,\phi}(q))$.

Readout parameters are fitted on past evidence and frozen before comparison; recalibration requires fresh acceptance evidence after fitting. Cue addressing reads retained codes, and its statistics count toward the state budget.

For this decoder, the invariance of an exact rewrite additionally requires unchanged parameters and decoded inputs, including any separate addressing of world and residual records. Preserving only their summed code does not by itself establish invariance for a decoder that treats those records differently.

Algorithm 1 RPC online memory update

Require: Initial state X_1^-, fixed epoch interface, capacity $B_{\max}$, and gate settings
for round $t = 1, 2, \ldots$ **do**
  Cache predictions from X_t^- and active candidate endpoints
  Observe y_t; update evidence for candidates formed before this outcome
  $X \leftarrow X_t^-$
  for each ranked prediction-changing rewrite $\rho : X \to X'$ **do**
    if its comparison is valid, equation 6 holds, and the peak cap is met **then**
      $X \leftarrow X'$; reset invalidated comparisons
    end if
  end for
  Form u_t from the cached prediction and y_t; construct h_t
  if probation and candidate workspace fit the peak cap **then**
    Stage h_t and its candidate scopes for subsequent validation
  end if
  Aggregate or absorb compatible modes under the exact resource rule
  Reset affected comparisons; set $X_{t+1}^- \leftarrow X$
end for

Exact resource rule. For invariant aggregation or absorption, the benefit is a resource difference:

$$s_t^{\text{exact}} = -\lambda\big(C(X') - C(X)\big) - \kappa\big(\Omega_t(X') - \Omega_t(X) + R_t(\rho)\big),$$

where $R_t(\rho)$ is the one-time rewrite cost. The rewrite requires $s_t^{\text{exact}} > 0$, representable coefficients, and sufficient peak capacity. Active evidence can be retained only if the rewrite leaves its paired predictions and resource differences unchanged; otherwise the rewrite is delayed or the affected comparison is reset.

B.6 State budgets and residual release

The complete resident count $C(X) \in \mathbb{N}$ includes world coefficients, explicit codes and keys, probation buffers, and decision state Ξ. The latter includes readout and addressing parameters, candidate descriptions, evidence statistics, and reserved caches. Every nonempty object costs at least $c_{\min} \in \mathbb{N}_{>0}$ words, and each round launches finitely many candidates. For an operation $\mathcal{A}$, $C_{\text{peak}}(\mathcal{A}; X)$ counts the maximum simultaneous allocation, including both endpoints and temporary workspace; shared allocations are counted once. Thus the cap bounds both endpoint states and the allocations needed to execute the transition.

A record reduces resident storage only when its allocation is released. The computation cost Ω_t includes deployed and candidate predictions, evidence updates, and candidate maintenance. Probation and comparison space are reserved before use; candidates that do not fit are discarded without changing deployed memory.

A finite-precision implementation may also propose a declared chain $\mathsf{q}_0(c) = c, \mathsf{q}_1(c), \ldots, \mathsf{q}_J(c) = 0$ with strictly decreasing encoded word costs. Each reduction must pass the same empirical acceptance rule. Quantization may change both cue matching and reconstruction, so the comparison uses the complete decoded endpoint. No monotonic fidelity assumption is required.

C Compression, Transfer, and Recall Guarantees

All expectations below are assumed finite. The query law and both compared endpoints are fixed. Population identities do not require independent observations; the finite-evidence selection result adds the explicit sampling assumptions in Appendix C.4.

C.1 Recursive decomposition and compression risk

Proposition C.1 (Exact recursive representation)**.** *For nested partitions with $Q_0 = I$, the components in equation 1 satisfy $h = \sum_{k<\ell} d^k + r^\ell$ at every depth ℓ. They are mutually orthogonal in $\mathcal{H}_Z$, and $Q_{k+1} d^k = 0$.*

Proof. For completeness, the fold is

$$(Q_k f)(z_i) = \frac{\sum_{z_j \in A_k(i)} \nu_j f(z_j)}{\nu(A_k(i))}, \qquad Q_k f = \mathbb{E}_\nu[f \mid \mathcal{G}_k]. \tag{22}$$

Conditional expectation onto $\mathcal{G}_k$ is the orthogonal projection onto fields constant on each cell of $\mathcal{P}_k$. Nestedness gives $Q_j Q_k = Q_k Q_j = Q_{\max(j,k)}$. Thus $Q_k - Q_{k+1}$ are pairwise orthogonal projections, also orthogonal to Q_ℓ for $k < \ell$. The telescoping identity

$$I = \sum_{k<\ell} (Q_k - Q_{k+1}) + Q_\ell$$

proves exact synthesis, while Pythagoras yields

$$\|h\|_\nu^2 = \sum_{k<\ell} \left\|d^k\right\|_\nu^2 + \left\|r^\ell\right\|_\nu^2. \tag{23}$$

Finally, $Q_{k+1}(Q_k - Q_{k+1})h = 0$. □

Proof of Theorem 3.1. Let $\eta = \eta_X(q)$, $\eta' = \eta_{X'}(q)$, and $\delta(q) = \eta' - \eta = B(q)e(z_{\pi(q)})$. Gradient Lipschitz continuity on the segment gives

$$f_q(\eta') - f_q(\eta) \le \nabla f_q(\eta)^\top \delta + \frac{L}{2} \|\delta\|^2.$$

Writing $p_i = \Pr(\pi(q) = i)$, the readout and coverage bounds imply

$$\mathbb{E}_q \|\delta(q)\|^2 \le b^2 \sum_i p_i \|e(z_i)\|^2 \le b^2 \gamma \|e\|_\nu^2. \tag{24}$$

Cauchy–Schwarz bounds the expected linear term in absolute value by

$$\left|\mathbb{E}_q \nabla f_q(\eta_X(q))^\top \delta(q)\right| \le \varepsilon_X \sqrt{\mathbb{E}_q \|\delta(q)\|^2} \le \varepsilon_X b \sqrt{\gamma}\, D.$$

Taking expectations proves the risk bound. If $e = -\sum_{k \in \mathcal{D}} d^k$ and other field contributions are unchanged, orthogonality gives the stated expression for D^2. The same argument bounds the absolute risk change by this upper bound.

Which distortion is controlled. The norm is taken on the actual synthesized field before the fixed readout. For overlapping records it is $\|\sum_i \Psi_{\zeta_i} \Delta c_i\|_\nu$, including cross terms. Masking can destroy orthogonality, and a general $B(q)$ need not preserve it; the proof does not sum separate output distortions. Since all $\nu_i > 0$, finite anchors ensure some γ exists, but a small reference weight on frequently queried anchors can make the bound weak. The analysis fixes the query law and therefore does not automatically bound control return when a changed policy alters future visitation.

C.2 True recall risk under log loss

For a sharper interpretation of generative recall, consider the conditional exponential family

$$p_\eta(y \mid q) = h_q(y) \exp\{\eta^\top T_q(y) - A_q(\eta)\}.$$

Assume common support and a twice differentiable log partition on a convex natural-parameter domain containing the compared segment. Let $\mu_*(q) = \mathbb{E}_*[T_q(Y) \mid q]$ and $g_X(q) = \nabla A_q(\eta_X(q)) - \mu_*(q)$. This conditional moment error is the gradient of true log-loss risk; it is not an empirical calibration score.

Proposition C.2 (Distribution change and true residual value)**.** *For $\delta = \eta_{X'} - \eta_X$, true expected log loss satisfies*

$$R(X') - R(X) = \mathbb{E}_q D_{\mathrm{KL}}(P_X \| P_{X'}) + \mathbb{E}_q[g_X^\top \delta], \tag{25}$$

where the KL divergence is conditional on the same query. If $0 \preceq \nabla^2 A_q \preceq LI$ along the segment and $\varepsilon_X^2 = \mathbb{E}_q \|g_X\|^2$, then

$$0 \le \mathbb{E}_q D_{\mathrm{KL}}(P_X \| P_{X'}) \le \frac{Lb^2\gamma}{2} D^2, \tag{26}$$

$$-\varepsilon_X b \sqrt{\gamma}\, D \le R(X') - R(X) \le \varepsilon_X b \sqrt{\gamma}\, D + \frac{Lb^2\gamma}{2} D^2. \tag{27}$$

Proof. For each query, write $\eta = \eta_X(q)$, $\eta' = \eta_{X'}(q)$. Taking the log likelihood ratio under the true outcome law gives

$$\begin{aligned}\mathbb{E}_*[-\log p_{\eta'}(Y \mid q) + \log p_\eta(Y \mid q) \mid q] &= A_q(\eta') - A_q(\eta) - \mu_*^\top \delta \\ &= D_{\mathrm{KL}}(p_\eta \| p_{\eta'}) + g_X^\top \delta.\end{aligned}$$

The equality follows from $\mathbb{E}_{p_\eta} T_q(Y) = \nabla A_q(\eta)$. Taylor's integral remainder yields

$$D_{\mathrm{KL}}(p_\eta \| p_{\eta'}) = \int_0^1 (1-t)\delta^\top \nabla^2 A_q(\eta + t\delta)\delta \, dt \le \frac{L}{2} \|\delta\|^2 .$$

Nonnegativity of KL, equation 24, and Cauchy–Schwarz give both bounds. □

When $g_X = 0$ almost surely, true risk change equals the expected KL and is nonnegative. This requires correct conditional sufficient-statistic means, not equality of the entire true and model distributions. Under the model's own law, it recovers the familiar residual-removal KL identity. In general the bias term can be negative: if $Y \sim \mathcal{N}(0,1)$, changing a prediction from $\mathcal{N}(1,1)$ to $\mathcal{N}(0,1)$ decreases true NLL by $1/2$, although the model-to-model KL is $1/2$. Here $g_X \delta = -1$. Thus distribution change alone does not determine whether forgetting helps; future outcomes assess the residual's actual value.

Readout conditions. A Gaussian mean with fixed variance and finite-category logits give standard smooth instances. For a calibrated readout, the theorem requires the complete map from the changed field to the scored parameter to remain fixed and linear, or a separately justified distortion bound for that map. Frozen model weights alone do not establish this property for changing addresses, clipped corrections, or decoded text inputs.

C.3 Scope expansion and transfer gain

Proof of Theorem 3.2. The field difference is

$$M^{\mathrm{jump}}_{m,k\to\ell} - M^{\mathrm{keep}}_{m,k} = (\chi_\ell - \chi_k) r^\ell_m. \tag{28}$$

Indeed, subtracting the kept contribution from equation 5 gives

$$\begin{aligned}M^{\mathrm{jump}}_{k\to\ell} - M^{\mathrm{keep}}_k &= \chi_k(r^k - r^\ell) + \chi_\ell r^\ell - \chi_k r^k \\ &= (\chi_\ell - \chi_k) r^\ell = \mathbf{1}_F r^\ell.\end{aligned}$$

The old cell and the region outside the new cell are unchanged. Fixed routing and readout therefore give identical predictions outside F. For $Z = \ell_{\mathrm{keep}} - \ell_{\mathrm{jump}}$, this implies $Z = \mathbf{1}_F Z$, so the law of total expectation gives the claimed gain identity when $p_F > 0$. For $p_F = 0$, $Z = 0$ almost surely and the gain is zero without conditioning on a null event.

For squared error, assume finite second moments and $\widehat{y}_X = \eta_X$, so $\widehat{y}_{\mathrm{jump}} = \widehat{y}_{\mathrm{keep}} + v$ on F. Then

$$\|Y - \widehat{y}_{\mathrm{keep}}\|^2 - \|Y - \widehat{y}_{\mathrm{keep}} - v\|^2 = 2(Y - \widehat{y}_{\mathrm{keep}})^\top v - \|v\|^2 .$$

Conditional expectation replaces $Y - \widehat{y}_{\mathrm{keep}}$ by $a(q)$, giving the squared-error expression for G. For positive p_F, the stated alignment condition is therefore both necessary and sufficient for positive gain.

Log-loss transfer. Under the exponential-family conditions above, the same frontier argument gives

$$G = p_F \mathbb{E}_q\left[-g_{\mathrm{keep}}^\top v - D_{\mathrm{KL}}(P_{\mathrm{keep}} \| P_{\mathrm{jump}}) \mid F\right], \tag{29}$$

where $v = B(q) r^\ell_m(q)$ is now a natural-parameter change. Consequently, for $p_F > 0$,

$$-\mathbb{E}[g_{\mathrm{keep}}^\top v \mid F] > \frac{L}{2}\mathbb{E}[\|v\|^2 \mid F]$$

is sufficient for beneficial reuse. Both loss-specific formulas evaluate alignment on the added region; positive graph edges alone do not imply that this condition holds for every ancestor.

C.4 FINITE EVIDENCE FOR CANDIDATE SCOPE SELECTION

The frontier identity isolates population transfer gain. The following fixed-block specialization quantifies how accurately subsequent evidence can select among candidate scopes. Condition on the formation history: the deployed endpoint, routing, readout, and $M \geq 1$ feasible candidate endpoints are fixed before validation. The candidate family and its validation state satisfy the declared peak cap. The block length $n \geq 1$ is also fixed in advance. The observations (q_i, Y_i) are independent draws from the common future query–outcome law, independent of formation data. Both endpoints are evaluated on each observation without updating them within the block.

For candidate ρ, define the paired gain

$$Z_{i,\rho} = \ell_{q_i}(\eta_X(q_i), Y_i) - \ell_{q_i}(\eta_{X^\rho}(q_i), Y_i), \qquad \widehat{G}_\rho = \frac{1}{n}\sum_{i=1}^{n} Z_{i,\rho}, \qquad G_\rho = R(X) - R(X^\rho).$$

Assume $|Z_{i,\rho}| \leq H$ almost surely for a fixed $H > 0$. Candidates may share observations and overlapping frontiers; no independence across candidates is required.

Proposition C.3 (Selection from subsequent evidence)**.** *For $0 < \delta < 1$, the following holds with conditional probability at least $1 - \delta$:*

$$\max_\rho |\widehat{G}_\rho - G_\rho| \leq \beta_n, \qquad \beta_n = H\sqrt{\frac{2\log(2M/\delta)}{n}}. \tag{30}$$

Choose $\widehat{\rho} \in \arg\max_\rho \widehat{G}_\rho$. Deploy this endpoint if $\widehat{G}_{\widehat{\rho}} > \beta_n$, and otherwise retain X. Its population gain satisfies equation 12 on the same event.

Proof. Hoeffding's inequality for variables in $[-H, H]$ gives

$$\Pr\big(|\widehat{G}_\rho - G_\rho| > \beta_n\big) \leq 2\exp\{-n\beta_n^2/(2H^2)\} = \delta/M.$$

A union bound gives the uniform deviation bound, conditional on the fixed formation history. On this event, let $\rho_* \in \arg\max_\rho G_\rho$. Empirical maximization gives

$$G_{\widehat{\rho}} \geq \widehat{G}_{\widehat{\rho}} - \beta_n \geq \widehat{G}_{\rho_*} - \beta_n \geq G_{\rho_*} - 2\beta_n.$$

If deployment occurs, the first bound also gives $G_{\widehat{\rho}} > 0$. Otherwise, every candidate has $G_\rho \leq \widehat{G}_\rho + \beta_n \leq 2\beta_n$, so retaining X with gain zero still satisfies the claimed deployment guarantee. □

The comparison is to the best member of the frozen, feasible candidate family. For a scope jump, gains vanish outside its frontier, so the estimate above targets the unconditional gain, including frontier probability. If only frontier observations are scored, their conditional mean must be combined with the frontier mass. Known resource penalties can be subtracted from both true and empirical gains without changing the uniform error bound, yielding the corresponding guarantee for net benefit. Estimated costs require their own error control.

This result uses a fixed sample size and bounded paired loss, conditions not supplied by an empirical standard-error penalty. Unbounded log loss, repeatedly inspected validation blocks, or adaptively changing endpoints require additional analysis. The experimental gates in Appendix B.4 retain their declared empirical scores; no statistical calibration or algorithm change is assumed here.

C.5 EXACT INTERNALIZATION AND RESOURCE BENEFIT

Proposition C.4 (Predictor-invariant internalization)**.** *Under the shared linear synthesis, compatibility, and allocation conditions of Section 2.3,*

$$F_{X^{\mathrm{agg}}} = F_X, \qquad C(X^{\mathrm{agg}}) = C(X) - (m-1)c_{\mathrm{mode}}(\zeta), \tag{31}$$

$$F_{X^{\mathrm{abs}}} = F_X, \qquad C(X^{\mathrm{abs}}) = C(X) - c_{\mathrm{mode}}(\zeta). \tag{32}$$

These operations preserve expected task and recall loss under any fixed future outcome law.

Proof. Compatible records share Ψ_ζ, so

$$\sum_{j=1}^{m} \Psi_\zeta c_j = \Psi_\zeta \sum_{j=1}^{m} c_j, \qquad \Psi_\zeta w_\zeta + \Psi_\zeta c = \Psi_\zeta (w_\zeta + c).$$

These identities preserve the combined field at every anchor and hence the full predictor under the fixed interface. Equal predictors have equal expected loss under any outcome law, irrespective of model bias. Aggregation releases $m - 1$ independent record allocations; absorption releases one while its world slot remains preallocated. With other allocations unchanged, these give the stated savings. □

The equalities assume exact evaluation of the shared linear maps and representable sums. Representability alone does not guarantee bitwise equality after floating-point reassociation. A calibrated decoder additionally requires unchanged parameters and full decoded inputs.

Proof of Proposition 3.3. The scope expansion has $R(X^+) - R(X) = -G$. The compression bound is evaluated at its actual input X^+, while exact internalization gives $R(\bar{X}) = R(X')$. Hence

$$R(\bar{X}) - R(X) = \big[R(X') - R(X^+)\big] + \big[R(X^+) - R(X)\big] \le \mathcal{B}_{X^+}(D) - G.$$

Adding the actual storage and computation differences proves the joint bound. With no scope expansion, $X^+ = X$ and $G = 0$, recovering the bound for lossy compression followed by exact internalization. With $D = 0$, the risk change equals $-G$: internalization preserves the entire transfer gain.

On the uniform event in Proposition C.3, a selected candidate's true gain can be replaced by its lower bound $\widehat{G}_{\hat{\rho}} - \beta_n$. Thus the same joint inequality links validation uncertainty, compression distortion, and resource savings. The compression term still depends on the stated smoothness, coverage, and predictor-bias quantities; it is a sufficient theoretical bound, not the empirical acceptance score.

If one-time rewrite work is charged to the comparison, its priced cost is added to the right-hand side. The storage saving is net of all resident allocations, and every executed step must satisfy the complete peak-capacity check. The bound therefore separates the cost of changing predictive content from the benefit of storing its unchanged influence more compactly.

D Experimental Setup

D.1 Environments and tasks

Symbolic Alchemy. Alchemy tests hidden-rule discovery through applying potions to stones and collecting rewards (Wang et al., 2021). A latent chemistry controls stone transformations and persists across trials, so earlier interventions can inform later decisions. We use the official symbolic interface with randomized perceptual mappings and bottlenecks. The main setting has five trials, three stones, and eight potions per trial; methods observe attributes and action outcomes, while chemistry remains hidden. A fixed intervention stream evaluates prediction and recall, and separate closed-loop episodes evaluate return. Longer runs use 10 and 20 trials with the same memory configuration and cap, retaining the reference controller for the horizon diagnostic.

Controlled synthetic tasks.

- **Predictive memory under a state budget.** We generate 32-dimensional queries and targets on a depth-three binary tree by summing Gaussian components along each leaf's path. Shared ancestors create transferable structure; leaf-specific corrections add stable local detail. Observations include noise and rare task-relevant perturbations with varying predictability. A fixed linear map converts targets into four-action scores, with larger utility weights on rare events. The 25 conditions vary shared structure, local detail, noise, prior bias, and rare-event properties. All methods receive the same data under 8, 16, and 32 KiB caps.

  Memory is formed from 128 observations drawn from one leaf of each sibling pair. Separate evidence and evaluation data test prediction on the other leaves and recall on the formation leaves. This distinguishes transfer to related contexts from retention of previously encountered information. Final evaluation data are held out from memory formation and evidence selection.

- **Readout across abstraction levels.** The multiscale task tests recovery of fine, intermediate, and coarse information when the generating hierarchy is known. Its complementary scales provide a controlled check of recursive representation and readout selection. It uses 12-dimensional codes arranged into three coarse groups, four intermediate groups per coarse group, and four leaves per intermediate group. Two leaves per intermediate group supply formation data. Opposing event residuals and centered local details cancel at their parent levels, giving known targets at each resolution. Methods receive perturbed query cues without hierarchy or event labels. At an 8 KiB cap, we compare adaptive readout with fixed scopes, a randomized hierarchy, and world-only readout.

LongMemEval-V2. LongMemEval-V2 tests memory of agent interactions with web and enterprise applications (Wu et al., 2026). Memory processes histories of instructions, interface states, actions, and outcomes, then supplies evidence for questions about past interactions. We evaluate all 451 official questions using a shared experience encoder, a 128 KiB cap for bounded methods, and a shared frozen QA reader, including the supplied images for 29 questions. Evaluation covers answer-content retention and downstream QA.

Two-Episode Procgen. Procgen contains 16 procedurally generated games with 64×64 RGB observations (Cobbe et al., 2020). Our protocol tests reuse of experience on a second attempt at the same level. The first attempt is shared across methods and supplies memory. Prediction is evaluated on a fixed second-attempt trajectory; control is evaluated in a separate closed-loop attempt. Methods share a frozen controller and a 64 KiB cap, with disjoint training, development, and evaluation levels. The evaluation tests adaptation within a previously unseen level by reusing its first attempt.

D.2 Comparison protocol

We compare memory organization, updating, and retrieval under a shared task model and state cap. We checked each baseline's core mechanisms and task performance to reproduce its capabilities as faithfully as possible within this interface. The implementations retain the method-specific operations in Table 2, with representations matched to each task. Numerical experience is supplied as predictive coordinates, so summary operations aggregate predictive values. For the bounded text comparison, extractive consolidation keeps summaries grounded in observed content for the shared reader. Representations, routing, empirical evidence scores, and readouts are specified in Appendix E.

Symbolic Alchemy. RPC and all baseline implementations use the same ten evaluation worlds, a 64 KiB memory-state cap, and the same frozen categorical prior. Prediction and recall use a shared, fixed intervention stream with identical observations, actions, outcomes, and recall queries. Closed-loop return is evaluated separately in the same worlds with a common public-value controller, action-scoring rule, and action-validity rules. Scores and paired differences are aggregated over these ten worlds.

The horizon diagnostic retains its fixed reference controller and measures shorter return prefixes within longer runs. Recall queries vary by horizon, so these are not fixed-test-set learning curves. The internalization case compares snapshots immediately before and after absorption, then probes an unseen stone–potion pairing in the same chemistry using frozen post-absorption memory. Component removal occurs only at readout. The native engine renders logged states with a presentation camera; these are state reconstructions rather than policy-camera frames.

Controlled synthetic. At each capacity, methods receive the same formation observations, evidence, and evaluation queries. Final evaluation data remain held out from memory formation and evidence selection.

LongMemEval-V2. Bounded methods share query-blind trajectory encodings capped at 2 KiB each, a 128 KiB state cap, and a readout of up to 4,096 tokens. Vector-based baselines use 128-dimensional frozen text embeddings, and Titans connects neural recall to bounded text retrieval. Method-private text, vectors, indices, and feedback state count toward the cap. Shared encoding and embedding caches are consumed at insertion and are not a read-time archive. Comparisons include budget-matched BM25 retrieval. Official AgentRunbook-C retrieves from complete trajectories and screenshots with up to 200K returned tokens (Wu et al., 2026); its scores therefore do not constitute budget-matched baselines for this bounded-memory evaluation.

Table 2: Baseline memory operations retained under the shared task interface, with settings and task-specific implementations.

Core operations and settings	Task-specific implementation
RGMem (Tian et al., 2025)	
Thresholded relation updates, common/specific synthesis, upward hierarchy updates, and multiscale retrieval. Inference/synthesis thresholds: 3/6; retrieval: top-3 numerical, top-6 text.	Numerical relation means integrate evidence; coordinate medians and a retained exemplar represent common and specific content. Preallocated concept and root nodes bound the hierarchy.
H-MEM (Yu et al., 2026)	
Temporal–semantic consolidation and hierarchical retrieval; text also retains entity-graph traversal and scope-aware query planning. Consolidation thresholds: 0.8, 0.7, 0.6.	Numerical tasks use the temporal–semantic branch, with observation order as time and count-weighted means as summaries. Entity-graph traversal and query planning operate on the text interface.
MemTree (Rezazadeh et al., 2025)	
Similarity-based insertion, leaf promotion, recursive ancestor aggregation, depth-dependent routing, and retrieval across levels. Base threshold: 0.4 (0.82 in Alchemy); depth rate: 0.5.	Numerical nodes aggregate descendant predictions using sample-weighted means; text nodes consolidate retained content at successive tree levels.
DeMem (Zou et al., 2026)	
Context routing, decision-conflict checks, witness-based slot splitting, reassignment, reconstruction, and split suppression at capacity. Eight slots; 64 in Alchemy.	Empirical conflict checks use numerical outcomes and action targets or text outcome/action-field probes. In controlled and Procgen comparisons, DeMem-cons. adds a standard-error penalty to the mean-gain gate used by DeMem-prac.
Titans (Behrouz et al., 2026)	
Associative gradient updates, learned initialization, and input-dependent learning rate, momentum, and forgetting.	The associative memory updates online alongside the shared frozen task model. In text tasks, neural recall ranks retained text for the shared reader.

Allocation, pruning, and saturation enforce the common cap. Method-private state and declared workspace count toward the cap.

Two-Episode Procgen. All methods receive the same first-attempt trajectory and share a frozen controller and a 64 KiB memory-state cap. Prediction uses a common, fixed second-attempt trajectory; control is evaluated in separate closed-loop attempts matched by game, level, and seed. This keeps the memory-formation stream and prediction targets common across methods while allowing their actions to affect subsequent control outcomes.

D.3 Evaluation metrics

Controlled synthetic. Future MSE evaluates new queries from non-formation leaves; seen-recall MSE evaluates new queries on formation leaves. Action NLL measures target-action log loss, decision accuracy counts correct choices, weighted utility emphasizes consequential choices, and rare-fact accuracy isolates rare events. In the multiscale task, NMSE divides each scale's MSE by its no-memory MSE. Macro NMSE averages the three scales equally; the plotted improvement is $-\log_{10}(\text{macro NMSE})$.

Symbolic Alchemy. Transition NLL scores the next hidden-rule outcome, future-recall NLL scores later reconstruction of revealed chemistry, and episode return measures closed-loop choices.

LongMemEval-V2. Answer-token reconstruction is the fraction of normalized gold answer terms in the memory readout; exact reconstruction requires the complete normalized answer to occur there. Both measure content retention on 226 phrase-answer questions. QA counts correct answers out of 451, using the benchmark's deterministic rules and semantic-judging prompts with a shared judge. Reconstruction intervals use question-level bootstrap over the 226 questions, with constructed memories held fixed. Questions can share histories; the intervals do not represent variation across independent memory-training runs.

Two-Episode Procgen. Next-latent MSE scores successor-state prediction; action NLL and reward MSE score the target action and immediate reward. Attempt-two return and success measure closed-loop performance, and adaptation return is the change from the first to the second attempt.

Online resources. Final state is the resident state after a stream. Peak state is the largest simultaneous allocation of live memory and declared workspace; its mean and maximum are reported across evaluation units. Shadow evaluations count additional online predictions. Query and update latency measure memory operations on the same implementation and stream, excluding shared-model inference.

E Models and Parameter Settings

Table 3 lists shared models and task-specific RPC settings; the readout and gate descriptions below specify the corresponding adaptations. Within each environment, we tune the hyperparameters of RPC and all baselines independently on the same development splits, using method-specific search spaces, equal search budgets, and the same predefined selection criteria. All selected hyperparameter configurations are fixed before evaluation, and test outcomes are not used for hyperparameter selection. Shared task models remain frozen. Titans uses a linear associative memory with projected key width 32 (8 in Alchemy). Its initial memory and update gates use separate formation streams, eight training epochs, and Adam at 10^{-3}. The Procgen controller uses approximately five million training frames, with checkpoint selection on development levels. LongMemEval-V2 uses a deterministic reader with at most 256 output tokens. Online memory updates use interaction histories, without reference answers.

Readout calibration. For the numerical readout, correction scales and temperatures can be fitted separately when base and memory-assisted predictions agree or conflict, followed by the retained-detail strength. Groups are determined from pre-outcome predictions; fitting uses past evidence and is frozen before evaluation. Cue addressing may use estimated predictive value from retained codes.

In the controlled suite, admitted codes mix their initial value with the mean of their assigned formation observations at weight 0.75. Recall reconstructs current world codes plus retained detail, including absorbed coefficients. All methods share task-calibration evidence, search grids, and utility-based action selection, with readout state and workspace counted toward the cap. The selected action receives the largest logit through a swap with the original winner. The multiscale synthetic task uses graph degree 8, two diffusion steps, four world slots, and a local-similarity threshold of 0.94.

Table 3: Shared models and final RPC configurations.

Shared model	Memory structure	Readout and gate
Controlled synthetic 8/16/32 KiB		
32-dimensional predictive state Four actions	Degree 4; 2 diffusion steps Depths $\{1, 2, 4, 8, 16\}$ 16 world slots Formation averaging 0.75	Margins 0.02 Resource 0.003 Release 0.40 Criticality 0.05
Symbolic Alchemy 64 KiB		
Frozen categorical prior Public-value controller	Four-transition blocks $k = 12$, $d = 8$ Scales $\{0.25, 1, 4\}$	$u = 48$, $\beta = 4$ Temperatures 1/1
LongMemEval-V2 128 KiB		
Shared frozen QA reader Shared baseline embeddings	Shared token counts and retained event text	Shared readout limit Future-evidence gate
Two-Episode Procgen 64 KiB		
IMPALA + GRU-256 Latent dimension 128	Depths $\{1, 4, 16\}$ Split-local GAE	Diagonal readout Ridge 1; clip 0.6 Gate 0.5

In the main Alchemy comparison and the sharing ablation, transition and future-recall NLL use the default temperature $T = 1$ for every method. In Alchemy closed-loop comparisons, every method uses the same public-value controller: it scores valid actions from current stone values and predicted

value increments, then selects the highest score. The prediction model and memory update rules remain fixed.

LongMemEval-V2 text adaptation. The first four admitted summaries define lexical anchors. Later summaries route by normalized token overlap. A frozen, symmetrized one-nearest-neighbor graph provides zero-, one-, and two-hop neighborhoods for each anchor. Shared state reserves 18% of the cap for bounded anchor-wise token counts; cue-addressable records retain event text. Readout combines supported shared phrases with selected event segments and supplies at most 4,096 tokens to the frozen QA reader. Increasing the active hop depth expands the text-sharing scope. Consolidation updates shared counts while retaining event payloads. This adaptation evaluates empirical text sharing, scope selection, content retention, and QA. The global projection and code-linear guarantees require the formal interface; exact coefficient absorption is examined in the controlled suite and Alchemy.

Gate settings.

- **Synthetic** prediction/task and recall checks use disjoint selection and acceptance evidence.
- **Alchemy** forms candidates from four-transition blocks. Strictly later affected transitions provide paired multiclass Brier gains on the observed outcome classes. The primary, task, and recall ledgers share these revealed outcomes. Admission and scope expansion require all three running means to be positive, with 0.03 added to the task and recall gains as non-inferiority tolerances. Each mean needs at least one eligible future observation; pending candidates expire after 16 comparison rounds.
- **Procgen** uses the first 60% of attempt one for formation and divides the remainder equally into selection and acceptance segments. The prediction score combines next-latent MSE, reward MSE, terminal cross-entropy, and advantage-weighted policy loss with weights $1, 0.25, 0.1, 0.1$. Selection maximizes mean gain; acceptance requires mean gain minus one standard error to be positive. The action check requires positive advantage-weighted log-probability gain minus 0.5 standard errors on the upper half of nonzero centered logit-change magnitudes, with at least four such transitions. Prediction needs at least four evidence transitions; advantages are computed within each segment.
- **LongMemEval-V2** proposes broader hop scopes every four admitted summaries and compares them on the next two admitted summaries, scoring each before adding its tokens to shared state. For a future summary, the two scores are the fractions of all summary terms and goal-prefix terms covered by the candidate scope. Relative to the active depth, both mean gains must be nonnegative and their sum must exceed 0.01 times the depth increase. The passing depth with the largest net gain is deployed. These checks use observed history text; benchmark answers and reader outputs are not gate evidence.

F Controlled synthetic results

Figure 7 compares all methods at 8, 16, and 32 KiB across prediction, decision, and recall metrics, allowing comparisons within each capacity.

G Text and visual environment results

Figure 8 combines the reported reconstruction, QA, prediction, and control endpoints for LongMemEval-V2 and Procgen. Alchemy's prediction, recall, and decision results are shown in Figure 2.

H Alchemy memory-update records

Table 4 summarizes saved updates for all ten Alchemy worlds, separately for the shared intervention and closed-loop streams. Means are per world, including worlds without absorption; events are not independent replicates. Scope expansions are accepted broader-scope updates, and local records are counted at stream end.

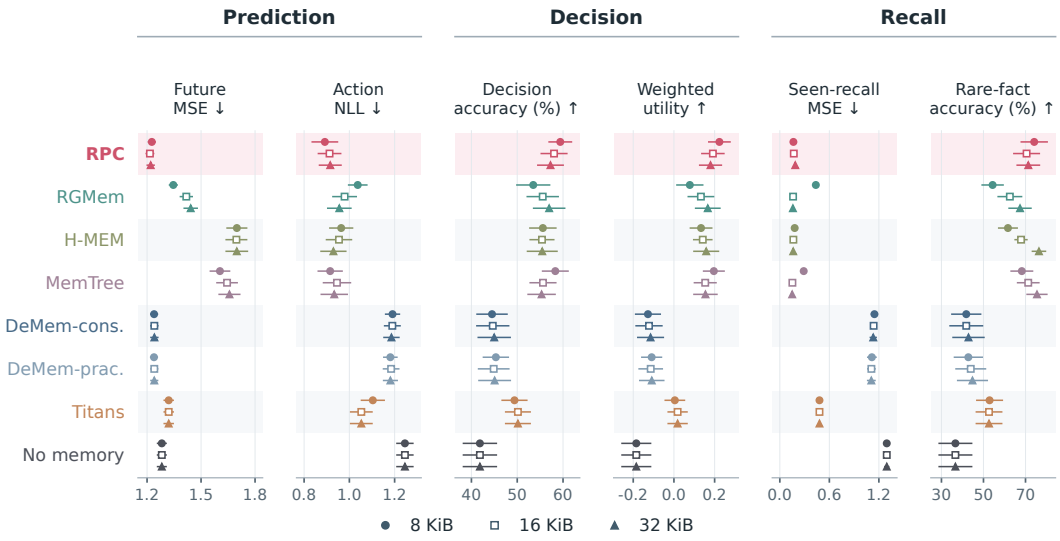

Figure 7: Controlled synthetic results across memory capacities. Colors denote methods; markers denote 8, 16, and 32 KiB. Points and horizontal intervals show means and 95% Student-t intervals over ten seeds, each averaged across 25 conditions. Panels use metric-specific scales; accuracies are percentages.

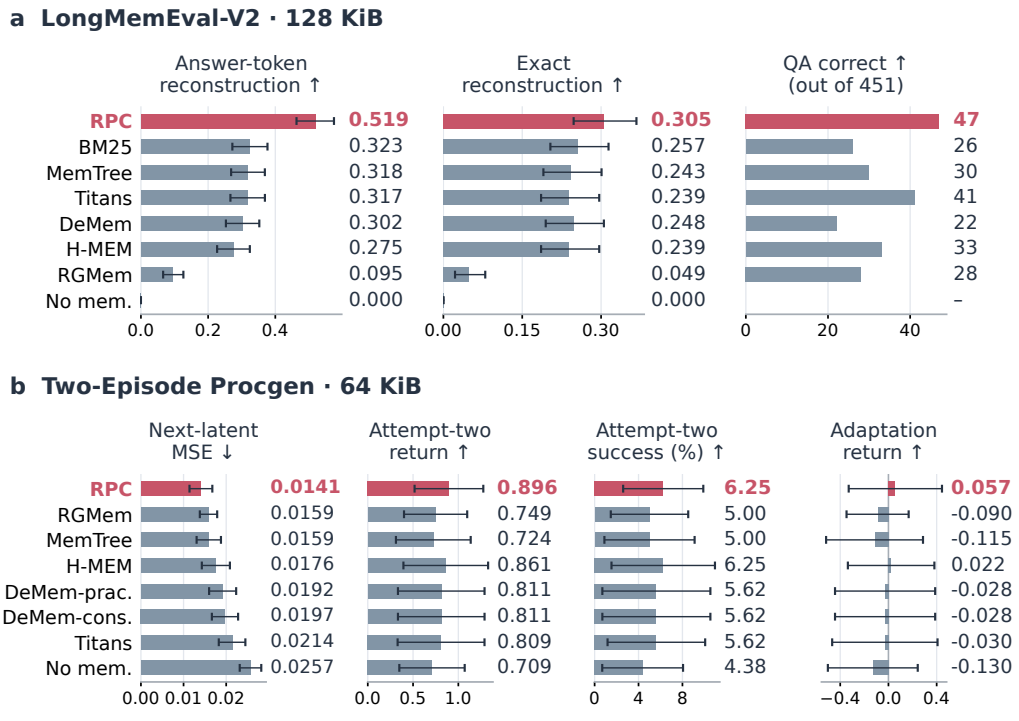

Figure 8: Text and visual results under matched state caps. **a,** LongMemEval-V2 at 128 KiB: reconstruction intervals use question bootstrap over 226 phrase-answer questions; QA reports correct answers out of 451 (not evaluated for no memory). **b,** Procgen at 64 KiB: means and 95% Student-t intervals over ten seeds. Methods follow the first metric's ranking within each environment; all panels use their own scales.

All six compatible absorptions have zero maximum absolute change in the complete correction across routing anchors, checked immediately before and after the move into world state. Figure 4

Table 4: Alchemy update records across ten worlds per stream. Means are per world; absorption totals and the maximum discrepancy summarize all realized events.

Quantity	Shared intervention	Closed-loop control
Validated scope expansions (mean)	7.9	5.3
Retained local records (mean)	21.9	16.0
Compatible absorptions (total)	3	3
Worlds with absorption	3/10	3/10
Maximum correction discrepancy	0	0

complements these preservation checks with a later reuse case: with post-absorption memory frozen, world state retains the released record's predictive contribution and improves prediction on an unseen stone–potion pairing in the same chemistry.

I Resource use

Table 5 reports online state and computation. State totals count packed method state and declared workspace, excluding shared models and host-runtime allocations. Memory-operation latencies exclude model inference.

Table 5: Measured online state and computation for the final configurations.

(a) Online state

Environment	Cap	Final (B)	Mean peak (B)	Max peak (B)
Controlled synthetic	8 KiB	2,884.8	5,966.5	6,891
Controlled synthetic	16 KiB	7,683.8	11,437.7	15,479
Controlled synthetic	32 KiB	17,126.3	22,177.0	31,952
Symbolic Alchemy	64 KiB	27,461	43,305.6	44,186
LongMemEval-V2	128 KiB	80,146.2	80,402.2	81,920
Two-Episode Procgen	64 KiB	35,423.1	36,226	63,226

(b) Online computation

Environment	Shadows or calls	Query	Update
Controlled synthetic (8 KiB)	12.0 gate shadows	32.5/77.4/44.7 μs	86.5 μs
Controlled synthetic (16 KiB)	15.8 gate shadows	44.3/78.7/59.7 μs	151.4 μs
Controlled synthetic (32 KiB)	15.7 gate shadows	52.2/86.0/71.7 μs	266.3 μs
Symbolic Alchemy	75.7 shadows	1.63 ms	0.563 ms
LongMemEval-V2	No extra memory calls	12.00 ms/q	0.031 ms/insert
Two-Episode Procgen	1 call/step	0.488 ms	2.20 ms

LongMemEval-V2 update time averages all insertion attempts. Synthetic query times are prediction/task/recall; update time includes formation and evidence fitting, amortized over 128 observations. Gate shadows count scope and release comparisons. Synthetic formation rereads an externally supplied input block, whose storage is outside the reported state cap; temporary workspace is 448 B at 8 KiB and 704 B at 16/32 KiB. Scratch reuse, graph release, and in-place residual-to-slot transfer reduce peak state without changing predictions or admission rules.

J Ablation and mechanism analysis

J.1 Ablation protocol

Each ablation uses the same data and state cap as full RPC. Controlled component ablations use 8 KiB and the main comparison's task-calibration procedure, refitted on the same evidence. The absorption comparison uses 32 KiB, 512 formation observations, and a shared NLL-calibrated readout; the other controlled ablations use 128 formation observations. Procgen world-only predictions are paired with the original full-memory predictions.

- **No hierarchical sharing** removes candidate grouping, cross-node sharing, and ancestor expansion while retaining the base prior and local memory. In the controlled suite, node admission, evidence

Table 6: Component ablations under matched state caps. Δ is full RPC minus the ablation, computed before rounding; brackets give paired 95% confidence intervals

Ablation	Diagnostic	Full RPC	Ablated	Δ [95% interval]
Controlled synthetic, 8 KiB; absorption, 32 KiB				
Fixed depth	Future MSE ↓	**1.226**	1.235	-0.009 $[-0.012, -0.006]$
Ungated	Future MSE ↓	**1.226**	1.429	-0.202 $[-0.237, -0.168]$
	Seen-recall MSE ↓	0.164	**0.162**	$+0.003$ $[+0.002, +0.004]$
World only	Seen-recall MSE ↓	**0.164**	1.095	-0.930 $[-0.998, -0.862]$
No absorption	Final state (B) ↓	**17,680**	21,253	-3572.4 $[-4057.5, -3087.3]$
No hierarchical sharing	Future MSE ↓	**1.226**	1.282	-0.056 $[-0.068, -0.043]$
	Action NLL ↓	**0.892**	0.984	-0.093 $[-0.130, -0.056]$
	Decision accuracy ↑	**0.594**	0.566	$+0.028$ $[+0.013, +0.043]$
	Seen-recall MSE ↓	**0.164**	0.165	-0.0007 $[-0.0011, -0.0002]$
	Rare-fact accuracy ↑	**0.741**	0.709	$+0.033$ $[-0.004, +0.069]$
	Weighted utility ↑	**0.223**	0.166	$+0.057$ $[+0.032, +0.083]$
Symbolic Alchemy, 64 KiB				
No hierarchical sharing	Future-recall NLL ↓	**0.951**	1.548	-0.597 $[-0.678, -0.516]$
	Transition NLL ↓	**1.615**	1.759	-0.144 $[-0.177, -0.111]$
LongMemEval-V2, 128 KiB				
No hierarchical sharing	Answer-token reconstruction ↑	**0.519**	0.318	$+0.200$ $[0.155, 0.247]$
World only	Answer-token reconstruction ↑	**0.519**	0.474	$+0.044$ $[0.024, 0.068]$
Two-Episode Procgen, 64 KiB				
World only	Next-latent MSE ↓	**0.0141**	0.0200	-0.0060 $[-0.0079, -0.0041]$

Synthetic component diagnostics use the shared task-calibration procedure specified in Appendix J.1. The absorption pair retains its reference layout and uses 512 formation observations; other synthetic pairs use 128. Intervals use paired Student-t estimates for synthetic, Alchemy, and Procgen results and question bootstrap for LongMemEval-V2.

thresholds, and applicable local checks are unchanged. Alchemy uses singleton routing with the same unit-temperature scoring and local evidence checks. In LongMemEval-V2, admitted records, summaries, and local readout settings are unchanged.

- **Fixed depth** uses scope size four in the synthetic suite.
- **Ungated** bypasses memory-acceptance checks while retaining candidate selection, capacity limits, numerical bounds, exact-rewrite checks, and the shared task-calibration accuracy constraints.
- **No absorption** disables the internalization path: learned world-state writes and subsequent exact absorption. The prior, grouping, and gates remain; shared predictions are reconstructed from explicit records.
- **World only** is fitted from the start in the synthetic suite, removing explicit residuals and their absorption, release, and criticality addressing. The Procgen and LongMemEval-V2 component comparisons instead remove components only at readout from the same fitted memory. In LongMemEval-V2, removing residual text also changes the readout length.

J.2 RESULTS

Table 6 presents paired tests of scope selection, evidence gating, sharing, residual detail, and internalization. Removing hierarchical sharing changes grouping, cross-node sharing, and ancestor expansion together; it measures their joint contribution. The text world-only comparison also changes readout length.

K LIMITATIONS AND BROADER IMPACT

Limitations. This paper studies memory under pre-specified memory budgets. It does not optimize how these budgets should adapt to task importance or resource availability, or how resources should be divided among retaining memories, performing computation, and reacquiring information. The study also does not jointly optimize memory management and active information gathering. Although memory informs action selection in the closed-loop experiments, the controller is not trained to seek evidence specifically to improve memory. A systematic investigation of adaptive resource allocation and active information gathering to improve memory is beyond the scope of the present study. We leave these questions for future work toward jointly optimizing memory management, resource allocation, and information gathering.

Broader impact. This work examines how agents can learn from accumulated experience under finite memory budgets. RPC internalizes reusable knowledge while retaining important event details, reducing the need to preserve every interaction as a separate record. This could support personal assistants that adapt through repeated use, scientific agents that reuse insights across investigations, and embodied systems that learn within practical memory budgets. Making predictive contributions explicit also helps examine what an agent retains, where that knowledge is reused, and how it affects later predictions. For applications involving personal data, deletion mechanisms must address both explicit records and their retained influence on consolidated knowledge.