CARE-MEM: DECISION-LEVEL COUNTERFACTUAL CREDIT FOR SCHEMA-FREE MEMORY REVISION

Anonymous authors
Paper under double-blind review

ABSTRACT

Agents with long-term memory must update outdated information while preserving useful memories. However, a single reward for an entire memory update can reinforce unnecessary edits alongside correct revisions. This paper introduces CARE-Mem, which learns how to revise memory using decision-level counterfactual credit. Given supplied or retrieved memories, CARE-Mem predicts node actions and propagation gates without predefined domain slots or an ontology. During training, it executes counterfactual proposals from the same memory snapshot and assigns the resulting utility differences to the corresponding decision spans in the sampled proposal. At inference, CARE-Mem executes a single proposal without counterfactual evaluation. This study also introduces CAREBench-240, a human-annotated benchmark for node actions, propagation gates, and replacement text. CARE-Mem achieves higher action and edge prediction scores than graph-memory baselines mapped to the same interface. In post hoc exploratory comparisons, CARE-Mem achieves strict replacement exact-match accuracy of 48.68%, versus 39.57% for the parent policy trained by supervised fine-tuning (SFT) and 40.77% for the global-plus-local control. On MemoryAgentBench FactConsolidation, CARE-Mem averages 44.00% accuracy across single- and multi-hop tasks, exceeding the SFT parent and global-plus-local control by 11.00 and 9.17 percentage points, respectively, without further parameter updates.

1 INTRODUCTION

Suppose an agent remembers that a user lives in New York, commutes to work from New York, and likes Italian cuisine. A new observation states that the user has moved to Paris. The stored residence information needs updating, the commuting information may need reconsideration, and the cuisine preference should remain unchanged. Prior residence in New York should remain accessible in the version history. This update involves two coupled decisions: which revision action to apply to each memory item and which dependency edges should permit propagation. Agents with long-term memory must make both decisions to keep useful information up to date (Wu et al., 2025; Hu et al., 2026).

Existing memory systems offer complementary tools. Structured consolidation supports state adjudication (Chao et al., 2026), while graph memories link and update notes (Xu et al., 2025). Memory-specific optimization provides a dedicated learning signal for memory writing alongside trajectory-level rewards (Yan et al., 2026). A further question remains: how much does each node action or propagation gate contribute to the utility of a structured update? A single proposal-level advantage gives both the same training signal. A correct update of the residence to Paris can therefore receive positive credit alongside an unnecessary change to the cuisine preference.

The central idea is to evaluate counterfactual alternatives at the level of executable decisions. **CARE-Mem** represents an update as node actions and propagation gates within a relation-typed candidate neighborhood. Its controls change a node action, flip an edge gate, or jointly change a small connected set of decisions, starting from the same pre-write snapshot. Utility combines current-state, retention, and history probe scores with an edit-cost penalty. Differences in utility between the sampled proposal and its controls provide advantages for the corresponding action or gate token spans. At inference, the executor applies one version-preserving proposal generated by the learned policy (Figure 1).

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.

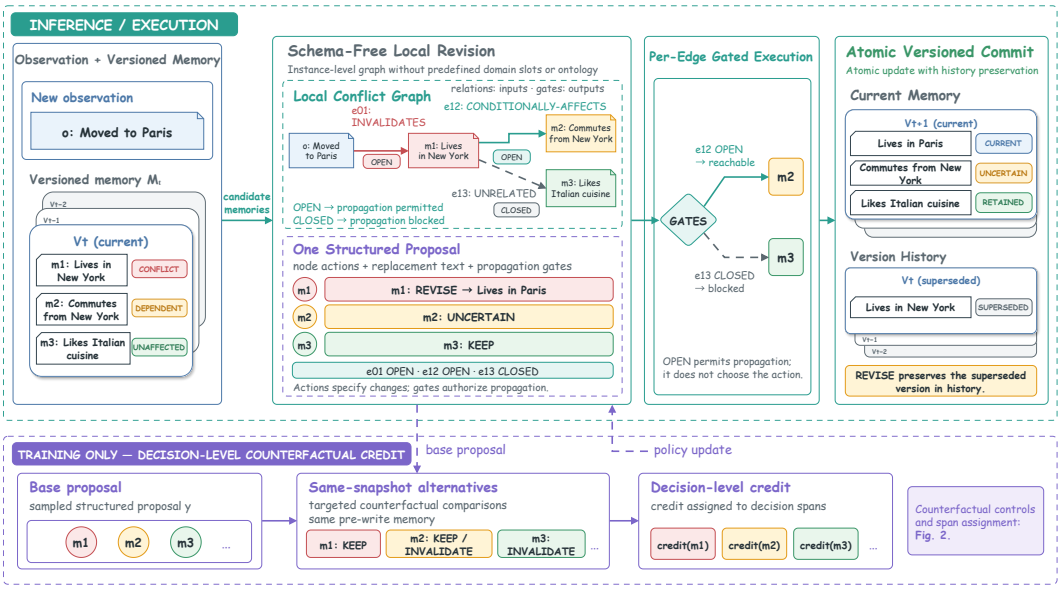


Figure 1: CARE-Mem overview. Relation labels are inputs. The policy outputs one structured proposal containing node actions, replacement text, and propagation gates. Gated execution preserves history when committing updates. The lower strip summarizes training-only counterfactual credit, detailed in Figure 2.

This study introduces **CAREBench-240** (CAREBench for short): 240 held-out cases with candidate neighborhoods and reference actions, gates, replacement text, and preservation-probe answers. MemoryAgentBench FactConsolidation evaluates models using the learned adapters on downstream single- and multi-hop questions.

Evaluation covers structural prediction, replacement generation, and downstream fact consolidation. CARE-Mem improves Action Macro-F1 and Edge F1 over CUPMem, whose native operations are mapped to the common action-and-gate interface, by 37.8 and 22.3 points on a 0–100 scale (Table 1). In post hoc exploratory comparisons, strict replacement exact-match accuracy is 48.7%, versus 39.6% for the SFT parent and 40.8% for the global-plus-local control (Table 3). On MemoryAgentBench FactConsolidation, average accuracy is $44.00 \pm 0.76\%$, exceeding these matched controls by 11.00 and 9.17 percentage points, respectively, without further parameter updates (Table 2).

This paper makes three contributions:

- Memory revision within candidate neighborhoods is formulated as coupled node actions and propagation gates, without predefined domain slots or an ontology.
- Counterfactual controls for node actions, edge gates, and small connected decision sets assign same-snapshot utility differences to decision-specific output spans during training.
- The human-annotated CAREBench-240 benchmark is introduced, and CARE-Mem is evaluated on structured revision and MemoryAgentBench FactConsolidation, with matched controls and ablation configurations.

2 Related Work

2.1 Dynamic Memory Revision

STALE studies implicit memory invalidation and introduces CUPMem, a prototype that uses a predefined hierarchical schema for state consolidation and propagation (Chao et al., 2026). PAMU tracks drift in predefined preference dimensions (Sun et al., 2026). CARE-Mem instead predicts actions and propagation for natural-language candidate memories without predefined domain slots.

2.2 GRAPH MEMORY AND ORGANIZATION

A recent survey organizes graph memory around extraction, storage, retrieval, and evolution (Yang et al., 2026). A-MEM links notes and updates them over time (Xu et al., 2025); GAM hierarchically consolidates event and topic graphs (Wu et al., 2026); and MRAgent actively reconstructs graph retrieval paths (Ji et al., 2026). MemGAS associates session, turn, summary, and keyword memories and uses entropy-guided routing with Personalized PageRank for query-time selection (Xu et al., 2026b). MemBuilder learns from downstream, session-level utility (Shen et al., 2026); MemCoE separates memory organization from memory writing (Xu et al., 2026a); and UMEM estimates marginal utility over semantic neighborhoods (Ye et al., 2026). These works establish that memory management can be learned, but their optimization operates at the level of memory candidates, individual dimensions, or complete updates. Spatial graph rectification uses version control and edge impact for low-cost repair (Zhang et al., 2026); CARE-Mem instead assigns same-snapshot credit to node actions and propagation gates.

2.3 DECISION-LEVEL CREDIT ASSIGNMENT AND EVALUATION

Source-attribution methods apply Shapley values and leave-one-out controls to retrieved documents (Nematov et al., 2025). TreeMem attributes rollout returns to memory-pipeline modules (Mao et al., 2026). MIRA derives a decaying utility signal from a memory graph built from large language model (LLM) suggestions and agent rollouts to shape advantage estimates in sparse-reward reinforcement learning (RL) (Nourzad & Joe-Wong, 2026), whereas HiMPO uses localized post hoc advantages for memory tokens to reduce blame leakage from tools or reasoning (Yan et al., 2026). CARE-Mem shares the counterfactual intuition but intervenes on executable node actions and propagation edges, and adds preservation-oriented probes to reveal the failure mode of excessive revision.

Memory benchmarks cover complementary capabilities. LongMemEval separates extraction, multi-session and temporal reasoning, knowledge updating, and abstention (Wu et al., 2025). BEAM combines conversations at a scale of 10 million tokens with 2,000 validated questions, while LIGHT combines episodic, working, and scratchpad memory (Tavakoli et al., 2026). AMemGym grounds simulated interactions in structured state evolution for on-policy diagnosis of memory writing, reading, and utilization (Cheng et al., 2026). MemoryAgentBench evaluates retrieval, test-time learning, long-range understanding, and selective forgetting; its FactConsolidation task evaluates single- and multi-hop questions over facts with multiple versions (Hu et al., 2026). These benchmarks measure end-task utility but do not jointly annotate state actions, propagation edges, minimal replacements, and preservation. CAREBench-240 provides this mechanism-level view; MemoryAgentBench tests the resulting adapters on external fact consolidation without further parameter updates.

3 PROBLEM FORMULATION

At time t, an agent maintains versioned memory $\mathcal{M}_t = (\mathcal{V}_t, \mathcal{E}_t)$. Here, $\mathcal{V}_t$ contains memory versions and $\mathcal{E}_t$ records their existing relations, including supersession links. Each node v_i stores natural-language text, provenance, event and write timestamps, and version status. No predefined domain slot is required. Given observation o_t and candidate nodes $\mathcal{I}_t \subseteq \mathcal{V}_t$, the policy generates

$$y_t = \left(\{a_i\}_{v_i \in \mathcal{I}_t}, \{b_e\}_{e \in \mathcal{C}_t}, \{x_i\}_{v_i \in \mathcal{I}_t:\, a_i = \textsc{Revise}}\right), \tag{1}$$

where $\mathcal{C}_t$ is the candidate dependency set for this update, distinct from stored relations $\mathcal{E}_t$. Each a_i is KEEP, INVALIDATE, REVISE, or UNCERTAIN; $b_e \in \{0, 1\}$ indicates whether propagation along e is permitted; and x_i supplies replacement text for REVISE.

The executor applies distinct transitions: KEEP preserves a node; INVALIDATE marks it invalid without a successor; REVISE supersedes it and creates an active replacement; UNCERTAIN records an unresolved state. In the running example, REVISE creates a Paris residence version and preserves the New York version as history. Gates permit propagation to dependent memories; unrelated nodes receive KEEP. Actions specify changes, and gates authorize propagation. Utility rewards conflict resolution and preservation and penalizes edits.

4 CARE-MEM

4.1 EXECUTABLE REVISION WITHIN CANDIDATE NEIGHBORHOODS

The policy predicts actions, replacement text, and gates; the executor commits authorized changes in one version-preserving transaction (Figure 1). Training assigns counterfactual utility differences to decision spans in the *sampled proposal* (Figure 2). Schema-free denotes revision without predefined domain slots or a domain ontology. CAREBench supplies identical candidates to all systems.

4.2 RELATION-TYPED CONFLICT GRAPH AND STRUCTURED PROPOSALS

Given o_t and $\mathcal{I}_t$, CARE-Mem constructs the instance-level directed graph

$$G_t = (\{o_t\} \cup \mathcal{I}_t, \mathcal{C}_t). \tag{2}$$

A candidate edge $e = (u, v, r_e)$ has source u, target v, and relation label r_e: SUPPORTS, INVALIDATES, CONDITIONALLY-AFFECTS, or UNRELATED. These labels are inputs; gates are policy outputs. The action INVALIDATE differs from the relation INVALIDATES.

Let $b = (b_e)_{e \in \mathcal{C}_t}$, and let $\mathcal{R}_t(b)$ denote the set of memory nodes reachable from o_t through open gates ($b_e = 1$). Closed gates have $b_e = 0$. The effective action after gating is

$$\widetilde{a}_i = \begin{cases} a_i, & a_i = \textsc{Keep} \text{ or } v_i \in \mathcal{R}_t(b), \\ \textsc{Keep}, & \text{otherwise.} \end{cases} \tag{3}$$

When a gate closes, the executor recomputes reachability and can suppress downstream changes. In Figure 1, e_{12} permits propagation to the commute memory; e_{13} blocks it to the cuisine memory. KEEP preserves its target regardless of reachability. Serialized decisions occupy identifiable training spans (Section 4.5).

4.3 CONFLICT-SPECIFIC PROBES

Three probe families evaluate the executed proposal $\widetilde{y}$: CURRENT assesses the updated state, RETENTION assesses unrelated valid memories, and HISTORY assesses access to superseded facts. Utility is

$$S(\widetilde{y}) = w_c \bar{s}_{\text{cur}} + w_r \bar{s}_{\text{ret}} + w_h \bar{s}_{\text{his}} - \lambda C(\widetilde{y}), \tag{4}$$

where each $\bar{s}$ averages the corresponding probe family for the case; w_c, w_r, w_h weight these scores, and λ weights edit cost. C counts effective non-KEEP actions plus selected edges. Experiments use $w_c = w_r = w_h = 1$ and $\lambda = 0.1$; Appendix A.6 specifies counting and empty-family rules. In the example, the families assess residence in Paris, the Italian cuisine preference, and prior residence in New York, respectively. The History score is unchanged across the analyzed training interventions and contributes zero to their local utility differences.

4.4 NODE, EDGE, AND CONNECTED COUNTERFACTUAL CREDIT

Unlike the correct update in Figure 1, the sampled proposal in Figure 2 correctly updates the stored residence information but incorrectly invalidates the cuisine preference. Controls execute independently from the same pre-write snapshot. Write $S(y)$ for utility after gated execution, omitting t. A node control $y^{\neg i}$ changes only the target action: KEEP becomes INVALIDATE; INVALIDATE or REVISE becomes KEEP, with any replacement text removed. The control's targeted action is applied regardless of reachability; untargeted actions remain gated. The base uses ordinary gating. Node credit is

$$\Delta_i = S(y) - S(y^{\neg i}). \tag{5}$$

Here $\neg i$ denotes a prescribed action substitution, not node deletion. For UNCERTAIN, credit is the minimum of the differences against KEEP and INVALIDATE controls (Appendix A.3). An edge control $y^{\neg e}$ flips only b_e to $1 - b_e$: relation labels, candidate edges, node actions, and replacement text remain fixed, and reachability is recomputed:

$$\Delta_e = S(y) - S(y^{\neg e}). \tag{6}$$

A positive difference favors the sampled decision; a negative difference favors its control.

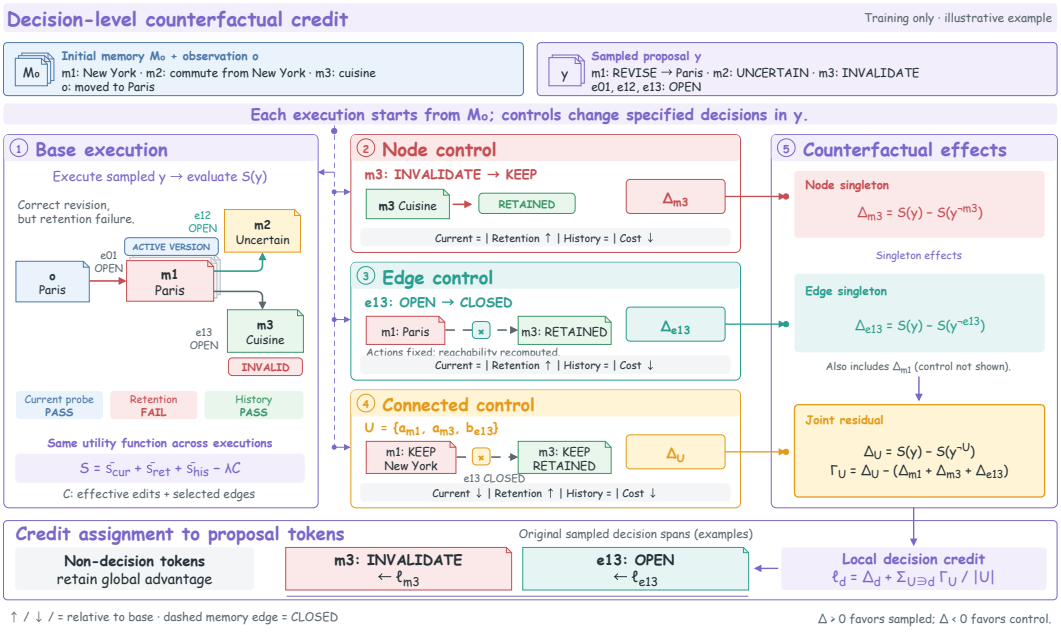


Figure 2: Illustrative counterfactual credit. Base and controls execute independently from M_0; changes are relative to the base. Subtracting all member singleton effects from Δ_U gives the joint residual. Local credit targets original sampled spans; non-decision tokens retain global advantage.

Connected controls jointly intervene on an edge gate and its endpoint actions. In sets, a_i and b_e denote *indexed decision items*, not their label or Boolean values. For $e = (u, v)$, suppressing r_e in the notation, define $U_e = \{b_e\} \cup \{a_i : v_i \in \{u, v\} \cap \mathcal{I}_t\}$ when $2 \le |U_e| \le 3$. The observation has no memory action. With $\Delta_U = S(y) - S(y^{\neg U})$, the residual is

$$\Gamma_U = \Delta_U - \sum_{d \in U} \Delta_d, \tag{7}$$

where Δ_d is the member singleton credit. Joint controls flip gates and substitute node actions, always replacing UNCERTAIN with KEEP; they do not enumerate alternatives. In Figure 2, the joint control for $U = \{a_{m1}, a_{m3}, b_{e13}\}$ keeps both nodes and closes e_{13}, reducing the Current score while improving the Retention score. Each of the $|U|$ members receives $\Gamma_U/|U|$; shares from overlapping sets are summed.

4.5 From Decision Credit to Policy Optimization

For each case, k indexes K retained valid proposals y_k. Their utilities define

$$A_k^{\text{global}} = \frac{S(y_k) - \mu_S}{\sqrt{\sigma_S^2 + \epsilon}}, \tag{8}$$

where μ_S and σ_S^2 are the mean and population variance of these K utilities, and $\epsilon = 10^{-8}$. Training uses $K = 2$; incomplete groups are not updated (Appendix A.3). For decision d, define $\ell_{k,d} = \Delta_{k,d} + \sum_{U \ni d} \Gamma_{k,U}/|U|$ and $\widehat{\ell}_{k,d} = \text{clip}(\ell_{k,d}, -5, 5)$ without normalization. Let $\mathcal{D}_{k,d}$ contain the token positions for decision d in proposal y_k, and $\mathcal{J}_k$ contain decisions with available local credit. CARE-Mem uses local override:

$$A_{k,\tau} = \begin{cases} \widehat{\ell}_{k,d}, & d \in \mathcal{J}_k,\ \tau \in \mathcal{D}_{k,d}, \\ A_k^{\text{global}}, & \tau \notin \bigcup_{d \in \mathcal{J}_k} \mathcal{D}_{k,d}. \end{cases} \tag{9}$$

Here τ is a token-position index. The decision spans are non-overlapping. Each node span covers the action value and any replacement text; each gate span covers the edge identifier and Boolean value. Local credit replaces global advantage on these spans; all remaining tokens retain global

advantage. Local, global, and interaction coefficients are one. The training implementation additionally scales edge-token advantages for repeated observations; write $\widetilde{A}$ for the resulting advantages (Appendix A.3).

The current policy minimizes a token-level clipped-surrogate loss using $\widetilde{A}$ and clipping parameter $\epsilon_{\text{clip}} = 0.2$ (Schulman et al., 2017), with sampled Kullback–Leibler (KL) penalties:

$$\mathcal{L}(\theta) = \mathcal{L}_{\text{clip}}(\theta; \widetilde{A}) + \beta_{\text{old}} \widehat{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{old}}) + \beta_{\text{ref}} \widehat{D}_{\text{KL}}(\pi_\theta \| \pi_{\text{parent}}). \tag{10}$$

Here θ denotes trainable parameters, π_θ the current policy, π_{old} the behavior policy generating the rollouts, and π_{parent} the frozen SFT parent. The negative clipped-surrogate loss and both sampled token-level penalties are averaged over valid proposal tokens. The two KL penalty weights are $\beta_{\text{old}} = \beta_{\text{ref}} = 0.01$. Appendix A.3 defines probability ratios, masks, averaging, and policy refresh. Invalid outputs are excluded from updates. At inference, CARE-Mem executes one gated proposal without probes or counterfactual evaluation.

5 CAREBENCH-240

5.1 DESIGN AND COVERAGE

CAREBench-240 contains 240 held-out cases from 13 predefined design strata. Analyses group them by conflict type, case shape, topic partition, and dependency depth; groups across dimensions overlap. Of the 240 cases, 120 concern topics present in the annotator qualification materials (qualification-covered), and 120 concern topics absent from those materials (qualification-uncovered). The direct-conflict, gradual-conflict, no-conflict, and propagation categories each contain 60 cases. Preassigned case-shape templates comprise 60 single-action, 75 multi-action, and 105 mixed cases (Appendix A.1). Reference annotations contain 793 node actions: 448 KEEP, 77 INVALIDATE, 139 REVISE, and 129 UNCERTAIN. Action Macro-F1 is the primary classification metric because the action labels are imbalanced; Table 23 reports per-class recall.

5.2 HUMAN CONSTRUCTION AND QUALITY CONTROL

Two case authors each wrote 120 cases. Each case was independently annotated by two members of a four-annotator pool. Annotators had no access to the case authors' reference annotations, private probe answers, or other annotators' labels. One of two adjudicators resolved each case using the annotation protocol; each adjudicator handled 120 cases. Averaged over six annotation batches, node-action Cohen's κ, edge micro-F1, and exact edge-set agreement were 0.9689, 0.9816, and 0.9667, respectively. Public candidate neighborhoods exclude reference actions, gates, replacements, and probe answers used for evaluation.

6 EXPERIMENTS

The experiments address three questions. **RQ1**: How does CARE-Mem compare with graph-memory systems on structured revision (Table 1)? **RQ2**: How do matched training objectives affect performance, and how does CARE-Mem's local credit align with training reference labels (Table 3 and Figure 4)? **RQ3**: How do the learned adapters perform on MemoryAgentBench FactConsolidation (Table 2; Hu et al., 2026)?

6.1 EXPERIMENTAL SETUP

Evaluation protocol. In CAREBench-240, systems receive identical candidate nodes and edges; references are used only for scoring. Policy conditions share a grammar-constrained lazy-greedy decoder, with no retries or repair. Execution failures receive zero probe scores; parseable structural fields are scored separately (Appendix A.6). Randomized conditions use Runs 0–2; deterministic systems run once. Action Macro-F1 and Edge F1 comparisons use 10,000 paired case-level bootstrap resamples and paired randomization tests with Holm correction, holding trained models fixed. Analyses of Edge-set Exact, strict replacement, and subgroups are post hoc exploratory; external comparisons are descriptive. Tables 1–3 report 0–100 scores as mean $\pm$ sample standard deviation

(SD). Bold and underlined values mark the highest and second-highest means, respectively, including ties; validity and zeros are not highlighted.

CAREBench systems. All policy-trained variants start from the same Qwen3-4B-Instruct-2507 SFT parent, using 16 training cases and two rollouts per case. A-MEM and CUPMem retain native operations: note or state changes map to node actions, and links to gates (Table 14). Append-only lacks comparable revision outputs. Direct Revision and MemCoE-MGI serve as interface diagnostics in the appendix; MGI is MemCoE's Memory Guideline Induction stage (Xu et al., 2026a). Global-only (A1) uses global advantage on all tokens without counterfactual (CF) controls. Global + local (A2) adds global advantage to local credit on decision spans; CARE-Mem uses local credit alone on those spans.

External evaluation. FactConsolidation contains 400 single-hop and 400 multi-hop questions; their accuracies are weighted equally. The learned adapters remain frozen. Previous facts sharing a claim key form the candidates; predicted actions update the active fact store (Appendix A.13).

6.2 CAREBENCH-240 MAIN RESULTS

Table 1: CAREBench-240 system comparison (mean ± sample SD, three runs). F1 scores are multiplied by 100; other columns are percentages. ‡: fixed mapping from native operations to the common interface. Strict replacement uses all reference REVISE nodes.

System	Action Macro-F1	Edge F1	Edge-set Exact	Strict replacement	Valid
A-MEM‡	18.8 ± 0.2	20.7 ± 3.1	24.3 ± 2.1	0.0	100.0
CUPMem‡	$\underline{27.8} \pm 0.8$	$\underline{38.5} \pm 1.5$	$\underline{28.2} \pm 1.3$	0.0	92.4 ± 0.2
CARE-Mem (ours)	$\mathbf{65.5} \pm 5.3$	$\mathbf{60.7} \pm 2.6$	$\mathbf{44.0} \pm 2.8$	$\mathbf{48.7} \pm 2.9$	100.0

CARE-Mem improves action and edge prediction over projected graph baselines. Table 1 reports Action Macro-F1 65.5, Edge F1 60.7, and Edge-set Exact 44.0%. The first two scores exceed CUPMem by 37.8 and 22.3 points, respectively, on the 0–100 scale; exactness requires every candidate-edge decision to match.

The 95% paired bootstrap confidence intervals (CIs) are $[32.7, 42.3]$ points for Action Macro-F1 and $[14.6, 29.6]$ for Edge F1; both Holm-adjusted p-values are below 0.0001. All 720 CARE-Mem predictions across three runs are executable.

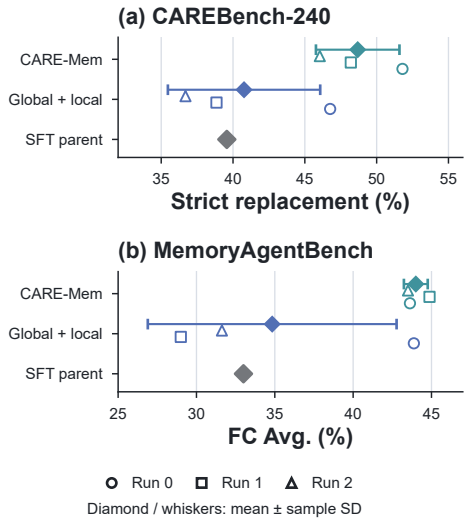


Figure 3: Matched outcomes across runs. Open markers identify Runs 0–2; diamonds and whiskers show mean ± sample SD. The SFT parent is evaluated once. (a) Strict replacement, exploratory. (b) FactConsolidation, descriptive.

6.3 MEMORYAGENTBENCH FACTCONSOLIDATION

The learned adapters improve downstream fact consolidation. Table 2 compares systems on the same 800 questions. CARE-Mem groups facts by claim key; HippoRAG 2 uses Qwen-adapted open information extraction (OpenIE) with the same reader (Gutiérrez et al., 2025). CARE-Mem reaches $44.0 \pm 0.8\%$ FC Avg., exceeding the append-only baseline, SFT parent, and global-plus-local control by 13.3, 11.0, and 9.2 percentage points, respectively. On average, CARE-Mem outperforms both matched controls on both tasks. Its FC Avg. exceeds the SFT parent's score in all three runs. A2 is slightly higher in Run 0 (43.88% versus 43.63%) but lower in Runs 1 and 2 (Figure 3b). All CARE-Mem and matched-control predictions are valid. Context-level results are in Appendix Figure 6.

Table 2: MemoryAgentBench results (%). Fact-SH and Fact-MH are single- and multi-hop Fact-Consolidation; FC Avg. is their unweighted mean accuracy. Valid is the protocol-valid output rate. Mean ± SD over three runs; †: deterministic; ‡: Qwen-adapted OpenIE. RAG: retrieval-augmented generation. All use Qwen3-4B.

System	Fact-SH	Fact-MH	FC Avg.	Valid
Memory systems				
CARE-Mem (ours)	**72.1** ± 0.9	**15.9** ± 0.9	**44.0** ± 0.8	100.0
Append-only†	55.3	6.3	30.8	100.0
BM25 RAG	37.3	3.8	20.5	100.0
Long context†	36.3	2.0	19.1	100.0
HippoRAG 2†‡	41.5	2.0	21.8	99.3
CUPMem	3.8 ± 0.7	0.1 ± 0.1	2.0 ± 0.4	91.7 ± 14.4
Matched training controls				
SFT parent†	57.0	9.0	33.0	100.0
Global + local (A2)	59.4 ± 11.3	10.3 ± 4.6	34.8 ± 7.9	100.0

6.4 Matched Controls and Credit Diagnostics

Local credit distinguishes aligned from misaligned decisions. Figure 4 examines 1,440 decision spans from 192 training rollouts. The area under the receiver operating characteristic curve (AUROC) is 0.864 for node credit and 0.675 for edge credit. Of 624 node and 816 edge decisions, 552 and 727 match the training reference labels, respectively. Eleven of 43 positive-global-advantage rollouts still contain a misaligned decision; 110/161 misaligned decisions receive negative local credit. These diagnostics measure agreement between credit and training labels. Full node and edge distributions and sign statistics appear in Appendix Figure 5 and Table 11.

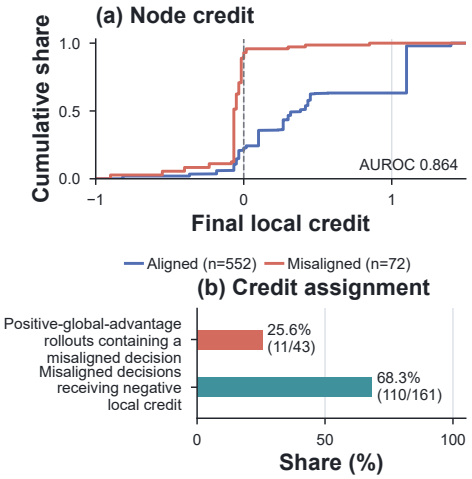


Figure 4: Training credit. (a) Node-credit cumulative distribution functions (CDFs) by training-label agreement. (b) Positive-global-advantage rollouts containing a misaligned decision (11/43), and misaligned decisions receiving negative credit (110/161).

The observed signal also depends on utility and serialization. In 75.25% of edge controls, the utility difference comes from edit cost alone; the History score is unchanged. In 7.6% of connected interventions, $|\Gamma_U| > |\sum_{d \in U} \Delta_d|$. REVISE constitutes 15.4% of node decisions but 50.4% of absolute node token-credit mass because replacement text occupies the same decision span as the action. Appendix C further analyzes these effects.

Matched comparisons show gains in replacement generation. Table 3 compares the SFT parent with policy variants trained on the same 16 cases. All use the same decoder, and every run produces 240/240 valid predictions.

CARE-Mem's strict replacement accuracy exceeds that of the SFT parent by 9.1 percentage points and that of A2 by 7.9 percentage points (Figure 3a; Holm-adjusted $p = 0.0151$ and 0.0028 across 14 post hoc exploratory tests). This endpoint requires the correct action label and an exact match to the reference replacement text. CARE-Mem does not differ significantly from either comparator in Action Macro-F1, Edge F1, or exploratory Edge-set Exact.

Ablations give metric-dependent rankings. CARE-Mem exceeds A1, A3, A4, and A6 in Action Macro-F1 by 6.6, 5.1, 7.1, and 8.7 points, respectively (Holm-adjusted $p \leq 0.028$).

Table 3: Matched CAREBench-240 conditions (mean ± sample SD, three runs; †: deterministic). F1 scores are multiplied by 100; exact-match rates are percentages. The variants without node CF or without edge CF also remove connected controls and assign global advantages to affected spans. Definitions and paired tests are in Tables 8, 9, and 16.

Condition	Action Macro-F1	Edge F1	Edge-set Exact	Strict replacement
Matched objectives				
CARE-Mem (ours)	65.5 ± 5.3	60.7 ± 2.6	$\underline{44.0} \pm 2.8$	$\underline{48.7} \pm 2.9$
SFT parent†	65.7	60.7	42.1	39.6
Global-only (A1)	58.9 ± 6.2	57.9 ± 2.1	40.3 ± 1.3	28.3 ± 11.6
Global + local (A2)	$\underline{67.2} \pm 2.7$	59.1 ± 2.4	40.4 ± 1.8	40.8 ± 5.3
Component removals				
Without node CF (A3)	60.5 ± 4.8	58.3 ± 3.2	40.0 ± 1.5	30.7 ± 11.5
Without edge CF (A4)	58.4 ± 9.2	$\mathbf{61.9} \pm 3.0$	$\mathbf{44.6} \pm 1.4$	29.3 ± 8.3
Without retention probe (A5)	$\mathbf{67.8} \pm 3.8$	60.3 ± 1.3	42.6 ± 1.0	$\mathbf{50.4} \pm 1.2$
Without connected CF (A6)	56.9 ± 8.6	$\underline{61.2} \pm 1.8$	41.7 ± 0.7	28.3 ± 7.9

The corresponding Edge F1 differences are not statistically significant. A3 and A4 also remove connected controls and change span credit, so they test combined changes. A4 has higher mean Edge F1 and Edge-set Exact than CARE-Mem. Removing Retention (A5) gives the highest Action Macro-F1 and replacement accuracy; its 2.3-point Macro-F1 gain is not significant after correction ($p = 0.061$; Tables 8–9).

On multi-action cases, joint action–edge exact-match accuracy is 38.2% for CARE-Mem, 29.3% for the SFT parent, and 31.6% for A2. For qualification-uncovered topics, the corresponding accuracies are 39.7%, 33.3%, and 33.6%. These post hoc comparisons require every action and gate to match the reference.

Policy training uses two optimizer updates and takes 0.556 ± 0.028 GPU-hours on one GPU per run, excluding SFT. At inference, CARE-Mem executes one proposal without counterfactual calls (Table 5).

7 Limitations

CAREBench-240 contains 240 synthetic cases from two authors with supplied candidates; baseline outputs require interface mappings. Policy training uses 16 cases and two updates. Case-level resampling holds trained models fixed; replacement and subgroup analyses are post hoc exploratory. MemoryAgentBench uses four shared contexts and a different candidate constructor and executor; Llama-3.1-8B is evaluated in only one run (Table 10).

Credit locality, scale, edit cost, connected residuals, and token-span weighting jointly affect the objective; isolating them requires matched estimators. Diagnostics measure credit-label agreement, not subsequent behavior. History scores are constant in these interventions. All evaluated policy conditions have zero Edge-set Exact on the 15 depth-3 cases. Probe-calibration limitations are discussed in Appendices A.7 and C; decoder revisions and complete reruns are documented in Appendix A.5.

8 Conclusion

CARE-Mem learns node actions and propagation gates for memory revision. It improves Action Macro-F1 and Edge F1 over mapped graph-memory baselines and, in exploratory comparisons, strict replacement exact-match accuracy over matched controls. On MemoryAgentBench FactConsolidation, average accuracy gains over the SFT parent and global-plus-local control are 11.00 and 9.17 percentage points, respectively, without further parameter updates. Same-snapshot counterfactuals supply training credit; inference executes one proposal.

AI USE STATEMENT

Generative AI tools were used as supplementary aids during the preparation of this work, primarily for language improvement, coding support, and routine consistency checks. All substantive aspects of the research, including study design, methodology, data analysis, interpretation, and conclusions, remained under the authors' direction and judgment. Any AI-assisted material was reviewed and verified by the authors before inclusion. The authors take full responsibility for the content and integrity of the submitted work.

ETHICS STATEMENT

CAREBench-240 was created by eight volunteer contributors with separate roles in case authoring, blind annotation, and adjudication. Released artifacts use anonymous aliases and contain no names, contact details, or other personally identifying information. The benchmark contains synthetic memory-revision scenarios rather than real user conversations; its cases are not used to infer contributor attributes. CAREBench-240 reference answers were withheld from the evaluated models. Case authoring, blind annotation, and adjudication were performed by human contributors. Only protocol-permitted, non-identifying data and statistics are released.

REPRODUCIBILITY STATEMENT

Section 6 describes the data, baselines, ablations, metrics, and statistical protocol. Appendix A records the dataset version, random seeds, training configurations, structured generation, failure handling, and artifact archiving. Deterministic scripts generate the tables in this paper from saved predictions and held-out evaluation results.

REFERENCES

Hanxiang Chao, Yihan Bai, Rui Sheng, Tianle Li, and Yushi Sun. STALE: Can LLM agents know when their memories are no longer valid? *arXiv preprint arXiv:2605.06527*, 2026. URL https://arxiv.org/abs/2605.06527.

Jiayang Cheng, Dongyu Ru, Lin Qiu, Yiyang Li, Xuezhi Cao, Yangqiu Song, and Xunliang Cai. AMemGym: Interactive memory benchmarking for assistants in long-horizon conversations. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/0856bc553d3e3b9827e5140d0ad3bf8d-Abstract-Conference.html.

Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. From RAG to memory: Non-parametric continual learning for large language models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 21497–21515. PMLR, 2025. URL https://proceedings.mlr.press/v267/gutierrez25a.html.

Yuanzhe Hu, Yu Wang, and Julian McAuley. Evaluating memory in LLM agents via incremental multi-turn interactions. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2507.05257.

Shuo Ji, Yibo Li, and Bryan Hooi. Memory is reconstructed, not retrieved: Graph memory for LLM agents. In *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2606.06036.

Marina Mao, Alexandr Liu, Pengbo Li, Siheng Li, Bo Zhou, and Xiang Wang. Tree-based credit assignment for multi-agent memory system. *arXiv preprint arXiv:2605.04811*, 2026. URL https://arxiv.org/abs/2605.04811.

Ikhtiyor Nematov, Tarik Kalai, Elizaveta Kuzmenko, Gabriele Fugagnoli, Dimitris Sacharidis, Katja Hose, and Tomer Sagi. Source attribution in retrieval-augmented generation. *arXiv preprint arXiv:2507.04480*, 2025. URL https://arxiv.org/abs/2507.04480.

Narjes Nourzad and Carlee Joe-Wong. MIRA: Memory-integrated reinforcement learning agent with limited LLM guidance. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/ca03ccaf61b63cf3ee5de18140f5885b-Abstract-Conference.html.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017. URL https://arxiv.org/abs/1707.06347.

Zhiyu Shen, Ziming Wu, Fuming Lai, Shaobing Lian, and Yanghui Rao. MemBuilder: Reinforcing LLMs for long-term memory construction via attributed dense rewards. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 27868–27887, 2026. doi: 10.18653/v1/2026.acl-long.1284. URL https://aclanthology.org/2026.acl-long.1284/.

Haoran Sun, Zekun Zhang, and Shaoning Zeng. Preference-aware memory update for long-term LLM agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 783–793, 2026. doi: 10.18653/v1/2026.findings-acl.38. URL https://aclanthology.org/2026.findings-acl.38/.

Mohammad Tavakoli, Alireza Salemi, Carrie Ye, Mohamed Abdalla, Hamed Zamani, and J. Mitchell. Beyond a million tokens: Benchmarking and enhancing long-term memory in LLMs. In *International Conference on Learning Representations*, 2026. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/d7f0cfa0fe759b033d5262e1bb7d4065-Abstract-Conference.html.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=pZiyCaVuti.

Zhaofen Wu, Hanrong Zhang, Fulin Lin, Wujiang Xu, Xinran Xu, Yankai Chen, Shaowen Chen, Henry Peng Zou, Weizhi Zhang, Xue Liu, Philip S. Yu, and Hongwei Wang. GAM: Hierarchical graph memory for LLM-based agents. In *International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=mmsVZGaYyp.

Derong Xu, Shuochen Liu, Pengfei Luo, Pengyue Jia, Yingyi Zhang, Yi Wen, Yimin Deng, Wenlin Zhang, Enhong Chen, Xiangyu Zhao, and Tong Xu. Learning how and what to memorize: Cognition-inspired two-stage optimization for evolving memory. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, pp. 44987–45011, 2026a. doi: 10.18653/v1/2026.acl-long.2084. URL https://aclanthology.org/2026.acl-long.2084/.

Derong Xu, Yi Wen, Pengyue Jia, Yingyi Zhang, Wenlin Zhang, Yichao Wang, Huifeng Guo, Ruiming Tang, Xiangyu Zhao, Enhong Chen, and Tong Xu. From single to multi-granularity: Toward long-term memory association and selection of conversational agents. In *International Conference on Learning Representations*, 2026b. URL https://proceedings.iclr.cc/paper_files/paper/2026/hash/93e444b4a1a64815631c3c74977ee200-Abstract-Conference.html.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. In *Advances in Neural Information Processing Systems*, 2025. URL https://arxiv.org/abs/2502.12110.

Jiangze Yan, Yi Shen, Wenjing Zhang, Jieyun Huang, Zhaoxiang Liu, Ning Wang, Kai Wang, and Shiguo Lian. HiMPO: Hindsight-informed memory policy optimization for less-entangled credit in long-horizon agents. *arXiv preprint arXiv:2606.16285*, 2026. URL https://arxiv.org/abs/2606.16285.

Chang Yang, Chuang Zhou, Yilin Xiao, Su Dong, Luyao Zhuang, Yujing Zhang, Zhu Wang, Zijin Hong, Zheng Yuan, Zhishang Xiang, Shengyuan Chen, Huachi Zhou, Qinggang Zhang, Ninghao Liu, Jinsong Su, Xinrun Wang, Yi Chang, and Xiao Huang. Graph-based agent memory: Taxonomy, techniques, and applications. *arXiv preprint arXiv:2602.05665*, 2026. URL https://arxiv.org/abs/2602.05665.

Yongshi Ye, Hui Jiang, Feihu Jiang, Tian Lan, Yichao Du, Biao Fu, Xiaodong Shi, Qianghuai Jia, Longyue Wang, and Weihua Luo. UMEM: Unified memory extraction and management framework for generalizable memory. In *International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=BoiXvrwtdi.

Puzhen Zhang, Xuyang Chen, Yu Feng, Yuhan Jiang, and Liqiu Meng. Constructing coherent spatial memory in LLM agents through graph rectification. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 48126–48146, 2026. doi: 10.18653/v1/2026.acl-long.2222. URL https://aclanthology.org/2026.acl-long.2222/.

A Reproducibility Details

Table 4: CAREBench-240 statistics.

Dimension	Category	Count
Topic partition	Qualification-covered / Uncovered	120 / 120
Conflict type	Direct / Gradual / No-conflict / Propagation	60 / 60 / 60 / 60
Case shape	Single / Multi / Mixed	60 / 75 / 105
Primary action	KEEP / INVALIDATE / REVISE / UNCERTAIN	60 / 45 / 60 / 75
Action occurrences	KEEP / INVALIDATE / REVISE / UNCERTAIN	448 / 77 / 139 / 129
Author batch	Batch 01–06	6×40

A.1 Dataset and Evaluation Split

The release artifact for CAREBench-240 is named `CAREBench-240-Sprint` (version `2026-07-28.v1`) and contains a single 240-case test split. All cases are held out from training and prompt demonstrations. Systems receive public candidate neighborhoods; the offline evaluator uses the separate reference actions, edges, replacements, and probe answers after predictions are fixed.

The three agreement statistics in Section 5 compare the two independent annotations before adjudication. Each of six 40-case batches yields node-action Cohen's κ over its pooled node labels, edge micro-F1 over its pooled edge decisions, and the fraction of cases with identical edge sets. The reported values are the unweighted means of these six batch statistics; author reference labels and adjudicated labels are not used in this agreement calculation.

The 13 predefined design strata P01–P13 partition the 240 cases. Case shape is a preassigned template category, not a function of model predictions. Single-action templates are P01, P02, and P08 (60 cases); multi-action templates are P03, P04, P05, P06, and P10 (75); mixed templates are P07, P09, P11, P12, and P13 (105). Mixed includes uncertainty propagation and no-edit scenarios, so it is not defined by a threshold on non-KEEP actions. Conflict, topic, depth, and case-shape analyses regroup the same cases. Dependency depth is the preassigned label in the locked design roster, not a distance recomputed from model predictions. In the depth-3 example in Table 13, $m_1 \to m_2 \to m_3 \to m_4$ has three memory-to-memory dependency edges; the observation-to-m_1 edge is separate. Qualification-covered and qualification-uncovered refer to the annotator qualification materials; they do not describe the model's pretraining exposure.

A.2 SFT Parent and Data Separation

The SFT parent is trained on separate bootstrap data containing 96 synthetic training cases and 32 development cases, constructed to cover four actions, graph shapes, and positive and negative edges. The training and development splits have disjoint persona, topic, and template identifiers. The parent model is fine-tuned on the 96 training cases for 10 epochs (120 optimizer steps), with low-rank adaptation (LoRA) checkpoints saved every two steps. Structured evaluation assesses the predeclared checkpoints 48, 54, 56, and 62 using eight fixed bootstrap training cases and four rollouts per case. Selection requires at least 95% valid outputs, at least two distinct canonical decisions in

at least 50% of cases, and a mean of at least 1.5 distinct decisions per case. All four checkpoints satisfy these predefined criteria; the earliest, checkpoint 48, is selected. The fixed training subset uses case suffixes 0000, 0018, 0024, 0030, 0054, 0060, 0084, and 0090.

The bootstrap data and CAREBench-240 use disjoint case and topic-family identifiers but share the same four-action interface, candidate-edge representation, and behavioral specification. The parent thus learns the structured task interface before policy optimization. Prompts, policy settings, and ablation configurations are fixed before private scoring. The deterministic parent baseline in Tables 2 and 3 uses this checkpoint without the two policy updates.

A.3 Model, Initialization, and Training

The main policy uses Qwen3-4B-Instruct-2507, initialized independently for each run from the same checkpoint-48 SFT adapter. LoRA uses rank 16, a scaling parameter of $\alpha = 32$, and a dropout rate of 0.05. Target modules are `q_proj`, `k_proj`, `v_proj`, `o_proj`, `gate_proj`, `up_proj`, and `down_proj`. Training uses bfloat16 precision and gradient checkpointing, with a maximum sequence length of 8,192 tokens; the model is not loaded in 4-bit precision. Runs 0, 1, and 2 use seeds 42, 43, and 44, respectively.

Each run uses 16 training cases, two epochs, and $K = 2$ valid rollouts per case. Sampling uses temperature 0.9, top-p 0.98, and `max_new_tokens`=768. Invalid samples are discarded; each of at most four collection rounds requests the missing rollouts. If any case still has fewer than K valid proposals, the update fails rather than using an incomplete group. Within a case,

$$\mu_S = \frac{1}{K}\sum_{k=1}^{K} S(y_k), \qquad \sigma_S^2 = \frac{1}{K}\sum_{k=1}^{K}(S(y_k) - \mu_S)^2, \qquad \epsilon = 10^{-8}.$$

The denominator is K, not $K-1$. Optimization uses learning rate 10^{-5}, weight decay 0.01, gradient accumulation of one, maximum gradient norm 1.0, and the proximal policy optimization (PPO) clipped-surrogate loss with $\epsilon_{\text{clip}} = 0.2$. Both KL penalty weights are 0.01. The local, global, and interaction coefficients are one; node, edge, and connected controls are enabled separately. Connected sets contain at most three decisions. Local credit is unnormalized and clipped to $[-5, 5]$.

For averaging across the update group, let j index cases, ξ_j be the observation and candidate-memory input, and $z_{j,k,\tau}$ be token τ of proposal $y_{j,k}$, with prefix $z_{j,k,<\tau}$. The main text suppresses j. Define

$$r_{j,k,\tau}(\theta) = \frac{\pi_\theta(z_{j,k,\tau} \mid \xi_j, z_{j,k,<\tau})}{\pi_{\text{old}}(z_{j,k,\tau} \mid \xi_j, z_{j,k,<\tau})}.$$

Let $m_{j,k,\tau}$ be one for valid proposal tokens and zero for prompt or padding positions, and define the token-weighted empirical mean

$$\widehat{\mathbb{E}}_m[f] = \frac{\sum_{j,k,\tau} m_{j,k,\tau} f_{j,k,\tau}}{\sum_{j,k,\tau} m_{j,k,\tau}}.$$

Within each optimizer update, $g(j)$ denotes the group containing case j whose observations are identical after case folding and whitespace normalization. Let $n_{g(j)}$ be the number of cases in that group; edge-token advantages are scaled by its reciprocal:

$$\widetilde{A}_{j,k,\tau} = \begin{cases} A_{j,k,\tau}/n_{g(j)}, & \tau \text{ belongs to an edge decision span,} \\ A_{j,k,\tau}, & \text{otherwise.} \end{cases}$$

This edge-family scaling is distinct from the unit local, global, and interaction coefficients. The policy loss is

$$\mathcal{L}_{\text{clip}}(\theta; \widetilde{A}) = -\widehat{\mathbb{E}}_m\left[\min\left(r_{j,k,\tau}\widetilde{A}_{j,k,\tau}, \text{clip}(r_{j,k,\tau}, 0.8, 1.2)\widetilde{A}_{j,k,\tau}\right)\right].$$

For $\pi_r \in \{\pi_{\text{old}}, \pi_{\text{parent}}\}$, set

$$\delta^{(r)}_{j,k,\tau} = \log \pi_r(z_{j,k,\tau} \mid \xi_j, z_{j,k,<\tau}) - \log \pi_\theta(z_{j,k,\tau} \mid \xi_j, z_{j,k,<\tau}).$$

The implemented sampled penalty is

$$\widehat{D}_{\text{KL}}(\pi_\theta \| \pi_r) = \widehat{\mathbb{E}}_m\left[\exp(\delta^{(r)}_{j,k,\tau}) - \delta^{(r)}_{j,k,\tau} - 1\right].$$

This is an empirical token average over the retained behavior-policy proposals under the stated sampling and structured-decoding settings. Each epoch collects all 16 case groups before one optimizer update. Behavior log probabilities are computed before that update and held fixed during it; the next epoch collects new rollouts from the updated policy. The SFT parent remains frozen. Micro-batch losses and KL penalties are weighted by their valid-token counts, giving one token average over the whole group rather than equal proposal averages. Experiments use Python 3.10, PyTorch 2.8.0+cu128, bfloat16, and one RTX 5090 per run. Configurations, predictions, and scores are retained in the artifacts.

Prescribed counterfactual controls. For an UNCERTAIN action, the singleton credit is

$$\Delta_i = \min\{S(y) - S(y^{i\to\textsc{Keep}}), S(y) - S(y^{i\to\textsc{Invalidate}})\}.$$

The higher-utility singleton control therefore determines the smaller difference. Joint interventions instead use a fixed UNCERTAIN→KEEP substitution. Their targeted node actions bypass reachability, while all untargeted actions remain gated. The base proposal uses ordinary gating. Joint controls do not select the higher-utility singleton branch or enumerate combinations. For sets containing UNCERTAIN actions, the residual is defined relative to the prescribed singleton credits and may also reflect differences between the singleton and joint substitution rules.

A.4 TRAINING AND INFERENCE EFFICIENCY

Table 5 summarizes CAREBench training and inference time. Policy GPU-hours measure the single-GPU time from the start of policy training to completion of the second optimizer update, excluding SFT training. A case exposure is one training case in one epoch. The CF count is measured per case exposure and includes controls for both retained rollouts, excluding their two base executions. CF controls are deterministic state executions with probe utility evaluation, not additional model generations. All three policy conditions use 16 cases, two epochs, and two rollouts per case.

Table 5: CAREBench policy-stage efficiency. Randomized conditions report mean ± sample SD over three seeds. All conditions use zero test-time counterfactual calls. N/A denotes no policy-training stage for the SFT parent.

Condition	Base rollouts/ case exposure	CF controls/ case exposure	Policy GPU-hours	Inference s/case	Test-time CF
SFT parent	N/A	N/A	N/A	29.99	0
Global-only (A1)	2.00	0.00	0.552 ± 0.002	29.49 ± 0.47	0
Global + local (A2)	2.00	24.68 ± 0.09	0.549 ± 0.016	29.50 ± 0.76	0
CARE-Mem	2.00	24.75 ± 0.08	0.556 ± 0.028	29.65 ± 0.33	0

In this short training configuration, policy training times remain similar across the matched conditions despite CARE-Mem's deterministic counterfactual evaluations, and inference retains zero counterfactual calls. Peak GPU memory is unavailable in the archived logs.

A.5 DECODING AND PROTOCOL AMENDMENTS

During training, generation constrains `replacement_text` to at most 256 characters and `proposal_id` to at most 128 characters using `lm-format-enforcer==0.11.2`. Reported CAREBench inference uses the grammar-constrained `lmfe_lazy_single_sentence_max256_v4` decoder, which enforces terminal marks and sentence boundaries during generation. All 22 runs of the parent, trained policies, and ablations each produce 240 executable predictions out of 240 cases. Evaluation uses no automatic retry, truncation, or output repair.

The decoding constraints were amended after failures in public test outputs. All 22 CAREBench runs were repeated under the revised decoding constraints. For MemoryAgentBench, a 128-character proposal-ID bound was likewise added, and all reported CARE-Mem runs were repeated in full. These amendments used public outputs, not private correctness labels. Reported results use complete runs under the final protocols, without mixing predictions across versions.

A.6 SCORING AND PROBE DEFINITIONS

The serialized interface includes a reserved `new_memory` list. It is empty in the inspected CARE-Mem training rollouts and all 5,280 predictions from the parent, trained policies, and ablations on CAREBench. The trigger observation is not inserted automatically; active successors are created by REVISE. Node and edge controls leave this reserved field unchanged.

The scorer separates readable structural decisions from a fully executable proposal. Once a proposal can be parsed and its node and edge identifiers and required fields pass schema validation, its action, edge, and replacement fields can contribute to structural endpoints. If subsequent sandbox application or state validation fails, the validity flag is set to false, but the parsed structural fields are retained. With no parseable proposal, actions are missing, selected edges are absent, and Edge-set Exact is false. Validity additionally requires complete execution on the versioned snapshot and a resulting probe state. This distinction explains why Direct Revision can have an Edge-set Exact score higher than its prediction-validity rate.

Probes use no auxiliary LLM. Each probe specifies protected node identifiers. Current and Retention read active protected nodes and active successors whose supersession links reference those identifiers. History reads the explicitly protected nodes regardless of status, rather than traversing every historical version. For a nonempty Current target set containing only UNCERTAIN nodes, the prediction is `<ABSTAIN>`. Unicode case-folding and token normalization precede the scoring rule fixed in each probe record: normalized exact match, substring match, set F1, or uncertainty-aware scoring. A non-executable prediction has no probe result and contributes zero under the fixed denominator.

For each probe family, $\bar{s}$ in Equation (4) is the sum of its scores divided by its number of probes; an empty family receives zero. Writing $\tilde{a}_i$ for the effective action after gating, edit cost is

$$C(\tilde{y}) = \sum_{v_i \in \mathcal{I}_t} \mathbf{1}[\tilde{a}_i \neq \textsc{Keep}] + \sum_{e \in \mathcal{C}_t} b_e.$$

Thus, every selected candidate edge contributes to cost, including an edge whose source is unreachable. Targeted node actions in counterfactual controls bypass reachability as specified in Appendix A.3.

A.7 BASELINES, METRICS, AND STATISTICS

Deterministic systems, including the SFT parent, run once; randomized systems use seeds 42, 43, and 44 where training or sampling is involved. Unsupported capabilities are reported as N/A. Trained ablations share the parent, training cases, decoder, and test input; their training changes are specified in Table 16. Reported endpoints include Action Macro-F1, Edge F1, Edge-set Exact, strict replacement exact-match accuracy, and prediction validity. Current, Retention, and History are executor diagnostics: executing adjudicated reference proposals yields Current = 0.5625, Retention = 0.4792, and History = 0.5000 (Table 22, All cases). Mean probe scores remain below one even for adjudicated reference proposals. They are training-utility components and descriptive execution diagnostics, rather than calibrated measures of revision correctness. Statistical units are cases; Action Macro-F1 and Edge F1 comparisons use paired bootstrap intervals and paired randomization tests with Holm adjustment.

A.8 PER-RUN RESULTS AND STATISTICAL FAMILIES

Tables 6–10 provide per-run structural, probe, and external results, together with paired comparisons. CAREBench uses 10,000 paired case-level bootstrap resamples for intervals and paired randomization tests for p-values. Holm correction covers Action Macro-F1 and Edge F1 within the CUPMem comparison and, separately, within each comparison with the parent or one of A1–A6 (seven separate two-test families). Matched Edge-set Exact and strict replacement tests form a separate post hoc 14-hypothesis exploratory family. MemoryAgentBench comparisons are descriptive. Complete stratified results appear in Appendix B, and probe sensitivity in Appendix C.

Table 6: Per-run CAREBench system results. A-MEM and CUPMem use fixed capability projections; Direct Revision and MemCoE-MGI are interface diagnostics. Probe columns are descriptive executor diagnostics. All scores use the 0–1 scale.

System	Run	Action Macro-F1	Edge F1	Edge-set Exact	Current	Retention	History	Valid
Direct Revision	0	0.0044	0.0225	0.0917	0.0000	0.0000	0.0000	0.0083
Direct Revision	1	0.0000	0.0058	0.1125	0.0000	0.0000	0.0000	0.0042
Direct Revision	2	0.0000	0.0000	0.1167	0.0000	0.0000	0.0000	0.0000
A-MEM	0	0.1872	0.2369	0.2333	0.1250	0.4792	0.5000	1.0000
A-MEM	1	0.1868	0.1743	0.2292	0.1250	0.4792	0.5000	1.0000
A-MEM	2	0.1899	0.2110	0.2667	0.1250	0.4792	0.5000	1.0000
CUPMem	0	0.2842	0.3931	0.2917	0.2083	0.3750	0.4542	0.9208
CUPMem	1	0.2689	0.3669	0.2875	0.1833	0.4042	0.4667	0.9250
CUPMem	2	0.2794	0.3938	0.2667	0.1917	0.4042	0.4750	0.9250
MemCoE-MGI	0	0.0071	N/A	N/A	0.0000	0.0042	0.0042	0.0125
MemCoE-MGI	1	0.0033	N/A	N/A	0.0000	0.0000	0.0000	0.0042
MemCoE-MGI	2	0.0129	N/A	N/A	0.0083	0.0125	0.0167	0.0208
CARE-Mem	0	0.7160	0.6318	0.4333	0.2958	0.4375	0.5000	1.0000
CARE-Mem	1	0.6340	0.6109	0.4708	0.2458	0.4667	0.5000	1.0000
CARE-Mem	2	0.6160	0.5796	0.4167	0.2333	0.4792	0.5000	1.0000

Table 7: Per-run CAREBench results for the policy conditions. The SFT parent is evaluated once with deterministic decoding; CARE-Mem per-run values appear in Table 6. CF denotes counterfactual controls; condition definitions appear in Table 16. All scores use the 0–1 scale.

(a) Structural prediction metrics

Condition	Run	Action Macro-F1	Edge F1	Edge-set Exact
SFT parent	0	0.6569	0.6072	0.4208
Global-only (A1)	0	0.6591	0.5997	0.3917
Global-only (A1)	1	0.5639	0.5802	0.4167
Global-only (A1)	2	0.5439	0.5584	0.4000
Global + local (A2)	0	0.7034	0.6163	0.3917
Global + local (A2)	1	0.6550	0.5882	0.4250
Global + local (A2)	2	0.6575	0.5694	0.3958
Without node CF	0	0.6600	0.6110	0.4125
Without node CF	1	0.5791	0.5901	0.4042
Without node CF	2	0.5747	0.5474	0.3833
Without edge CF	0	0.6892	0.6521	0.4292
Without edge CF	1	0.5214	0.5940	0.4542
Without edge CF	2	0.5409	0.6123	0.4542
Without retention probe	0	0.7197	0.6169	0.4208
Without retention probe	1	0.6435	0.5913	0.4208
Without retention probe	2	0.6721	0.6014	0.4375
Without connected CF	0	0.6673	0.6319	0.4250
Without connected CF	1	0.5248	0.5969	0.4125
Without connected CF	2	0.5138	0.6077	0.4125

(b) Executor diagnostics and prediction validity

Condition	Run	Current	Retention	History	Valid
SFT parent	0	0.2542	0.4500	0.5000	1.0000
Global-only (A1)	0	0.2542	0.4417	0.5000	1.0000
Global-only (A1)	1	0.2167	0.4583	0.5000	1.0000
Global-only (A1)	2	0.2125	0.4625	0.5000	1.0000
Global + local (A2)	0	0.3292	0.4292	0.5000	1.0000
Global + local (A2)	1	0.2458	0.4542	0.5000	1.0000
Global + local (A2)	2	0.2458	0.4583	0.5000	1.0000
Without node CF	0	0.2625	0.4458	0.5000	1.0000
Without node CF	1	0.2208	0.4542	0.5000	1.0000
Without node CF	2	0.2208	0.4625	0.5000	1.0000
Without edge CF	0	0.3167	0.4375	0.5000	1.0000
Without edge CF	1	0.1958	0.4708	0.5000	1.0000
Without edge CF	2	0.2083	0.4625	0.5000	1.0000
Without retention probe	0	0.3042	0.4375	0.5000	1.0000
Without retention probe	1	0.2500	0.4667	0.5000	1.0000
Without retention probe	2	0.2625	0.4583	0.5000	1.0000
Without connected CF	0	0.2750	0.4333	0.5000	1.0000
Without connected CF	1	0.2083	0.4625	0.5000	1.0000
Without connected CF	2	0.2125	0.4625	0.5000	1.0000

Table 8: Paired CARE-Mem-minus-comparator differences on CAREBench. Intervals are pointwise 95% paired-case bootstrap intervals, with trained models held fixed. Differences and intervals use the 0–1 scale. Holm correction covers Action Macro-F1 and Edge F1 within each comparator family; intervals are not multiplicity-adjusted.

	Action Macro-F1			**Edge F1**		
Comparator	Δ	95% CI	Holm p	Δ	95% CI	Holm p
CUPMem (projected)	0.3778	$[0.3275, 0.4231]$	0.00004	0.2229	$[0.1458, 0.2962]$	0.00004
SFT parent	-0.0015	$[-0.0436, 0.0408]$	1.00000	0.0002	$[-0.0285, 0.0281]$	1.00000
Global-only (A1)	0.0664	$[0.0256, 0.1066]$	0.00452	0.0280	$[-0.0008, 0.0566]$	0.05898
Global + local (A2)	-0.0167	$[-0.0510, 0.0166]$	0.35904	0.0161	$[-0.0077, 0.0394]$	0.35904
Without node CF	0.0507	$[0.0112, 0.0894]$	0.02796	0.0246	$[-0.0024, 0.0515]$	0.07446
Without edge CF	0.0715	$[0.0339, 0.1096]$	0.00060	-0.0120	$[-0.0423, 0.0163]$	0.43512
Without retention probe	-0.0231	$[-0.0430, -0.0045]$	0.06120	0.0042	$[-0.0118, 0.0197]$	0.61085
Without connected CF	0.0867	$[0.0502, 0.1245]$	0.00004	-0.0047	$[-0.0350, 0.0248]$	0.76017

Table 9: Post hoc exploratory CARE-Mem-minus-comparator differences. Differences and intervals use the 0–1 scale. Holm correction covers all 14 hypotheses for Edge-set Exact and strict replacement accuracy. Pointwise 95% paired-case bootstrap intervals hold trained models fixed and are not multiplicity-adjusted.

	Edge-set Exact			**Strict replacement**		
Comparator	Δ	95% CI	Holm p	Δ	95% CI	Holm p
SFT parent	0.0194	$[-0.0194, 0.0583]$	0.81046	0.0911	$[0.0380, 0.1515]$	0.01512
Global-only (A1)	0.0375	$[0.0056, 0.0694]$	0.19110	0.2038	$[0.1344, 0.2767]$	0.00028
Global + local (A2)	0.0361	$[0.0056, 0.0667]$	0.19110	0.0791	$[0.0378, 0.1290]$	0.00280
Without node CF	0.0403	$[0.0083, 0.0736]$	0.13072	0.1799	$[0.1182, 0.2440]$	0.00028
Without edge CF	-0.0056	$[-0.0361, 0.0264]$	0.81046	0.1942	$[0.1228, 0.2731]$	0.00028
Without retention probe	0.0139	$[-0.0069, 0.0361]$	0.81046	-0.0168	$[-0.0375, 0.0000]$	0.77138
Without connected CF	0.0236	$[-0.0083, 0.0556]$	0.77138	0.2038	$[0.1330, 0.2791]$	0.00028

Table 10: Per-run MemoryAgentBench results for the reported systems and backbone comparison. Values are percentages; † denotes the single-run evaluation with Llama-3.1-8B.

System	Run	Fact-SH	Fact-MH	FC Avg.	Valid
CARE-Mem (Qwen3-4B)	0	71.25	16.00	43.63	100.00
CARE-Mem (Qwen3-4B)	1	73.00	16.75	44.88	100.00
CARE-Mem (Qwen3-4B)	2	72.00	15.00	43.50	100.00
BM25 RAG	0	37.25	3.75	20.50	100.00
BM25 RAG	1	37.25	3.75	20.50	100.00
BM25 RAG	2	37.25	3.75	20.50	100.00
CUPMem	0	4.25	0.00	2.13	100.00
CUPMem	1	4.25	0.25	2.25	100.00
CUPMem	2	3.00	0.00	1.50	75.00
Global + local (A2)	0	72.25	15.50	43.88	100.00
Global + local (A2)	1	50.75	7.25	29.00	100.00
Global + local (A2)	2	55.25	8.00	31.63	100.00
CARE-Mem (Llama-3.1-8B)†	0	61.25	10.25	35.75	100.00

A.9 Decision-Level Credit Analysis

Credit diagnostics recompute interventions from saved training rollouts and compare sampled decisions with bootstrap training labels. Benchmark stratification uses held-out CAREBench labels during offline analysis of the completed predictions. A node decision is aligned when its action type matches the training label; an edge decision is aligned when its Boolean matches. Replacement text is not part of this alignment label. Final local-credit sign accuracy uses the raw, unclipped $\ell_{k,d}$, including connected-residual shares: aligned decisions must have positive credit and misaligned decisions negative credit. Zero credit counts as incorrect in the denominators of 624 node and 816 edge

decisions. The Spearman statistic correlates this raw final credit with the singleton Δ; UNCERTAIN uses the smaller of its two control differences.

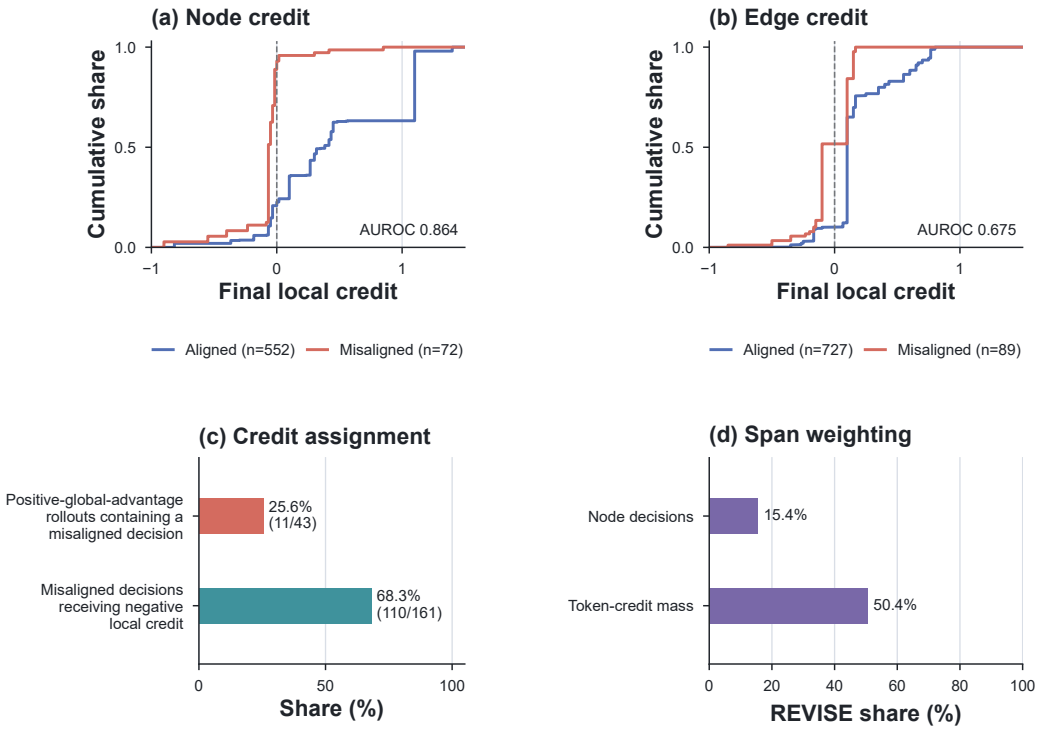


Figure 5: Complete decision-level credit diagnostics on bootstrap training traces. (a,b) Empirical cumulative distributions for gold-aligned and gold-misaligned decisions. (c) The first fraction is positive-global-advantage rollouts containing a misaligned decision (11/43); the second is misaligned decisions receiving negative local credit (110/161). (d) REVISE's share of node decisions and absolute token-credit mass.

Table 11: Decision-level credit diagnostics over 1,440 training decision spans. Gold-aligned means that the sampled decision matches the training label.

Metric	Node	Edge
Decision count	624	816
$P(\Delta > 0 \mid \text{gold-aligned})$	0.690	0.879
$P(\Delta < 0 \mid \text{misaligned})$	0.778	0.539
Final local-credit sign accuracy	0.787	0.858
AUROC for gold-aligned decisions	0.864	0.675
Spearman(final credit, singleton Δ)	0.747	0.809

Among 43 positive-global-advantage rollouts, 11 contain at least one misaligned decision (25.6% of these rollouts), and local credit assigns negative values to 68.3% of misaligned decisions. The connected residuals have a numerically zero median absolute value; 62 of 816 (7.6%) exceed the magnitude of the summed singleton effects, $|\Gamma_U| > |\sum_{d \in U} \Delta_d|$. Singleton statistics use the smaller delta for the two UNCERTAIN controls. The output span also couples REVISE with its replacement: REVISE accounts for 15.4% of node decisions but 50.4% of absolute node token-credit mass because its mean span is 21.34 tokens, compared with 1–3 for other actions. Across the 2,376 interventions, the History score never changes. Edge controls change the Current score in 24.75% of records and never change Retention or History scores. In 75.25% of edge interventions, the utility difference is

nonzero solely because of edit cost. Current and edit cost therefore supply the observed edge-control signal, while node and connected controls can also change Retention.

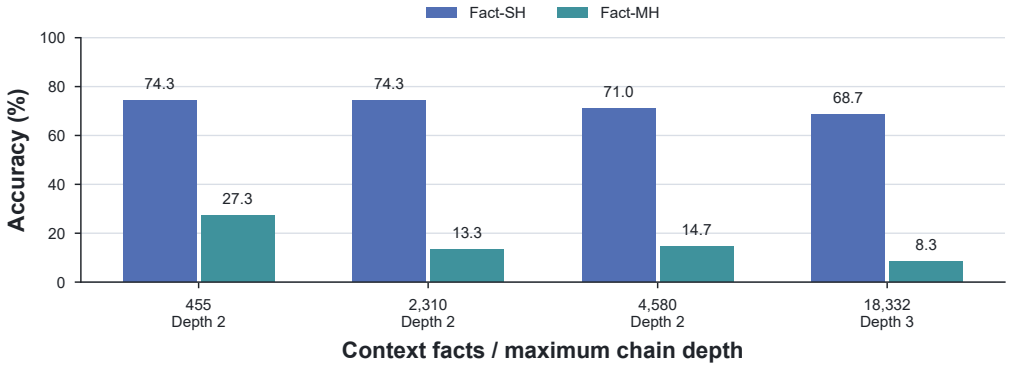


Figure 6: CARE-Mem fact consolidation across four contexts. Bars show mean accuracy over three runs, with 100 questions per task in each context. Context size and maximum chain depth vary jointly; the comparison is descriptive.

Figure 6 groups questions according to the fact count and maximum chain depth of the context shared by each group. Per-question hop count and predecessor-chain length are unavailable in the scored records.

A.10 Prediction and Execution Case Studies

These post hoc examples contrast structural success (case 0089) with propagation failure (case 0106), using Run 0 (seed 42). Case 0106 is the first depth-3 case in case-ID order. Both use the same decoder as the reported evaluation; Figure 2 is a separate constructed training example. Post-update states are obtained by replaying the saved proposals. Node identifiers are abbreviated, and replacement strings are reproduced verbatim.

Table 12: Structural success: `carebench-240-test-0089`, Run 0. All three CARE-Mem runs match the reference actions, gates, and replacement strings for this case.

Field	Prediction and execution outcome
Observation	"A repair note replaces component Hazel-3 with component Hazel-8."
Memory	m_1: repair uses Hazel-3; m_2: calibration follows its installation; m_3: completion card records that calibration; m_4: ivory lampshade.
Actions	Reference and prediction: REVISE m_1, m_2, m_3; KEEP m_4.
Open gates	Reference and prediction: $o \to m_1$, $m_1 \to m_2$, $m_2 \to m_3$. The other three candidate gates are closed.
Replacements	"A fictional floor-lamp repair uses component Hazel-8." "The calibration test begins after component Hazel-8 is installed." "The completion card records the calibration test for Hazel-8."
Execution	Three active version-2 successors contain Hazel-8. Their prior Hazel-3 versions are superseded and retained; the ivory-shade node stays active.
Recorded scores	Valid; 4/4 action labels, 6/6 gate Booleans, and 3/3 replacement strings match. Current/Retention/History probe scores are 0/0/0.

In case 0089 (Table 12), the replay preserves the expected current and historical texts. Its recorded zero probe scores illustrate the calibration limitation discussed in Appendices A.7 and C, despite exact matches for the action labels, gate Booleans, and replacement strings.

Table 13: Propagation failure: `carebench-240-test-0106`, Run 0. All three runs have zero Edge-set Exact scores for this case; the displayed actions and execution are specifically from Run 0.

Field	Prediction and execution outcome
Observation	"A fictional supply delivery may arrive late, so the Meadow Bell setup shift may be delayed; the arrival time is unknown."
Memory	m_1: setup at 08:00; m_2: wayfinding follows setup; m_3: check-in follows wayfinding; m_4: orientation follows check-in. m_5 is an unrelated garden-count duty.
Reference	UNCERTAIN for m_1–m_4, KEEP for m_5. Open $o \to m_1 \to m_2 \to m_3 \to m_4$; close $o \to m_5$.
Prediction	UNCERTAIN for all five nodes; all five candidate gates closed. No replacement text is emitted or required.
Execution	No node is reachable from the observation. All predicted non-KEEP actions are suppressed; all five original nodes remain active with unchanged text.
Recorded scores	Valid; action accuracy 0.8000, Edge F1 0.0000, Edge-set Exact 0.0000. Current/Retention/History are 0/1/1. Replacement is not applicable.

Case 0106 (Table 13) separates predicting a plausible action from authorizing its execution: four of five action labels match the reference, but closed gates prevent the predicted uncertainty updates from being applied along the dependency chain.

A.11 BASELINE CAPABILITY MAPPING

Table 14 makes the common CAREBench interface explicit. Mappings use each system's post-update memory state without reference labels or scores. The projected action and edge predictions support comparison through a common interface; they are not the external systems' native outputs.

Table 14: CAREBench baseline capability mapping. Unsupported endpoints are N/A, not zero.

System	Node-action projection	Candidate-edge projection
Append-only	Leaves historical memories unchanged; actions are N/A	N/A
Direct Revision	Native four-way action	Native edge Boolean
A-MEM	Removed/changed/unchanged notes map to INVALIDATE/REVISE/KEEP	Select when an official note link matches the candidate directed pair (derived)
CUPMem	Same-track successor, unknown-current, still-active, and all-stale states map to REVISE/UNCERTAIN/KEEP/INVALIDATE	Select when an official stale-support link maps to the candidate session pair (derived)
MemCoE-MGI	The adapted Memory Guideline Induction stage updates a profile; a projector using the same backbone maps it to node actions	Non-native endpoint, reported N/A

A.12 REPLACEMENT RESULTS

Table 15 reports replacement endpoints for the matched policy conditions. Coverage is the fraction of gold REVISE nodes with a generated replacement; strict accuracy is exact match to the annotated text. Both use all 139 reference REVISE nodes as their denominator, including nodes without a generated replacement. CARE-Mem exceeds the SFT parent, global-only and global-plus-local controls, and the variants without node, edge, or connected counterfactuals; removing the retention probe gives the highest point estimate.

Table 15: Complete per-run replacement results. Coverage and strict exact-match accuracy use the 139 reference REVISE nodes. The deterministic parent is evaluated once; dashes mark the unused replicate columns. Scores use the 0–1 scale; summary cells are mean ± sample SD.

Condition	Run 0	Run 1	Run 2	Mean ± SD
Coverage				
CARE-Mem	0.6547	0.6187	0.6187	$\underline{0.6307} \pm 0.0208$
SFT parent	0.5971	–	–	0.5971
Global-only (A1)	0.6187	0.3165	0.3094	0.4149 ± 0.1766
Global + local (A2)	0.6691	0.5540	0.5036	0.5755 ± 0.0848
Without node CF	0.6619	0.3453	0.3885	0.4652 ± 0.1717
Without edge CF	0.5755	0.3165	0.3597	0.4173 ± 0.1388
Without retention probe	0.6691	0.6547	0.6259	$\mathbf{0.6499} \pm 0.0220$
Without connected CF	0.5468	0.3525	0.3165	0.4053 ± 0.1238
Strict replacement exact-match accuracy				
CARE-Mem	0.5180	0.4820	0.4604	$\underline{0.4868} \pm 0.0291$
SFT parent	0.3957	–	–	0.3957
Global-only (A1)	0.4173	0.2158	0.2158	0.2830 ± 0.1163
Global + local (A2)	0.4676	0.3885	0.3669	0.4077 ± 0.0530
Without node CF	0.4388	0.2302	0.2518	0.3070 ± 0.1147
Without edge CF	0.3885	0.2446	0.2446	0.2926 ± 0.0831
Without retention probe	0.5180	0.4964	0.4964	$\mathbf{0.5036} \pm 0.0125$
Without connected CF	0.3741	0.2446	0.2302	0.2830 ± 0.0792

A.13 MEMORYAGENTBENCH ADAPTATION

For each public fact, the FactConsolidation runtime turns active same-key predecessors into nodes and adds candidate INVALIDATES edges. Its active-fact executor retains facts assigned KEEP, removes facts assigned INVALIDATE or UNCERTAIN, and inserts replacement facts for REVISE; it does not retain CAREBench version history. The trained adapters are evaluated on MemoryAgentBench without further parameter updates.

The evaluation uses frozen adapters, fixed public inputs, and the same answer-generation protocol across conditions; all 800 questions remain in the accuracy denominator. Private labels and aggregate accuracy are reserved for scoring and do not enter prompts, training, or checkpoint selection. Appendix A.5 describes the public-output-driven decoding amendment and complete reruns.

A.14 EXACT POLICY CONDITIONS AND EXTERNAL RUNTIME

Table 16 specifies the training changes behind the condition names. The global-plus-local control (A2) retains local credit and adds the standardized global advantage on decision spans. Thus, its decision advantage is $A_k^{\text{global}} + \widehat{\ell}_{k,d}$, whereas CARE-Mem uses $\widehat{\ell}_{k,d}$. Both retain global advantage on non-decision tokens. The comparison tests additive mixing against local override. These expressions define the advantages before the common edge-family scaling in Appendix A.3. That scaling applies to edge-token advantages in every policy-trained variant, including spans that fall back to global advantage.

Table 16: Policy conditions. Trained variants share the SFT parent, 16 cases, two epochs, and two optimizer steps. "Global" here denotes standardized proposal utility, not an external answer reward. All trained variants use the common edge-family scaling defined in Appendix A.3.

Condition	Change relative to CARE-Mem
CARE-Mem	Node, edge, and connected controls; local override on credited decision spans.
SFT parent	No policy updates; deterministic inference from checkpoint 48.
Global-only (A1)	Same probe/edit-cost utility; no counterfactual controls. Global advantage trains all proposal tokens.
Global + local (A2)	Local and global advantages are added on decision spans; both weights are one.
Without node CF (A3)	Node singleton controls disabled; connected controls are also omitted by the scheduler. Node spans fall back to global advantage.
Without edge CF (A4)	Edge singleton controls disabled; connected controls are also omitted. Edge spans fall back to global advantage.
Without retention probe (A5)	Retention removed from training utility; other utility terms and controls retained. Evaluation still records Retention.
Without connected CF (A6)	Connected controls disabled; node and edge singleton controls retained.

Connected controls require both node and edge singleton families, because their residual subtracts the corresponding singleton effects. A3 and A4 therefore remove a family together with its connected residuals; A6 isolates the removal of connected controls while retaining both singleton families. When local credit is unavailable for a span, the implementation retains its global advantage. The configuration overlays are in `code/configs/ablations/`; scheduling and token assignment are implemented in `counterfactual.py`, `rollouts.py`, and `losses.py` under `code/src/caremem/`.

The Qwen MemoryAgentBench runtime groups complete sentences with a cumulative 4,096-token budget, starting a new chunk when the next sentence would exceed the budget. It batches 32 independent conflict-revision problems, each producing one complete proposal, and reuses each of the four unique fact contexts for all associated questions. Proposal IDs are nonempty and bounded to 128 characters; replacement text is bounded to 256 characters. Unlike the CAREBench decoder, this external evaluation protocol permits replacement text containing multiple sentences. All 800 questions remain in the accuracy denominator. Candidate construction, active-fact execution, and protocol amendments are described in Appendices A.13 and A.5.

B Complete Stratified CAREBench Results

Tables 17–20 cover every category of conflict type, case shape, topic partition, and dependency depth for all eight matched conditions and capability-projected CUPMem. Each dimension separately partitions the same 240 cases, so the four analyses reuse cases rather than forming disjoint datasets. Metrics are computed within each category for each run, then averaged across runs; the parent is evaluated once.

The analyses are post hoc descriptive; inferential comparisons remain those in Tables 8–9. Macro-F1 uses four action labels. Joint exactness requires every action and gate to match; strict replacement uses reference Revise nodes, with N/A where that subset is empty. Counts are unique cases, not seed–case repetitions.

Table 17: Breakdown by conflict type on CAREBench-240 (post hoc descriptive). Scores use the 0–1 scale and are averaged over three runs; the parent is evaluated once. N counts unique cases. The highest and second-highest means in each category are bold and underlined, respectively, including ties. N/A denotes no reference REVISE nodes.

System	Action Macro-F1	Edge F1	Edge-set Exact	Joint action–edge exact	Strict replacement
Direct ($N = 60$)					
CARE-Mem	0.6958	<u>0.7882</u>	**0.4833**	<u>0.4167</u>	**0.5444**
SFT parent	0.6774	0.7632	0.4167	0.3667	0.5000
Global + local (A2)	<u>0.6962</u>	0.7603	0.3944	0.3556	0.4556
Global-only (A1)	0.6355	0.7449	0.3889	0.3611	0.3556
Without node CF	0.6339	0.7470	0.3944	0.3611	0.3778
Without edge CF	0.6578	0.7681	<u>0.4444</u>	**0.4222**	0.4222
Without retention probe	**0.7033**	**0.7883**	**0.4833**	0.3833	<u>0.5333</u>
Without connected CF	0.6527	0.7664	0.4333	0.4111	0.4222
CUPMem	0.2439	0.5279	0.3111	0.0611	0.0000
Gradual ($N = 60$)					
CARE-Mem	0.4384	0.4646	0.4778	0.4222	<u>0.3712</u>
SFT parent	0.4281	0.4776	0.4500	0.4000	0.2727
Global + local (A2)	<u>0.4568</u>	0.4731	<u>0.4889</u>	**0.4556**	0.3106
Global-only (A1)	0.3627	0.4706	0.4778	0.4056	0.1515
Without node CF	0.3909	0.5027	<u>0.4889</u>	0.4111	0.1894
Without edge CF	0.3725	**0.5613**	**0.5444**	<u>0.4389</u>	0.1439
Without retention probe	**0.4675**	0.4702	0.4444	0.4111	**0.4015**
Without connected CF	0.3381	<u>0.5290</u>	<u>0.4889</u>	0.3611	0.1212
CUPMem	0.2986	0.2698	0.3556	0.2056	0.0000
No conflict ($N = 60$)					
CARE-Mem	0.2632	0.2254	0.4500	0.4111	N/A
SFT parent	<u>0.2752</u>	0.1702	<u>0.4667</u>	<u>0.4333</u>	N/A
Global + local (A2)	**0.2950**	0.2205	0.4222	0.4056	N/A
Global-only (A1)	0.2517	0.1038	0.4278	0.4222	N/A
Without node CF	0.2518	0.1242	0.4056	0.3889	N/A
Without edge CF	0.2621	<u>0.2392</u>	**0.4722**	**0.4444**	N/A
Without retention probe	0.2742	0.2094	0.4000	0.3667	N/A
Without connected CF	0.2604	0.1962	0.4056	0.3778	N/A
CUPMem	0.2570	**0.4102**	0.4500	0.2278	N/A
Propagation ($N = 60$)					
CARE-Mem	0.6869	0.6581	<u>0.3500</u>	<u>0.2889</u>	<u>0.5385</u>
SFT parent	0.7089	<u>0.6640</u>	<u>0.3500</u>	0.2667	0.4308
Global + local (A2)	<u>0.7153</u>	0.6319	0.3111	0.2667	0.4513
Global-only (A1)	0.6380	0.6272	0.3167	0.2278	0.3385
Without node CF	0.6477	0.6225	0.3111	0.2333	0.3538
Without edge CF	0.6205	0.6377	0.3222	0.2222	0.3333
Without retention probe	**0.7253**	0.6601	**0.3778**	**0.3000**	**0.5590**
Without connected CF	0.6164	**0.6641**	0.3389	0.2333	0.3282
CUPMem	0.2387	0.3544	0.0111	0.0000	0.0000

Table 18: Breakdown by case shape on CAREBench-240 (post hoc descriptive). Scores use the 0–1 scale and are averaged over three runs; the parent is evaluated once. N counts unique cases. The highest and second-highest means in each category are bold and underlined, respectively, including ties. N/A denotes no reference REVISE nodes.

System	Action Macro-F1	Edge F1	Edge-set Exact	Joint action–edge exact	Strict replacement
Single action ($N = 60$)					
CARE-Mem	0.5045	<u>0.6753</u>	<u>0.5222</u>	0.4556	N/A
SFT parent	<u>0.5342</u>	0.6667	0.4833	0.4667	N/A
Global + local (A2)	**0.5497**	0.6725	0.4889	0.4722	N/A
Global-only (A1)	0.5222	0.6585	0.4889	0.4833	N/A
Without node CF	0.5210	0.6728	0.5056	<u>0.4944</u>	N/A
Without edge CF	0.4983	0.6704	0.5167	**0.5111**	N/A
Without retention probe	0.5189	**0.6892**	**0.5389**	0.4444	N/A
Without connected CF	0.5001	0.6508	0.4833	0.4778	N/A
CUPMem	0.2304	0.5845	0.4778	0.2111	N/A
Multi-action ($N = 75$)					
CARE-Mem	<u>0.5606</u>	<u>0.8030</u>	<u>0.4578</u>	**0.3822**	<u>0.4868</u>
SFT parent	0.5133	0.7915	0.4133	0.2933	0.3957
Global + local (A2)	0.5224	0.7731	0.3911	0.3156	0.4077
Global-only (A1)	0.4684	0.7660	0.3822	0.2400	0.2830
Without node CF	0.4789	0.7606	0.3733	0.2489	0.3070
Without edge CF	0.4818	0.8021	0.4444	0.2756	0.2926
Without retention probe	**0.5622**	0.7987	0.4489	<u>0.3689</u>	**0.5036**
Without connected CF	0.4733	**0.8205**	**0.4622**	0.2800	0.2830
CUPMem	0.1746	0.3370	0.0089	0.0000	0.0000
Mixed ($N = 105$)					
CARE-Mem	0.2984	0.1409	0.3810	0.3460	N/A
SFT parent	<u>0.3301</u>	0.1497	<u>0.3905</u>	<u>0.3619</u>	N/A
Global + local (A2)	**0.3341**	0.1513	0.3651	0.3524	N/A
Global-only (A1)	0.2993	0.0994	0.3683	<u>0.3619</u>	N/A
Without node CF	0.2998	0.1468	0.3587	0.3365	N/A
Without edge CF	0.2914	<u>0.1729</u>	**0.4063**	**0.3841**	N/A
Without retention probe	0.3244	0.1553	0.3460	0.3175	N/A
Without connected CF	0.2822	0.1452	0.3460	0.3175	N/A
CUPMem	0.2719	**0.3564**	0.3651	0.1619	N/A

Table 19: Breakdown by topic partition on CAREBench-240 (post hoc descriptive). Scores use the 0–1 scale and are averaged over three runs; the parent is evaluated once. N counts unique cases. The highest and second-highest means in each category are bold and underlined, respectively, including ties. N/A denotes no reference REVISE nodes.

System	Action Macro-F1	Edge F1	Edge-set Exact	Joint action–edge exact	Strict replacement
Qualification-covered topics ($N = 120$)					
CARE-Mem	0.6340	0.5496	0.4222	0.3722	<u>0.3744</u>
SFT parent	<u>0.6926</u>	<u>0.5795</u>	0.4333	0.4000	0.3562
Global + local (A2)	**0.6969**	0.5561	0.4278	**0.4056**	0.3562
Global-only (A1)	0.6375	0.5406	0.4167	0.4000	0.3105
Without node CF	0.6485	0.5648	0.4333	0.4000	0.3105
Without edge CF	0.6165	0.5662	<u>0.4389</u>	<u>0.4028</u>	0.3059
Without retention probe	0.6670	0.5518	0.4139	0.3722	**0.3790**
Without connected CF	0.5916	**0.5857**	**0.4444**	0.3917	0.3014
CUPMem	0.3261	0.4703	0.3194	0.1444	0.0000

Table 19 (continued)

System	Action Macro-F1	Edge F1	Edge-set Exact	Joint action–edge exact	Strict replacement
Qualification-uncovered topics ($N = 120$)					
CARE-Mem	0.6635	0.6611	**0.4583**	**0.3972**	0.6111
SFT parent	0.6012	0.6333	0.4083	0.3333	0.4394
Global + local (A2)	0.6297	0.6243	0.3806	0.3361	0.4646
Global-only (A1)	0.5189	0.6154	0.3889	0.3083	0.2525
Without node CF	0.5442	0.5993	0.3667	0.2972	0.3030
Without edge CF	0.5374	**0.6695**	0.4528	0.3611	0.2778
Without retention probe	**0.6674**	0.6500	0.4389	0.3583	**0.6414**
Without connected CF	0.5337	0.6362	0.3889	0.3000	0.2626
CUPMem	0.2284	0.2876	0.2444	0.1028	0.0000

Table 20: Breakdown by dependency depth on CAREBench-240 (post hoc descriptive). Scores use the 0–1 scale and are averaged over three runs; the parent is evaluated once. N counts unique cases. The highest and second-highest means in each category are bold and underlined, respectively, including ties. N/A denotes no reference REVISE nodes.

System	Action Macro-F1	Edge F1	Edge-set Exact	Joint action–edge exact	Strict replacement
Depth 0 ($N = 120$)					
CARE-Mem	0.4803	0.5207	0.4861	0.4333	N/A
SFT parent	0.5055	0.5132	0.4750	0.4500	N/A
Global + local (A2)	**0.5159**	0.5171	0.4556	0.4389	N/A
Global-only (A1)	0.4780	0.4850	0.4583	0.4528	N/A
Without node CF	0.4815	0.4887	0.4556	0.4417	N/A
Without edge CF	0.4639	**0.5281**	**0.4944**	**0.4778**	N/A
Without retention probe	0.4909	0.5151	0.4694	0.4056	N/A
Without connected CF	0.4610	0.4860	0.4444	0.4278	N/A
CUPMem	0.2460	0.5109	0.4639	0.2194	N/A
Depth 1 ($N = 50$)					
CARE-Mem	**0.4205**	**0.7653**	**0.5333**	**0.5067**	0.5423
SFT parent	0.3782	0.7111	0.4400	0.3400	0.4179
Global + local (A2)	0.3791	0.6888	0.4267	0.3733	0.4179
Global-only (A1)	0.3568	0.6708	0.4133	0.3400	0.3333
Without node CF	0.3510	0.6624	0.4000	0.3267	0.3333
Without edge CF	0.3784	0.7079	0.4733	0.4067	0.3582
Without retention probe	0.4193	0.7499	0.5200	0.4667	**0.5572**
Without connected CF	0.3751	0.6919	0.4333	0.3733	0.3582
CUPMem	0.2170	0.3920	0.2333	0.0667	0.0000
Depth 2 ($N = 55$)					
CARE-Mem	0.6023	0.6940	0.3758	0.2727	0.4352
SFT parent	0.6078	0.7200	0.4000	0.3091	0.3750
Global + local (A2)	**0.6364**	0.6994	0.3818	**0.3212**	0.3981
Global-only (A1)	0.5349	0.6984	0.3818	0.2485	0.2361
Without node CF	0.5650	0.7081	0.3879	0.2606	0.2824
Without edge CF	0.5280	0.7420	0.4364	0.2545	0.2315
Without retention probe	0.6335	0.6959	0.3636	0.2848	**0.4537**
Without connected CF	0.4938	**0.7561**	**0.4545**	0.2364	0.2130
CUPMem	0.2178	0.2929	0.0061	0.0000	0.0000

Table 20 (continued)

System	Action Macro-F1	Edge F1	Edge-set Exact	Joint action–edge exact	Strict replacement
Depth 3 ($N = 15$)					
CARE-Mem	0.2828	0.0000	**0.0000**	**0.0000**	N/A
SFT parent	**0.3338**	0.0000	**0.0000**	**0.0000**	N/A
Global + local (A2)	0.3289	0.0000	**0.0000**	**0.0000**	N/A
Global-only (A1)	0.2867	0.0000	**0.0000**	**0.0000**	N/A
Without node CF	0.2861	0.0000	**0.0000**	**0.0000**	N/A
Without edge CF	0.2756	0.0000	**0.0000**	**0.0000**	N/A
Without retention probe	0.3195	0.0000	**0.0000**	**0.0000**	N/A
Without connected CF	0.2756	0.0000	**0.0000**	**0.0000**	N/A
CUPMem	0.2391	**0.3653**	**0.0000**	**0.0000**	N/A

C PROBE SENSITIVITY AND CREDIT SOURCES

Table 21 decomposes score variation across 744 node, 816 edge, and 816 connected interventions reconstructed from the 192 CARE-Mem training rollouts. These are executions, distinct from the 1,440 decision spans: the 624 node decisions generate 744 controls because each UNCERTAIN node decision has two singleton controls. A change can increase or decrease a probe score. The percentages describe which utility terms vary, not the proportion of positive credit supplied by each term.

Table 21: Probe-score sensitivity in 2,376 reconstructed training interventions from 192 rollouts. Cells show the percentage of interventions with a score change in either direction. The cost-only column requires nonzero utility difference with no probe-score change; columns are not additive.

Control	N	Current	Retention	History	Any probe	Cost-only
Node	744	25.13	25.81	0.00	50.94	31.45
Edge	816	24.75	0.00	0.00	24.75	75.25
Connected	816	40.32	32.35	0.00	66.42	29.04

The History score changes in none of the 2,376 interventions. Retention scores change in node and connected controls but not edge controls. In 614 of 816 edge controls (75.25%), utility changes despite unchanged probe scores, so the difference comes from edit cost alone. These observations identify the utility terms contributing to edge credit.

Among the 816 connected interventions, 62 (7.60%) have residuals satisfying $|\Gamma_U| > |\sum_{d \in U} \Delta_d|$, using the absolute value of the sum of singleton effects. Node singleton credit uses the smaller of the two UNCERTAIN control deltas, as specified in Section 4.4.

Table 22: Probe scores after executing adjudicated reference proposals. Counts refer to the primary-action case categories. These are deterministic executor diagnostics, not model predictions.

Primary action	Cases	Current	Retention	History
KEEP	60	0.5000	0.4167	0.5000
INVALIDATE	45	0.0000	0.4889	0.4889
REVISE	60	0.5000	0.5000	0.5000
UNCERTAIN	75	1.0000	0.5067	0.5067
All cases	240	0.5625	0.4792	0.5000

Table 22 evaluates the adjudicated reference proposals with the existing executor and deterministic probe scorers. Scores below one even for reference proposals identify a mismatch between struc-

tured revision correctness and these probe readouts. The probes remain training utility components and descriptive execution diagnostics. Action, edge, and replacement endpoints separately evaluate the structured outputs.

D ACTION-CLASS RECALL

Table 23 supplies the class-level results introduced in Section 5. Recall is the fraction of reference nodes in an action class for which the model predicts that action; missing predictions remain in the denominator. The same 793 reference nodes are scored in every run. Recall is computed from the same predictions as Table 3 and summarized descriptively.

Table 23: Action-class recall for the matched policy conditions. Values are on the 0–1 scale, reported as mean ± sample SD across Runs 0–2 (seeds 42, 43, and 44); the SFT parent is evaluated once with deterministic decoding. Header counts are reference nodes per class, not seed–node repetitions. CF denotes counterfactual controls; definitions are in Table 16.

Condition	KEEP $N = 448$	INVALIDATE $N = 77$	REVISE $N = 139$	UNCERTAIN $N = 129$
CARE-Mem	0.8780 ± 0.0935	0.8442 ± 0.0130	0.6307 ± 0.0208	0.3359 ± 0.3416
SFT parent	0.8460	0.8961	0.5971	0.3721
Global-only (A1)	0.8765 ± 0.0287	0.9610 ± 0.0130	0.4149 ± 0.1766	0.2558 ± 0.0620
Global + local (A2)	0.8371 ± 0.0591	0.9394 ± 0.0198	0.5755 ± 0.0848	0.4341 ± 0.1527
Without node CF	0.8713 ± 0.0357	0.9524 ± 0.0270	0.4652 ± 0.1717	0.2610 ± 0.0792
Without edge CF	0.8802 ± 0.0742	0.9740 ± 0.0000	0.4173 ± 0.1388	0.2713 ± 0.2755
Without retention probe	0.8475 ± 0.0965	0.8268 ± 0.0270	0.6499 ± 0.0220	0.4419 ± 0.3035
Without connected CF	0.8713 ± 0.0819	0.9654 ± 0.0270	0.4053 ± 0.1238	0.2584 ± 0.2864