# AMORTIZING CAUSAL MEMORY VERIFICATION VIA EVIDENCE REUSE IN LONG-HORIZON LLM AGENTS

**Anonymous authors**
Paper under double-blind review

## ABSTRACT

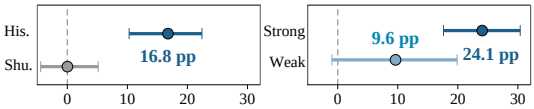
Long-horizon large language model (LLM)-based agents increasingly rely on external memory, but relevance-based retrieval does not establish whether memory exposure improves agent behavior. Causal Memory Intervention (CMI) provides a stronger behavioral criterion through matched interventions, but applying Full-CMI to every candidate incurs repeated verification costs. We study whether this cost can be amortized across sequential memory decisions. We propose the **C**ausal **E**valuation **R**euse **P**olicy (CERP), a selective verification policy that fits a rejector with history-conditioned features on an initial Full-CMI-evaluated prefix and applies the frozen policy to subsequent candidates, rejecting candidates before utility verification. On Causal-LoCoMo, CERP achieves 12.7% net call savings relative to Always-Verify after accounting for the initial Full-CMI cost, with a pooled rejection error rate of 8.8%, a 0.02 decrease in the extractive task proxy, and a 0.051 decrease in Useful-Memory F1. Matched random rejection and direct nearest-label reuse incur higher rejection error (16.3–23.7% vs. 8.8%). Budget and horizon analyses characterize when the initial verification cost is recovered.

## 1 INTRODUCTION

Long-horizon large language model (LLM)-based agents increasingly rely on external memory to reuse information and experience across interactions (Tan et al., 2025; Kang et al., 2025). Existing memory systems address how to store, organize, retrieve, and select experience (Du et al., 2026), yet relevance alone does not guarantee behavioral utility. Even highly similar retrieved experiences can propagate errors or provide limited value for subsequent decisions (Xiong et al., 2026). Therefore, reliable memory requires determining not only what experience is relevant, but also whether exposing that experience improves subsequent behavior (Ma et al., 2026).

Causal Memory Intervention (CMI) (Srivastava, 2026) provides such a criterion by comparing matched executions under no-memory, original-memory, and perturbed-memory conditions. We call the complete procedure Full-CMI. A candidate is retained when memory exposure improves the task outcome and the improvement remains stable under perturbation. Because Full-CMI requires multiple executions for each candidate, repeated application becomes increasingly costly as memory decisions accumulate over long trajectories.

A long-running agent repeatedly evaluates related memory candidates, while each completed evaluation contributes information about earlier decisions. Prior work shows that task-solving experience can transfer across tasks (Gao et al., 2025) and that task evaluations can signal the quality of stored experience (Xiong et al., 2026). This raises a different question from memory retrieval itself: whether information accumulated from earlier evaluations can reduce the need for repeated verification without incurring excessive selection error.




(a) Historical outcome correspondence (b) Historical match strength

Figure 1: Motivation diagnostic. (a) Historical Full-CMI outcomes (His.) are associated with a 16.8-pp difference in later rejection probability; outcome shuffling (Shu.) removes this separation. (b) The difference is 9.6 pp for weak matches and 24.1 pp for strong matches. Error bars denote clustered 95% confidence intervals.

***Can the cost of Full-CMI-based verification be amortized across sequential memory decisions?***



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 provides an initial diagnostic. Historical Full-CMI outcomes are associated with later Full-CMI decisions, with a 16.8-pp separation that disappears after outcome shuffling. This correspondence motivates testing whether evaluation history can support later verification decisions.

We propose the **C**ausal **E**valuation **R**euse **P**olicy (CERP), a history-conditioned selective verification policy. CERP acquires Full-CMI supervision from an initial prefix of candidates and uses the resulting evaluation history to identify later candidates that can be rejected before utility verification. The remaining candidates continue to utility verification, while Full-CMI serves as the reference criterion. On Causal-LoCoMo, CERP achieves $12.7\%$ net call savings relative to Always-Verify after accounting for the initial Full-CMI cost, with a pooled rejection error rate of $8.8\%$, a $0.02$ decrease in the extractive task proxy, and a $0.051$ decrease in Useful-Memory F1. Matched random rejection and direct nearest-label reuse incur higher rejection error ($16.3$-$23.7\%$). Utility-only supervision performs comparably on this benchmark, so the results do not establish an advantage of Full-CMI supervision itself. Further analyses show that amortization depends on the initial supervision budget, the reuse horizon, and the responder. Our contributions are as follows:

- **Problem formulation.** We formulate repeated Full-CMI verification as an amortization problem over sequential memory decisions, shifting focus from whether to verify a memory to whether repeated verification cost can be recovered over a long decision horizon. This makes the trade-off between initial supervision cost and later avoided verification explicit.
- **History-conditioned selective verification.** We propose CERP, a selective verification policy that uses an evaluated prefix and accumulated evaluation history to identify later candidates for rejection before utility verification. CERP preserves Full-CMI as the reference criterion while reducing later verification calls through history-conditioned routing.
- **Empirical characterization of amortization.** Across Causal-LoCoMo and CausalMemBench, we quantify the resulting efficiency-quality-error trade-off and examine how amortization varies with the initial supervision budget, reuse horizon, rejection operating point, and responder model. The analysis also distinguishes CERP from matched random rejection and direct label reuse, and shows that utility-only supervision can perform comparably to Full-CMI supervision in the evaluated setting.

## 2 Related Work

### 2.1 Long-Horizon Memory Selection and Causal Verification

Long-horizon LLM agents increasingly rely on external memory to preserve and reuse experience across interactions. Earlier systems (Park et al., 2023; Zhao et al., 2024) focused on persistent storage, organization, and retrieval, including MemoryBank (Zhong et al., 2024), RMM (Tan et al., 2025), A-MEM (Xu et al., 2026), and MemoryOS (Kang et al., 2025). More recent work shifts attention toward what information should be retained or retrieved. Nemori (Ma et al., 2026) models future utility for adaptive memory distillation, while MemGuide (Du et al., 2026) selects memories according to task intent rather than semantic similarity alone. Xiong et al. (2026) shows that retrieved experience can directly shape later behavior, including error propagation and misaligned experience replay. However, these methods improve memory construction and selection but do not estimate the behavioral contribution of an individual candidate through matched intervention. CMI proposed by Srivastava (2026) addresses this distinction by comparing no-memory, original-memory, and perturbed-memory executions. Therefore, CMI incurs repeated executions of the responder for each evaluated candidate. Our work studies whether the cost of repeated Full-CMI verification can be amortized across sequential memory decisions.

### 2.2 Selective Evaluation under Cost

Prior work examines the selective use of expensive evaluation or inference. Selective prediction and generation allocate additional effort according to uncertainty or confidence (Srinivasan et al., 2024; Lee et al., 2024; Geifman & El-Yaniv, 2019; Gupta et al., 2024), while cascaded evaluation escalates difficult cases to stronger evaluators (Jung et al., 2025). Experimental design methods reduce supervision costs by selecting informative examples (Bhatt et al., 2024), and RouteLLM (Ong et al., 2025) routes requests across models with different costs and capabilities. These approaches

allocate computation using signals from the current instance. CERP instead conditions on prior memory-evaluation history to amortize verification cost across a sequence of decisions.

# 3 PRELIMINARIES

## 3.1 LONG-HORIZON MEMORY-AUGMENTED AGENTS

We consider an LLM-based agent operating over a task sequence $\mathcal{X} = (x_1, \ldots, x_T)$ with a fixed policy $\pi(\cdot; \theta_{\text{agent}})$. Cross-task adaptation therefore relies on external memory rather than parameter updates. Before processing $x_t$, the agent has access to a memory bank $\mathcal{M}_t = \{m_1, \ldots, m_{N_t}\}$, where $N_t = |\mathcal{M}_t|$. A task-conditioned retriever selects a candidate subset $\mathcal{C}_t = \mathcal{R}(x_t, \mathcal{M}_t) = \{m_{t,1}, \ldots, m_{t,K_t}\} \subseteq \mathcal{M}_t$. The candidates are evaluated before memory exposure, yielding $\mathcal{S}_t \subseteq \mathcal{C}_t$, and the agent executes the current task as $\pi(x_t, \mathcal{S}_t; \theta_{\text{agent}})$. The resulting interaction may update the memory state for subsequent tasks, $\mathcal{M}_t \to \mathcal{M}_{t+1}$. Hence, the long-horizon memory pipeline considered in this work is $\mathcal{M}_t \xrightarrow{\mathcal{R}(x_t,\cdot)} \mathcal{C}_t \longrightarrow \mathcal{S}_t \longrightarrow \pi(x_t, \mathcal{S}_t; \theta_{\text{agent}}) \longrightarrow \mathcal{M}_{t+1}$.

## 3.2 CAUSAL MEMORY INTERVENTION

CMI evaluates a candidate $m \in \mathcal{C}_t$ by varying the memory context while keeping the task $x_t$ fixed. Using the policy defined above, the three intervention conditions are $Y_{\varnothing}(x_t) = \pi(x_t, \varnothing; \theta_{\text{agent}}), Y_m(x_t) = \pi(x_t, \{m\}; \theta_{\text{agent}}), Y_{\widetilde{m}}(x_t) = \pi(x_t, \{\widetilde{m}\}; \theta_{\text{agent}})$, corresponding to no memory, the original memory, and a perturbed version of the same memory. Given an evaluator $s(\cdot, x_t)$, CMI measures utility and stability by

$$\tau(m, x_t) = s(Y_m(x_t), x_t) - s(Y_{\varnothing}(x_t), x_t)\,, \tag{1}$$

$$\rho(m, x_t) = s(Y_m(x_t), x_t) - s(Y_{\widetilde{m}}(x_t), x_t)\,. \tag{2}$$

The utility effect $\tau$ measures improvement over the no-memory condition, while the stability effect $\rho$ tests whether this improvement depends on the original memory content. Full-CMI retains a candidate only when both conditions are satisfied, that is

$$y^I(m, x_t) = \mathbb{1}\left[\tau(m, x_t) > 0 \wedge \rho(m, x_t) \geq 0\right]. \tag{3}$$

Utility verification omits the perturbation condition and uses only

$$y^V(m, x_t) = \mathbb{1}\left[\tau(m, x_t) > 0\right]. \tag{4}$$

Hence, $y^I(m, x_t) = 1 \implies y^V(m, x_t) = 1$, while the two decisions differ when a candidate has positive utility, but fails the stability condition. The no-memory execution $Y_{\varnothing}(x_t)$ is shared across candidates for the same task. We therefore count only additional candidate-level responder calls. Full-CMI (action $F$) evaluates both $Y_m$ and $Y_{\widetilde{m}}$, utility verification $V$ evaluates only $Y_m$, and pre-verification rejection $R$ invokes neither: $F:\ c(F) = 2, V:\ c(V) = 1, R:\ c(R) = 0$.

# 4 METHOD

## 4.1 CERP FRAMEWORK

We index retrieved task-candidate pairs by their evaluation order as $\{(x_i, m_i)\}_{i=1}^{N}$, where $x_i$ denotes the task associated with the $i$-th candidate decision and $m_i$ the corresponding candidate memory. Figure 2 summarizes the CERP workflow, and Algorithm 1 gives the complete procedure.

For the first $B$ candidates, CERP constructs a pre-decision representation $\phi_i$ using only information available before evaluation and applies Full-CMI to obtain the retention label $y_i^I$. These pairs form the initial supervision set

$$\mathcal{D}_B = \left\{\left(\phi_i, y_i^I\right)\right\}_{i=1}^{B}.$$

CERP fits a selective verification model on $\mathcal{D}_B$ and selects a rejection threshold $\eta$ using the same initial supervision set. The model and threshold are then fixed.

For each subsequent candidate $i > B$, the frozen model estimates whether Full-CMI would reject the candidate. Candidates exceeding the rejection threshold are discarded without further verification; all others undergo utility verification and are retained according to $y_i^V$. Each paid evaluation updates the historical state used to construct later representations, while rejected candidates do not.

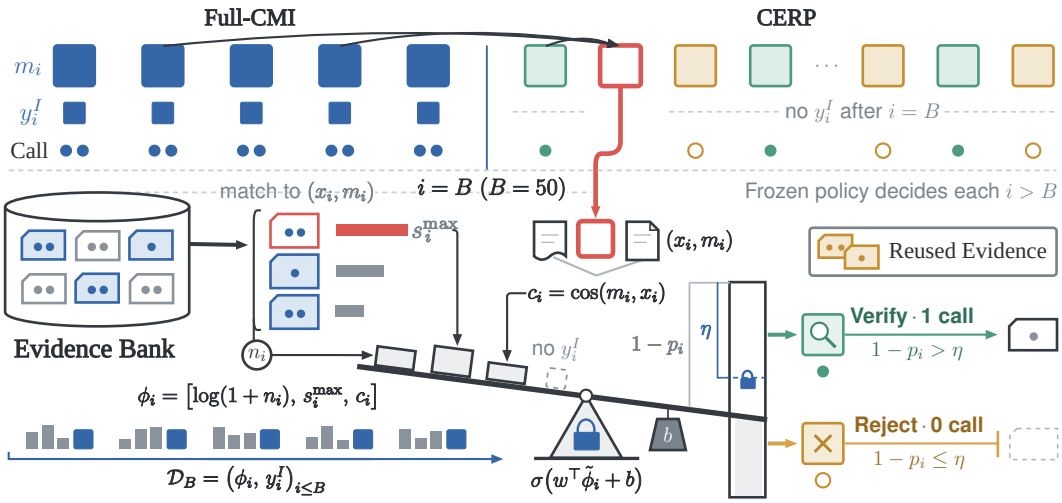



Figure 2: Overview of CERP. The first $B$ candidates undergo Full-CMI to form $\mathcal{D}_B = \{(\phi_i, y_i^I)\}_{i=1}^B$, where $\phi_i = [\log(1 + n_i), s_i^{\max}, c_i]$ summarizes historical support and candidate-task compatibility. A rejector estimates $p_i = \Pr(y_i^I = 1 \mid \phi_i)$, after which the model and rejection threshold $\eta$ are fixed. For $i > B$, candidates with $1 - p_i \ge \eta$ are rejected before verification; the rest undergo utility verification. Paid evaluations update the historical state used by later decisions.

### 4.2 History-Conditioned Rejection and Amortized Verification

For each candidate $i$, CERP constructs a pre-decision representation from the evaluation history available before the current candidate is assessed. Let $\mathcal{H}_{i-1}$ denote the historical state accumulated from previous paid evaluations. CERP summarizes the history relevant to the current task-candidate pair as below

$$\phi_i = [\log(1 + n_i),\ s_i^{\max},\ c_i]\,, \tag{5}$$

where $n_i$ measures the amount of historical support associated with candidate memory $m_i$, $s_i^{\max}$ is the strongest task match among the corresponding historical rules, and $c_i$ captures current candidate-task compatibility. All three quantities are computed before the current evaluation outcome is observed. Appendix B provides implementation details for historical-state construction and matching.

Using the first $B$ evaluated candidates, CERP fits an L2-regularized logistic regression rejector,

$$p_i = \Pr\left(y_i^I = 1 \mid \phi_i; \psi\right) = \sigma\left(\mathbf{w}^\top \tilde{\phi}_i + b\right), \tag{6}$$

where $\psi = (\mathbf{w}, b)$ and $p_i$ estimates the probability that Full-CMI would retain $m_i$. Full-CMI labels supervise model fitting but are not used as runtime input features or directly transferred to later candidates. For each subsequent candidate $i > B$, CERP applies

$$a_i = \begin{cases} R, & 1 - p_i \ge \eta, \\ V, & 1 - p_i < \eta, \end{cases} \tag{7}$$

where the actions $R$ and $V$ follow the definitions in Section 3.2. The threshold $\eta$ is selected on the initial supervision set $\mathcal{D}_B$ to maximize empirical rejection coverage subject to a training-side rejection-error target $\delta_{\text{train}}$. We use $\delta_{\text{train}} = 0.02$. This criterion governs threshold selection on $\mathcal{D}_B$ and does not bound rejection error for subsequent candidates. Appendix B provides details on threshold selection and fallback behavior. Let

$$N_R = \sum_{i=B+1}^{N} \mathbb{1}[a_i = R]$$

denote the number of subsequent candidates rejected before utility verification. Using the action costs defined in Section 3.2, CERP incurs

$$C_{\text{CERP}} = 2B + (N - B - N_R) = N + B - N_R. \tag{8}$$

Since Always-Verify incurs $N$ calls, the net saving is

$$N - C_{\mathrm{CERP}} = N_R - B.$$

Hence, the initial Full-CMI cost is recovered when $N_R \geq B$, and positive net savings require $N_R > B$. Amortization therefore depends on the initial supervision budget, the number of later candidates rejected before verification, and the remaining decision horizon.

# 5 EXPERIMENTS

We organize our experiments around four research questions (RQs). **RQ1**: How much evaluation cost can CERP save relative to Always-Verify, and what quality trade-off does this entail? **RQ2**: What information supports selective rejection, and can simpler reuse strategies reproduce its error-quality trade-off? **RQ3**: When does CERP recover its initial Full-CMI cost and achieve positive amortization? **RQ4**: How does amortization vary across LLM responders? Section 5.1 describes the experimental setup.

## 5.1 EXPERIMENTAL SETUP

**Benchmarks and replay protocol.** Our primary experiments use the Causal-LoCoMo intervention benchmark distributed with CMI. The evaluation pool contains 100 tasks from 10 conversations, with five candidate memories per task, yielding 500 candidate-level decisions. Sections 5.2-5.4 use offline sequential replay over fixed Full-CMI outcomes on Causal-LoCoMo. We construct 10 conversation-atomic orderings while preserving task and candidate order within each conversation. We additionally use CausalMemBench for additional cross-model evaluation. Full benchmark versions and evaluation details are provided in Appendix B.

**Compared methods.** We compare three evaluation strategies. **Full-CMI** applies the complete intervention procedure and serves as the reference criterion. **Always-Verify** applies utility verification to every candidate. **CERP** acquires Full-CMI supervision from an initial prefix and selectively rejects later candidates before utility verification. Unless otherwise specified, we use an initial supervision budget of $B = 50$ and select the rejection threshold on $\mathcal{D}_B$ with $\delta_{\mathrm{train}} = 0.02$. Full-CMI-supervised CERP is the primary setting because our central question concerns amortizing the cost of repeated Full-CMI verification. Section 5.3 separately examines utility-only supervision.

**Metrics and uncertainty.** We measure efficiency by nominal call reduction relative to Full-CMI and net savings relative to Always-Verify. We evaluate memory quality using Useful-Memory F1 and the deterministic extractive task-performance proxy, both averaged over tasks (Appendix B). The rejection error rate is

$$\mathrm{RER} = \frac{\sum_{i>B} \mathbb{1}[a_i = R \wedge y_i^I = 1]}{\sum_{i>B} \mathbb{1}[a_i = R]}, \tag{9}$$

pooled across replay orderings. Agreement with the Full-CMI reference is

$$\mathrm{Agreement} = \frac{1}{N} \sum_{i=1}^{N} \mathbb{1}[\hat{y}_i = y_i^I]. \tag{10}$$

We report the SD across ten replay orderings and paired 95% confidence intervals from a conversation-clustered bootstrap.

**Responders.** Section 5.5 instantiates the same amortization protocol with six LLM responders. Qwen2.5-7B-Instruct (Qwen2.5-7B), Llama-3.1-8B-Instruct (Llama-8B), and Mistral-7B-Instruct-v0.3 (Mistral-7B) are served locally, while Kimi-K3, DeepSeek-V4.1-Flash (D.S.-V4.1), and GLM-5.3 are accessed through cloud APIs. Responder-specific configurations are provided in Appendix B. We also evaluate the same six responder families on CausalMemBench in Appendix C.3.

## 5.2 AMORTIZED CAUSAL MEMORY VERIFICATION

To answer **RQ1**, we compare CERP with Full-CMI and Always-Verify on evaluation cost, rejection error, and memory quality. Table 1 summarizes the main results. For a 500-candidate stream, Full-CMI requires 1,000 nominal candidate-level calls, whereas Always-Verify requires 500.

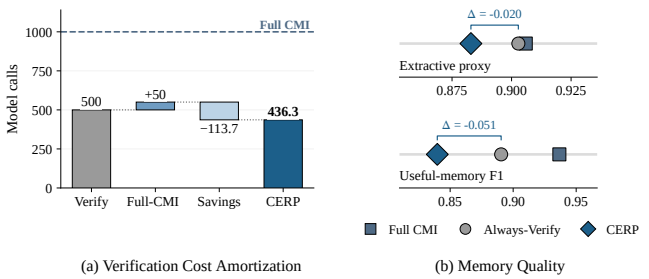
Table 1: Overall comparison of memory evaluation strategies. Nominal calls are reported per 500-candidate stream under the candidate-level cost model, and net savings are measured relative to Always-Verify after accounting for the initial Full-CMI cost. CERP values are MEAN±SD across ten replay orderings, while RER is pooled over all pre-verification rejections. The final row reports conversation-clustered 95% CIs: CERP−Always-Verify for calls, Proxy, F1, and Agreement, and absolute intervals for net savings and pooled RER.

| Strategy | Calls ↓ | Net savings ↑ | Proxy ↑ | Useful-Memory F1 ↑ | Agreement ↑ | RER (%) ↓ |
|---|---|---|---|---|---|---|
| Full-CMI | 1000 | – | 0.906 | 0.9367 | 1 | – |
| Always-Verify | 500 | 0% | 0.9029 | 0.8905 | 0.97 | – |
| CERP | 436.3±66.3 | 12.7% | 0.8829±0.0108 | 0.8398±0.0291 | 0.9548±0.0102 | 8.8 |
| 95% CI | $[-97.6, -28.1]$ | $[5.6, 19.5]$% | $[-0.029, -0.012]$ | $[-0.084, -0.020]$ | $[-0.026, -0.005]$ | $[5.8, 11.7]$ |

Table 2: Information-source ablations and simpler reuse baselines. Similarity-only uses current candidate-task compatibility, History-only uses historical matching state, and Full CERP combines both. Random rejection matches Full CERP's rejection coverage within each replay ordering and therefore has the same nominal cost. The 1-NN and 5-NN baselines directly reuse Full-CMI labels from the initial $B = 50$ evaluated candidates. RER is pooled across the ten replay orderings.

| Variant | Nominal Calls ↓ | Net savings ↑ | RER (%) ↓ | Useful-Memory F1 ↑ | Proxy ↑ |
|---|---|---|---|---|---|
| Similarity-only | 489.3 | 2.1% | 9.6 | 0.862 | 0.890 |
| History-only | 441.5 | 11.7% | 13.9 | 0.799 | 0.876 |
| Full CERP | 436.3 | 12.7% | 8.8 | 0.840 | 0.883 |
| Random rejection | 436.3 | 12.7% | 23.7 | 0.704 | 0.837 |
| Strongest-match label (1-NN) | 227.8 | 54.4% | 18.7 | 0.482 | 0.764 |
| 5-NN label vote | 397.9 | 20.4% | 16.3 | 0.739 | 0.853 |

CERP reduces the mean cost to $436.3 \pm 66.3$ calls, yielding 12.7% net call savings relative to Always-Verify (95% CI $[5.6, 19.5]$%; positive in 7/10 orderings). It also achieves an Agreement of $0.9548 \pm 0.0102$ with the Full-CMI reference and a Useful-Memory F1 of $0.8398 \pm 0.0291$.




(a) Verification Cost Amortization (b) Memory Quality

Figure 3: Amortizing Full-CMI-based verification cost (means over ten orderings). (a) Always-Verify cost (500 calls), the initial Full-CMI cost relative to Always-Verify (+50), avoided utility-verification calls from pre-verification rejections (−113.7), and CERP's total cost (436.3), yielding 63.7 net calls saved per stream. (b) Efficiency-quality trade-off among Full-CMI, Always-Verify, and CERP; Δ denotes CERP minus Always-Verify.

Figure 3(b) shows the resulting efficiency-quality trade-off. Relative to Always-Verify, CERP decreases the extractive task proxy by 0.020 (95% CI $[0.012, 0.029]$) and Useful-Memory F1 by 0.051 (95% CI $[0.020, 0.084]$). At $B = 50$, the initial Full-CMI cost adds 50 calls relative to Always-Verify, while pre-verification rejection avoids 113.7 utility-verification calls on average, leaving 63.7 net calls saved per 500-candidate stream (Figure 3(a)).

### 5.3 Information Sources and Simpler Reuse Baselines

To answer **RQ2**, we examine which information supports selective rejection and whether simpler reuse strategies reproduce CERP's error-quality trade-off. We first ablate the two components of its deployed representation: historical matching state and current candidate-task compatibility. **Similarity-only** uses only candidate-task compatibility, **History-only** uses only historical matching state, and **Full CERP** combines both. We additionally compare against **Random rejection**, matched to Full CERP's rejection coverage within each replay ordering, and two baselines that directly reuse Full-CMI labels from the initial evaluated prefix.

Table 3: Effect of the initial Full-CMI supervision budget $B$. Calls and net calls saved are means per 500-candidate ordering; Positive orderings counts orderings with net savings $> 0$, and RER is pooled across orderings. † denotes the primary setting fixed independently of this sweep.

| $B$ | Mean calls | Mean net saved | Net savings (%) | Positive orderings | RER (%) | F1 | Proxy |
|---|---|---|---|---|---|---|---|
| 25 | 351.2 | 148.8 | 29.8 | 9/10 | 12.7 | 0.7563 | 0.8534 |
| $50^{\dagger}$ | **436.3** | **63.7** | **12.7** | **7/10** | **8.8** | **0.8398** | **0.8829** |
| 75 | 496.2 | 3.8 | 0.8 | 5/10 | 8.1 | 0.8590 | 0.8906 |
| 100 | 552.4 | −52.4 | −10.5 | 1/10 | 8.6 | 0.8722 | 0.8941 |

Table 2 shows that the two information sources play different roles. Similarity-only yields $2.1\%$ net savings with an RER of $9.6\%$, whereas History-only increases net savings to $11.7\%$ but also raises RER to $13.9\%$ and reduces Useful-Memory F1 to $0.799$. Combining both sources increases net savings to $12.7\%$ while reducing RER to $8.8\%$ and improving F1 to $0.840$ relative to History-only. Thus, historical matching state provides most of the rejection coverage, while current candidate-task compatibility helps avoid erroneous rejections.

Random rejection isolates candidate selection from rejection volume. Because both methods match Full CERP's rejection coverage within each ordering, they incur the same nominal cost. However, random rejection raises RER from $8.8\%$ to $23.7\%$ and reduces Useful-Memory F1 from $0.840$ to $0.704$. The savings of CERP therefore depend not only on rejecting candidates, but on identifying which candidates can be rejected with lower error.

Direct label reuse does not reproduce this trade-off. Reusing the label of the strongest historical match (1-NN) yields $54.4\%$ net savings but an RER of $18.7\%$ and F1 of $0.482$. A 5-NN label vote reduces net savings to $20.4\%$ but still yields an RER of $16.3\%$ and F1 of $0.739$. Both baselines exhibit higher rejection error and lower memory quality than Full CERP, showing that direct label transfer does not reproduce CERP's observed error–quality trade-off.

### 5.4 Amortization Conditions and Error-Quality Trade-offs

To answer **RQ3**, we examine when CERP recovers its initial Full-CMI cost and achieves positive net savings. From equation 8, positive amortization requires $N_R > B$: the number of later utility-verification calls avoided by early rejection must exceed the additional cost of the initial Full-CMI evaluations. We therefore study how this condition is realized under different initial supervision budgets $B$, reuse horizons $N$, and rejection operating points.

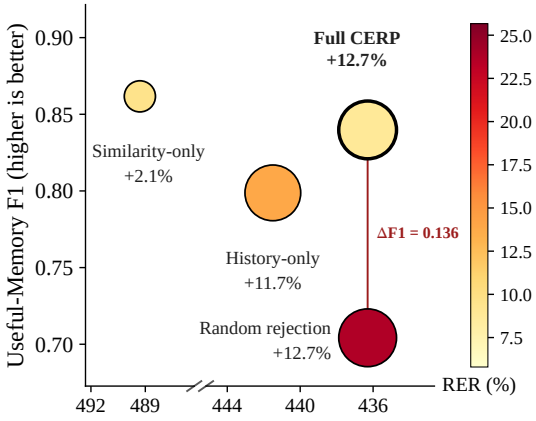



Figure 4: Efficiency, quality, and rejection error under different information sources and matched random rejection.

**Initial supervision budget.** Table 3 reports the cost-error-quality trade-off across four initial supervision budgets. Smaller budgets recover their initial Full-CMI cost more readily but are associated with lower memory-selection quality: $B = 25$ yields $29.8\%$ mean net savings and positive amortization in 9/10 orderings, whereas the primary setting $B = 50$ yields $12.7\%$ net savings with a lower RER of $8.8\%$ and higher Useful-Memory F1. As the tested budget increases further, net savings approach break-even at $B = 75$ and become negative at $B = 100$, although the latter attains the highest F1 and proxy among the tested settings.

Figure 5 connects these results to the break-even condition in Eq. (8). Positive amortization requires $N_R > B$, or equivalently $N_R/N > B/N$. The observed rejection fraction exceeds this requirement on average for $B = 25, 50, 75$, but falls below it at $B = 100$, matching the transition from positive to negative mean net savings. Across the same tested budgets, the rejection fraction decreases while the mean extractive proxy increases, illustrating the observed efficiency-quality trade-off.

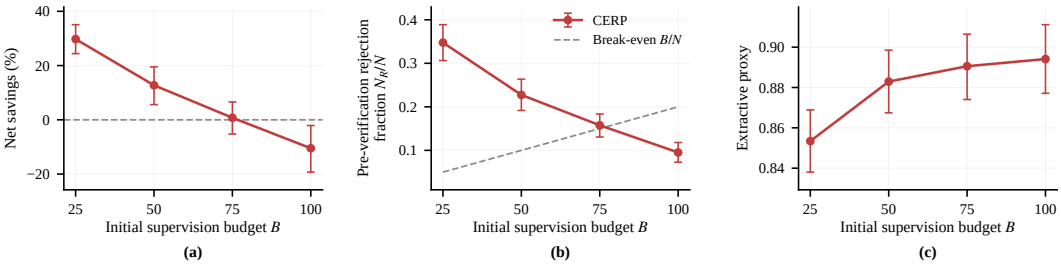



Figure 5: Effect of the initial supervision budget $B$ on CERP. (a) Net call savings relative to Always-Verify. (b) Pre-verification rejection fraction $N_R/N$, with $B/N$ denoting the break-even requirement for positive amortization. (c) Extractive task-performance proxy. Points report means over ten replay orderings.

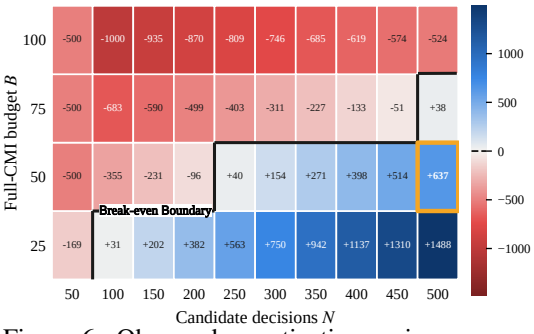



Figure 6: Observed amortization regimes across initial supervision budgets $B$ and candidate horizons $N$. Cell values report net calls saved relative to Always-Verify, pooled over 10 orderings; dividing by 10 yields the mean per ordering. When $N \leq B$, all candidates lie within the initial Full-CMI prefix, and no subsequent selective verification occurs.

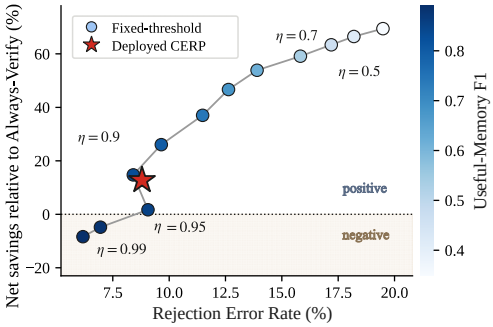



Figure 7: Efficiency-error trade-off across fixed rejection thresholds. The star marks the primary operating point selected from $\mathcal{D}_B$ separately for each ordering (12.7% net savings and 8.8% RER), and the horizontal line marks zero net savings.

**Reuse horizon.** Figure 6 jointly varies the initial supervision budget $B$ and candidate horizon $N$ to examine when the upfront Full-CMI cost is recovered. With $B = 25$, pooled savings become positive by $N = 100$ and increase to 1,488 calls, or 148.8 per ordering, at $N = 500$. With $B = 50$, savings remain negative at $N = 200$, turn positive at $N = 250$, and reach 637 calls, or 63.7 per ordering, at $N = 500$. For $B = 75$, positive amortization appears only at $N = 500$, while $B = 100$ remains net negative throughout the observed horizon. These results show that longer decision horizons provide more opportunities to recover the initial Full-CMI cost through later rejections, whereas larger supervision budgets require longer horizons to break even.

**Rejection operating point.** Figure 7 evaluates fixed rejection thresholds and traces the resulting efficiency-error-quality trade-off. At $\eta = 0.99$, pooled RER is 6.2%, but net savings remain negative because few utility-verification calls are avoided. Lower thresholds produce more early rejections and higher observed savings, but increase RER and reduce Useful-Memory F1. The star marks the primary CERP operating point, whose threshold is selected separately for each replay ordering from the initial supervision set $\mathcal{D}_B$ using $\delta_{\text{train}} = 0.02$. It achieves a mean net savings of 12.7% with a pooled RER of 8.8%; subsequent outcomes are not used to select this operating point.

Across these analyses, amortization is governed by a simple balance: the initial Full-CMI cost must be offset by a sufficient number of later pre-verification rejections. The supervision budget determines the cost to be recovered, the horizon determines how many later decisions are available for recovery, and the rejection operating point governs the observed trade-off between efficiency, error, and quality.

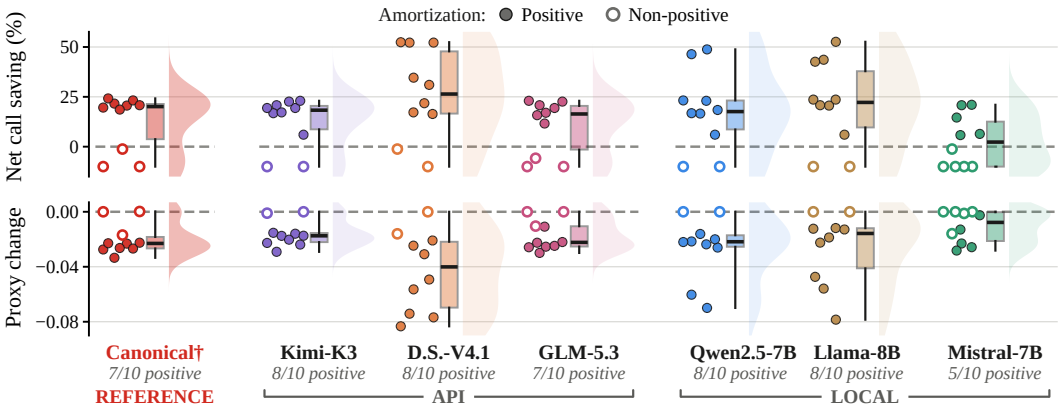



Figure 8: CERP across responder models on Causal-LoCoMo. Six LLM responders are compared with the deterministic replay reference (†, no LLM responder). Net savings and proxy changes are measured relative to Always-Verify. Llama-8B denotes Llama-3.1-8B; D.S.-V4.1 denotes DeepSeek-V4.1-Flash.

### 5.5 Amortization Across Responder Models

To answer **RQ4**, we apply the same CERP protocol to the six LLM responders described in Section 5.1. Mean net savings are positive for all six responders on Causal-LoCoMo, ranging from $2.7\%$ to $26.7\%$, as shown in Figure 8. However, the magnitude and consistency of amortization vary across responders: positive amortization occurs in only 5–8 of the 10 replay orderings. Proxy changes are negative for all six models, ranging from $-0.011$ for Mistral-7B to $-0.043$ for DeepSeek-V4.1-Flash (D.S.-V4.1). Thus, positive mean savings do not imply that the initial Full-CMI cost is recovered consistently across evaluation histories. The same broad pattern appears on the CausalMemBench cross-model subset (Appendix C.3), where mean net savings range from $19.1\%$ to $54.9\%$ across the six responders, with positive amortization in 4/10 to 10/10 orderings. These results provide additional evidence that CERP's amortization behavior is responder-dependent: the method reduces mean verification cost across the evaluated responder families, but the magnitude and ordering-level consistency of the savings are not responder-invariant.

## 6 Conclusion

We study whether the cost of repeated Full-CMI-based memory verification can be amortized across sequential memory decisions. We introduce CERP, a history-conditioned selective verification policy that learns from an initial Full-CMI-evaluated prefix and rejects selected later candidates before utility verification. On Causal-LoCoMo, CERP achieves $12.7\%$ net call savings relative to Always-Verify, with a pooled rejection error rate of $8.8\%$, a $0.020$ decrease in the extractive task proxy, and a $0.051$ decrease in Useful-Memory F1. Matched random rejection and direct nearest-label reuse yield substantially higher rejection error, while budget, horizon, and operating-point analyses show that positive amortization depends on whether later avoided verification calls recover the initial Full-CMI cost. Across six LLM responders on Causal-LoCoMo, mean net savings remain positive, although positive amortization occurs in only 5-8 of the 10 replay orderings, depending on the responder. CausalMemBench provides additional cross-benchmark evidence of the same amortization phenomenon. Utility-only supervision performs comparably to Full-CMI supervision on Causal-LoCoMo, so the current results do not establish a unique advantage of Full-CMI labels for learning the rejection policy. The study is limited to offline sequential replay, a deterministic downstream proxy, six responder families, and two benchmark settings, and its empirically selected rejection threshold provides no guarantee of subsequent rejection error. Future work can test whether amortized verification extends to broader agent architectures, longer deployment horizons, and alternative memory-verification procedures.

### AI USE STATEMENT

Generative AI tools were used to assist with analysis-code implementation, figure editing, and manuscript drafting and revision. They also assisted with running analysis scripts over fixed benchmark outcomes, including baseline comparisons and statistical uncertainty estimation. Generative AI was not used to create benchmark instances or to modify experimental responder outputs outside the evaluated LLMs' role as responder models. The authors reviewed all code, experimental results, figures, and manuscript text and take full responsibility for the content of this paper.

### ETHICS STATEMENT

This work evaluates memory-verification efficiency using Causal-LoCoMo and CausalMemBench resources distributed with CMI (Srivastava, 2026). We collect no new human-subject data, conduct no user study, and release no additional user data. The primary risk introduced by CERP is erroneous pre-verification rejection: a candidate that Full-CMI would retain may instead be rejected without further verification. We quantify such errors using the rejection error rate (RER) together with downstream quality measures. Because the rejection threshold is selected empirically from a limited initial supervision set, the observed rejection error does not constitute a formal guarantee for subsequent decisions. Applying CERP under different memory environments, responder models, or task distributions therefore requires re-evaluating the operating point and its associated rejection error.

### REPRODUCIBILITY STATEMENT

The offline replay experiments in Sections 5.2–5.4 use fixed Causal-LoCoMo intervention outcomes and require no responder generation during replay. The CERP rejector and rejection threshold are fixed after the initial Full-CMI prefix. The historical state may continue to evolve through subsequent paid evaluations, but no additional Full-CMI supervision or model updating is performed after the initial prefix. Threshold selection uses the empirical rejection-error target $\delta_{\mathrm{train}} = 0.02$; historical-state construction, matching, threshold selection, and metric definitions are described in Section 4 and Appendix B.

Responder experiments evaluate six LLM responders under local and API-based inference settings, with additional experiments on CausalMemBench. The experiment package records the corresponding model, prompting, decoding, and inference configurations. Appendix B provides the experimental and statistical protocols, and Appendix C reports additional diagnostics and cross-benchmark analyses.

## REFERENCES

Gantavya Bhatt, Yifang Chen, Arnav Das, Jifan Zhang, Sang Truong, Stephen Mussmann, Yinglun Zhu, Jeff Bilmes, Simon Du, Kevin Jamieson, et al. An experimental design framework for label-efficient supervised finetuning of large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 6549–6560, 2024.

Yiming Du, Bingbing Wang, Yang He, Bin Liang, Baojun Wang, Zhongyang Li, Lin Gui, Jeff Z Pan, Ruifeng Xu, and Kam-Fai Wong. MemGuide: Intent-driven memory selection for goal-oriented multi-session LLM agents. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 30584–30592, 2026.

Jinglong Gao, Xiao Ding, Lingxiao Zou, Bibo Cai, Bing Qin, and Ting Liu. ExpeTrans: LLMs are experiential transfer learners. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10577–10616, 2025.

Yonatan Geifman and Ran El-Yaniv. Selectivenet: A deep neural network with an integrated reject option. In *International conference on machine learning*, pp. 2151–2159. PMLR, 2019.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond. In *International Conference on Learning Representations*, volume 2024, pp. 4147–4180, 2024.

Jaehun Jung, Faeze Brahman, and Yejin Choi. Trust or escalate: LLM judges with provable guarantees for human agreement. In *International Conference on Learning Representations*, volume 2025, pp. 3101–3125, 2025.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25972–25981, 2025.

Minjae Lee, Kyungmin Kim, Taesoo Kim, and Sangdon Park. Selective generation for controllable language models. *Advances in Neural Information Processing Systems*, 37:50494–50527, 2024.

Wenquan Ma, Jiayan Nan, and Wenlong Wu. What deserves memory: Adaptive memory distillation for LLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 34789–34812, 2026.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E Gonzalez, Mohammed Kadous, and Ion Stoica. RouteLLM: Learning to route LLMs from preference data. In *International Conference on Learning Representations*, volume 2025, pp. 34433–34448, 2025.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pp. 1–22, 2023.

Tejas Srinivasan, Jack Hessel, Tanmay Gupta, Bill Yuchen Lin, Yejin Choi, Jesse Thomason, and Khyathi Chandu. Selective "selective prediction": Reducing unnecessary abstention in vision-language reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 12935–12948, 2024.

Saksham Sahai Srivastava. Causal intervention-based memory selection for long-horizon LLM agents. *arXiv preprint arXiv:2605.17641*, 2026.

Zhen Tan, Jun Yan, I-Hung Hsu, Rujun Han, Zifeng Wang, Long Le, Yiwen Song, Yanfei Chen, Hamid Palangi, George Lee, et al. In prospect and retrospect: Reflective memory management for long-term personalized dialogue agents. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 8416–8439, 2025.

Zidi Xiong, Yuping Lin, Wenya Xie, Pengfei He, Zirui Liu, Jiliang Tang, Himabindu Lakkaraju, and Zhen Xiang. How memory management impacts LLM agents: An empirical study of experience-following behavior. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 623–645, 2026.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-MEM: Agentic memory for LLM agents. *Advances in Neural Information Processing Systems*, 38:17577–17604, 2026.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. ExpeL: LLM agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 19632–19642, 2024.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. MemoryBank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 19724–19731, 2024.

**Algorithm 1** CERP: Causal Evaluation Reuse Policy for Amortized Memory Verification

**Require:** Ordered candidates $\{(x_i, m_i)\}_{i=1}^{N}$, initial supervision budget $B$, empirical rejection-error target $\delta_{\text{train}}$
**Ensure:** Retained set $\mathcal{S}$

Initialize historical state $\mathcal{H} \leftarrow \varnothing$, supervision set $\mathcal{D} \leftarrow \varnothing$, and retained set $\mathcal{S} \leftarrow \varnothing$
**for** $i = 1, \ldots, B$ **do**
  $\phi_i \leftarrow \textsc{ConstructFeatures}(x_i, m_i, \mathcal{H})$
  $y_i^I \leftarrow \textsc{FullCMI}(x_i, m_i)$
  $\mathcal{D} \leftarrow \mathcal{D} \cup \{(\phi_i, y_i^I)\}$
  **if** $y_i^I = 1$ **then**
    $\mathcal{S} \leftarrow \mathcal{S} \cup \{m_i\}$
  **end if**
  $\mathcal{H} \leftarrow \textsc{UpdateHistory}(\mathcal{H}, x_i, m_i, y_i^I)$
**end for**
$(\psi, \eta) \leftarrow \textsc{TrainRejectionPolicy}(\mathcal{D}, \delta_{\text{train}})$
**for** $i = B + 1, \ldots, N$ **do**
  $\phi_i \leftarrow \textsc{ConstructFeatures}(x_i, m_i, \mathcal{H})$
  $p_i \leftarrow \Pr(y_i^I = 1 \mid \phi_i; \psi)$
  **if** $1 - p_i \geq \eta$ **then**
    Reject $m_i$ without verification
  **else**
    $y_i^V \leftarrow \textsc{UtilityVerify}(x_i, m_i)$
    **if** $y_i^V = 1$ **then**
      $\mathcal{S} \leftarrow \mathcal{S} \cup \{m_i\}$
    **end if**
    $\mathcal{H} \leftarrow \textsc{UpdateHistory}(\mathcal{H}, x_i, m_i, y_i^V)$
  **end if**
**end for**
**return** $\mathcal{S}$

# A CERP Algorithm

Algorithm 1 summarizes the CERP procedure described in Section 4.1. For the first $B$ candidates, CERP constructs the pre-decision representation $\phi_i$ from the available historical state and applies Full-CMI to obtain the reference retention label $y_i^I$. The resulting pairs form the initial supervision set used to fit the rejection model and select the threshold $\eta$. The fitted model and threshold are then fixed. For each subsequent candidate, CERP constructs $\phi_i$, estimates the Full-CMI retention probability $p_i$, and applies the rejection rule in Eq. (7). Candidates not rejected before verification undergo utility verification. Only evaluated candidates update the historical state; rejected candidates contribute no additional feedback. Historical-state construction and threshold-selection details are provided in Appendix B.

# B Supplementary Experimental Details

**Benchmarks and evaluation pools.** Our primary experiments use the Causal-LoCoMo intervention benchmark distributed with CMI (Srivastava, 2026). We use the 100-task release, comprising 10 conversations and five candidate memories per task, for a total of 500 candidate-level decisions. For the offline replay experiments in Sections 5.2–5.4, Full-CMI outcomes are precomputed for all candidates using the benchmark evaluator and remain fixed across policies. We additionally evaluate CERP on CausalMemBench, the template-based benchmark generator distributed with CMI. The full generated pool contains 1,000 tasks with seven candidate memories per task. For cross-model evaluation, we use a fixed subset of 200 tasks spanning eight task families, with 25 tasks per family, yielding 1,400 candidate-level decisions per stream. The same subset is used across responders.

**Replay protocol.** For Causal-LoCoMo, we construct ten replay orderings by permuting conversations while preserving task and candidate order within each conversation. For CausalMemBench,

Table 4: Responder configurations used in the cross-model experiments.

| Model | Serving | Decoding |
|---|---|---|
| Qwen2.5-7B-Instruct | Local | $T = 0$ |
| Llama-3.1-8B-Instruct | Local | $T = 0$ |
| Mistral-7B-Instruct-v0.3 | Local | $T = 0$ |
| Kimi-K3 | API | $T = 0$ |
| DeepSeek-V4.1-Flash | API | $T = 0$ |
| GLM-5.3 | API | $T = 0$ |

the 200 tasks are independently permuted while preserving the seven candidates within each task. These orderings vary the historical state available to CERP without changing the underlying candidate pool. The policy observes only feedback revealed by the action it selects. Full-CMI labels are unavailable unless Full-CMI is executed, and utility verification does not reveal additional Full-CMI labels. The rejection model and threshold remain fixed after the initial Full-CMI prefix.

**CERP implementation.** CERP uses the representation defined in Section 4 and an $L_2$-regularized logistic regression rejector. Unless otherwise specified, the initial supervision budget is $B = 50$. The first $B$ Full-CMI evaluations provide the feature–label pairs used to fit the rejector and select its rejection threshold. Candidate–task and historical matching scores are computed using the same fixed deterministic representation across all experiments. After the initial prefix, the fitted rejector and threshold remain fixed: no additional Full-CMI supervision or model updating is performed. The historical state, however, continues to evolve through subsequent paid evaluations and is used to construct the pre-decision representation for later candidates. This separates adaptation of the evaluation history from adaptation of the rejection policy itself.

**Historical-state construction and matching.** The historical state contains one record for each candidate that undergoes Full-CMI or utility verification; candidates rejected before verification do not update the state. For a current candidate, CERP first links it to the corresponding previously evaluated memory when available and otherwise to the most similar historical memory under the fixed representation. The feature $n_i$ counts prior evaluations associated with the linked memory, while $s_i^{\max}$ records the strongest task match among the corresponding historical records. Together, these quantities summarize the amount and strength of historical support available before the current decision. Stored outcome values are not included in the runtime representation $\phi_i$; Full-CMI labels affect later decisions through the initially fitted rejector rather than through direct label transfer. Thus, later decisions may depend on the accumulated evaluation history without accessing unavailable Full-CMI labels for previously utility-verified candidates.

**Threshold selection.** For the initial supervision set $\mathcal{D}_B$, CERP selects the threshold $\eta$ that maximizes empirical rejection coverage subject to an empirical rejection error no greater than $\delta_{\text{train}} = 0.02$. Candidate thresholds must reject at least five examples, preventing the operating point from being determined by only a few rejected observations. If no threshold satisfies these conditions, CERP uses a no-rejection fallback and routes all subsequent candidates to utility verification. Once selected, the threshold remains fixed together with the rejector and is not revised using subsequent outcomes. The constraint is used only to select an empirical operating point on the initial supervision set and does not provide a rejection-error guarantee for later candidates.

**Responder-model evaluation.** We evaluate six LLM responders: Qwen2.5-7B, Llama-3.1-8B, Mistral-7B, Kimi-K3, DeepSeek-V4.1-Flash, and GLM-5.3. The first three are evaluated locally, and the latter three through cloud APIs. All responders use the same prompting protocol and deterministic decoding ($T = 0$). Table 4 summarizes the configurations used in the experiments.

For CausalMemBench, tasks yielding no usable responder output are excluded at the task level using a fixed screening rule. This removes 39 tasks for DeepSeek-V4.1-Flash and 4 tasks for Kimi-K3; no such exclusions occur on Causal-LoCoMo. All reported cross-model results use the resulting fixed evaluation sets for the corresponding responder.

**Prompt.** All responders receive the same user-level prompt:

> You are a careful assistant. Complete the current task using only memories that are relevant and reliable. If the necessary information is unavailable, say so.
> Retrieved memories:
> {memory_1}
> {memory_2}
> Current task:
> {task}
> Return only the final response.

**Evaluation metrics.** Efficiency follows the cost model in Section 3: Full-CMI costs two nominal evaluation calls, utility verification one, and pre-verification rejection zero. Call reduction is measured relative to Full-CMI, whereas net savings are measured relative to Always-Verify after accounting for the initial Full-CMI cost. RER and Agreement follow the definitions in Section 5.1.

*Useful-Memory F1.* For task $t$, let $G_t$ denote the benchmark-annotated set of useful memories and $S_t$ the set retained by the evaluated policy. We compute

$$P_t = \frac{|S_t \cap G_t|}{|S_t|}, \qquad R_t = \frac{|S_t \cap G_t|}{|G_t|}, \qquad F1_t = \frac{2P_tR_t}{P_t + R_t}, \tag{11}$$

with the usual empty-set conventions. Useful-Memory F1 therefore measures whether the policy retains the memories annotated as useful while avoiding irrelevant candidates. We compute F1 independently for each task and macro-average it over tasks, followed by averaging across replay orderings.

*Extractive task-performance proxy.* For each task, we construct a proxy response directly from the memories retained by the policy by concatenating their text in retrieval order. No LLM generation is used. This response is scored against the task's benchmark criteria using the same deterministic evaluator supplied with Causal-LoCoMo. The resulting task score measures how much task-relevant information remains available after memory selection, without introducing additional responder variability. Scores are averaged over tasks and then over replay orderings.

CausalMemBench uses the same definitions of evaluation cost, net savings, pre-verification rejection fraction, and RER. Because responder outputs are available in its cross-model experiments, task-quality change is computed as CERP minus Always-Verify within each responder.

**Statistical analysis.** The ten replay orderings share the same underlying candidate pool and are therefore used to characterize ordering sensitivity rather than treated as independent samples. Reported SDs are computed across replay orderings. For Causal-LoCoMo, paired uncertainty is estimated with a conversation-clustered bootstrap, with conversations as the resampling units. For CausalMemBench, each generated example is treated as an independent bootstrap cluster. RER is recomputed from the pooled rejection error and rejection counts within each bootstrap resample. Unless otherwise specified, confidence intervals are based on 10,000 bootstrap iterations.

# C Additional Experiments

## C.1 Motivation diagnostic details

Figure 1 uses only Full-CMI outcomes from the deployed $B = 50$ bootstrap. For each post-bootstrap candidate $i$, we identify its strongest-matched bootstrap candidate $j$ and measure

$$\Delta_{\text{hist}} = P(y_i^I = 0 \mid y_j^I = 0) - P(y_i^I = 0 \mid y_j^I = 1), \tag{12}$$

where $j$ is restricted to the first $B = 50$ decisions. The current Full-CMI label $y_i^I$ serves only as an offline diagnostic target and is never used as an online CERP feature.

Table 5 reports the detailed statistics. Across 4,500 post-bootstrap candidate-ordering observations (450 candidates per ordering across 10 replay orderings), $\Delta_{\text{hist}} = 0.1676$, with a conversation-clustered 95% confidence interval of $[0.1029, 0.2242]$. The separation is $0.0963$ under weak historical match (95% CI $[-0.0097, 0.1988]$) and $0.2406$ under strong match. A within-order permutation of bootstrap labels reduces the mean separation to $0.0002$, with a 95% permutation range of $[-0.0444, 0.0513]$ and a two-sided permutation $p = 0.0005$.

Table 5: Detailed statistics for the motivation diagnostic in Figure 1. Observed intervals are conversation-clustered 95% confidence intervals, while the shuffled interval is the 95% permutation range.

| Setting | $n$ | $P(y_i^I = 0 \mid y_j^I = 0)$ | $P(y_i^I = 0 \mid y_j^I = 1)$ | $\Delta_{\text{hist}}$ |
|---|---|---|---|---|
| Overall | 4500 | 0.8132 | 0.6455 | 0.1676 |
| Weak match | 2226 | 0.8146 | 0.7183 | 0.0963 |
| Strong match | 2274 | 0.8118 | 0.5712 | 0.2406 |

*Notes.* 95% CIs for $\Delta_{\text{hist}}$: Overall $[0.1029, 0.2242]$, Weak match $[-0.0097, 0.1988]$, and Strong match $[0.1761, 0.3039]$. The label-shuffle control uses 2,000 permutations and yields 0.0002; $[-0.0444, 0.0513]$.

### C.2 Offline Replay Diagnostics and Amortization

This subsection provides additional diagnostics for the offline replay results in Sections 5.2 and 5.4. All call counts follow the candidate-level cost model defined in Section 3.

**Threshold sweep and operating point.** Table 6 reports the fixed-threshold sweep used to characterize the efficiency–error–quality tradeoff. The rejection model and initial supervision remain fixed while the rejection threshold $\eta$ varies. The primary CERP operating point is reported separately because its threshold is selected from the initial supervision set independently for each replay ordering. At the primary operating point, CERP achieves 12.7% mean net savings relative to Always-Verify with a pooled RER of 8.8%. Table 7 decomposes this result into initial Full-CMI evaluations, utility verifications, pre-verification rejections, and disagreements with the Full-CMI reference. Across the ten replay orderings, CERP rejects 1,137 of 5,000 candidate decisions before verification and saves 637 net calls in total (63.7 per ordering), with an overall disagreement rate of 4.5%.

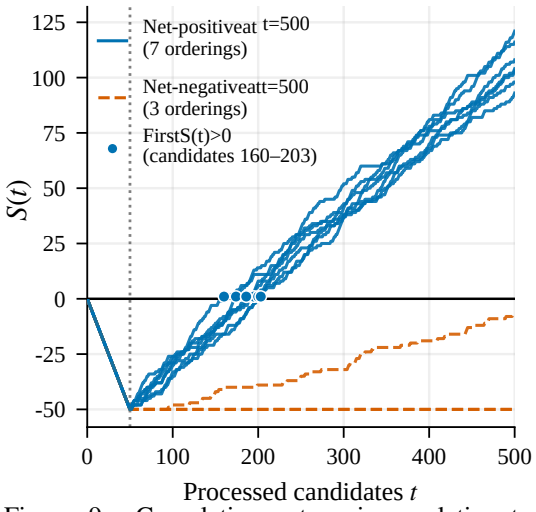



Figure 9: Cumulative net savings relative to Always-Verify across the ten replay orderings.

**Cumulative amortization.** Figure 9 tracks cumulative net savings relative to Always-Verify,

$$S(t) = t - \sum_{i \le t} c_i, \tag{13}$$

where $c_i$ is the nominal evaluation cost of candidate $i$. After the first $B = 50$ Full-CMI evaluations, all orderings have $S(50) = -50$. Among the seven orderings that eventually achieve positive net savings, the initial Full-CMI cost is recovered between candidate positions 160 and 203; the remaining three do not reach break-even within the 500-candidate horizon. These trajectories describe the observed replay orderings rather than a guaranteed amortization horizon.

### C.3 Evaluation on CausalMemBench

We further evaluate CERP on CausalMemBench to examine whether the amortization patterns observed on Causal-LoCoMo extend to a distinct memory-evaluation setting. As described in Appendix B, CausalMemBench contains independently generated tasks rather than conversation-level trajectories. We use both the full benchmark for detailed analysis with the primary Qwen2.5-7B responder and a fixed 200-task cross-model subset for comparison across all six responder families.

Table 6: Threshold sweep with a fixed rejector and initial supervision over ten replay orderings. The primary CERP operating point is selected independently from the initial supervision set and is reported separately from the fixed-threshold sweep.

| $\eta$ | Red. | Net | Proxy | Cov. | RER | Errors / rejects |
|---|---|---|---|---|---|---|
| **CERP** | $56.4 \pm 6.6$ | 12.7 | $0.8829 \pm 0.0108$ | 22.7 | 8.8 | 100/1,137 |
| 0.50 | $84.7 \pm 1.6$ | 69.4 | $0.7121 \pm 0.0303$ | 79.4 | 19.5 | 774/3,972 |
| 0.55 | $83.2 \pm 1.5$ | 66.5 | $0.7345 \pm 0.0272$ | 76.5 | 18.2 | 696/3,825 |
| 0.60 | $81.7 \pm 1.9$ | 63.4 | $0.7534 \pm 0.0291$ | 73.4 | 17.2 | 631/3,670 |
| 0.65 | $79.6 \pm 2.1$ | 59.1 | $0.7806 \pm 0.0262$ | 69.1 | 15.8 | 547/3,457 |
| 0.70 | $76.9 \pm 2.6$ | 53.9 | $0.8095 \pm 0.0298$ | 63.9 | 13.9 | 444/3,194 |
| 0.75 | $73.3 \pm 3.6$ | 46.7 | $0.8295 \pm 0.0313$ | 56.7 | 12.6 | 358/2,834 |
| 0.80 | $68.5 \pm 4.4$ | 37.0 | $0.8480 \pm 0.0284$ | 47.0 | 11.5 | 270/2,351 |
| 0.85 | $63.0 \pm 4.6$ | 26.1 | $0.8693 \pm 0.0144$ | 36.1 | 9.7 | 174/1,803 |
| 0.90 | $57.4 \pm 4.2$ | 14.7 | $0.8815 \pm 0.0091$ | 24.7 | 8.4 | 104/1,237 |
| 0.95 | $50.8 \pm 2.8$ | 1.7 | $0.8900 \pm 0.0044$ | 11.7 | 9.1 | 53/585 |
| 0.975 | $47.6 \pm 1.9$ | $-4.8$ | $0.8989 \pm 0.0044$ | 5.2 | 6.9 | 18/259 |
| 0.99 | $45.8 \pm 1.1$ | $-8.4$ | $0.9019 \pm 0.0034$ | 1.6 | 6.2 | 5/81 |

*Notes.* Red. denotes nominal call reduction relative to Full-CMI, Net denotes net call savings relative to Always-Verify, and Cov. denotes the pre-verification rejection fraction. Red., Net, Cov., and pooled RER are reported as percentages. Errors / rejects gives the number of rejection errors over all pre-verification rejections. Red. and Proxy report mean $\pm$ SD across ten replay orderings; Net and Cov. report means.

Table 7: CERP evaluation costs and decision errors across ten replay orderings comprising 5,000 candidate decisions.

| Quantity | Count | Rate |
|---|---|---|
| Initial Full-CMI evaluations | 500 | — |
| Utility verifications | 3,363 | — |
| Pre-verification rejections | 1,137 | — |
| CERP nominal calls | 4,363 | — |
| Always-Verify nominal calls | 5,000 | — |
| Net calls saved | 637 | 12.7 |
| Rejection errors (RER) | 100/1,137 | 8.8 |
| Verification disagreements | 126/3,363 | 3.7 |
| Overall disagreements | 226/5,000 | 4.5 |

*Notes.* Rates are reported as percentages. Net savings are measured relative to Always-Verify. Error rates use the corresponding decision counts as denominators, and disagreements are defined relative to Full-CMI. The RER denominator is the number of pre-verification rejections.

### C.3.1 Full-benchmark amortization

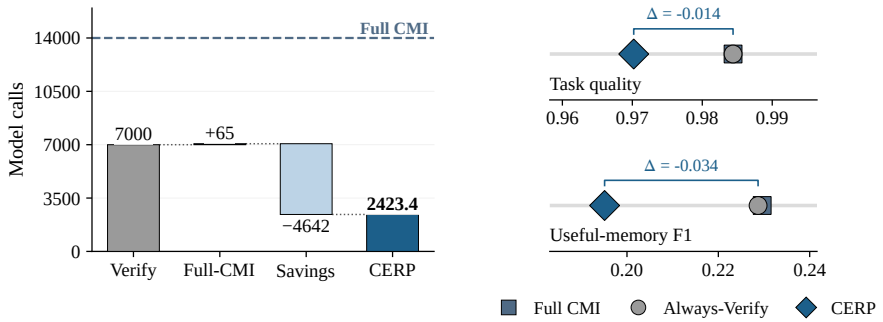
We first evaluate CERP on the full CausalMemBench benchmark with Qwen2.5-7B. Each evaluation stream contains 1,000 tasks with seven memory candidates per task, yielding 7,000 candidate-level decisions. Table 8 reports overall efficiency, selection quality, and rejection-risk results. Relative to Always-Verify, CERP achieves $65.4\%$ mean net call savings, reducing the mean evaluation cost from 7,000 to $2423.4$ calls per stream, with a pooled RER of $4.3\%$. Task quality decreases from $0.9844$ to $0.9702$, while Useful-Memory F1 decreases from $0.2287$ to $0.1951$.

Figure 10 (a) makes the amortization accounting explicit. The initial Full-CMI supervision adds 65 calls on average relative to Always-Verify, while subsequent pre-verification rejections avoid $4{,}642$ utility-verification calls, yielding 4,576.6 net calls saved over the 7,000-candidate stream. Figure 10 (b) shows the associated quality trade-off: relative to Always-Verify, CERP decreases task quality by $0.014$ and Useful-Memory F1 by $0.034$. Thus, within this longer evaluation stream, the avoided later verification cost is sufficient to recover the initial Full-CMI cost.

Table 8: Overall comparison of memory evaluation strategies on the full CausalMemBench benchmark with a Qwen2.5-7B-Instruct (Qwen2.5-7B) responder. Each stream contains 7,000 candidate decisions. Task quality is scored using CausalMemBench's upstream evaluation rule.

| Strategy | Calls ↓ | Net savings ↑ | Task Quality ↑ | Useful-Memory F1 ↑ | RER (%) ↓ |
|---|---|---|---|---|---|
| Full-CMI | 14,000 | – | 0.9844 | 0.2296 | – |
| Always-Verify | 7,000 | 0% | 0.9844 | 0.2287 | – |
| CERP | $2423.4 \pm 2535.7$ | 65.4% | $0.9702 \pm 0.0141$ | $0.1951 \pm 0.0467$ | 4.3 |

*Notes.* The nominal initial supervision target is $B = 50$. When additional Full-CMI evaluations are required before the rejection model can be fitted, their cost is included in the reported CERP calls and net savings. CERP values report mean $\pm$ SD across ten replay orderings.




(a) Verification cost amortization (b) Memory Quality

Figure 10: Verification-cost amortization and memory quality on full CausalMemBench with Qwen2.5-7B (means over ten orderings). (a) Always-Verify requires 7,000 calls; CERP adds 65 initial Full-CMI calls but avoids 4,642 later utility-verification calls, reducing total cost to 2,423.4 calls and yielding 4,576.6 net calls saved per stream. (b) Task quality and Useful-Memory F1. $\Delta$ denotes CERP minus Always-Verify.

For this experiment, $B = 50$ is the nominal initial supervision target. If the initial Full-CMI labels do not permit a non-degenerate rejection model, additional Full-CMI evaluations are acquired before the policy is fixed. Across the ten replay orderings, seven use 50 initial Full-CMI labels, while the remaining three require 75, 100, and 125 labels, respectively, giving an average of 65. The cost of all acquired Full-CMI labels is included in the reported CERP call count and net-savings calculation.

### C.3.2 Amortization across responder models

We evaluate the fixed CausalMemBench cross-model subset with six LLM responders under ten global task orderings. Each stream contains 1,400 candidate decisions, except for Kimi-K3 (1,372) and DeepSeek-V4.1-Flash (1,127) after the task-level exclusions described in Appendix B. Table 9 and Figure 11 show positive mean net savings for all six responders, ranging from $19.1\%$ to $54.9\%$. Positive amortization occurs in $4/10$ to $10/10$ orderings depending on the responder, while absolute task-quality differences from Always-Verify remain within $0.019$. These results show substantial responder-dependent variation in both the magnitude and consistency of amortization.

### C.3.3 Initial supervision budget sensitivity

Table 10 and Figure 12 examine how CERP behaves across nominal initial supervision budgets $B \in \{25, 50, 75, 100\}$ on the full CausalMemBench stream. Mean net savings remain positive in all four settings, ranging from $62.7\%$ to $76.8\%$. Because some replay orderings require additional Full-CMI evaluations before the rejection model can be fitted, the actual initial supervision cost can exceed the nominal budget. The mean numbers of acquired Full-CMI evaluations are $50.0$, $65.0$, $82.5$, and $102.5$, respectively, and all acquired evaluations are included in the reported net-savings calculation (Table 10).

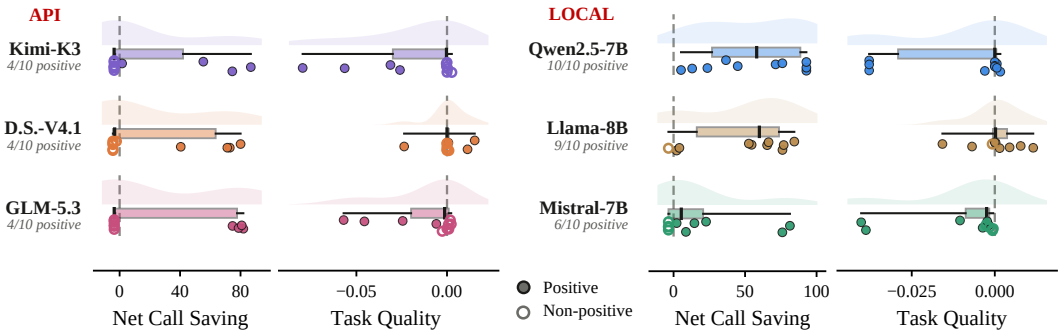



Figure 11: CERP across responder models on the fixed CausalMemBench cross-model subset. Net savings and task-quality changes are measured relative to Always-Verify on each responder's evaluated task set (Table 9).

Table 9: CERP across six LLM responders on the fixed CausalMemBench cross-model subset. $N$ denotes the number of candidate decisions per replay stream after the task-level exclusions described in Appendix B. Net savings are measured relative to Always-Verify; rejection fraction is the proportion of candidates rejected before verification. ΔQuality is CERP minus Always-Verify task quality. RER is pooled across the ten replay orderings for each responder.

| Responder | $N$ | Net savings (%) ↑ | Rejection frac. | ΔQuality | Useful-Memory F1 ↑ | RER (%) ↓ |
|---|---|---|---|---|---|---|
| Qwen2.5-7B | 1400 | 54.9 | 0.594 | −0.011 | 0.203 | 4.3 |
| Llama-3.1-8B | 1400 | 47.9 | 0.515 | +0.000 | 0.253 | 13.6 |
| Mistral-7B | 1400 | 19.1 | 0.227 | −0.010 | 0.298 | 15.9 |
| Kimi-K3 | 1372 | 19.7 | 0.233 | −0.019 | 0.219 | 25.3 |
| DeepSeek-V4.1-Flash | 1127 | 24.0 | 0.285 | +0.000 | 0.252 | 21.3 |
| GLM-5.3 | 1400 | 29.5 | 0.330 | −0.013 | 0.173 | 26.6 |

*Notes.* The +0.000 entries denote observed quality differences that round to zero and should not be interpreted as quality improvements. The primary nominal initial supervision budget is $B = 50$; all acquired Full-CMI evaluations are included in the reported net-savings calculation.

Figure 12 relates these results to the amortization condition. Let $L$ denote the actual number of initial Full-CMI evaluations acquired in an ordering. Positive amortization requires $N_R > L$, or equivalently $N_R/N > L/N$. Across all four tested budgets, the observed pre-verification rejection fraction remains well above this break-even requirement, consistent with the positive mean net savings throughout the 7,000-candidate stream. Task quality varies only within the range reported in the table and shows no monotonic pattern across the tested budgets.

Compared with the 500-candidate Causal-LoCoMo setting, where the tested $B = 100$ configuration does not recover its initial Full-CMI cost within the observed horizon, all four tested CausalMemBench settings remain above break-even. This is consistent with the amortization analysis in Section 5.4: a longer stream provides more subsequent decisions over which an initial verification cost can be recovered, although the two benchmarks differ in other respects.

### C.4 Representative CERP Decisions

We provide representative candidate-level decisions from the canonical Causal-LoCoMo replay orderings to illustrate how CERP operates at different points in a sequential evaluation stream. The examples cover an initial Full-CMI evaluation, a subsequent pre-verification rejection, and a subsequent candidate routed to utility verification. For each decision type, we select the earliest occurrence under a fixed scan order over replay orderings and candidate positions. This selection rule is used only for exposition and does not affect model fitting, threshold selection, or any reported aggregate result.

For candidates after the initial supervision prefix, the Full-CMI outcome reported below is used only as an offline reference for interpreting the decision. It is not available to CERP at decision time. The rejector and threshold are fixed after the initial prefix, while the historical state used to construct later pre-decision representations may continue to evolve through paid evaluations.

Table 10: Initial supervision budget sensitivity on full CausalMemBench with Qwen2.5-7B. $B$ denotes the nominal supervision budget; acquired Full-CMI evaluations report the mean [range] across ten replay orderings and include any additional initial supervision required before the policy is fixed. All acquired evaluations are included in the cost calculation.

| Nominal $B$ | Acquired Full-CMI | Net savings ↑ | Rejection frac. | Task quality |
|---|---|---|---|---|
| 25 | 50.0 [25–125] | 62.7% | 0.635 | 0.9720 |
| **50**† | 65.0 [50–125] | **65.4%** | **0.663** | **0.9702** |
| 75 | 82.5 [75–125] | 76.8% | 0.780 | 0.9656 |
| 100 | 102.5 [100–125] | 69.7% | 0.712 | 0.9679 |

*Notes.* † Primary setting fixed independently of this sweep.

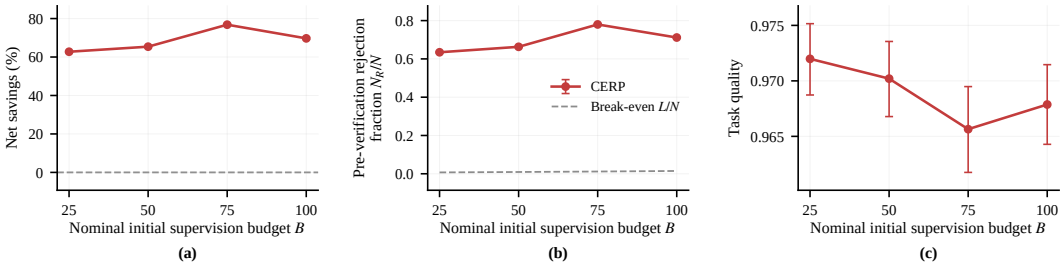



Figure 12: Effect of the nominal initial supervision budget $B$ on full CausalMemBench. (a) Net savings relative to Always-Verify. (b) Pre-verification rejection fraction $N_R/N$, with $L/N$ denoting the break-even requirement based on the actual number $L$ of acquired initial Full-CMI evaluations. (c) Task quality. Points report means over ten replay orderings; error bars denote 95% bootstrap confidence intervals with generated examples as resampling units.

**Initial Full-CMI evaluation.** The first candidate in seed 0 pairs the task "what are John's goals with regards to his basketball career?" with an unrelated memory about cooking. At this point no prior evaluation history is available, and the candidate lies within the initial supervision prefix. CERP therefore constructs its pre-decision representation and applies Full-CMI, which rejects the memory ($y_i^I = 0$). The resulting representation–label pair is added to the initial supervision set used to fit the rejection policy. This example illustrates how Full-CMI supervision is acquired before selective verification is applied to later candidates.

**Pre-verification rejection.** The earliest subsequent rejection under a finite threshold occurs at candidate $i = 51$ in seed 1 for the same task–memory pair. By this point, the rejection model and threshold have been fixed from the initial supervision prefix. Using the historical state available before candidate $i = 51$, CERP constructs $\phi_i$ and estimates

$$p_i = 0.029,$$

where $p_i$ is the predicted probability that Full-CMI would retain the candidate. The selected rejection threshold is $\eta = 0.908$. Since

$$1 - p_i = 0.971 \geq \eta,$$

CERP rejects the candidate before utility verification and therefore incurs no additional evaluation call for this decision. The offline Full-CMI reference also rejects the candidate ($y_i^I = 0$), so this pre-verification rejection agrees with the reference criterion.

**Utility verification.** A nearby decision in the same replay ordering illustrates the opposite branch. For the same task, candidate $i = 54$ is another unrelated memory concerning surfing. CERP again constructs the pre-decision representation from the currently available historical state and applies the same fixed rejector and threshold, obtaining

$$p_i = 0.094.$$

In this case,

$$1 - p_i = 0.906 < \eta,$$

so the pre-verification rejection rule is not triggered. CERP instead routes the candidate to utility verification, which returns $y_i^V = 0$; the memory is therefore not retained. The offline Full-CMI

reference also rejects this candidate ($y_i^I = 0$). Thus, although candidates $i = 51$ and $i = 54$ belong to the same replay ordering, concern the same task, and use the same fixed threshold, their pre-decision representations place them on opposite sides of the rejection boundary.

These examples illustrate the distinction between acquiring initial Full-CMI supervision and using the resulting fixed policy for later selective verification. They also show that routing a candidate to utility verification does not imply that the candidate will ultimately be retained; it means only that CERP does not have sufficient score-based evidence to reject it before verification. The same local sequence also contains a false rejection at $i = 53$: CERP rejects a gold memory with $p_i = 0.044$, whereas the offline Full-CMI reference retains it. Such decisions constitute rejection errors and are counted by the RER reported in Table 7.