IMPRINT: RETHINKING RETRIEVAL FOR LONG-TERM CONVERSATIONAL AGENT MEMORY

Anonymous authors
Paper under double-blind review

ABSTRACT

Long-term memory systems for conversational agents differ in how they organize memory but share one retrieval primitive: dense embeddings ranked by cosine similarity. We argue that this primitive, not the memory structure, is the bottleneck, because personal recall hinges on rare user-specific identifiers whose exact identity mean pooling averages away. We reframe recall as *imprint-following*, where each rare token is a trace pointing back to its source session, and instantiate it in **IMPRINT**, whose single *Token Profile Table* serves as an IDF-weighted inverted index and drives an online consolidation pool, with no retrieval training, no LLM calls at indexing, and no GPU at query time. We prove that rarity-weighted exact matching keeps a separation margin independent of the retrieval-unit length, whereas the dense margin decays as $L^{-1/2}$; the same result explains why session-level sparse retrieval was dismissed by prior work. On LoCoMo (27 sessions) and LongMemEval (500 sessions), IMPRINT beats the strongest prior memory system by up to +15.8 Recall@3 and +3.9 QA-F1, outperforms dense retrieval under an identical pipeline, and produces the best answers on LoCoMo. [1]

1 INTRODUCTION

Long-term conversational agents must remember what users tell them across hundreds of dialogue sessions and surface that information on demand. As interaction histories grow, the quality of this *personal memory recall* increasingly determines the user experience.

A growing literature on agent memory has responded with increasingly elaborate designs, spanning hierarchical summaries (Zhong et al., 2024), knowledge graphs (Gutierrez et al., 2024), structured note linking (Xu et al., 2026b), multi-granularity routing (Xu et al., 2026a), and tree-structured recursive summarization (Sarthi et al., 2024). These systems differ sharply in how they *organize* memory, yet they converge on a single design choice at the retrieval stage: encode each memory unit as a dense embedding and rank by cosine similarity. The implicit assumption is that long-term memory recall is fundamentally a semantic-matching problem.

This assumption mischaracterizes what personal memory actually retrieves. Dense embeddings project rare, user-specific identifiers (proper nouns, dates, place names) into continuous semantic neighborhoods where their unique fingerprints are averaged away (Figure 1, left). The unresolved issue is therefore not the *structure* of memory but its *retrieval primitive*: however elaborately memory is organized, the final ranking step dilutes the rare token signals on which recall depends.

This observation motivates a simple but powerful reframing: treat each rare token as an *imprint*, a distinctive trace that points to the original session, so that recall reduces to *following these imprints to their source*. We instantiate this idea in **IMPRINT** (*IDF-Modulated Profiles of Rare Interaction-ngram Tokens*), which scores sessions by IDF-weighted exact-token overlap with the query, letting rare tokens dominate while common tokens contribute nothing at all. The mechanism preserves the exact identity of the user's words rather than projecting them into a semantic neighborhood populated by distractors, and it requires no retrieval-specific training and no LLM-driven structuring: the index is a table of token statistics that grows incrementally as the conversation proceeds.

[1] Code is available at Supplementary Material.

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.

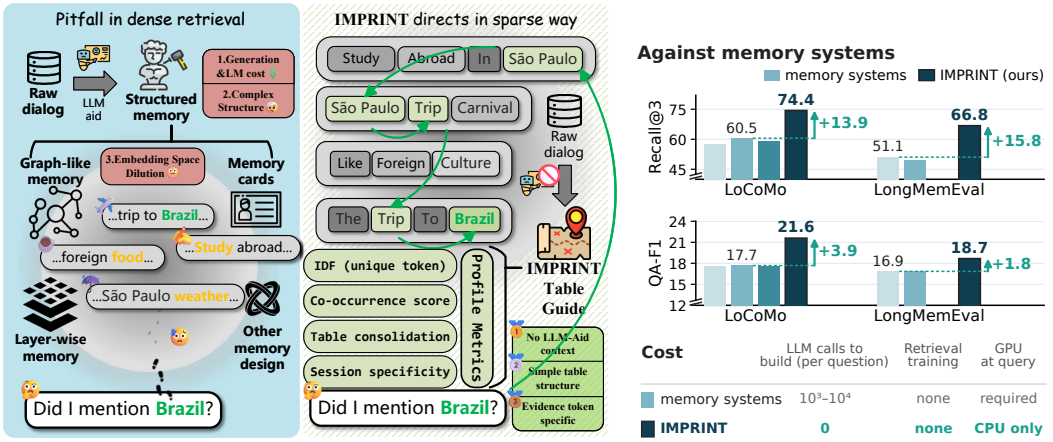


Figure 1: **IMPRINT at a glance. Left:** dense memory systems dilute imprints. **Middle:** Token Profile Table lookup. **Right:** gains over memory systems without LLM, training or query-time GPU.

Two issues stand in the way. **Challenge 1: imprint dilution.** A rare token is only useful if it is not drowned out by the rest of the unit it sits in. In a mean-pooled dense embedding its contribution is divided among all L tokens of the unit, so the margin that separates the source session from a distractor shrinks as $L^{-1/2}$ (Theorem 5.1); a classical sparse scorer tested on whole sessions suffers a similar dilution from common terms, which is why prior work dismissed sparse retrieval. **Challenge 2: verbose context.** Even accurate retrieval returns long, noisy dialogue that obscures the key facts.

To address imprint dilution, IMPRINT maintains a *Token Profile Table* of per-token rarity statistics and scores a session by the rarity-weighted exact overlap of its tokens with the query: common tokens contribute nothing, so a rare imprint keeps a separation margin that does not decay with the length of the retrieval unit (Theorem 5.1), and the index can operate at the session level without losing turn-level precision. The same lens explains a historical negative result: dense scoring recovers only with turn-level units, and so, empirically, does BM25, so sparse retrieval was dismissed not because matching exact tokens is wrong, but because it was weighted and segmented wrongly (§6.2).

To address verbose context, IMPRINT introduces **statistical consolidation**, where the same Token Profile Table drives online merging. When a new turn is similar enough to an existing memory unit, they are merged and pruned by a four-factor score combining IDF and session specificity with a consolidation count and a co-occurrence richness. At generation, these consolidated entries are placed ahead of the raw dialogue as importance-weighted, deduplicated context, so that the rare identifiers the generator needs are the first thing it sees.

Contributions. (1) We characterize personal memory recall as *imprint-following* and prove that rarity-weighted exact matching keeps a separation margin independent of the unit length while mean-pooled dense scoring decays as $L^{-1/2}$; with our granularity study, this explains why session-level sparse retrieval was prematurely dismissed. (2) The *Token Profile Table* serves as both an IDF-weighted inverted index and the driver of online consolidation; it is built incrementally, never re-processes past sessions, and needs no learned retriever, knowledge graph, or LLM-based memory extraction. (3) On LoCoMo and LongMemEval, IMPRINT improves over the strongest prior memory system by up to +15.8 Recall@3 and +3.9 QA-F1 with no training and no GPU at query time, outperforms turn-level Contriever and the BM25 + Contriever hybrid under an identical pipeline, and on LoCoMo turns its consolidated units into the best answers of any retriever we compare.

2 Related Work

Memory systems for conversational agents. Persistent memory for LLM agents (Hu et al., 2025; Zhang et al., 2025) has been organized as OS-style hierarchies (Packer et al., 2023; Kang et al., 2025), extracted facts (Chhikara et al., 2025; Xu et al., 2026b), forgetting curves (Zhong et al., 2024), compressed or staged stores (Liu et al., 2026; Fang et al., 2025a; Nan et al., 2025), knowledge graphs (Gutierrez et al., 2024; Gutiérrez et al., 2025; Anokhin et al., 2025; Edge et al., 2024; Guo et al., 2025; Wu et al., 2025b), trees (Sarthi et al., 2024; Rezazadeh et al., 2025), and multi-granularity

routing (Xu et al., 2026a); other lines train the memory policy (Yu et al., 2026; Yan et al., 2025; Wang et al., 2025b;c; Zhao et al., 2024; Fang et al., 2025b) or replace external memory by long context and compression (Bulatov et al., 2022; Wang et al., 2024; Lee et al., 2024; Ge et al., 2024; Jiang et al., 2023; Pan et al., 2024; Jiang et al., 2024; Bai et al., 2024; Fountas et al., 2025), and cognitive theories of memory inform the framing (Atkinson & Shiffrin, 1968; Tulving, 1972; 2002; Baddeley, 2012; Kumaran et al., 2016; Spens & Burgess, 2024). Whatever the structure, all of these systems retrieve with dense embeddings, which is where IMPRINT intervenes.

Sparse, dense, and hybrid retrieval. BM25 (Robertson & Zaragoza, 2009) stays competitive with dense retrievers zero-shot (Thakur et al., 2021); learned sparse models (Dai & Callan, 2020; Formal et al., 2021b;a; Gao et al., 2021) close the gap with supervision, and dense retrievers (Karpukhin et al., 2020; Izacard et al., 2022) back nearly every memory system within the RAG paradigm (Lewis et al., 2020; Asai et al., 2024). In conversational memory, Pan et al. (2025) first reported BM25 gains over dense retrieval on LoCoMo. IMPRINT is training-free, with term weights taken from corpus statistics alone. We evaluate on LoCoMo (Maharana et al., 2024) and LongMemEval (Wu et al., 2025a); MemoryAgentBench (Hu et al., 2026) offers a complementary multi-turn probe.

3 PRELIMINARY

Problem definition. A long-term conversational agent interacts with a user across N chronologically ordered sessions $\mathcal{S} = \{S_1, S_2, \ldots, S_N\}$, where each session $S_i = (u_1^{(i)}, u_2^{(i)}, \ldots, u_{|S_i|}^{(i)})$ is an ordered sequence of dialogue turns. The full memory bank is $\mathcal{M} = \bigcup_{i=1}^{N} \{u_j^{(i)} : 1 \le j \le |S_i|\}$. Given a query q, the *retrieval* task returns the top-k sessions: $\mathcal{S}_k(q) = \arg\text{top-k}_{S \in \mathcal{S}}\ s(q, S)$, where $s(q, S)$ is a query-session scoring function. The *generation* task then produces an answer $\hat{a} = g(q, \mathcal{S}_k(q))$.

Tokenization. We tokenize every turn with a fixed tokenizer $\mathcal{T}$ that maps a string to a multiset of tokens from vocabulary $\mathcal{V}$. For a turn u, $\mathcal{T}(u) \subseteq \mathcal{V}$ denotes its token multiset; for the query q, $\mathcal{T}(q) = \{t_1, \ldots, t_m\}$ denotes its (typically deduplicated) token set. All sparse statistics are computed with respect to this single tokenizer.

Dense baseline. We consider a standard dense retriever $\phi : \mathcal{V}^* \to \mathbb{R}^d$ that maps a string to a d-dimensional embedding. The dense session score is

$$s_{\text{dense}}(q, S_i) = \cos\big(\phi(q), \phi(S_i)\big) = \frac{\phi(q)^\top \phi(S_i)}{\|\phi(q)\|\,\|\phi(S_i)\|}. \tag{1}$$

This is the retrieval primitive shared by essentially all prior agent memory systems (Zhong et al., 2024; Gutierrez et al., 2024; Xu et al., 2026b;a), and the object that imprint-following replaces.

4 METHOD

IMPRINT is built around a single data structure, the *Token Profile Table* $\mathcal{P}$, that simultaneously serves as an inverted index for retrieval and as a statistical driver for memory consolidation. Figure 2 illustrates the pipeline. The remainder of this section formalizes each component: §4.1 defines the four token statistics in $\mathcal{P}$; §4.2 derives the retrieval scoring function; §4.3 describes the consolidation mechanism; §4.4 specifies the generation pipeline.

4.1 THE TOKEN PROFILE TABLE

The Token Profile Table $\mathcal{P} : \mathcal{V} \to \mathbb{R}^4$ associates each token $t \in \mathcal{V}$ with four statistics: $\mathcal{P}(t) = (\text{df}(t),\ \text{sess}(t),\ \kappa(t),\ \mathcal{N}(t))$, where $\text{df}(t) \in \mathbb{N}$ is the *document frequency* (number of distinct sessions containing t, i.e. $|\text{sess}(t)|$); $\text{sess}(t) \subseteq \{1, \ldots, N\}$ is the *session index set*; $\kappa(t) \in \mathbb{N}$ is the *consolidation count* (times t has been retained across pool merges); and $\mathcal{N}(t) \subseteq \mathcal{V}$ is the *co-occurrence neighborhood* (tokens sharing at least one turn with t). We also write $\text{tf}(t, S_i) = \sum_j \mathbb{1}[t \in \mathcal{T}(u_j^{(i)})] \cdot \#(t, u_j^{(i)})$ for the term frequency of t in session S_i.

Incremental construction. For each newly arrived turn $u_j^{(i)}$, we update $\mathcal{P}$ in $O(|\mathcal{T}(u_j^{(i)})|)$ time:

$$\text{sess}(t) \cup\!= \{i\}, \qquad \text{tf}(t, S_i)\ +\!= \#(t, u_j^{(i)}), \qquad \mathcal{N}(t) \cup\!= \mathcal{T}(u_j^{(i)}) \setminus \{t\}, \qquad \forall\, t \in \mathcal{T}(u_j^{(i)}). \tag{2}$$

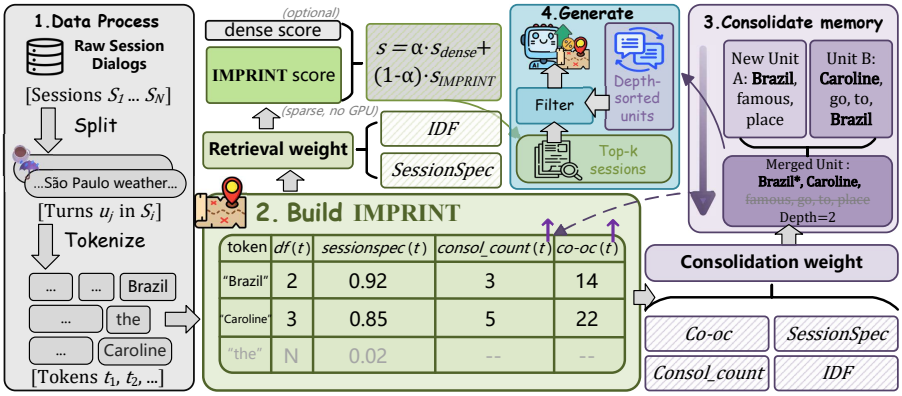


Figure 2: **Pipeline of IMPRINT.** The Token Profile Table records four token statistics: two define the retrieval weight (Eq. 5); all four define the consolidation weight (Eq. 12).

The consolidation count $\kappa(t)$ is updated separately during pool merging (§4.3). Construction is strictly incremental: past sessions are never reprocessed.

Derived imprint statistics. From df and sess we derive two scalars; the first is the IDF:

$$\mathrm{IDF}(t) = \log \frac{N+1}{\mathrm{df}(t)+1}, \tag{3}$$

which grows as t becomes rarer across sessions; a token present in every session receives $\mathrm{IDF}(t) = 0$. The second is the *session specificity*, derived from the binary entropy of the session-incidence proportion $p(t) = |\mathrm{sess}(t)|/N$:

$$\mathrm{SessSpec}(t) = 1 - \frac{H_b(p(t))}{\log 2}, \qquad H_b(p) = -p \log p - (1-p)\log(1-p). \tag{4}$$

$\mathrm{SessSpec}(t) \in [0,1]$ is minimized at $p(t) = 1/2$ (maximum uncertainty) and maximized at $p(t) \in \{0,1\}$; the $p(t) \to 1$ branch is neutralized by the IDF factor, which is exactly 0 for a token present in every session, so the product below rewards tokens concentrated in few sessions; both statistics are parameter-free.

4.2 Imprint-Following Retrieval

Retrieval weight. Combining rarity and session-distribution concentration multiplicatively:

$$w(t) = \mathrm{IDF}(t) \times \rho(t), \qquad \rho(t) = \frac{1 + \mathrm{SessSpec}(t)}{2}. \tag{5}$$

A token must be rare in both senses to receive a large weight. The rescaling $\rho(t) \in [1/2, 1]$ ensures session specificity acts as a bonus multiplier rather than a gate, adding no tunable hyperparameter.

Sparse session score. Given query q with token set $\mathcal{T}(q)$, the session S_i imprint-following score is:

$$s_{\text{IMPRINT}}(q, S_i) = \sum_{t \in \mathcal{T}(q)} w(t) \cdot \log\big(1 + \mathrm{tf}(t, S_i)\big). \tag{6}$$

Rare query tokens with strong imprints dominate the sum and pull their source sessions to the top; term frequencies are accumulated at the session level from turn-level units, and §5 shows that the resulting margin does not depend on the length of the unit over which they are aggregated. Evaluation proceeds by posting-list lookup with cost $O(\sum_{t \in \mathcal{T}(q)} |\mathrm{sess}(t)|)$ and no neural inference.

Optional dense complement. The sparse score can be blended with a dense score. Consistent with §5, which shows that mean-pooled dense scoring needs short units, the dense channel scores individual turns and max-pools them to their session, $s_{\text{dense}}(q, S_i) = \max_{u \in S_i} \cos(\phi(q), \phi(u))$. We min-max normalize each score over the candidate set $\mathcal{S}$:

$$\tilde{s}(q, S_i) = \frac{s(q, S_i) - \min_j s(q, S_j)}{\max_j s(q, S_j) - \min_j s(q, S_j)}, \tag{7}$$

and combine with a single mixing coefficient $\alpha \in [0, 1]$:

$$s(q, S_i) = \alpha \cdot \tilde{s}_{\text{dense}}(q, S_i) + (1-\alpha) \cdot \tilde{s}_{\text{IMPRINT}}(q, S_i). \tag{8}$$

Setting $\alpha = 0$ recovers pure imprint-following; $\alpha = 1$ recovers the dense baseline. We use $\alpha = 0.3$ throughout. Because s_{IMPRINT} depends only on $\mathcal{P}$ and shares no state with other components, it can also augment an arbitrary external memory system $\mathcal{A}$:

$$s_{\mathcal{A}+\text{IMPRINT}}(q, S_i) = \alpha \cdot \tilde{s}_{\mathcal{A}}(q, S_i) + (1 - \alpha) \cdot \tilde{s}_{\text{IMPRINT}}(q, S_i), \tag{9}$$

making IMPRINT a drop-in sparse complement requiring no retraining or architectural coupling.

4.3 Statistical Consolidation and the Memory Pool

Pool entries. The consolidated memory pool $\mathcal{C}$ contains entries $e = (T_e, R_e, \sigma_e, d_e)$, where $T_e \subseteq \mathcal{V}$ is the retained token set, R_e is the raw source text, $\sigma_e \in \{1, \ldots, N\}$ is the source-session index, and $d_e \in \mathbb{R}_{\geq 0}$ is the *consolidation depth* (initialized to 0, incremented by $\delta = 0.5$ on each merge). Each entry has an IDF-weighted embedding:

$$\phi(e) = \frac{1}{\sum_{t \in T_e} \text{IDF}(t)} \sum_{t \in T_e} \text{IDF}(t) \cdot h(t), \tag{10}$$

where $h(t)$ is the dense encoder's hidden state for token t.

Online merge rule. Each new turn $u_j^{(i)}$ forms a candidate entry e_{new} with nearest neighbor $e^\star = \arg\max_{e \in \mathcal{C}} \cos\big(\phi(e_{\text{new}}), \phi(e)\big)$ in $\mathcal{C}$. If $\cos(\phi(e_{\text{new}}), \phi(e^\star)) > \tau$ ($\tau = 0.75$), they are merged:

$$T_{e'} = \text{Prune}(T_{e_{\text{new}}} \cup T_{e^\star}), \qquad R_{e'} = R_{e^\star} \,\|\, R_{e_{\text{new}}}, \qquad d_{e'} = d_{e^\star} + \delta, \tag{11}$$

where $\|$ denotes text concatenation. For every token $t \in T_{e_{\text{new}}} \cup T_{e^\star}$, $\kappa(t)$ += 1. Otherwise, e_{new} is added to $\mathcal{C}$ unchanged. Consolidation is thus a streaming operation: each turn triggers exactly one nearest-neighbor lookup and at most one merge. Two remarks. The merge test is a *unit-to-unit* near-duplicate decision, for which whole-unit semantic similarity over IDF-weighted token states is the appropriate signal; it is not the query-to-session recall step whose dilution §1 criticizes. And consolidation never changes the ranking: κ enters only the retention weight, so Table 1's retrieval columns are identical with or without it, and its effect is confined to the generation context (§4.4).

Four-factor consolidation weight. Pruning extends the retrieval weight with two relational factors:

$$w_{\text{cons}}(t) = \underbrace{w(t)}_{\text{retrieval}} \times \underbrace{(1 + \beta_c\, c(t))}_{\text{reinforcement}} \times \underbrace{(1 + \beta_r\, r(t))}_{\text{co-occurrence richness}}, \tag{12}$$

where the bonuses are bounded saturating functions:

$$c(t) = 1 - \frac{1}{1 + \kappa(t)} \in [0, 1), \qquad r(t) = \min\left(\frac{|\mathcal{N}(t)|}{C_{\max}}, 1\right) \in [0, 1], \tag{13}$$

with a fixed saturation point $C_{\max} = 20$ and dampening coefficients $\beta_c = 0.3$, $\beta_r = 0.2$. Both factors are bounded in $[0, 1]$, so $w_{\text{cons}}(t)$ ranges from $w(t)$ to at most $1.56 \cdot w(t)$. Consolidation thus retains tokens that are rare, repeatedly re-encountered, and connected to diverse contexts.

Pruning operator. Given a merged token set and a fixed capacity $K_{\max} = 120$ (pruning is triggered only when the merged set exceeds it):

$$\text{Prune}(T \cup T') = \arg\underset{t \in T \cup T'}{\text{top-K}_{\max}}\ w_{\text{cons}}(t). \tag{14}$$

The same Token Profile Table feeds two different weight formulas: two factors for retrieval and four for consolidation. Retrieval benefits from per-token independence, so strong imprints are not contaminated by neighbors' statistics; consolidation benefits from relational reinforcement, so tokens seen repeatedly across contexts persist. One $\mathcal{P}$ thus serves both roles without interference.

4.4 Generation with Consolidated Context

At query time, after retrieval returns $\mathcal{S}_k(q)$, we collect the *consolidated* pool entries whose source sessions intersect $\mathcal{S}_k(q)$, i.e. entries that have been merged at least twice and thus span at least three turns ($d_e \geq 1$ with $\delta = 0.5$; at most five per session): $\mathcal{C}_k(q) = \{e \in \mathcal{C} : \sigma_e \cap \mathcal{S}_k(q) \neq \emptyset,\ d_e \geq 1\}$, and order them by consolidation depth in descending order: $\pi(q) = (e_{(1)}, e_{(2)}, \ldots)$ with $d_{e_{(1)}} \geq d_{e_{(2)}} \geq \cdots$ (de-duplicated, at most $U_{\max} = 60$ texts). The generation context concatenates depth-ordered pool source texts followed by raw retrieved sessions as supplementary context:

$$\text{Ctx}(q) = \underbrace{R_{e_{(1)}} \,\|\, \cdots \,\|\, R_{e_{(|\mathcal{C}_k(q)|)}}}_{\text{consolidated, depth-ordered}} \,\|\, \underbrace{S_{i_1} \,\|\, \cdots \,\|\, S_{i_k}}_{\text{raw retrieved sessions}}. \tag{15}$$

Pool entries with higher depth, those most reinforced by repeated user mentions, are placed earliest where the generator's attention budget is largest.

Filter–generate pipeline. Following Xu et al. (2026a), we adopt a two-stage generator $g = g_{\text{gen}} \circ g_{\text{filt}}$. The filter stage extracts query-relevant evidence: $z = g_{\text{filt}}(q, \text{Ctx}(q))$, and the generation stage produces the final answer: $\hat{a} = g_{\text{gen}}(q, z)$. The generation prompt is shared with every baseline, and the filter prompt differs from the baselines' only by the additional *Memory Units* block that exposes the pool (App. B.6), so any improvement is attributable to the context, not to prompt engineering.

Implementation details. We use the Contriever tokenizer for $\mathcal{T}$ and frozen Contriever hidden states for h and the dense channel. The merge threshold $\tau = 0.75$, depth increment $\delta = 0.5$, mixing coefficient $\alpha = 0.3$, and affine floor $1/2$ are fixed across all experiments and datasets. IMPRINT's scoring introduces no learned parameters and requires no retrieval-specific training: the sparse channel runs on CPU, and the only neural components are the frozen off-the-shelf encoder and the generator shared with all baselines.

5 Theoretical Analysis

We formalize why rarity-weighted scoring separates a target unit from distractors in a way that does not decay with the length of the retrieval unit, whereas mean-pooled dense scoring does. Extended discussion, the multi-imprint case, and the role of session specificity are deferred to App. A.

Setup. Units are blocks of L i.i.d. tokens from a background distribution μ ($L = L_S$ for session-level, $L \ll L_S$ for turn-level scoring); the memory holds M candidate units. The query carries one *imprint* $t^\star$ with $\mu(t^\star) = \varepsilon \ll 1$ (several imprints: App. A.3); the *target* u^+ has $t^\star$ planted in one slot, a *distractor* u^- contains it only by chance, with probability at most εL. s_{IMP} is Eq. (6) restricted to $t^\star$, with $w(t^\star) > 0$ computed from the realized memory (Eqs. (3)–(5)). For the dense rule, $s_{\text{dense}}(q, u) = \langle h^\star, \phi(u)\rangle / \|\phi(u)\|$ with $\phi(u) = \frac{1}{L}\sum_i h_i$, where the planted slot has $h_i = h^\star$ and all other slot vectors are i.i.d. uniform on $\mathbb{S}^{d-1}$ (chance copies of $t^\star$ change the margin by $O(\varepsilon)$ and are neglected). This *bag-of-random-vectors* model has no semantic structure by design: it measures how much of a rare identifier survives mean pooling, not how embeddings confuse related topics.

Theorem 5.1 (Length-robust imprint margin)**.** (a) Imprint scoring. *For every distractor that does not contain $t^\star$,*

$$s_{\text{IMP}}(q, u^+) - s_{\text{IMP}}(q, u^-) \;\geq\; w(t^\star)\log 2 \;>\; 0 \qquad \textit{for every } L, \tag{16}$$

and the target is outranked only by distractors that contain $t^\star$ by chance, whose expected number is at most $M\varepsilon L$. (b) Dense scoring. *With probability $1 - e^{-\Omega(d)}$ over the embeddings,*

$$\mathbb{E}\big[s_{dense}(q, u^+) - s_{dense}(q, u^-)\big] = \frac{1}{\sqrt{L}}\big(1 + O(d^{-1/2})\big), \qquad \text{sd}\big[s_{dense}(q, u^-)\big] = \Theta(d^{-1/2}), \tag{17}$$

so the margin-to-noise ratio of the dense rule is $\Theta(\sqrt{d/L})$ and decreases with the unit length, whereas that of the imprint rule is infinite (zero noise against non-matching distractors) at every L.

Proof sketch. (a) Restricted to $t^\star$, $s_{\text{IMP}}(q, u) = w(t^\star)\log(1 + \text{tf}(t^\star, u))$; the target has $\text{tf} \geq 1$ and a non-matching distractor has $\text{tf} = 0$, and a union bound over M units of L slots bounds the expected number of chance matches. (b) The planted slot contributes $1/L$ to $\langle h^\star, \phi(u^+)\rangle$, the other slots are zero-mean, and $\|\phi(u)\|$ concentrates at $L^{-1/2}$ because the slot vectors are independent and isotropic; a distractor's numerator is a mean of L terms of variance $1/d$, i.e. $(Ld)^{-1/2}$ before and $d^{-1/2}$ after normalization. Full proof in App. A.1. □

Interpretation. The theorem compares margin to noise, the quantity that determines a ranking, and it holds irrespective of the min-max normalization of Eq. (7), which rescales both scorers by a constant per query. The imprint margin does not depend on L, so IMPRINT can aggregate term frequencies at the session level (Eq. 6) without losing turn-level precision; the only price of a single imprint is the tie with the $\leq M\varepsilon L$ units that contain the same rare token, which additional imprints resolve additively (App. A.3). The dense margin shrinks as $L^{-1/2}$ because mean pooling divides the identifier's contribution among all L tokens while the per-distractor noise stays at $d^{-1/2}$, so dense retrieval *needs* short units. The model deliberately omits semantic structure and encoder anisotropy

Table 1: **Main results.** Best in **bold**, second best underlined; Δ: gain over the strongest baselines.

Method	Retrieval						Generation				
	R@3	N@3	R@5	N@5	R@10	N@10	F1	B-4	R-1	R-L	BS
LoCoMo											
Contriever	49.90	52.15	58.26	56.29	71.80	60.92	15.66	2.67	16.01	15.00	84.65
BM25	59.11	61.28	65.06	64.14	74.42	67.42	17.16	3.81	17.61	16.74	84.98
MPC	49.50	51.47	57.45	55.53	71.85	60.47	14.81	1.99	15.10	14.13	84.43
RecurSum	47.23	48.99	59.01	54.58	74.97	60.07	9.14	0.99	9.82	8.98	83.45
SeCom	55.24	57.90	64.80	62.36	78.30	66.97	13.79	2.30	14.28	13.30	84.04
HippoRAG 2†	60.52	62.33	70.69	66.99	82.02	70.89	17.68	3.65	18.07	17.02	85.03
RAPTOR†	54.78	55.82	64.05	60.46	78.10	65.34	16.29	3.50	16.65	15.72	84.83
A-Mem†	58.81	59.74	67.57	63.88	81.02	68.59	17.56	3.83	17.96	16.95	85.05
MemGAS	57.30	58.76	67.32	63.62	81.82	68.42	17.66	3.61	18.00	16.99	85.13
IMPRINT	**74.42**	**76.97**	**80.77**	**79.98**	**87.76**	**82.22**	**21.60**	**4.76**	**21.79**	**20.53**	**85.70**
Δ	+13.90	+14.64	+10.08	+12.99	+5.74	+11.33	+3.92	+0.93	+3.72	+3.51	+0.57
LongMemEval											
Contriever	44.26	56.11	53.40	59.44	65.96	62.74	11.88	1.66	12.56	11.02	83.28
BM25	43.40	54.58	49.57	56.95	55.96	59.21	16.35	3.46	16.96	15.34	84.45
MPC	35.96	47.51	42.55	50.36	54.26	53.88	11.28	1.37	11.93	10.57	82.98
RecurSum	23.19	32.10	31.70	36.99	43.62	41.40	10.04	1.70	10.89	9.21	83.12
SeCom	44.26	56.61	55.32	60.76	66.60	63.78	11.33	1.79	12.03	10.49	83.36
HippoRAG 2†	49.57	60.45	57.45	63.82	67.87	66.79	16.84	**3.83**	17.60	15.87	84.67
MemGAS	51.06	61.36	63.62	66.07	77.02	69.46	16.85	3.39	17.60	16.14	84.69
IMPRINT	**66.81**	**74.66**	**77.02**	**77.96**	**84.04**	**80.08**	**18.68**	3.69	**19.39**	**17.92**	**84.89**
Δ	+15.75	+13.30	+13.40	+11.89	+7.02	+10.62	+1.83	-0.14	+1.79	+1.78	+0.20

(App. A.6), so its constants are not predictive; its content is the direction of the dependence on L, which §6.2 confirms. It is stated for imprint versus dense scoring; the granularity dependence of BM25 is an empirical finding, discussed in App. A.7.

6 Experiments

Datasets. We evaluate on two benchmarks: **LoCoMo** (Maharana et al., 2024) (27 sessions per conversation) and **LongMemEval** (Wu et al., 2025a) (about 500 sessions per question), an 18× range in conversation length; §6.4 additionally uses nested 50–350-session subsets of LongMemEval. Following LongMemEval, its 30 abstention questions are excluded from retrieval metrics.

Baselines. Memory systems evaluated under the unified setup of Xu et al. (2026a), whose reported numbers we use: session-level Contriever (Izacard et al., 2022), RecurSum (Wang et al., 2025a), MPC (Lee et al., 2023), SeCom (Pan et al., 2025), and MemGAS (Xu et al., 2026a). HippoRAG 2 (Gutierrez et al., 2024), RAPTOR (Sarthi et al., 2024) and A-Mem (Xu et al., 2026b) are run by us with their official code under the same protocol (marked †; App. B.8); RAPTOR and A-Mem are reported on LoCoMo only because of their LLM cost at 500 sessions. Under IMPRINT's tokenizer, evaluation code and generation pipeline we also run session-level BM25 (Robertson & Zaragoza, 2009) (Table 1) and the turn-level retrievers of App. C.5.

Metrics. Retrieval: Recall@k and NDCG@k for $k \in \{3, 5, 10\}$. Generation: QA-F1 (F1), BLEU-4 (B-4), ROUGE-1 (R-1), ROUGE-L (R-L), BERTScore (BS).

Implementation. IMPRINT uses the Contriever tokenizer; the dense complement uses turn-level Contriever embeddings with $\alpha = 0.3$. Index construction needs no training and no LLM calls, only one frozen-encoder pass over each new turn (for the dense channel and the consolidation pool; App. B.9); retrieval scoring runs on CPU and needs no GPU at query time, and with $\alpha = 0$ the ranking uses no vectors at all (72.2 R@3 on LoCoMo). Every method we run shares one generation pipeline: top-3 sessions, a filter and a generate stage with GPT-4o-mini, and identical prompts except that IMPRINT's filter prompt additionally exposes the consolidated memory units (App. B.6). All hyperparameters ($\tau = 0.75$, $\delta = 0.5$, $\alpha = 0.3$, floor $= 1/2$) are fixed. Appendix B has more detail.

6.1 Main Results

Table 1 reports retrieval and generation. IMPRINT leads every baseline on every retrieval metric and, except for BLEU-4 on LongMemEval, on every generation metric: $+13.90$ / $+15.75$ Recall@3 and $+3.92$ / $+1.83$ QA-F1 over the strongest memory system (HippoRAG 2 on LoCoMo, MemGAS on LongMemEval); every LoCoMo gain over MemGAS re-run under our pipeline is significant at $p < 10^{-4}$ (paired bootstrap, 1,974 questions, App. C.4). Against turn-level Contriever, the strongest training-free dense alternative (Table 11), the lead is $+21.9$ / $+1.9$ Recall@3 and $+5.1$ / $+0.15$ QA-F1; IMPRINT also exceeds turn-level BM25 and the BM25 + Contriever hybrid that uses the same units and α ($+0.6$ / $+5.5$ Recall@3; App. C.5). Consolidated Memory Units contribute $+1.0$ QA-F1 on LoCoMo and $+0.2$ on LongMemEval, in line with how often turns merge (App. C.3).

6.2 The Granularity–Scoring Interaction

Figure 3 crosses segmentation granularity (session vs. turn, turn scores max-pooled to their session) with the scoring rule (dense, BM25, uniform, IMPRINT) under the same tokenizer, no dense complement and no truncation. Imprint weighting alone reaches 72.4 / 60.2 R@3 from a session-level index (the full system with $\alpha = 0$ gives 72.2 on LoCoMo; the two implementations differ only in unit filtering and tokenization details) and moves by only $+1.1$ / -3.2 when the unit shrinks to a turn, whereas turn-level units gain dense scoring $+4.0$ / $+29.6$ and BM25 $+13.1$ / $+15.1$: dense scoring needs short units, as Theorem 5.1 predicts, and empirically so does BM25 (App. A.7), whereas rarity weighting does not. Session-level BM25, the configuration tested by prior work, is therefore the one setting in which sparse retrieval fails. App. C.5 compares IMPRINT with the strongest training-free dense configuration, turn-level Contriever (Table 11), and gives the full analysis.

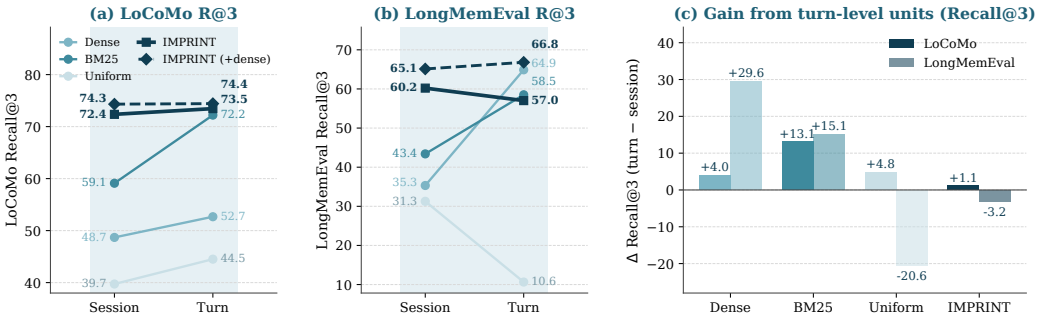


Figure 3: **Granularity × scoring interaction.** (a,b) Recall@3 at session vs. turn units. (c) Gain from turn-level units: large for dense and BM25, negligible for imprint scoring ("+dense": full system).

6.3 Ablations

We isolate three design choices: the weight formula, the dense–sparse mixing α, and per-category retrieval patterns. Figure 4 reports all three.

Weight factors. Removing all weighting (uniform) collapses R@1 by 25.1 (57.05 $\rightarrow$ 31.92); removing only session specificity costs 1.0; using the four-factor relational weight *at retrieval* costs 0.2, consistent with the two-vs-four split described in §4.3. The three rarity-weighted variants are within 0.1 of each other on Recall@3, so the choice among them matters only for the top rank.

Mixing coefficient. Pure imprint-following ($\alpha = 0$) already achieves R@3 $= 72.2$ on LoCoMo; the turn-level dense complement adds up to 2.3 points with a broad optimum around our a-priori $\alpha = 0.3$, and dense-dominant mixing ($\alpha = 0.7$) drops 6.5 points.

Per-category breakdown. Single-hop QA-F1 reaches 28.33 (EM 48.75), the highest of any category, because rare-entity matching is the most direct fit. Adversarial reaches 33.18 EM, indicating that sharper retrieval helps abstention. Temporal questions reach 16.1 F1, while open-domain questions remain challenging because they require reasoning *over* retrieved evidence beyond entity matching.

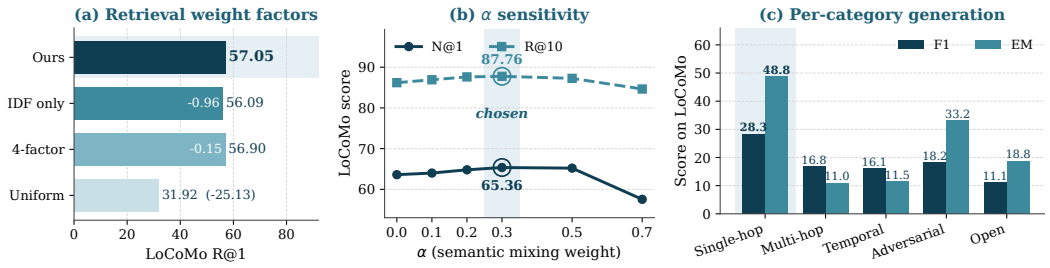
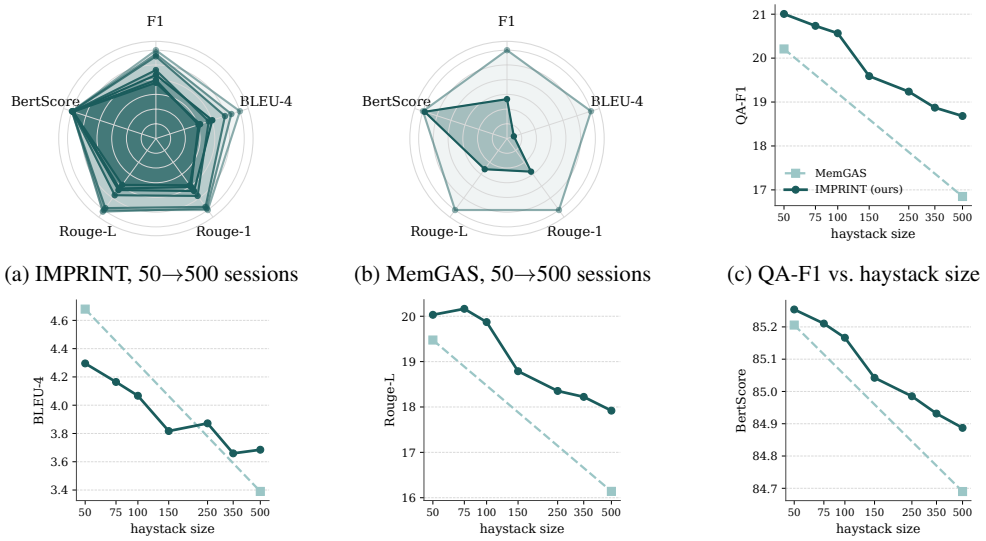


Figure 4: **Ablations.** (a) Weight variants on LoCoMo R@1. (b) α sensitivity on LoCoMo. (c) Per-category generation on LoCoMo.

6.4 Scaling with Conversation Length

We test nested LongMemEval subsets at $50, 75, 100, 150, 250, 350, 500$ sessions (Figure 5). (i) Every retriever we run loses 18–21 Recall@3 from 50 to 500 sessions (IMPRINT $85.1 \rightarrow 66.8$, turn-level Contriever $85.1 \rightarrow 64.9$, BM25 $76.6 \rightarrow 58.5$) with a stable ordering, so the loss is a property of the haystack. (ii) IMPRINT's lead over MemGAS is $+4.7$ Recall@3 at 50 sessions and $+15.8$ at 500, and its QA-F1 falls by 2.3 ($21.0 \rightarrow 18.7$) against MemGAS's 3.4 ($20.2 \rightarrow 16.9$; MemGAS's 50-session point is our re-run on LongMemEval's 50-session variant, its 500-session values are those reported by Xu et al. (2026a)). (iii) At 500 sessions HippoRAG 2 needs two LLM calls for each of 51,694 unique sessions, RAPTOR about 10^3 summaries and A-Mem about 5×10^3 sequential LLM calls per haystack (App. B.8); IMPRINT's table of token counts is extended, never rebuilt.



(a) IMPRINT, 50→500 sessions (b) MemGAS, 50→500 sessions (c) QA-F1 vs. haystack size

(d) BLEU-4 vs. haystack size (e) ROUGE-L vs. haystack size (f) BERTScore vs. haystack size

Figure 5: **Scaling 50–500 sessions.** (a,b) Radar plots, 50→500 sessions. (c–f) Per-metric trajectories.

7 Conclusion and Limitations

The bottleneck of long-term agent memory is the retrieval primitive, not the memory structure: we reframe recall as *imprint-following*, where rare user-specific tokens, rather than dense embeddings, carry the retrieval signal, and instantiate it in IMPRINT, whose single Token Profile Table serves as both an IDF-modulated inverted index and the driver of an online consolidation pool. Across LoCoMo and LongMemEval, IMPRINT improves over the strongest prior memory system by up to +15.75 Recall@3 and +3.92 QA-F1 while requiring no training, no LLM calls at indexing and no GPU at retrieval time, and it also outperforms BM25 and BM25–dense hybrids under an identical pipeline.

Three limitations point to next steps: retrievers trained for retrieval (SPLADE, ColBERTv2) may keep a recall edge that a training-free index cannot fully close; imprint-following helps least on temporal and open-domain questions, whose bottleneck is reasoning *over* the evidence; and the token statistics depend on a fixed English subword tokenizer, so adaptive or multilingual tokenization is future work.

AI USE STATEMENT

In this work, we used generative AI tools to formulate mathematical claims, provide ingredients for proving them, and implement methods. We also used them to polish the paper for readability. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including any text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

The retrieval side of IMPRINT is deterministic and requires no training. Appendix B lists every hyperparameter (Table 2), the complete construction and consolidation algorithms (Algorithms 1 and 2), the exact prompts of the two-stage generator, and the evaluation protocol; Appendix D documents dataset processing and statistics. All numbers in the main tables are produced by two scripts (`retrieval.py` and `generation.py`) contained in the supplementary code, together with the baseline runner used for the sparse and late-interaction comparisons.

ETHICS STATEMENT

This work uses only publicly available benchmarks (LoCoMo and LongMemEval) and introduces no new human-subject data. Because IMPRINT stores rare user-specific identifiers verbatim, a deployed Token Profile Table is personally identifying data; Appendix E discusses privacy risks and deployment-time mitigations (data minimization, user-controlled deletion, and audit logging).

REFERENCES

Petr Anokhin, Nikita Semenov, Artyom Y. Sorokin, Dmitry Evseev, Andrey Kravchenko, Mikhail Burtsev, and Evgeny Burnaev. Arigraph: Learning knowledge graph world models with episodic memory for LLM agents. In *Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2025, Montreal, Canada, August 16-22, 2025*, pp. 12–20. ijcai.org, 2025. doi: 10.24963/IJCAI.2025/2. URL https://doi.org/10.24963/ijcai.2025/2.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024. URL https://openreview.net/forum?id=hSyW5go0v8. Oral.

R.C̃. Atkinson and R.M̃. Shiffrin. Human memory: A proposed system and its control processes. In *The Psychology of Learning and Motivation: Volume II*. Academic Press, 1968. doi: 10.1016/S0079-7421(08)60422-3.

Alan Baddeley. Working memory: Theories, models, and controversies. *Annual Review of Psychology*, 63:1–29, 2012. doi: 10.1146/annurev-psych-120710-100422.

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. Longbench: A bilingual, multitask benchmark for long context understanding. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 3119–3137. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.172. URL https://doi.org/10.18653/v1/2024.acl-long.172.

Aydar Bulatov, Yuri Kuratov, and Mikhail Burtsev. Recurrent memory transformer. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=Uynr3iPhksa.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. Mem0: Building production-ready AI agents with scalable long-term memory. In Inês Lynce, Nello Murano, Mauro Vallati, Serena Villata, Federico Chesani, Michela Milano, Andrea Omicini, and Mehdi Dastani (eds.), *ECAI 2025 - 28th European Conference on Artificial Intelligence, 25-30 October 2025,*

Bologna, Italy - Including 14th Conference on Prestigious Applications of Intelligent Systems (PAIS 2025), Frontiers in Artificial Intelligence and Applications, pp. 2993–3000. IOS Press, 2025. doi: 10.3233/FAIA251160. URL https://doi.org/10.3233/FAIA251160.

Zhuyun Dai and Jamie Callan. Context-aware term weighting for first stage passage retrieval. In Jimmy X. Huang, Yi Chang, Xueqi Cheng, Jaap Kamps, Vanessa Murdock, Ji-Rong Wen, and Yiqun Liu (eds.), *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval, SIGIR 2020, Virtual Event, China, July 25-30, 2020*, pp. 1533–1536. ACM, 2020. doi: 10.1145/3397271.3401204. URL https://doi.org/10.1145/3397271.3401204.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, and Jonathan Larson. From local to global: A graph RAG approach to query-focused summarization. *CoRR*, abs/2404.16130, 2024. doi: 10.48550/ARXIV.2404.16130. URL https://doi.org/10.48550/arXiv.2404.16130.

Jizhan Fang, Xinle Deng, Haoming Xu, Ziyan Jiang, Yuqi Tang, Ziwen Xu, Shumin Deng, Yunzhi Yao, Mengru Wang, Shuofei Qiao, Huajun Chen, and Ningyu Zhang. Lightmem: Lightweight and efficient memory-augmented generation. *CoRR*, abs/2510.18866, 2025a. doi: 10.48550/ARXIV.2510.18866. URL https://doi.org/10.48550/arXiv.2510.18866.

Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. Memp: Exploring agent procedural memory. *CoRR*, abs/2508.06433, 2025b. doi: 10.48550/ARXIV.2508.06433. URL https://doi.org/10.48550/arXiv.2508.06433.

Thibault Formal, Carlos Lassance, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE v2: Sparse lexical and expansion model for information retrieval. *CoRR*, abs/2109.10086, 2021a. URL https://arxiv.org/abs/2109.10086.

Thibault Formal, Benjamin Piwowarski, and Stéphane Clinchant. SPLADE: sparse lexical and expansion model for first stage ranking. In Fernando Diaz, Chirag Shah, Torsten Suel, Pablo Castells, Rosie Jones, and Tetsuya Sakai (eds.), *SIGIR '21: The 44th International ACM SIGIR Conference on Research and Development in Information Retrieval, Virtual Event, Canada, July 11-15, 2021*, pp. 2288–2292. ACM, 2021b. doi: 10.1145/3404835.3463098. URL https://doi.org/10.1145/3404835.3463098.

Zafeirios Fountas, Martin Benfeghoul, Adnan Oomerjee, Fenia Christopoulou, Gerasimos Lampouras, Haitham Bou-Ammar, and Jun Wang. Human-inspired episodic memory for infinite context llms. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=BI2int5SAC.

Luyu Gao, Zhuyun Dai, and Jamie Callan. COIL: revisit exact lexical match in information retrieval with contextualized inverted list. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tür, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2021, Online, June 6-11, 2021*, pp. 3030–3042. Association for Computational Linguistics, 2021. doi: 10.18653/V1/2021.NAACL-MAIN.241. URL https://doi.org/10.18653/v1/2021.naacl-main.241.

Tao Ge, Jing Hu, Lei Wang, Xun Wang, Si-Qing Chen, and Furu Wei. In-context autoencoder for context compression in a large language model. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=uREj4ZuGJE.

Zirui Guo, Lianghao Xia, Yanhua Yu, Tu Ao, and Chao Huang. Lightrag: Simple and fast retrieval-augmented generation. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 10746–10761. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.findings-emnlp.568/.

Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/6ddc001d07ca4f319af96a3024f6dbd1-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 Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 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://openreview.net/forum?id=DT7JyQC3MR.

Yuyang Hu, Shichun Liu, Yanwei Yue, Guibin Zhang, Boyang Liu, Fangyi Zhu, Jiahang Lin, Honglin Guo, Shihan Dou, Zhiheng Xi, Senjie Jin, Jiejun Tan, Yanbin Yin, Jiongnan Liu, Zeyu Zhang, Zhongxiang Sun, Yutao Zhu, Hao Sun, Boci Peng, Zhenrong Cheng, Xuanbo Fan, Jiaxin Guo, Xinlei Yu, Zhenhong Zhou, Zewen Hu, Jiahao Huo, Junhao Wang, Yuwei Niu, Yu Wang, Zhenfei Yin, Xiaobin Hu, Yue Liao, Qiankun Li, Kun Wang, Wangchunshu Zhou, Yixin Liu, Dawei Cheng, Qi Zhang, Tao Gui, Shirui Pan, Yan Zhang, Philip Torr, Zhicheng Dou, Ji-Rong Wen, Xuanjing Huang, Yu-Gang Jiang, and Shuicheng Yan. Memory in the age of AI agents. *CoRR*, abs/2512.13564, 2025. doi: 10.48550/ARXIV.2512.13564. URL https://doi.org/10.48550/arXiv.2512.13564.

Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. Unsupervised dense information retrieval with contrastive learning. *Trans. Mach. Learn. Res.*, 2022, 2022. URL https://openreview.net/forum?id=jKN1pXi7b0.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Llmlingua: Compressing prompts for accelerated inference of large language models. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pp. 13358–13376. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.EMNLP-MAIN.825. URL https://doi.org/10.18653/v1/2023.emnlp-main.825.

Huiqiang Jiang, Qianhui Wu, Xufang Luo, Dongsheng Li, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. Longllmlingua: Accelerating and enhancing llms in long context scenarios via prompt compression. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 1658–1677. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.91. URL https://doi.org/10.18653/v1/2024.acl-long.91.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. Memory OS of AI agent. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pp. 25961–25970. Association for Computational Linguistics, 2025. doi: 10.18653/V1/2025.EMNLP-MAIN.1318. URL https://doi.org/10.18653/v1/2025.emnlp-main.1318.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. Dense passage retrieval for open-domain question answering. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu (eds.), *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020*, pp. 6769–6781. Association for Computational Linguistics, 2020. doi:

10.18653/V1/2020.EMNLP-MAIN.550. URL https://doi.org/10.18653/v1/2020.emnlp-main.550.

Dharshan Kumaran, Demis Hassabis, and James L. McClelland. What learning systems do intelligent agents need? complementary learning systems theory updated. *Trends in Cognitive Sciences*, 20 (7):512–534, 2016. doi: 10.1016/j.tics.2016.05.004.

Gibbeum Lee, Volker Hartmann, Jongho Park, Dimitris Papailiopoulos, and Kangwook Lee. Prompted llms as chatbot modules for long open-domain conversation. In Anna Rogers, Jordan L. Boyd-Graber, and Naoaki Okazaki (eds.), *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, Findings of ACL, pp. 4536–4554. Association for Computational Linguistics, 2023. doi: 10.18653/V1/2023.FINDINGS-ACL.277. URL https://doi.org/10.18653/v1/2023.findings-acl.277.

Kuang-Huei Lee, Xinyun Chen, Hiroki Furuta, John F. Canny, and Ian Fischer. A human-inspired reading agent with gist memory of very long contexts. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 26396–26415. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/lee24c.html.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Jiaqi Liu, Yaofeng Su, Peng Xia, Siwei Han, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. Simplemem: Efficient lifelong memory for LLM agents. *CoRR*, abs/2601.02553, 2026. doi: 10.48550/ARXIV.2601.02553. URL https://doi.org/10.48550/arXiv.2601.02553.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. Evaluating very long-term conversational memory of LLM agents. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pp. 13851–13870. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.ACL-LONG.747. URL https://doi.org/10.18653/v1/2024.acl-long.747.

Jiayan Nan, Wenquan Ma, Wenlong Wu, and Yize Chen. Nemori: Self-organizing agent memory inspired by cognitive science. *CoRR*, abs/2508.03341, 2025. doi: 10.48550/ARXIV.2508.03341. URL https://doi.org/10.48550/arXiv.2508.03341.

Charles Packer, Vivian Fang, Shishir G. Patil, Kevin Lin, Sarah Wooders, and Joseph E. Gonzalez. Memgpt: Towards llms as operating systems. *CoRR*, abs/2310.08560, 2023. doi: 10.48550/ARXIV.2310.08560. URL https://doi.org/10.48550/arXiv.2310.08560.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Menglin Xia, Xufang Luo, Jue Zhang, Qingwei Lin, Victor Rühle, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Dongmei Zhang. Llmlingua-2: Data distillation for efficient and faithful task-agnostic prompt compression. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, Findings of ACL, pp. 963–981. Association for Computational Linguistics, 2024. doi: 10.18653/V1/2024.FINDINGS-ACL.57. URL https://doi.org/10.18653/v1/2024.findings-acl.57.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H. Vicky Zhao, Lili Qiu, and Jianfeng Gao. Secom: On memory construction and retrieval for personalized conversational agents. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=xKDZAW0He3.

Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for llms. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025. URL https://openreview.net/forum?id=moXtEmCleY.

Stephen E. Robertson and Hugo Zaragoza. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389, 2009. doi: 10.1561/1500000019. URL https://doi.org/10.1561/1500000019.

Keshav Santhanam, Omar Khattab, Jon Saad-Falcon, Christopher Potts, and Matei Zaharia. ColBERTv2: Effective and efficient retrieval via lightweight late interaction. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT)*, pp. 3715–3734, 2022.

Parth Sarthi, Salman Abdullah, Aditi Tuli, Shubh Khanna, Anna Goldie, and Christopher D. Manning. RAPTOR: recursive abstractive processing for tree-organized retrieval. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net, 2024. URL https://openreview.net/forum?id=GN921JHCRw.

Eleanor Spens and Neil Burgess. A generative model of memory construction and consolidation. *Nature Human Behaviour*, 8(3):526–543, 2024. doi: 10.1038/s41562-023-01799-z.

Nandan Thakur, Nils Reimers, Andreas Rücklé, Abhishek Srivastava, and Iryna Gurevych. BEIR: A heterogeneous benchmark for zero-shot evaluation of information retrieval models. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2021.

Endel Tulving. Episodic and semantic memory. In Endel Tulving and Wayne Donaldson (eds.), *Organization of Memory*, pp. 381–403. Academic Press, 1972.

Endel Tulving. Episodic memory: From mind to brain. *Annual Review of Psychology*, 53:1–25, 2002.

Qingyue Wang, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing*, 639:130193, 2025a. doi: 10.1016/J.NEUCOM.2025.130193. URL https://doi.org/10.1016/j.neucom.2025.130193.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, Jingbo Shang, and Julian J. McAuley. MEMORYLLM: towards self-updatable large language models. In Ruslan Salakhutdinov, Zico Kolter, Katherine A. Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pp. 50453–50466. PMLR / OpenReview.net, 2024. URL https://proceedings.mlr.press/v235/wang24s.html.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian J. McAuley, and Xiaojian Wu. Mem-α: Learning memory construction via reinforcement learning. *CoRR*, abs/2509.25911, 2025b. doi: 10.48550/ARXIV.2509.25911. URL https://doi.org/10.48550/arXiv.2509.25911.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. Agent workflow memory. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu (eds.), *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net, 2025c. URL https://proceedings.mlr.press/v267/wang25bx.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, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net, 2025a. URL https://openreview.net/forum?id=pZiyCaVuti.

Yaxiong Wu, Yongyue Zhang, Sheng Liang, and Yong Liu. Sgmem: Sentence graph memory for long-term conversational agents. *CoRR*, abs/2509.21212, 2025b. doi: 10.48550/ARXIV.2509.21212. URL https://doi.org/10.48550/arXiv.2509.21212.

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 *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=i2yIvZARnG.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for LLM agents. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026b. URL https://openreview.net/forum?id=FiM0M8gcct.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Hinrich Schütze, Volker Tresp, and Yunpu Ma. Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *CoRR*, abs/2508.19828, 2025. doi: 10.48550/ARXIV.2508.19828. URL https://doi.org/10.48550/arXiv.2508.19828.

Hongli Yu, Tinghong Chen, Jiangtao Feng, Jiangjie Chen, Weinan Dai, Qiying Yu, Ya-Qin Zhang, Wei-Ying Ma, Jingjing Liu, Mingxuan Wang, and Hao Zhou. Memagent: Reshaping long-context LLM with multi-conv RL-based memory agent. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=k5nIOvYGCL.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. A survey on the memory mechanism of large language model-based agents. *ACM Trans. Inf. Syst.*, 43(6):155:1–155:47, 2025. doi: 10.1145/3748302. URL https://doi.org/10.1145/3748302.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. Expel: LLM agents are experiential learners. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pp. 19632–19642. AAAI Press, 2024. doi: 10.1609/AAAI.V38I17.29936. URL https://doi.org/10.1609/aaai.v38i17.29936.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. Memorybank: Enhancing large language models with long-term memory. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan (eds.), *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pp. 19724–19731. AAAI Press, 2024. doi: 10.1609/AAAI.V38I17.29946. URL https://doi.org/10.1609/aaai.v38i17.29946.

A Extended Theoretical Discussion

This appendix gives the full proof of Theorem 5.1 (§A.1–A.2), extends it to multi-imprint queries (§A.3), spells out the role of session specificity (§A.4), and discusses the tokenizer (§A.5), the scope of the dense model (§A.6), BM25 (§A.7) and the consolidation weight (§A.8).

A.1 Proof of Theorem 5.1

Proof of Theorem 5.1. (a) Imprint. Restricting Eq. (6) to $t^\star$ gives $s_{\text{IMP}}(q,u) = w(t^\star)\log(1+\mathrm{tf}(t^\star,u))$ with $w(t^\star)=\mathrm{IDF}(t^\star)\rho(t^\star)$ computed on the realized memory; $\mathrm{IDF}(t^\star)=\log\frac{N+1}{\mathrm{df}(t^\star)+1}>0$ as soon as $\mathrm{df}(t^\star)<N$, and $\rho(t^\star)\geq\frac{1}{2}$, so $w(t^\star)>0$. The target contains the planted copy, hence $\mathrm{tf}(t^\star,u^+)\geq 1$ and $s_{\text{IMP}}(q,u^+)\geq w(t^\star)\log 2$; a distractor without $t^\star$ has $\mathrm{tf}=0$ and scores exactly 0. This proves Eq. (16) deterministically, for every L. A distractor contains $t^\star$ in a given slot with probability ε, so by a union bound over its L slots it contains $t^\star$ with probability at most εL, and the expected number of such distractors among M candidates is at most $M\varepsilon L$. Only these can tie with or outrank the target: chance copies raise a distractor to $w(t^\star)\log(1+\mathrm{tf})$, while the target scores $w(t^\star)\log(2+X)$ with X its own number of chance copies, and neither quantity depends on the $L-1$ (respectively L) non-matching slots. No limit in N is required.

(b) Dense. Write $\phi(u)=\frac{1}{L}\sum_i h_i$. In u^+ the planted slot contributes $\langle h^\star,h^\star\rangle/L=1/L$ to the numerator and each other slot contributes $\langle h^\star,h_i\rangle/L$, which has mean 0 and variance $1/(dL^2)$ by isotropy; in u^- all L slots are of the latter kind. Hence $\mathbb{E}\langle h^\star,\phi(u^+)\rangle-\mathbb{E}\langle h^\star,\phi(u^-)\rangle=1/L$ exactly. Because the slot vectors are independent and isotropic, $\mathbb{E}\|\phi(u)\|^2=\frac{1}{L^2}\sum_i\|h_i\|^2=1/L$ for both unit types (the planted vector $h^\star$ is independent of the other slots), and §A.2 shows $\|\phi(u)\|=L^{-1/2}\big(1+O(d^{-1/2})\big)$ with probability $1-e^{-\Omega(d)}$. Dividing the numerator gap by this norm gives the first part of Eq. (17). For the noise, $\langle h^\star,\phi(u^-)\rangle$ is a mean of L independent terms of variance $1/d$, so its standard deviation is $(Ld)^{-1/2}$, which becomes $\Theta(d^{-1/2})$ after division by $\|\phi(u^-)\|=\Theta(L^{-1/2})$. The margin-to-noise ratio is therefore $\Theta(L^{-1/2}/d^{-1/2})=\Theta(\sqrt{d/L})$. □

A.2 Norm Concentration for Mean-Pooled Units

Let $\bar h=\frac{1}{L}\sum_{i=1}^L h_i$ with h_i independent and uniform on $\mathbb{S}^{d-1}$ (one slot may instead be the fixed unit vector $h^\star$, which changes nothing below). Then

$$\mathbb{E}\|\bar h\|^2=\tfrac{1}{L^2}\Big(\sum_{i=1}^L\|h_i\|^2+2\sum_{i<j}\mathbb{E}\langle h_i,h_j\rangle\Big)=\tfrac{1}{L^2}\cdot L=\tfrac{1}{L},$$

because the cross terms vanish by isotropy and independence. For concentration, each $\langle h_i,h_j\rangle$ $(i\neq j)$ is sub-Gaussian with variance proxy $1/d$ by Lévy's lemma and the map $(h_1,\dots,h_L)\mapsto\|\bar h\|$ is L^{-1}-Lipschitz in each argument, hence $\big|\|\bar h\|-L^{-1/2}\big|=O(d^{-1/2})$ with probability $1-e^{-\Omega(d)}$. The bag-of-random-vectors model assigns independent vectors to slots even when two slots carry the same token; with a fixed embedding per token type, repeated tokens add non-negative cross terms, which *increase* $\|\bar h\|$ and therefore only shrink the dense margin further, so Eq. (17) is an upper bound on the dense margin in that case.

A.3 Multi-Imprint Queries

The main theorem treats a single imprint. Practical queries (e.g., *"dentist appointment three weeks ago"*) carry several. Let q contain m imprints $t_1^\star,\dots,t_m^\star$ with masses ε_j, all planted in u^+, and assume no chance repetitions, i.e. $\mathrm{tf}(t_j^\star,u)\leq 1$ for every imprint and unit (the event that a unit repeats an imprint has probability $O(\varepsilon_j^2L^2)$).

Corollary A.1 (Additive composition)**.** *For every distractor that contains none of the imprints, $s_{\text{IMP}}(q,u^+)-s_{\text{IMP}}(q,u^-)\geq\log 2\sum_{j=1}^m w(t_j^\star)$, independently of L, and a distractor can tie with the target only if it contains every imprint, an event of probability at most $\prod_j\varepsilon_jL$. Under the dense model with $d\gg m$, the expected margin is $\Theta(\sqrt{m/L})$ while the per-distractor noise stays $\Theta(d^{-1/2})$.*

Proof. s_{IMP} is a sum of per-token contributions, so the bound of Theorem 5.1(a) applies to each imprint and adds up; with $\mathrm{tf} \le 1$ every imprint contributes either 0 or $w(t_j^\star)\log 2$, so a distractor that misses one imprint trails the target by $w(t_j^\star)\log 2$ on that token and can match it nowhere else. For the dense rule, the query embedding is $\phi(q) = \frac{1}{m}\sum_j h_j^\star$ with $\|\phi(q)\| = \Theta(m^{-1/2})$ under near-orthogonality ($d \gg m$); the normalized query has m matching contributions of size $\Theta(1/(\sqrt{m}L))$ in $\langle \cdot, \phi(u^+)\rangle$, and dividing by $\|\phi(u^+)\| = \Theta(L^{-1/2})$ gives $\Theta(\sqrt{m/L})$; the noise computation is unchanged. □

Additional imprints therefore both enlarge the sparse margin additively and shrink the set of units that can tie with the target multiplicatively, whereas they help the dense rule only as $\sqrt{m}$ and do not remove its $L^{-1/2}$ decay.

A.4 Effect of Session Specificity

The bound keeps $\rho(t^\star) \in [\frac{1}{2}, 1]$ as a factor. Substituting the explicit form sharpens it: if $t^\star$ appears in k of N sessions, $\rho(t^\star) = \frac{1}{2} + \frac{1}{2}\big(1 - H_b(k/N)/\log 2\big)$, so $\rho \to 1$ as $k/N \to 0$ (the personal-memory regime, where each rare token marks one episode) and $\rho = \frac{1}{2}$ at the maximum-entropy point $k/N = \frac{1}{2}$. The two factors of $w(t)$ thus reinforce each other precisely for session-concentrated imprints, while a token spread over half of the sessions is down-weighted by a factor of 2 on top of its already small IDF; a token present in every session has $\mathrm{IDF} = 0$ and drops out of the score altogether.

A.5 Role of the Tokenizer

The analysis is stated for whatever token inventory the tokenizer produces: an imprint is a *token* that is rare in the memory. Changing the tokenizer changes which strings are imprints. A rare word that one tokenizer keeps whole may be split by another into common subwords ("Zanzibar" → "Zan"+"zi"+"bar"), in which case its imprint disappears and only the co-occurrence of the pieces, captured by the multi-token score, remains; conversely a tokenizer with a larger vocabulary creates more imprints. We make no robustness claim across tokenizers. All results use the Contriever WordPiece tokenizer, which is also the tokenizer of the dense channel, so both channels see the same units.

A.6 Scope of the Dense Model

The bag-of-random-vectors model isolates one mechanism, dilution by mean pooling, and deliberately omits two properties of real encoders. First, it has no semantic structure: distinct tokens are near-orthogonal, so the model cannot describe how an encoder places a rare identifier next to topically similar distractors; this is why the model's constants are not predictive and why, on LoCoMo, turn-level dense scoring still trails imprint scoring by 22 Recall@3 although turns are short (App. C.5). Second, BERT-style encoders are anisotropic: token states share a large common component, so $\|\phi(u)\|$ does not shrink as $L^{-1/2}$ and the per-distractor noise is not $d^{-1/2}$. The argument survives after removing the common component (mean-centering, as is standard practice), because it is the centered residual that carries the identifier; the qualitative content, that the identifier's share of the pooled vector decays as $1/L$ while the distractor noise does not, is what §6.2 tests, and the numbers there (dense $+4.0$ / $+29.6$ R@3 from turn-level units, imprint $+1.1$ / -3.2) agree with the direction, not with any constant. Eq. (17) is stated for mean pooling, the standard for session encoding in MemoryBank, A-Mem, and MemGAS. Max or attention pooling can in principle preserve a $\Theta(1)$ signal independent of L, but at the cost of either losing multi-topic capacity in a single vector or requiring per-query attention over the full session, which raises retrieval cost from $O(1)$ to $O(L)$ per session.

A.7 Empirical Note on BM25

Theorem 5.1 compares imprint scoring with dense scoring; it makes no statement about BM25 or TF–IDF, and for a single-imprint query a BM25 score is also length-robust up to its length normalization. The strong granularity dependence of BM25 that we observe (Fig. 3: $+13.1$ and

+15.1 R@3 from turn-level units) is therefore an empirical finding. Two of its ingredients can be named. Under BM25 the contribution of the rare match is multiplied by the length-normalization factor $(k_1 + 1)/(1 + k_1(1 - b + bL/\overline{L}))$, so a target session longer than average is penalized on exactly the token that identifies it, and the term-frequency factor of common query tokens saturates at $k_1 + 1$, so a session that happens to contain several of them collects a score comparable to one rare match. Under IMPRINT there is no length normalization, the rare match contributes $w(t^\star)\log(1+\mathrm{tf})$ regardless of L, and a token present in every session receives weight 0 under Eq. (3). The uniform scorer in Fig. 3 (all weights equal) is the extreme case in which common tokens dominate, and it benefits most from fine granularity while remaining far below every rarity-weighted rule. Which of these ingredients dominates on a given corpus we do not claim to know; the evidence for the ordering of the scorers is the experiment.

A.8 Connection to the Consolidation Weight

Theorem 5.1 concerns the retrieval weight $w(t)$. The consolidation weight $w_{\text{cons}}(t) = w(t)(1 + \beta_c c(t))(1 + \beta_r r(t))$ of Eq. (12) is used only to decide which tokens survive a merge (Algorithm 2) and never enters retrieval scoring. Because the two bonuses are bounded, $w_{\text{cons}}(t) \in [w(t),\ (1 + \beta_c)(1 + \beta_r)\, w(t)] = [w(t),\ 1.56\, w(t)]$ for $\beta_c = 0.3$, $\beta_r = 0.2$, a common token can only displace a rare token from a merged unit if their retrieval weights are already within a factor 1.56 of each other. Retention therefore preserves, up to this constant, the tokens with the largest imprint margin, and among tokens of comparable rarity it prefers those the user has returned to ($\kappa(t) > 0$) and those that connect diverse contexts.

B Implementation Details

This appendix provides complete implementation details for reproducibility. All hyperparameters are fixed across datasets unless stated otherwise. Section references point to the main paper; the released code base is ∼1,500 lines of Python.

B.1 Hyperparameter Settings

Table 2 lists every tunable quantity in IMPRINT. All values are held fixed across LoCoMo and LongMemEval (and all of its nested subsets).

Note

The four primary parameters $\{\alpha, \tau, \delta, \text{floor}\}$ are the only quantities that need attention before running IMPRINT. All are fixed at the values above. The remaining knobs (consolidation dampening, co-occurrence saturation, generation budgets) act on auxiliary parts of the pipeline and were not tuned per dataset; sensitivity analyses for α and the affine floor are reported in §C.1.

B.2 Token Profile Table Construction

The Token Profile Table $\mathcal{P}$ is built in a single left-to-right pass over the session stream. Algorithm 1 gives the complete procedure.

Complexity. Each turn insertion costs $O(|\mathcal{T}(u)|)$ for profile updates and $O(|\mathcal{C}|)$ for the nearest-neighbour lookup in the pool. Total construction time is $O\Big(\sum_{i,j} |\mathcal{T}(u_j^{(i)})| + M \cdot |\mathcal{C}|\Big)$, where M is the total number of turns. The profile itself occupies $O(|\mathcal{V}| \cdot N)$ memory; in practice $|\mathcal{V}| \ll 30{,}000$ for a single conversation.

B.3 Consolidation Pool Algorithm

Table 2: **Complete hyperparameter table.** Colours indicate the component each parameter belongs to. All values are fixed; no dataset-specific tuning was performed.

Component	Parameter	Value	Description
	Mixing coefficient α	0.3	Dense weight in Eq. (8); α=0 gives sparse-only
	Affine floor	0.5	$\rho(t) = \frac{1+\mathrm{SessSpec}(t)}{2}$; see Eq. (5)
	Max query tokens	128	Tokenizer truncation for query encoding
	Max unit tokens	128	Tokenizer truncation per memory unit
	Dense channel unit	turn	Turn embeddings max-pooled per session (Eq. 8)
	Merge threshold τ	0.75	Cosine similarity gate in Eq. (11)
	Depth increment δ	0.5	Additive depth growth per merge
	Token capacity $K_{\max}$	120	Max tokens retained per pool unit after pruning
	Consol. dampening β_c	0.3	Coefficient on $c(t)$ in $w_c(t)$ (Eq. (12))
	Co-occ. dampening β_r	0.2	Coefficient on $r(t)$ in $w_c(t)$ (Eq. (12))
	Co-occ. saturation $C_{\max}$	20	Denominator in $r(t) = \min(\lvert\mathcal{N}(t)\rvert/C_{\max}, 1)$
	Top-k sessions	3	Sessions passed to filter+generate pipeline
	Min. surfaced depth	1.0	Only consolidated entries ($d_e \geq 1$, i.e. ≥ 2 merges) become Memory Units
	Max pool texts (filter) $U_{\max}$	60	Depth-sorted, de-duplicated pool texts fed to the filter stage
	Max pool units per session	5	Deepest pool units logged per retrieved session
	Generator max tokens	4,000	LLM generation budget (both stages)
	Dense backbone	Contriever	`facebook/contriever`
	Embedding dimension d	768	Hidden state dimension
	Embedding batch size	64	Texts per forward pass

Auxiliary factors. The consolidation count $c(t)$ and co-occurrence richness $r(t)$ are defined as

$$c(t) \;=\; \frac{\kappa[t]}{1+\kappa[t]}\,, \qquad r(t) \;=\; \min\!\big(\,|\mathcal{N}(t)|/C_{\max},\,1\big)\,, \tag{18}$$

where $\kappa[t]$ is the number of merges in which t has participated, $\mathcal{N}(t)$ is the set of tokens that have ever co-occurred with t inside a unit, and $C_{\max} = 20$ is a fixed saturation point. Both $c(t)$ and $r(t)$ lie in $[0, 1]$.

Dampening coefficients. The dampening coefficients $\beta_c = 0.3$ and $\beta_r = 0.2$ keep the multiplicative bonuses close to unity, so that the auxiliary factors shape token retention but do not dominate the IDF-driven base weight. Using the four-factor weight at retrieval time instead of the 2-factor weight costs 0.2 R@1 (Fig. 4a), so keeping the bonuses small and confining them to token retention is the conservative choice.

Retrieval vs. consolidation weight. Algorithm 2 uses $w_c(t)$ *only* to decide which tokens to retain inside a merged unit. The retrieval scoring at query time uses the strictly simpler 2-factor weight $w(t) = \mathrm{IDF}(t) \cdot \rho(t)$ from Eq. (5); this separation is empirically validated by the weight-factor ablation in the main text.

Algorithm 1 Incremental Token Profile Table Construction

Require: Sessions $\{S_1, \ldots, S_N\}$; tokenizer $\mathcal{T}$; dense encoder ϕ with hidden states $h(\cdot)$
Ensure: Profile $\mathcal{P}$; consolidated pool $\mathcal{C}$

Initialise $\mathcal{P}$: freq $\leftarrow \mathbf{0}$, sess $\leftarrow \emptyset$, $\kappa \leftarrow \mathbf{0}$, $\mathcal{N} \leftarrow \emptyset$, $\mathcal{C} \leftarrow \emptyset$
for $i = 1 \to N$ **do**
  for each turn $u \in S_i$ **do**
    $\{t_1, \ldots, t_\ell\} \leftarrow \mathcal{T}(u)$ ▷ Subword tokenisation, strip special tokens
    $h_1, \ldots, h_\ell \leftarrow \phi(u)$ ▷ Token-level hidden states, §B.4
    for $t \in \{t_1, \ldots, t_\ell\}$ **do**
      $\text{freq}[t] \mathrel{+}= 1$
      $\text{sess}[t] \cup= \{i\}$
      $\mathcal{N}[t] \cup= \{t_1, \ldots, t_\ell\} \setminus \{t\}$
    end for
    $e_{\text{new}} \leftarrow \textsc{MakeUnit}(u, \{t_j\}, \{h_j\}, i)$
    $\mathcal{C} \leftarrow \textsc{Consolidate}(\mathcal{C}, e_{\text{new}}, \mathcal{P})$ ▷ Algorithm 2
  end for
  $\mathcal{P}.n_\text{sessions} \leftarrow i$
  $\textsc{RecomputeEmbeddings}(\mathcal{C}, \mathcal{P})$ ▷ Refresh IDF-weighted pool embeddings after each session
end for

Algorithm 2 Online Pool Consolidation

Require: New unit e_{new}; pool $\mathcal{C}$; profile $\mathcal{P}$; threshold τ; capacity $K_{\max}$; depth increment δ; dampening coefficients β_c, β_r
Ensure: Updated pool $\mathcal{C}$; updated $\mathcal{P}.\kappa$

if $|\mathcal{C}| = 0$ **then**
  $\mathcal{C} \leftarrow \{e_{\text{new}}\}$; **return**
end if
$e^\star \leftarrow \arg\max_{e \in \mathcal{C}} \cos\big(\phi(e_{\text{new}}),\, \phi(e)\big)$
if $\cos\big(\phi(e_{\text{new}}),\, \phi(e^\star)\big) > \tau$ **then**
  $T_{\text{merged}} \leftarrow T_{e^\star} \cup T_{e_{\text{new}}}$
  if $|T_{\text{merged}}| > K_{\max}$ **then**
    $w_c(t) \leftarrow w(t) \cdot \big(1 + \beta_c\, c(t)\big) \cdot \big(1 + \beta_r\, r(t)\big)$ ▷ 4-factor consolidation weight, Eq. (12)
    $T_{\text{merged}} \leftarrow \text{top-}K_{\max}\big(\{(t, w_c(t))\}_{t \in T_{\text{merged}}}\big)$
  end if
  for $t \in T_{e^\star} \cup T_{e_{\text{new}}}$ **do**
    $\kappa[t] \mathrel{+}= 1$
  end for
  $e^\star.T \leftarrow T_{\text{merged}}$; $e^\star.R \leftarrow e^\star.R \,\|\, e_{\text{new}}.R$
  $e^\star.d \leftarrow e^\star.d + \delta$; $e^\star.n_\text{cons} \mathrel{+}= 1$
  $e^\star.\text{sessions} \cup= e_{\text{new}}.\text{sessions}$
  $\textsc{RecomputeEmbedding}(e^\star, \mathcal{P})$
else
  $\mathcal{C} \leftarrow \mathcal{C} \cup \{e_{\text{new}}\}$
end if

IDF-weighted embedding. After each merge or session boundary, pool embeddings are recomputed as:

$$\phi(e) = \frac{\sum_{t \in T_e} \text{IDF}(t) \cdot h(t)}{\sum_{t \in T_e} \text{IDF}(t)}, \tag{19}$$

where $h(t)$ is the dense encoder's last hidden state for token t, extracted once at unit-creation time and cached. Recomputation uses updated IDF statistics after each session so that pool geometry remains aligned with the evolving profile.

B.4 Encoder Details

Token hidden states. Per-token hidden states $h(t)$ are obtained by a single forward pass through Contriever. Special tokens (`[CLS]`, `[SEP]`, `[PAD]`) are stripped before indexing into the hidden-state matrix; only content tokens contribute to pool embeddings and IDF-weighted unit embeddings.

Turn embeddings. For the optional dense channel, every turn unit is embedded once at indexing time by mean-pooling its non-special token hidden states with attention-mask weighting (128-token truncation) followed by ℓ_2 normalisation (standard Contriever inference), in batches of 64. A session's dense score is the maximum cosine over its turns.

Alternative encoders. The codebase exposes Contriever via the alias `contriever` and accepts any other HuggingFace encoder path through the `-retriever` flag; every result in this paper uses Contriever.

B.5 Retrieval Scoring Pipeline

At query time, given a query string q:

1. **Tokenise.** $\{t_1, \ldots, t_m\} \leftarrow \mathcal{T}(q)$, stripping all special tokens.
2. **Sparse score.** For each unique query token t, look up its posting list $\mathrm{tf}(t, S_i)$ for all i, accumulate $s_{\text{IMP}}(q, S_i)$ $+= w(t) \cdot \log(1 + \mathrm{tf}(t, S_i))$. Cost: $O(\sum_t |\mathrm{sess}(t)|)$, no GPU required.
3. **Dense score** (if $\alpha > 0$). Encode q (one forward pass); turn embeddings are precomputed at indexing time (App. B.4). The session score is the maximum cosine over its turns.
4. **Normalise.** Min-max normalise sparse and dense score vectors independently over all N candidate sessions.
5. **Combine.** $s(q, S_i) = \alpha\, \tilde{s}_{\text{dense}} + (1 - \alpha)\, \tilde{s}_{\text{IMP}}$.
6. **Rank and collect pool context.** Sort sessions by $s(q, \cdot)$ descending; for the top-k sessions, retrieve their pool units sorted by depth descending.

B.6 Generation Prompts

Every method in Table 1 that we run ourselves (IMPRINT and the sparse, hybrid, and late-interaction baselines) uses the same two-stage filter–generate pipeline, the same generator (GPT-4o-mini, temperature 0), the same top-3 retrieved sessions, and the same Stage-2 `PROMPT_GENERATE`. The Stage-1 `PROMPT_FILTER` of IMPRINT differs from the baselines' filter prompt only by the additional *Memory Units* block that exposes the consolidated pool to the LLM (the baseline prompt omits that block and the two sentences that refer to it); this is the only prompt-level difference in our pipeline, and it is necessary because the pool is a new artefact produced by IMPRINT. All prompts are given below.

Stage 1 — Filter Prompt (`PROMPT_FILTER`)

```
You are an intelligent dialog bot.  You will be shown Consolidated
Memory Units and History Dialogs from past conversations.

The Memory Units are pre-extracted key information, sorted by
importance.  The History Dialogs provide full context.

Filter and combine the Memory Units and History Dialogs to extract
only the parts directly relevant to the Question.  Prioritize Memory
Units as they contain the most important information.  Preserve
original tokens, do not paraphrase.
```

```
Memory Units: {pool_context}
History Dialogs: {session_context}
Question Date: {question_date}
Question: {question}
Answer:
```

Stage 2 — Generate Prompt (`PROMPT_GENERATE`)

```
You are an intelligent dialog bot. You will be shown relevant
information from past conversations. Please read, memorize, and
understand the given information, then generate one concise, coherent
and helpful response for the Question.
{retrieved_texts}
Question Date: {question_date}
Question: {question}
```

Pool context format. The `{pool_context}` field is assembled by iterating over the depth-sorted pool units of the top-k retrieved sessions, deduplicating source texts, and prepending a depth tag:

```
- [depth=2.0] Caroline mentioned a two-week trip to Brazil.
```

Units with depth 0 (never merged) carry no depth tag. Only consolidated units ($d_e \geq 1$) whose source sessions intersect the top-k retrieved sessions are used (at most five per session, the deepest first); their source texts are de-duplicated and the first $U_{\max} = 60$ texts are kept to bound context length. When no unit qualifies (frequent on LongMemEval, whose single-topic assistant exchanges rarely merge), the block reads `(none available)` and the prompt otherwise equals the baseline prompt. Section C.3 reports the sensitivity of QA-F1 to this assembly rule.

B.7 Evaluation Protocol

Answer post-processing. All model outputs are post-processed with the `get_answer` function before metric computation. The function performs two passes: (i) strip a fixed list of leading/trailing artefacts (`Dialogs:`, `[bot]:`, `Assistant:`, `Review:`, and trailing newlines), (ii) truncate at the first occurrence of any conversation-continuation marker (`\n[human]:`, `\nQuestion:`, `\nQ:`).

Retrieval metrics. Recall@k (any) is 1 if *any* gold session appears in the top-k; Recall@k (all) requires *all* gold sessions to appear. We report Recall@k (all) as the primary retrieval metric following Xu et al. (2026a). NDCG@k uses binary relevance with the standard DCG formulation (position-discounted cumulative gain, ideal DCG as normaliser). For LoCoMo, gold sessions are extracted from the `evidence` field; for LongMemEval, from the per-turn `has_answer` flag.

Generation metrics. All generation scores are computed by `generation.py` from the same answer strings with the libraries and settings of Table 3; QA-F1 and EM use the SQuAD normalisation, and the same functions are reused by the significance test.

B.8 Official Implementations of LLM-Based Memory Systems

The HippoRAG 2, RAPTOR and A-Mem rows of Table 1 (marked †) are produced by us with the authors' released code rather than taken from Xu et al. (2026a), so that both benchmarks are scored by the same code, tokenizer, top-3 session protocol and filter–generate pipeline as every other row. We keep the MemGAS protocol: the memory unit is a session, Contriever encodes memory text, and `gpt-4o-mini` is the LLM for memory construction. Every question is answered against its own haystack (one index per LoCoMo conversation, one per LongMemEval question), and each method's native retrieval score is mapped to a full session ranking as described below.

Table 3: Generation-metric implementations.

Metric	Library	Configuration
QA-F1	custom	Token-level F1 after `normalize_answer` (lowercase, remove articles/punctuation)
EM	custom	Best-subspan exact match (`normalize_answer2`, SQuAD protocol)
BLEU-4	NLTK	`sentence_bleu`, weights $(0.25, 0.25, 0.25, 0.25)$, smoothing method 1
ROUGE-L	`rouge` (Python pkg)	F1 of ROUGE-L (LCS-based, default settings)
BERTScore	`bert_score`	`lang=en` (RoBERTa-large default), F1; GPU batch 64 with CPU fallback batch 32

HippoRAG 2 (`OSU-NLP-Group/HippoRAG`, v2.0.0a5). Passages are sessions; OpenIE, the recognition-memory fact filter and personalized PageRank use the released prompts and defaults (linking top-k 5, damping 0.5, passage-node weight 0.05). Two adaptations were necessary for long dialogue sessions: the NER and triple output limits were raised from 512/2,048 to 2,048/4,096 tokens, and because roughly a quarter of the LongMemEval sessions (ShareGPT/UltraChat conversations) make the model answer the dialogue instead of extracting entities, which aborts the released indexer, a chunk whose first response is not the expected JSON is retried once with the passage quoted and an explicit instruction not to follow it; the few chunks that still fail are indexed without entities. OpenIE results are cached by session content, so a session shared by several LongMemEval haystacks is extracted once (51,694 unique sessions, about 6k prompt tokens each). Sessions are ranked by the PageRank passage scores.

RAPTOR (`parthsarthi03/raptor`). Leaves are 100-token chunks of each session (the released default), clustered and summarized with the released `ClusterTreeBuilder` and summarization prompt (100-token summaries), embeddings from Contriever. Retrieval uses the released collapsed-tree scoring over all leaves and summaries; a session's score is the best score of any node covering it (a leaf covers its session, a summary covers the sessions of its descendants). A LoCoMo conversation yields 140–270 leaves and 36–70 summaries; a single LongMemEval haystack yields 12,204 leaves, 1,063 summaries, 1.2M summarization tokens and 17 minutes of clustering, so the 500-question benchmark would require about 5×10^5 LLM calls and we report RAPTOR on LoCoMo only, as Xu et al. (2026a) did.

A-Mem (`WujiangXu/AgenticMemory`, the authors' LoCoMo reproduction). One note per dialogue turn, inserted in order through `add_note` (LLM keyword/context/tag analysis followed by LLM-driven evolution of the note and its five nearest neighbours), with the released keyword expansion of the question as the query and Contriever as the note encoder. The released `memory_layer.py` never imports `re`, so every note analysis silently fell back to empty metadata; we restore the intended path. A session's score is the best cosine score of its notes. Insertion is inherently sequential (two LLM calls per note, about 10 s per turn); a LongMemEval haystack has about 2,600 turns, i.e. roughly 9 hours per question, so A-Mem is likewise reported on LoCoMo only.

B.9 Compute Budget

Table 4: Approximate wall-clock times and API costs per dataset. All GPU times measured on a single NVIDIA A100-80GB. LLM costs are estimates at mid-2025 API pricing.

Dataset	Unit decomp.	TPT build	Retrieval	Generation	API cost
LoCoMo	$<$1 min	$\sim$3 min	$<$1 min	$\sim$10 min	$\sim$\$0.5
LongMemEval	$\sim$2 min	$\sim$35 min	$\sim$3 min	$\sim$60 min	$\sim$\$8

Note

The retrieval side of IMPRINT is **fully deterministic**: it uses no random seeds, no stochastic sampling, and no learned parameters, so re-running `retrieval.py` on the same inputs reproduces every reported retrieval number bit-for-bit. The only non-determinism in the end-to-end pipeline lies in the LLM generation calls (GPT-4o-mini, temperature $= 0$, but no API-side seeding). This determinism enables the per-question paired bootstrap protocol described in §C.4.

B.10 Software and Reproducibility

Key library versions.

Package	Version	Package	Version
Python	3.9	`transformers`	$\geq$4.38
PyTorch	2.3.1	`rouge`	latest
CUDA	12.1	`bert_score`	latest
`sentence-transformers`	$\geq$2.2	`openai`	$\geq$1.30

Directory layout. The complete pipeline executes from the repository root in five steps:

```
# 1.  Install dependencies
conda create -n imprint python=3.9 && conda activate imprint
pip install torch==2.3.1 ...  -r requirements.txt

# 2.  Prepare datasets (see data/README.md for download URLs)
cd data/ && python dataprocess.py

# 3.  Turn-level memory units
cd src/retrieval/ && python ablation_raw_session.py -dataset locomo10

# 4.  Build the Token Profile Table + pool, retrieve
python retrieval.py -dataset locomo10 -alpha 0.3

# 4b.  Baselines (same output format)
python baselines.py -dataset locomo10 -method splade_turn

# 5.  Filter-generate + evaluate (same script for every method)
cd ../generation/ && python generation.py -dataset locomo10 \
       -method imprint -model gpt-4o-mini -topk 3
```

Running steps 3–5 on both datasets on a single A100 GPU requires approximately 2 GPU-hours of TPT construction and encoder inference (reused across all ablations via the on-disk cache) plus API call time. The complete TPT cache is portable: once built, all ablation experiments read from it without rerunning the encoder.

C Additional Results

This appendix collects the result tables and analyses that supplement §6; implementation choices are documented in App. B.

C.1 Sensitivity to the Two Continuous Knobs

The only continuous-valued hyperparameters in IMPRINT that one might reasonably tune are the affine floor in the weight formula and the sparse-dense mixing coefficient α. We document their sensitivity here so that reviewers can verify that no validation-set tuning was performed.

Affine floor. We sweep the floor in $\{0.25, 0.5, 0.75\}$ on LoCoMo with all other settings fixed (run `src/retrieval/floor_sensitivity.py` to reproduce; the sweep now runs through `retrieval.py -floor`). Recall@1 varies by 0.35 points across the range (56.75, 57.05, 56.70). The choice 0.5 is the natural symmetric midpoint that gives IDF and session specificity equal maximum-to-minimum dynamic range, and we use it without further tuning.

Mixing coefficient α. We sweep $\alpha \in \{0.0, 0.1, 0.2, 0.3, 0.5, 0.7\}$ on LoCoMo. Recall@1 varies by less than 0.8 points across the interior range $\alpha \in [0.1, 0.5]$, with $\alpha = 0.3$ marginally the best. We did not retune α on LongMemEval; the same value of 0.3 is used everywhere.

Affine floor (full sweep). Table 5 reports the sweep over the affine floor of $\rho(t)$ on LoCoMo (final configuration, $\alpha = 0.3$ with the turn-level dense channel; the retrieval log of every row is produced by `retrieval.py -floor`, so the 0.5 row is exactly Table 1). Recall@1 varies by at most 1.8 points between the pure gate (0) and the default, and by 0.35 points inside $[0.25, 0.75]$.

Table 5: Floor sensitivity on LoCoMo ($\alpha = 0.3$). Floor $= 0.5$ is the default; 0 turns session specificity into a hard gate and 1 removes it (IDF only).

Floor	R@1	N@1	R@3	N@3	R@10	N@10
0.0	55.29	63.14	72.31	74.40	87.21	80.41
0.25	56.75	64.85	73.67	76.08	88.02	81.79
0.5	57.05	65.36	74.42	76.97	87.76	82.22
0.75	56.70	64.90	74.57	77.08	87.92	82.30
1.0	56.09	64.30	74.47	76.81	88.02	82.09

C.2 Robustness to Lexical Mismatch

Imprint-following relies on exact subword overlap, so we measure its degradation when the query no longer repeats the memory's surface form. We perturb every query with GPT-4o-mini in three controlled ways (entity aliases and abbreviations; paraphrases that replace distinctive terms by descriptions; a deterministic one-character typo on a content word) and evaluate IMPRINT ($\alpha = 0.3$), turn-level Contriever, and turn-level BM25 on *identical* perturbed queries (Table 6). IMPRINT degrades less than BM25 under aliases and paraphrases and equally under typos, because the dense complement and shared subwords partially recover the match, but more than pure dense retrieval under aliases and paraphrases; it nonetheless remains the best method under every condition.

Table 6: Recall@3 on LoCoMo under query perturbations (change from the original query in parentheses). All three columns are scored by `lexical_mismatch.py` with the final configuration (turn-level dense channel); its original-query row reproduces Table 1 and Table 11 within 0.3 points.

Perturbation	IMPRINT	Contriever (turn)	BM25 (turn)
Original	74.47	52.77	72.21
Alias / abbreviation	52.11 (-22.36)	43.86 (-8.91)	47.58 (-24.62)
Paraphrase	58.61 (-15.86)	46.63 (-6.14)	52.32 (-19.89)
Typo (1-char edit)	66.52 (-7.96)	43.40 (-9.37)	64.25 (-7.96)

Merge threshold τ and depth increment δ. Retrieval scoring is invariant to τ and δ by construction (registration in $\mathcal{P}$ precedes merging), and we verified that every (τ, δ) combination reproduces the retrieval numbers of Table 1 exactly. The two knobs act on generation through the pool. Table 7 re-runs the full pipeline on LoCoMo for six combinations around the default $(0.75, 0.5)$, which was fixed *a priori* ($\tau = 0.75$ is a common clustering threshold, $\delta = 0.5$ requires two merges before an entry is surfaced). QA-F1 varies within 1.0 point, and stricter merging ($\tau = 0.85$) would give $+0.7$ F1 on LoCoMo; we nonetheless keep the a-priori default on all datasets rather than tune on a test set.

Table 7: Sensitivity of LoCoMo QA-F1 to the merge threshold τ and depth increment δ (consolidated-only Memory Units; development ranking with the session-level dense channel, whose default gives 21.57 against 21.60 for the final ranking of Table 1). Default in **bold**.

$\tau \setminus \delta$	0.25	0.5	1.0
0.65	21.44	–	21.48
0.75	21.90	**21.57**	21.29
0.85	22.33	–	22.28

No validation-set tuning. None of the values in Table 2 were tuned against the LoCoMo or LongMemEval test splits. The IDF form, the sparse-score TF saturation, and the entropy-based session specificity are all parameter-free statistics. The remaining quantities were either fixed at standard values from the literature ($\tau = 0.75$ is a widely used clustering threshold; $K_{\max} = 120$ comes from the typical encoder context budget for a single dialogue turn) or inherited unchanged from the MemGAS evaluation harness ($\alpha = 0.3$).

C.3 Context Assembly Sensitivity

The generator sees the consolidated pool through the *Memory Units* block. Table 8 varies how that block is assembled, with the retrieval ranking, the raw session context, the prompts, and the generator held fixed. Surfacing every logged unit helps on LoCoMo, whose user–user turns merge often (11.4% of turns end up inside consolidated units of depth $d_e \geq 1$), but hurts on LongMemEval, where merging is rarer (5.5% of turns) and most units merely duplicate the session text. Restricting the block to consolidated entries ($d_e \geq 1$) recovers most of the LoCoMo gain while staying within 0.2 F1 of the session-only baseline on LongMemEval (where consolidation is essentially inactive: fewer than one in five queries has any consolidated unit), and it is the rule used in every main result; with the final retrieval ranking it gives $+1.0$ F1 on LoCoMo and $+0.2$ on LongMemEval over raw sessions (Table 9). Variants that were only run on the LoCoMo development of the rule are marked "–". Table 9 isolates the resulting effect of consolidation on generation: identical retrieval, identical session text, Memory Units present or absent.

Table 8: Context-assembly variants for the Memory Units block (QA-F1). All rows use the same retrieval ranking (the development ranking with a session-level dense channel; the final turn-level channel gives 21.60 / 18.68 for the chosen rule and 20.61 / 18.49 without units). "Top-k" restricts units to the three retrieved sessions; "logged" uses the ten sessions stored in the retrieval log; "cap" is $U_{\max}$.

Memory Units block	LoCoMo	LongMemEval
None (sessions only)	20.77	**18.63**
Logged sessions, rank order, cap 30	21.14	–
Top-k sessions, rank order, cap 30	20.96	–
Top-k sessions, depth order, cap 30	21.04	–
Top-k sessions, depth order, cap 60	21.38	–
Top-k sessions, depth order, no cap	20.92	–
Logged sessions, depth order, cap 60	**21.64**	–
Consolidated only ($d_e \geq 0.5$), depth order, cap 60	21.57	–
Consolidated only ($d_e \geq 1$), depth order, cap 60 (ours)	21.57	18.41

C.4 Statistical Significance

The IMPRINT retrieval pipeline is deterministic (no random seeds, no stochastic sampling, no learned parameters), so per-question scores are exact and cross-run variance is zero. To test significance against the strongest published memory system, we re-run MemGAS (Xu et al., 2026a) end-to-end under the same deterministic harness so that per-question scores are aligned across both systems on

Table 9: Consolidation ablation: identical IMPRINT retrieval and session context; the generator either receives the consolidated Memory Units or not.

	LoCoMo			LongMemEval		
Generator context	F1	EM	R-L	F1	EM	R-L
Raw sessions only	20.61	31.07	19.92	18.49	36.80	17.70
+ consolidated Memory Units (ours)	**21.60**	**32.43**	**20.53**	**18.68**	**37.60**	**17.92**

Table 10: Statistical significance on LoCoMo (IMPRINT vs. MemGAS re-run under our pipeline). Paired bootstrap ($B = 10{,}000$) over per-question scores. Δ = IMPRINT $-$ MemGAS; 95% CIs are percentile bootstrap.

Metric	IMPRINT	MemGAS	Δ	95% CI	p
Retrieval ($n = 1974$)					
R@1	56.94	35.31	+21.63	[+19.25, +23.91]	$< 10^{-4}$
R@3	74.27	57.65	+16.62	[+14.39, +18.79]	$< 10^{-4}$
R@5	80.65	68.24	+12.41	[+10.44, +14.39]	$< 10^{-4}$
R@10	87.69	81.51	+6.18	[+4.46, +7.95]	$< 10^{-4}$
N@1	65.30	42.05	+23.25	[+20.62, +25.79]	$< 10^{-4}$
N@3	76.85	59.41	+17.45	[+15.33, +19.56]	$< 10^{-4}$
N@5	79.87	64.30	+15.57	[+13.75, +17.43]	$< 10^{-4}$
N@10	82.13	68.73	+13.40	[+11.81, +15.00]	$< 10^{-4}$
Generation ($n = 1974$)					
QA-F1	21.55	17.01	+4.54	[+3.78, +5.33]	$< 10^{-4}$
EM	32.47	24.72	+7.75	[+5.98, +9.52]	$< 10^{-4}$

the same question set. We then conduct a paired bootstrap test ($B = 10{,}000$ resamples; percentile-method 95% CI; $H_0 \colon \mathbb{E}[\Delta] = 0$) on per-question score differences. Generation outputs are produced by GPT-4o-mini at temperature 0. We report this analysis on LoCoMo ($n = 1{,}974$ distinct questions; duplicated questions are collapsed, which is why the means differ slightly from Table 1). The MemGAS column is our re-run of its released code under our deterministic harness and generation pipeline, so its values differ from the point estimates reported by Xu et al. (2026a) that Table 1 lists. Table 10 reports the per-metric bootstrap results.

C.5 Granularity–Scoring Interaction: Full Analysis

This section gives the complete discussion summarised in §6.2; the numbers refer to Figure 3 and Table 11.

Table 11: **Turn-level dense vs. IMPRINT** (identical pipeline). Δ: gain vs. Contriever (turn); the session-level Contriever row is the number reported by Xu et al. (2026a).

	LoCoMo		LongMemEval	
Method	R@3	F1	R@3	F1
Contriever (session)	49.90	15.66	44.26	11.88
Contriever (turn)	52.48	16.53	64.89	18.53
IMPRINT	**74.42**	**21.60**	**66.81**	**18.68**
Δ	+21.94	+5.07	+1.92	+0.15

This section explains a historical negative result rather than the source of IMPRINT's gains: prior work tested sparse retrieval on whole sessions, found it no better than dense retrieval, and moved on. We contrast two factors, segmentation granularity (session vs. turn, with turn scores max-pooled to their session) and scoring rule (dense, BM25, uniform, IMPRINT), using the same tokenizer, no dense complement and no truncation for all sparse rules. Figure 3(a,b) presents the interaction effect.

Weighting dominates granularity: uniform scoring stays far below every rarity-weighted rule at either granularity, while the IMPRINT weight lifts the same session-level index to 72.4 R@3 on LoCoMo and 60.2 on LongMemEval before any dense complement. Granularity matters exactly for the length-sensitive scorers: dense cosine, whose margin shrinks as $L^{-1/2}$ (Theorem 5.1), gains $+4.0$ R@3 on LoCoMo and $+29.6$ on LongMemEval from turn-level units, and BM25 gains $+13.1$ and $+15.1$ because of its length normalization and saturated term frequencies (App. A.7); imprint scoring moves by only $+1.1$ on LoCoMo and -3.2 on LongMemEval, and the full system with its dense channel at session vs. turn granularity moves by $+0.1$ and $+1.7$. Theorem 5.1 thus offers two routes to a length-robust margin, shrinking L or weighting by rarity, and the two benchmarks show when each suffices (Table 11). On LongMemEval, whose user–assistant sessions are long and single-topic, shrinking L is almost enough: turn-level Contriever alone (64.9) already beats every memory system in Table 1, including MemGAS, which indexes turns among its granularities yet recovers only 51.1 through its routing; imprint scoring adds $+1.9$ on top of it. On LoCoMo, whose turns are already short (73 tokens on average), shrinking L gains dense scoring only $+4.0$ and leaves it 21.9 points behind IMPRINT: the queries hinge on rare identifiers that a short dense vector still places next to topically similar distractors, so exact matching, not segmentation, is what closes the gap. Two further caveats: pure imprint scoring at session level (60.2) trails turn-level dense on LongMemEval, which is why IMPRINT keeps a light dense channel and reaches 66.8 with it; and the uniform rule collapses under turn-level max-pooling on LongMemEval ($31.3 \rightarrow 10.6$) because, with all weights equal, the longest turn of a session wins regardless of the query. The conclusion stands: sparse scoring does not need short documents once rare tokens are weighted correctly, which is why session-level BM25, the configuration tested by prior work, was dismissed while the same tokens with the imprint weight succeed.

Turn-level retrievers under the identical pipeline. For completeness we list the training-free turn-level retrievers that share IMPRINT's tokenizer, units, top-3 sessions and generation pipeline (Recall@3 / QA-F1 on LoCoMo and on LongMemEval): turn-level Contriever 52.48 / 16.53 and 64.89 / 18.53; turn-level BM25 72.21 / 20.00 and 58.51 / 18.00; the BM25 + Contriever hybrid with IMPRINT's $\alpha = 0.3$ blend 73.82 / 20.38 and 61.28 / 18.50; IMPRINT 74.42 / 21.60 and 66.81 / 18.68. IMPRINT therefore exceeds the hybrid by $+0.6$ and $+5.5$ Recall@3 and by $+1.2$ and $+0.2$ QA-F1 with the same units and blend, so the rarity weight rather than sparsity or granularity per se drives the gains of Table 1. As a reference point only, retrievers trained on MS MARCO and run at the same turn granularity with the same pipeline reach 78.65 / 20.91 and 68.30 / 18.94 (SPLADE v2 (Formal et al., 2021a)) and 77.24 / 20.97 and 67.66 / 19.31 (ColBERTv2 (Santhanam et al., 2022)); they retain a 1.5–4 point Recall@3 edge that IMPRINT approaches with no training data and no GPU at retrieval time, while IMPRINT gives the best LoCoMo answers. We do not treat them as baselines, since they are trained specifically for retrieval and the memory systems of Table 1 are not.

D Dataset Details

IMPRINT is evaluated on two publicly available long-term conversational benchmarks spanning an $18\times$ range in conversation length and two distinct interaction paradigms (user–user and user–AI), plus nested subsets of LongMemEval for the scaling analysis. This section provides complete per-dataset statistics (§D.1), a detailed description of the question-type taxonomy (§D.2), documentation of the data processing pipeline (§D.3), and unit-decomposition statistics (§D.4).

D.1 Per-Dataset Statistics

Table 12 summarises aggregate corpus statistics. Three structural differences drive the experimental design.

1. **Conversation length.** LoCoMo averages $\sim$27 sessions per conversation; LongMemEval uses about 500 per question (502 on average), an $18\times$ increase. For the scaling analysis we draw nested subsets of 50, 75, 100, 150, 250, and 350 sessions from each LongMemEval haystack (evidence sessions always kept, distractors sampled with a fixed seed, $s_{50} \subset s_{75} \subset \cdots \subset s_{350} \subset$ full), so that memory size varies without changing any other experimental variable.
2. **Interaction paradigm.** LoCoMo captures two users reminiscing (*user–user*), yielding longer, narrative-rich turns. LongMemEval captures a single user talking with an AI assistant (*user–AI*):

short user turns followed by long assistant turns, so the merged exchange is longer (424 vs. 73 tokens) but single-topic and rarely revisits earlier sessions.

3. **Answer evidence.** LoCoMo questions require evidence from a median of 1 session (adversarial category up to 3); LongMemEval explicitly annotates `has_answer` at turn level, enabling precise session-level recall evaluation.

Table 12: **Corpus statistics.** Token counts use the Contriever tokenizer; a turn is one merged user/assistant exchange as produced by the processing pipeline (App. D.3); "Sessions/conv", "Turns/session" and "Tokens/session" are means. LongMemEval numbers are per question (haystack); the corpus contains 51,694 distinct sessions shared across the 500 haystacks.

Dataset	Convs	Sessions/conv	Turns/session	Tokens/session	Total tokens	QA pairs	Paradigm	Labels
LoCoMo	10	27.2	11.1	~806	0.22M	1,986	user–user	session
LongMemEval	500	502	5.2	~2,214	114M (1.1M per question)	500	user–AI	session

D.2 Question-Type Taxonomy

Each dataset uses a different taxonomy of question types, reflecting the distinct evaluation focuses of the benchmark designers. We document them here for completeness; the per-type generation results referenced in the main paper (Figure 4c) map onto these categories directly.

LoCoMo — 5 question types (integer codes 1–5)

Code	Type	Description	Freq.
1	Multi-hop retrieval	Requires synthesising evidence from ≥2 distinct sessions.	~18%
2	Temporal reasoning	Questions involving time ("when", "before/after", "how long").	~22%
3	Open-domain knowledge	Mixes retrieved dialogue facts with world knowledge.	~20%
4	Single-hop retrieval	Answer is contained in exactly one session; entity-grounded.	~28%
5	Adversarial	Gold answer is an `adversarial_answer` that contradicts a plausible surface-level guess; tests robustness to misleading distractors.	~12%

Note: adversarial answers are extracted via the `adversarial_answer` field in `dataprocess.py:process_locomo10()` and fall back to `answer` when absent.

LongMemEval — 6 question types

Type	Description	Freq.
`single-session-user`	The answer is grounded in a single user turn.	~18%
`single-session-assistant`	The answer is grounded in a single AI turn.	~17%
`knowledge-update`	User changes a stated fact (e.g. job, address) in a later session; the answer should reflect the most recent value.	~16%
`temporal-reasoning`	Requires identifying time references or ordering events across sessions.	~18%
`multi-session`	Evidence is distributed across ≥2 sessions.	~18%
`single-session-preference`	Answer is a preference or opinion stated once by the user.	~13%

Note: answer session IDs are extracted from `has_answer=True` turns at the turn level and then collapsed to session level in `dataprocess.py:process_longmemeval()`.

D.3 Data Processing Pipeline

Both datasets are normalised to a common JSON schema by `data/dataprocess.py` before any further processing. The schema for each entry is:

```
{
"conversation_id": str,
"sessions": List[List[str]], # outer = sessions; inner = merged
turns
"sessions_ids": List[str],
"sessions_dates": List[str | None],
"qa": [{
"question": str,
"question_type": str | int | None,
"question_date": str | None,
"answer": str,
"answer_session_ids": List[str] | None
}]
}
```

The **turn-merging** step is common to both datasets: consecutive user and AI turns are joined into a single string with a newline separator—`"[User]: ...\n[AI]: ..."`—forming the atomic session entry. This mirrors the input format expected by `retrieve.py`. Dataset-specific processing notes follow.

LoCoMo. Sessions are stored as nested dicts keyed `session_1`, `session_2`, ... alongside `session_{i}_date_time`. Image captions (field `blip_caption`) are appended inline: `"The image Caption: {caption}"`. Evidence turn IDs are converted to session IDs by stripping the colon-delimited turn index and normalising `D{k}` → `session_{k}`.

LongMemEval. Answer session IDs are identified via `has_answer=True` at the turn level, then deduplicated to session granularity. The `answer_` prefix is stripped from `haystack_session_ids` to align keys with `sessions_ids`. Each question forms its own top-level conversation entry.

Session key format. The unit JSONL files consumed by `retrieval.py` use *different* key formats for the two datasets:

- **LongMemEval:** key = `{session_id}` (e.g. `session_42`)
- **LoCoMo:** key = `convid-{conv_id}-sessid-{session_id}`

The branching condition `"longmemeval" in dataset` in `retrieval.py` and `ablation_raw_session.py` implements this distinction.

D.4 Unit Decomposition Statistics

IMPRINT consumes one of two unit granularities produced by `ablation_raw_session.py`: **`raw_turn`** (one memory unit per merged dialogue turn, the default for all main results) and **`raw_session`** (one unit per full session, used only for the session-level statistics reported here). Table 13 reports the resulting unit counts and mean token lengths at each granularity.

Table 13: **Unit decomposition statistics** over distinct sessions (LongMemEval haystacks share sessions: 250,948 session slots and 1.30M turn slots across the 500 questions). Token counts use the Contriever tokenizer; a `raw_turn` unit is one merged user/assistant exchange.

	`raw_turn` (turn-level)			`raw_session` (session-level)		
Dataset	**Total units**	**Units/session**	**Tokens/unit**	**Total units**	**Units/session**	**Tokens/unit**
LoCoMo	3,011	11.1	73	272	1	806
LongMemEval	267,206	5.2	424	51,694	1	2,214

Filtering. `ablation_raw_session.py:convert_per_turn()` applies a minimum length filter (`len(turn_text) > 10`) to exclude degenerate turns such as back-channel acknowledgements (*"OK."*, *"Sure."*). No additional preprocessing (lowercasing, punctuation removal, or stopword filtering) is applied; the Contriever tokenizer handles normalisation downstream.

Unit length and Theorem 5.1. The $11\times$ ratio between `raw_session` and `raw_turn` token length in LoCoMo (806 vs. 73 tokens/unit) is the empirical counterpart of the parameter L in Theorem 5.1. It is this gap that makes dense and BM25 scoring depend on turn-level units (§6.2), whereas imprint scoring is insensitive to it

E Broader Impacts

Positive impacts. Long-term conversational memory is a foundational capability for personal assistants, educational tutoring, and accessibility tools that interact with the same user across hundreds of sessions. By making accurate recall both cheaper (CPU-only retrieval, no learned parameters) and more robust across conversation lengths, IMPRINT lowers the barrier to deploying helpful long-term agents in resource-constrained settings such as on-device assistants and educational platforms in low-compute environments.

Privacy considerations. The mechanism that gives IMPRINT its retrieval signal is also the source of its primary risk: rare user-specific identifiers — proper nouns, dates, place names — are preserved *verbatim* in the Token Profile Table by design, rather than being projected into a continuous semantic neighborhood as in dense retrievers. This makes the memory store a sensitive artifact: an adversary who exfiltrates the table can recover identifying details with high fidelity. Deployments should treat the Token Profile Table as personally identifying data, encrypt it at rest, restrict access to the user it belongs to, and support user-initiated deletion that propagates through both the table and the consolidated pool.

Mitigation practices. We recommend three deployment-time practices for systems built on imprint-following memory: (i) *data minimization* — only register turns the user has explicitly authorized for long-term retention; (ii) *user-controlled deletion* — expose a per-conversation `forget` operation that removes affected entries from $\mathcal{P}$ and $\mathcal{C}$ in a single deterministic pass, which the incremental construction in Algorithm 1 already supports in $O(|\mathcal{T}(u)|)$ time per turn; (iii) *audit logging* — record retrieval queries against the table so users can inspect what their agent recalls about them.

Foreseeable misuse. Because IMPRINT is a retrieval mechanism rather than a generation system, it does not directly create new risks of disinformation, deepfakes, or content manipulation. The most plausible misuse is surveillance: an operator with access to a user's conversation logs could use IMPRINT to efficiently search for sensitive personal references across long histories. This is a deployment-policy concern that applies to any long-term memory system; IMPRINT does not introduce capabilities beyond those already available with dense retrieval, but its higher precision on rare identifiers makes such queries more effective. We do not release any user data or pretrained models that would aid such misuse.