

# Age Is Not Evidence of Staleness: A Recency Decay Is a One-Rate Survival Model, the Rate Belongs to the Fact Rather Than the Clock, and the Silent Change Only a Rate Could Catch Is the Case No Agent-Memory Benchmark Scores

Pranay Mahendrakar

SONYTECH · ORCID 0009-0003-7224-029X

pranaymahendrakar@sonytech.in · mahendrakarpranay@gmail.com · +91 6361723454

**Abstract.** Language agents that keep memories across sessions usually discount a record by how long ago it was written or last read. That discount was introduced to decide what an agent attends to, but it is increasingly read as a statement about whether a record is still true. This paper examines that second reading. Taken as a validity claim, an exponential recency term is a survival model with a constant hazard and a single rate shared by every record in the store. The time-sensitive knowledge literature, together with more than two decades of web-crawl measurement, contradicts the single rate: how fast an answer goes out of date differs by orders of magnitude across kinds of fact, and on the benchmark that splits questions by that rate, accuracy falls sharply with it. The paper then complicates the obvious remedy of attaching the rate to the predicate. Two published operationalisations of volatility, one by relation type and one by per-fact edit history, reach opposite conclusions on the same question, and within-class variation is large. It shows by standard derivation that pooling heterogeneous rates makes the pooled hazard fall with age, so a long-unchanged value is evidence of stability. Age since last confirmation and duration held before it therefore point in opposite directions, and a single timestamp cannot separate them. Finally, it observes that the systems and conversational-memory benchmarks located here invalidate a memory only when a later observation contradicts it. The case where only a volatility prior could help, a change the agent never hears about, is scored by no conversational-memory benchmark located here, and the one controlled study that withholds a change and budgets verification does not vary elapsed time or how volatile the fact is. It closes by stating a decision structure as an algorithm and specifying a benchmark that would settle the open parts.

**Keywords:** agent memory; temporal knowledge; staleness; knowledge volatility; survival analysis; recency ranking; web crawling freshness; retrieval-augmented generation

## 1. Introduction

An assistant that remembers its user writes records such as "lives in Lisbon", "was born in Porto", "is training for a marathon in April" and "prefers aisle seats". Months later it retrieves some of them to answer a question. Before it uses one, something has to decide how far to trust it. In most published agent memories that decision is folded into a retrieval score, and the term that carries time is a decay over elapsed hours or days. Park et al. (2023) scored recency as an exponential decay over the sandbox hours since a memory was last retrieved, with a decay factor of 0.995, and added it with equal weight to relevance and importance. Zhong et al. (2023) borrowed the Ebbinghaus curve, retaining a memory with strength  $R = \exp(-t/S)$  and strengthening it each time it is recalled. Wei et al. (2026) generalised the curve to a stretched exponential whose rate depends on an importance score built from relevance, access frequency and recency.

None of these designs claims that an old record is a false one. The first two were built to decide what an agent keeps in view or keeps at all, and the third says it is aimed at storage economy and retrieval precision. But the same number is now routinely asked to do a third job. When a system surfaces the newer of two records, or

down-weights a memory "because it is old", it is using age as evidence that the content may no longer hold. The author's earlier paper on the recency rule (Mahendrakar, 2026a) separated these three jobs - retention, salience and truth - and examined the case where two records disagree. This paper takes the case that one left open: a single record, uncontradicted, whose age is the only temporal fact the store holds. The question is what that age is evidence of.

The answer argued here has three parts.

First, read as a validity claim, an exponential recency term is a survival function with a constant hazard, and because one decay constant is applied to every record, it asserts that every kind of fact goes stale at the same rate. The literature on time-sensitive knowledge in language models says otherwise and says so with numbers. For each of the five models plotted in Figure 1a from those evaluated on FreshQA, accuracy falls by a factor of four or more between questions whose answers almost never change and questions whose answers change within a year (Vu et al., 2023). A web cache fitted to observed page changes needed half-lives from three hours to twenty-two days across four fitted freshness classes, with a fifth given a 30-second prior (Mansoor et al., 2026). A temporal knowledge-graph memory that learns one

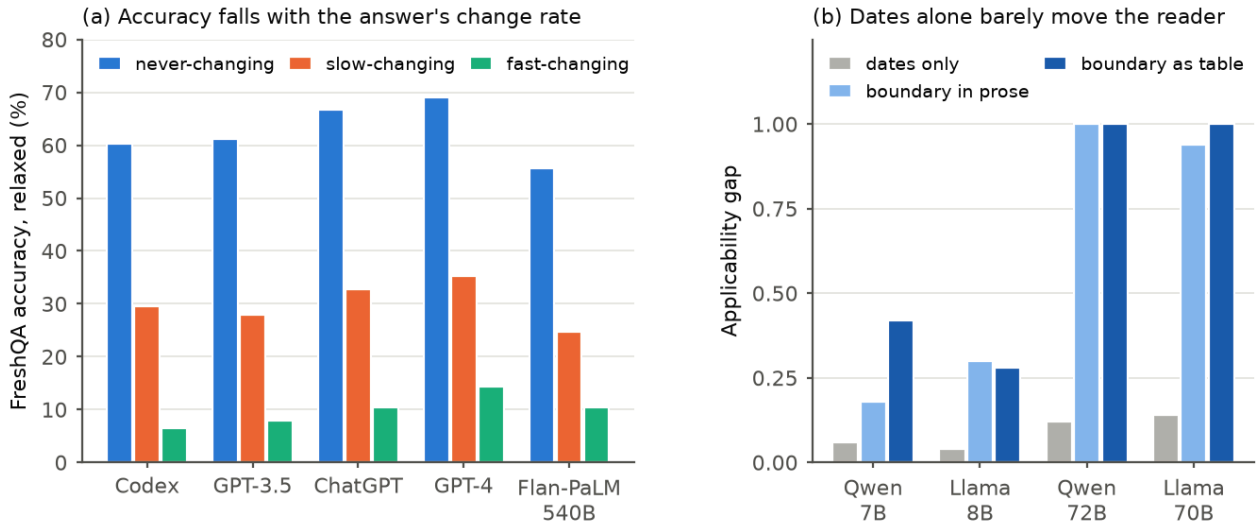


**Figure 1.** Two published results on temporal information. (a) FreshQA accuracy under relaxed evaluation, by how often the answer changes, for five models benchmarked on 26 April 2023 without search (Vu et al., 2023, Table 4). (b) Applicability gap - the rate of giving the old answer while the evidence is valid minus the rate after it has been superseded - on 50 verified reversals, when only the evaluation date differs and when the validity boundary is stated in prose or as a table (Ahmed et al., 2026, Table 7). Each panel re-plots the values it cited source reports.

rotation speed per relation lost two thirds of its retrieval quality when the per-relation speed was replaced by a shared one (Li et al., 2026).

Second, the obvious repair - give each predicate its own rate - is less clean than the premise suggests. Fierro et al. (2024) key mutability on the relation type and find mutable facts easier to update in context; Marjanovic et al. (2024) key it on a fact's own Wikidata edit count and find the opposite; the reproduction that ran each study's evaluation on the other's data (Dey et al., 2026) traces the disagreement partly to that definitional choice. Web measurement has long found large variation within any class one might name (Fetterly et al., 2003; Adar et al., 2009). A predicate rate is a prior, and the evidence that should move it is record-level.

Third, the case in which any rate model earns its keep is the one the field has barely tested. Every agent-memory system examined here that retires records on grounds of validity does so when a newer observation contradicts them (Rasmussen et al., 2025; Chhikara et al., 2025; Zhou et al., 2026; Yadav, 2026). Every conversational-memory benchmark examined here that scores staleness supplies that newer observation, explicitly or implicitly (Wu et al., 2025; Chao et al., 2026). The one located instrument that withholds a change and scores budgeted verification is a controlled study of a six-memory store (Nakayashiki, 2026); it has no conversational history and does not vary elapsed time or how volatile the withheld fact is. The world, however, changes without telling the agent. A user who moves house and never mentions it leaves a record that no contradiction-driven system will ever retire, and that only a prior over how often addresses change - applied between observations - could flag for re-verification. In the language of state

estimation, the published systems implement the correction step and omit the prediction step.

The paper also locates the one published result that says what form a validity estimate must take to matter. When the same outdated document was shown to models with only a changed evaluation date, the models barely moved; when the document's validity boundary was stated outright, the two largest models switched to the current answer in 47 to 50 of 50 cases (Ahmed et al., 2026). A timestamp handed to a reader is close to inert. An explicit horizon is not.

The contribution is analytical. Section 3 states what a recency term says when it is read as a validity claim. Section 4 assembles the evidence that change rates differ by kind of fact, and Section 5 the evidence that "kind of fact" is an unstable unit. Section 6 derives, from standard survival arguments, why age can count for or against a record depending on which clock is read, and why a published duration-based discount is exact only under a condition that is not stated with it. Section 7 identifies the missing prediction step and the silent-change case. Section 8 turns to how readers use temporal information. Section 9 gives the strongest evidence that plain recency works. Section 10 states a decision structure as Algorithm 1. Section 11 specifies the benchmark that would score it, and Sections 12 to 14 give the case against this paper, what is not known, and the conclusion.

## 2. Scope, and what this paper does not claim

The paper is about records in an agent's own memory store, written from interaction, that assert something about a changeable world: a user's circumstances, a

project's state, an external fact the agent was told. It asks what the elapsed time since such a record was written licenses the agent to believe about whether it still holds.

Four neighbouring questions are excluded.

It does not treat conflict resolution between two records. When a store holds "lives in Lisbon" and later "lives in Berlin", deciding which wins is the subject of the author's earlier paper (Mahendrakar, 2026a), which argued that preferring the newer record rests on three unstated presuppositions. That paper's constraint is binding here: nothing below is an argument about which record wins. The present question arises when there is only one record.

It does not treat retention or salience on their own terms. A decay that decides what to evict under a storage budget, or what to show a model first, can be justified by cost or by need regardless of whether older records are truer. Anderson and Schooler (1991) showed that the probability a memory will be needed tracks the frequency, recency and pattern of past exposure in several real environments, and argued that human forgetting is adapted to those statistics. That is a salience argument about need, not a validity argument about truth, and the paper cites it only to mark the boundary.

It does not treat staleness of a model's parametric knowledge as such. The time-sensitive QA literature is used here as evidence about how fast different kinds of fact change and about how models read temporal information. The remedies for parametric staleness - continual pretraining, editing, temporal alignment - are outside scope. A recent survey organises temporal question answering around the interaction of corpus temporality, question temporality and model capability (Piryan et al., 2026); the present question, how a memory store should discount an uncontradicted record of its own, is not the question that literature is organised to answer, and the paper draws on it for evidence rather than position.

It does not propose a system. The decision structure in Section 10 names the quantities a validity-aware read would need and the order in which they enter; it is offered as a specification, to be scored on the benchmark that Section 11 specifies.

The claims come in three evidence classes, kept apart throughout. Published measurements are reported at the value and scope their sources give. Results that follow from probability alone - the hazard of a mixture, the residual life of a fixed term - are stated flatly and derived in the text. Interpretations that connect the two are marked as such.

### 3. What a recency term says when it is read as a validity claim

#### 3.1 Three published forms

Park et al. (2023) define recency as an exponential decay over the number of sandbox game hours since a memory was last retrieved, with a decay factor of 0.995. The three components are min-max normalised and summed with all weights set to one. Two features matter here. The clock is time since last retrieval, not since writing, so reading a record resets it. And the rate is one constant for every record in the stream.

Zhong et al. (2023) model retention as  $R = \exp(-t/S)$ , where  $t$  is time since the item was learned and  $S$  a strength initialised to one at first mention; recalling the item increments  $S$  and resets  $t$  to zero. The authors describe this as "an exploratory and highly simplified memory updating model", and add that "The forgetting curve will look different for different people and different types of information." The second sentence is the premise of the present paper, stated by the designers of one of the systems it examines and then set aside for simplicity.

Wei et al. (2026) replace the single exponential with  $v(t) = v(0) \exp(-\lambda(t - \tau)^\beta)$ , where  $\lambda$  falls with an importance score and  $\beta$  is 0.8 in a long-term layer and 1.2 in a short-term layer. At zero importance and  $\lambda_{\text{base}} = 0.1$ , they report half-lives of about 11.25 days and 5.02 days for the two layers. This is the most differentiated of the three, and the differentiation is worth reading closely: the rate varies with importance, and importance is built from relevance to recent context, access frequency and recency. Nothing in it refers to how often the content of the record changes in the world. A user's current employer, mentioned often and judged important, would be among the slowest to decay.

#### 3.2 The reading this paper examines

Suppose one takes any of these terms at its word as a probability that a record still holds after elapsed time  $t$ . An exponential decay  $\exp(-\lambda t)$  is then the survival function of an exponential lifetime, and its hazard - the instantaneous rate at which still-valid records become invalid - is the constant  $\lambda$ . Three consequences follow without any empirical input.

The model is memoryless: the probability that a record valid at time  $t$  is still valid at  $t + s$  equals the probability that a fresh record survives  $s$ . How long the record has already survived carries no information.

The model is homogeneous: because one  $\lambda$  is applied to every record, the ordering of records by validity is exactly the ordering by age. A record written a year ago about where the user was born ranks below a record written yesterday about which hotel the user is in, whatever the content.

The model's clock is the one the system happens to keep. For Park et al. (2023) it is time since last retrieval, so a record that is read often looks fresh. As the author's earlier paper on write-time importance noted (Mahendrakar, 2026b), that makes recency a read-time quantity; for the validity reading it is worse than that, because being retrieved is not being re-verified. Frequent retrieval of an unverified record raises its recency score without adding any evidence that it is still true.

To make the implied rate concrete: a decay factor of 0.995 per hour corresponds to  $\lambda = -\ln(0.995)$ , about 0.0050 per hour, and a half-life of  $\ln(2)/0.0050$ , about 138 hours or 5.8 days. Park et al. did not intend this as a validity half-life, and this paper does not attribute that intent to them. The arithmetic shows only what the term would assert if a downstream component treated it as one: that any memory, of any kind, has an even chance of having stopped holding six days after it was last used.

### 3.3 Why this reading is not a straw man

The validity reading enters in three ways that do not require any designer to intend it.

It enters through conflict handling. Where two records disagree, systems that prefer the newer one are using write order as a proxy for validity; the earlier paper traced that pattern across seven systems (Mahendrakar, 2026a). Once recency is established as the tie-breaker for truth in the two-record case, the same number is available as a discount in the one-record case.

It enters through composition. When recency is added to relevance and importance with equal weight, as in Park et al. (2023), a stale-but-relevant record and a fresh-but-less-relevant one trade off in a single score. Whatever the designer intended, the score's output is what decides whether the content reaches the model, and a record that no longer holds is as harmful to the answer when it arrives through a salience score as when it arrives through a truth score.

It enters through the literature's own framing of the problem. Recent systems describe their goal as keeping outdated information out of the agent's context: TEPA frames the failure as stale memories that "remain retrievable and pollute the prompt" (Zhou et al., 2026); HiCoMER criticises flat pools ranked "by semantic relevance, importance, or recency without modeling hierarchical structure or evolving validity" (Shi et al., 2026); RoMem names the "static-dynamic dilemma" that timestamp sorting creates (Li et al., 2026). In each, recency is the incumbent that validity-aware designs are measured against.

The rest of the paper asks what a validity-aware discount would have to know that a uniform decay does not.

## 4. Where volatility lives: the evidence that change rates differ by kind of fact

### 4.1 Time-sensitive question answering

The cleanest single demonstration comes from FreshQA (Vu et al., 2023), whose 600 questions are divided by how often the answer changes: never-changing ("the answer almost never changes"), slow-changing (changes over several years), fast-changing (changes within a year or less), plus false-premise questions. The benchmark was evaluated by human raters in more than 50,000 judgements, in a relaxed mode that checks only the main answer and a strict mode that penalises any outdated or hallucinated claim. Under relaxed evaluation on 26 April 2023, GPT-4 answered 69.0 percent of never-changing, 35.2 percent of slow-changing and 14.4 percent of fast-changing questions correctly; Codex scored 60.3, 29.6 and 6.4 on the same three categories (Figure 1a). The authors describe flat scaling on fast-changing questions: larger models did not reliably do better there.

These are accuracies of a model's parametric knowledge, not survival curves of facts, and the ordering is confounded with other properties of the questions. What the result does establish is that "how often does this kind of answer change", assigned per question by the benchmark's designers, separates accuracy by a wide margin for every model plotted. It is not the only variable that does: in the same table, whether the needed knowledge dates from before or after 2022 separates accuracy by a comparable margin (80.9 against 14.9 percent for GPT-4 under relaxed evaluation). That second split is about the model's training cutoff rather than about the fact, and the two are entangled, since fast-changing answers are disproportionately recent ones.

Several other datasets are built on the same distinction. Dhingra et al. (2022) open with the observation that "Many facts come with an expiration date", and probe models on a diagnostic set of facts that change over time. Chen et al. (2021) mine time-evolving facts from Wikidata with their validity periods; the best system they tested reached 46 percent against 87 percent for humans. Park et al. (2025) separate knowledge that evolves (personal histories, amended laws) from knowledge that stays constant (mathematical truths, commonsense facts) and find that models often recall chronological knowledge only in part, often stopping short at the edge of a time span. Jang et al. (2022a) frame continual knowledge learning as three separable targets - retaining time-invariant knowledge, updating outdated knowledge, acquiring new knowledge - which presupposes that the invariant part can be identified. Mousavi et al. (2024) check model outputs against Wikidata's current values and its record of each outdated value's validity interval, and report that two recent chat models gave outdated answers to around 30 percent of 130 time-sensitive questions about heads of state,

athletes' clubs and company officers.

How much of ordinary question traffic is exposed? Zhang and Choi (2021) found that roughly 16.5 percent of NQ-Open questions have answers that depend on temporal or geographic context, and that models trained on older data lost about 15 points of accuracy answering them as of the present even with an updated evidence corpus. Zhang and Choi (2023) report that 48 percent of the SituatedQA questions they used had answers updated within the three-year gap from 2018 to 2021. Those numbers bound nothing about a personal-assistant memory store, but they show that time-dependence is neither rare nor uniform in the one kind of traffic where it has been counted.

#### **4.2 Change-rate measurement outside language models**

The most mature evidence that change rates are item-specific comes from web crawling, where the question "how likely is my copy still current" has been asked, with money attached, for more than twenty-five years.

Brewington and Cybenko (2000) studied the frequency and nature of page modifications and used a metric of up-to-dateness to estimate how often a search engine must re-index. Cho and Garcia-Molina (2000) and Cho and Garcia-Molina (2003b) formalised two freshness metrics for a local copy of an autonomous source, proposed a Poisson process as the change model, and reported that it describes web-page changes well. The same authors (Cho and Garcia-Molina, 2003a) then treated the change frequency itself as the unknown and developed estimators for it, reporting that a crawler could gain 35 percent in freshness by adopting one. Avrachenkov et al. (2020) supply online estimators that need only one bit per visit - whether the page has changed since the last crawl - with convergence guarantees.

The measured rates are wide. Fetterly et al. (2003) summarise Cho and Garcia-Molina's crawl of 720,000 pages over four months as finding that 40 percent of pages changed within a week and 23 percent of .com pages changed daily. Their own weekly crawl of about 150 million pages over eleven weeks found that the degree of change "varies widely across top-level domains" and that larger pages change more often and more severely. Adar et al. (2009), crawling 55,000 pages chosen by how people actually visit them, found change rates higher than in random samples, with a large portion of pages changing more than hourly, and identified stable and dynamic content within the same page. Olston and Pandey (2008) drew the practically decisive distinction: ephemeral content, such as a quote of the day, is not worth fetching because it is gone before it is indexed, while persistent content that survives many page updates is.

Mansoor et al. (2026) bring this tradition directly to retrieval-augmented generation. Their cache estimates,

before reusing a cached result, the probability that it is stale, using a half-life fitted per freshness class by maximum likelihood from re-fetches at one hour, twelve hours, a day and a week. The fitted half-lives were 22 days for timeless queries, 16 days for slow, 15 hours for medium and 3 hours for fast; a real-time class was given a 30-second prior because its pages could not be observed changing at the windows used. That is a spread of more than two orders of magnitude between the timeless and fast classes on a single benchmark. Two further details in the same paper anticipate the next sections: only 34.3 percent of the page changes their hashes detected were judged to affect the cached answer, and the authors note that a single exponential rate cannot jointly fit short-term churn and slower long-term drift.

#### **4.3 Stationarity as a commonsense property**

The temporal-commonsense literature names the same property from the language side. Zhou et al. (2019) define five temporal properties, one of which is stationarity: whether a state holds for a very long time or indefinitely. Zhou et al. (2020) acquire duration and frequency knowledge from explicit and implicit mentions in a large corpus. Almquist and Jatowt (2019) posed the task of predicting validity periods of sentences; Leblay and Chekol (2018) posed the knowledge-graph version, predicting time validity for edges that lack a time annotation; and Wenzel and Jatowt (2024) extended the sentence task to the case in which additional context from the same text stream changes how long a statement should be expected to remain valid.

The closest published precedent for the argument of this paper is Zhang and Choi (2023), who propose fact duration prediction - estimating the length of time for which a stated fact stays true - and use it to discount a QA system's confidence under temporal misalignment. Trained only with distant supervision, their systems predicted the duration of 65 percent of temporally dependent facts from real search queries to within three years, against 11 percent for a baseline that predicts the average duration, and cut expected calibration error by 50 to 60 percent relative to using the QA system's confidence alone. They also report that models trained for temporal commonsense and models trained for fact duration do not transfer well to each other, because commonsense separates orders of magnitude while the durations of real queried facts cluster between one and ten years. Section 6.4 returns to one condition under which their discount is exact.

#### **4.4 A memory system that learned the rate per relation**

Li et al. (2026) built the first agent-memory module located here that learns volatility per relation rather than assuming it. RoMem maps each relation's text embedding through a small network to a scalar between zero and one that sets how fast facts with that relation rotate in a

complex embedding space; the gate is pretrained on transitions mined from a political events graph, where a changed counterpart signals high volatility and an unchanged one low volatility. At query time a fact's phase moves by an amount proportional to that scalar and to the time since the fact's anchor, so a newer fact with a volatile relation outranks an older competitor, while facts with stable relations are never "buried by recency". On a temporal reasoning benchmark, removing the gate - using one shared speed for all relations - lowered mean reciprocal rank from 0.3371 to 0.1044.

That ablation is the strongest single datum located for the premise of this paper: in a system otherwise held fixed, a shared rate lost about two thirds of the retrieval quality that per-relation rates achieved. Two limits of the result matter for what follows, and both are stated by, or visible in, the source. The gate values "follow the expected volatility ordering" but "do not measure calibration over all relations"; the rate is a rotation speed, not a hazard, and no probability of validity is read from it. And the temporal score enters retrieval as a multiplicative amplifier on a semantic score, with strength 0.3, so it reorders candidates that are already plausible; it does not produce a statement that an uncontradicted record should be doubted. RoMem answers "which of these competing facts is current" with per-relation rates. It does not address "is this lone fact still current", which is the question here.

#### **4.5 Retrieval research learned the same lesson about relevance**

Information retrieval met a version of this problem earlier, for relevance rather than validity, and resolved it the same way. Li and Croft (2003) identified a type of query that favours very recent documents and built time-based language models for those queries, rather than applying a recency prior to every query. Dong et al. (2010) detect recency-sensitive queries with a classifier and send only those to a ranker trained to use freshness features. Efron and Golovchinsky (2011) went a step further and estimated the rate parameter of the temporal prior per query, reporting that the query-specific models better captured the temporal variability of relevance associated with some topics. In each case the field moved from one decay for everything to a decay whose existence and rate depend on what is being asked about. The analogue for a memory store is a decay whose rate depends on what the record says.

Table 1 consolidates the measured values that the argument of Sections 4 to 8 rests on.

### **5. Why "kind of fact" is an unstable unit**

Section 4 supports the negative half of the premise: one rate for all records is wrong. The positive half - that volatility is a property of the predicate and can be looked up by predicate - does not survive the same literature intact. Four findings complicate it.

#### **5.1 Two definitions of volatility, two opposite conclusions**

Fierro et al. (2024) built MuLan from 35 Wikidata relations sorted by mutability at the level of the relation: 12 immutable one-to-one relations, 10 immutable one-to-many, and 13 mutable, with about 47,000 queries. They report that models are less confident on mutable facts, that a linear probe on the final-layer representation separates mutable from immutable one-to-one facts with 0.85 to 0.89 accuracy across six models (against 0.64 for a bag-of-words classifier on the templates alone), and that mutable facts are updated more consistently by in-context evidence.

Marjanovic et al. (2024) built DynamicQA with the opposite unit. A fact is labelled temporal if its Wikidata record has been edited more than once and static if it has never been edited; disputable facts are mined from reverted edits on controversial Wikipedia articles. The resulting 11,378 questions include 2,495 temporal and a sample of 2,500 static. They find that models show more intra-memory conflict on dynamic facts, that static facts are the most easily persuaded by counter-context, and that the number of edits is the strongest single predictor of persuasion, with more edits meaning less persuasion. They note the conflict with Fierro et al. directly and attribute it to the fact that one study "determine[s] mutability via relationship type" while the other verifies it with per-fact proxies.

Dey et al. (2026) ran each study's evaluation on the other's dataset and extended both to three sizes of one model family. MuLan's findings held under both evaluation frameworks; applying MuLan's evaluation to DynamicQA gave mixed results; and within the Gemma 3 family the MuLan pattern was non-monotonic in size, with mutable facts updated less successfully than immutable one-to-one facts, by small margins, at 1B and 12B and more successfully at 4B. They conclude that whether external context reliably resolves temporal conflicts "remains an open question". Two of their audit findings bear directly on this paper. DynamicQA used the same relations for static and temporal facts and relied on edit counts alone, which they say misclassifies inherently stable relations whose apparent variation reflects disambiguation rather than change. And 1,369 of the 2,495 temporal facts have an edit count of exactly two - one more than the threshold - so that the difference "may instead reflect corrections or synonyms".

The lesson is not that one definition is right. It is that the two obvious units for volatility - the predicate, and the record's own history - measure different things, fail in different ways, and have already produced opposite published conclusions on a question as basic as whether volatile facts are easier to update.

**Table 1.** Published measurements bearing on whether elapsed time predicts staleness, and on what agents and readers do with temporal information. All values are as reported by the named source.

| Source                                               | Setting                                      | Reported value                                                              | Bears on                             |
|------------------------------------------------------|----------------------------------------------|-----------------------------------------------------------------------------|--------------------------------------|
| Vu et al. (2023), Table 4                            | FreshQA, relaxed, GPT-4, no search           | 69.0 / 35.2 / 14.4 percent on never / slow / fast-changing                  | Error scales with change rate        |
| Mansoor et al. (2026)                                | Web cache, 5 freshness classes               | Fitted half-lives 22 d, 16 d, 15 h, 3 h; 30 s prior                         | Rates span orders of magnitude       |
| Fetterly et al. (2003), citing Cho and Garcia-Molina | 720,000 pages, daily, 4 months               | 40 percent changed within a week; 23 percent of .com daily                  | Rates differ by item                 |
| Zhang and Choi (2023)                                | SituatedQA, 2018 to 2021                     | 48 percent of answers updated; duration within 3 years for 65 vs 11 percent | Per-fact duration is predictable     |
| Li et al. (2026)                                     | RoMem, MultiTQ ablation                      | MRR 0.3371 with per-relation gate, 0.1044 with one speed                    | Shared rate is costly                |
| Fierro et al. (2024), Table 2                        | Probe, mutable vs immutable-1                | Accuracy 0.85 to 0.89; template-only baseline 0.64                          | Models encode mutability             |
| Dey et al. (2026)                                    | DynamicQA re-audit                           | 1,369 of 2,495 "temporal" facts have edit count 2                           | Per-fact proxies are noisy           |
| Mansoor et al. (2026)                                | 396 hash-detected changes                    | 34.3 percent affected the answer                                            | Change is not invalidation           |
| Zhou et al. (2026)                                   | TEPA, full reversal, 50 seeds                | Append-only and last-write-wins 0.210; no memory 0.309; TEPA 0.950          | Stale memory worse than none         |
| Yadav (2026)                                         | RAG over evolving facts                      | Serves superseded values 15 to 40 percent of the time                       | Contradiction handling matters       |
| Chao et al. (2026)                                   | STALE, 400 scenarios                         | Best model 55.2 percent overall                                             | Acting on updates is hard            |
| Nakayashiki (2026)                                   | Inherited constraint, budget 2 of 6          | Stale-consistent decisions 77.3 percent; provenance read about 1 in 5       | Settled-looking records go unchecked |
| Ahmed et al. (2026), Table 7                         | 50 reversals, date only vs explicit boundary | Gap 0.12 vs 1.00 (Qwen-72B); 0.14 vs 0.94 to 1.00 (Llama-70B)               | Readers ignore dates, use horizons   |
| Wu et al. (2025)                                     | LongMemEval, time-aware expansion            | Temporal-reasoning recall up by an average of 6.8 to 11.3 percent           | Where plain time signals help        |

## 5.2 A record's history mixes change with correction

The edit-count problem is not specific to Wikidata. Any history of a stored value records every time the value was rewritten, and a rewrite can mean that the world changed or that the earlier value was wrong. The author's earlier paper argued that these two cases, which belief revision separates, demand opposite resolution rules (Mahendrakar, 2026a). For volatility estimation the consequence is narrower but just as direct: a per-record change count, the natural input to any rate estimator of the Cho and Garcia-Molina (2003a) or Avrachenkov et al. (2020) type, inflates the apparent rate of exactly those records that were written carelessly. Mansoor et al. (2026) show the web version of the same confound from the other side: of the page changes their hashes detected, only 34.3 percent changed the answer, so a rate fitted to detected changes overstates the rate of answer-relevant change by about a factor of three for that benchmark. Dong et al. (2009) had to model the same distinction in data integration, characterising each source over time by coverage, exactness and freshness rather than by any single error rate.

## 5.3 Within-class variation is large and does not transfer

Web measurement found early that variation inside any natural class is substantial: degree of change varied widely across top-level domains and with page size (Fetterly et al., 2003), and stable and dynamic content coexisted within single pages (Adar et al., 2009). Mansoor et al. (2026) report the modern version. Their

learned model beat the per-class rule chiefly by finding stable queries inside volatile classes, using features such as answer type, per-domain empirical volatility and entity volatility; but they also report that this within-class discrimination did not transfer to out-of-distribution query sets. There the class-level rule degraded more gracefully (16.3 percent cache-induced stale error at a 25.1 percent hit rate on DailyQA), while the within-class refinement's hit rate dropped sharply.

In question answering the same pattern appears in the structure of the data. DyKnow's time-sensitive facts share a handful of predicates - head of state, head of government, sports team, directorial role - across dozens of subjects (Mousavi et al., 2024), and the rate at which a country changes its head of state is not one number. Zhang and Choi (2023) make the point about knowledge: the duration of "which team does this athlete play for" requires knowing the entity, not only the relation, and they report that commonsense knowledge alone is insufficient to separate one-year from two-year durations.

## 5.4 Validity can depend on other attributes

Finally, some records are invalidated without their own predicate changing. Chao et al. (2026) distinguish co-referential implicit conflicts, where a later observation updates the same attribute without explicit negation (a new lease in another city), from propagated conflicts, where a later observation updates a different attribute whose consequences cascade (a leg injury that makes a cycling-commute memory inapplicable). They call the

propagated type the harder one because the dependency is never stated. Wenzel and Jatowt (2024) show the same at the level of text: other statements in a stream can change how long a given statement should be expected to hold. No per-predicate rate, however well estimated, captures a hazard that is driven by a different predicate.

### 5.5 What survives

The premise survives in a narrowed form. Change rates differ by orders of magnitude, and the kind of fact carries enough of that difference to matter (Section 4). But "the kind of fact" is best treated as a prior rate class, whose assignment is itself uncertain, to be updated by record-level evidence that is noisy and, where it has been tested, fails to transfer; and some records are exposed to hazards set by other records. What does not survive is the tidy version in which a lookup table from predicate to half-life replaces the uniform decay. A lookup table is an improvement on one constant. It is not the answer.

Survival analysis has the standard vocabulary for the shape that remains. Cox (1972) took the hazard to be an unknown baseline function of time multiplied by a function of explanatory variables, which lets a record's features scale a shared time profile without fixing that profile in advance. In this setting the predicate class would supply the baseline and record-level evidence the covariates. The form is offered as vocabulary rather than as a recommendation: Section 5.4's propagated hazards do not factor that way, and nobody has fitted such a model to memory data.

## 6. Two clocks: why age can count for or against a record

This section derives four results from standard probability. None is new as mathematics; each has a consequence for agent memory that the systems in Section 3 do not reflect.

### 6.1 Under one known rate, age is the whole story and says nothing new

If every record's lifetime is exponential with a known common rate  $\lambda$ , then the probability that a record written  $e$  time units ago still holds is  $\exp(-\lambda e)$ , and the ordering of records by that probability is the ordering by  $e$ . Memorylessness means that how long the value had already held before it was written is irrelevant. This is the model a uniform recency decay implements, and inside that model age is exactly as informative as the decay says and no more.

### 6.2 Under many rates, the pooled hazard falls with age

Now let each record belong to one of  $K$  classes, with class weights  $w_k$  and rates  $\lambda_k$ , and suppose the store does not know which class a record is in. The

pooled survival function is  $S(t) = \sum_k w_k \exp(-\lambda_k t)$ , and the pooled hazard is

$$h(t) = \sum_k w_k \lambda_k \exp(-\lambda_k t) / S(t).$$

The right-hand side is the average of  $\lambda_k$  under weights proportional to  $w_k \exp(-\lambda_k t)$ , which are the posterior class probabilities for a record that has survived to  $t$ . So  $h(t) = E[\lambda_k | \text{survived to } t]$ , and its derivative is minus the posterior variance of  $\lambda_k$ , which cannot be positive. The pooled hazard is non-increasing, and strictly decreasing whenever the rates differ.

Proschan (1963) gave this explanation of an observed decreasing failure rate in pooled air-conditioner failures across an aircraft fleet: each aircraft's failures were exponential, but at a different rate, and the mixture therefore looked as if old systems were more reliable. Vaupel et al. (1979) made the same point for human mortality, arguing that life tables that ignore heterogeneity underestimate individual aging. The consequence here is that a single decay fitted to a store whose records have heterogeneous volatility is misspecified in a known direction. Survival itself is evidence of membership in a stable class, so a long-surviving value should become more trustworthy per unit of further time, not less. Mansoor et al. (2026) report that in their data a single exponential could not jointly fit short-term churn and long-term drift; that shows one rate misfitting, not which way the pooled hazard moves with age.

A direct consequence is a testable disagreement between the two models. Take a record  $b$ , written  $t_b$  ago with rate  $\lambda_b$ , and a record  $c$ , written  $t_c$  ago with rate  $\lambda_c$ , with  $t_b > t_c$ . The uniform decay always prefers  $c$ . A rate-aware model prefers  $b$  whenever  $\lambda_c t_c > \lambda_b t_b$ , that is, whenever the ratio of rates exceeds the ratio of ages. For a birthplace record a year old against a current-location record a day old, the ratio of ages is 365, and the ratio of rates is effectively unbounded because a birthplace does not change. Section 11 proposes to score exactly such inversions.

### 6.3 Two clocks, pointing in opposite directions

The derivation in 6.2 separates two quantities that a record's timestamp conflates. Call  $h$  the time the value had already held when it was last confirmed, and  $e$  the time elapsed since that confirmation. Under the mixture, the class posterior at confirmation is proportional to  $w_k \exp(-\lambda_k h)$ , and the probability that the value still holds now is

$$P(\text{valid now} | h, e) = \sum_k w_k(h) \exp(-\lambda_k e),$$

with  $w_k(h)$  proportional to  $w_k \exp(-\lambda_k h)$ .

This is decreasing in  $e$ , as every recency decay assumes, and increasing in  $h$ , which no agent-memory decay

located here represents. Long elapsed time since confirmation is evidence against the record; long tenure before confirmation is evidence for it.

HTTP caching already implements both clocks. When a response carries no explicit expiry, RFC 9111 encourages a cache to use "a heuristic expiration value that is no more than some fraction of the interval since" the Last-Modified time, adding that "A typical setting of this fraction might be 10%" (Fielding et al., 2022). The interval since Last-Modified at fetch time is  $h$ . The resulting rule - treat the copy as fresh while  $e$  is below one tenth of  $h$  - lets the tolerable elapsed time grow with demonstrated tenure. It is a two-clock rule, and under the mixture model it is the qualitatively right shape. The DNS takes the other route: the owner of a record sets its time to live, "the time interval (in seconds) that the resource record may be cached before it should be discarded" (Mockapetris, 1987). The party that knows the volatility states it.

An agent's memory has neither. A record such as "lives in Lisbon" written today almost never says since when, so  $h$  is unknown; and no owner attaches a horizon. Some stores do extract the start of a fact's validity when the conversation states it: Zep's graph keeps, for each edge, the interval over which the fact held alongside the times the edge was created and expired in the store, and it resolves relative expressions such as "two weeks ago" to do so (Rasmussen et al., 2025). As described, those intervals serve temporal queries and are closed by a contradicting edge; the paper does not describe using the start of validity to estimate whether an uncontradicted edge still holds. The raw material for the second clock exists in at least one deployed design. It is not read as evidence.

The reader should not over-read this. Under a single known rate,  $h$  carries no information (6.1). The tenure clock matters in proportion to the uncertainty about which rate applies, which is to say in proportion to exactly the heterogeneity Section 5 documented. That is also the condition under which the HTTP heuristic is justified; for a resource whose volatility the cache knows, an explicit lifetime supersedes it, as the RFC requires.

#### **6.4 Scheduled facts, residual life, and a condition on a published discount**

Not every predicate changes at random. Offices have terms, competitions have cycles, seasons end. Zhang and Choi (2023) give the example of the last Summer Olympics host, with a duration of four years. For such facts the hazard is not constant; it is near zero during the term and concentrated at its end. Dean and Kanazawa (1989) had framed persistence this way for reasoning about change: an agent must predict how long a proposition that has become true will continue to be so, using knowledge about "the tendency for propositions to persist or not as a function of time passing", encoded in a

probabilistic network. Persistence, in that framing, is knowledge the agent has about the proposition, not a property of the clock.

For a fact with total duration  $D$ , observed at an age  $a$  into that duration, the probability that it still holds after a further time  $m$  is the residual survival  $P(D > a + m \mid D > a)$ , not the total-duration survival  $P(D > m)$ . The two coincide trivially when  $a = 0$ , and they coincide for every  $a$  and  $m$  only when  $D$  is exponential. For a fixed four-year term observed at a uniformly random point within it, the residual life is uniform on zero to four years, so the probability of still holding after  $m$  years is  $1 - m/4$ ; the total-duration survival  $P(D > m)$  is one for every  $m$  below four.

This bears on the one confidence discount built from fact durations that was located for this paper. Zhang and Choi (2023) scale a QA system's confidence by the predicted probability that the duration is at least the misalignment  $m$ ,  $c$  times the sum over  $d$  of  $P(d \mid q, a)$  for  $d$  at least  $m$ . For their knowledge-base training source, durations were derived as the time from the start of one answer to the start of the next. That is the total-duration survival. It is exact if the fact began at the training snapshot or if durations are exponential; otherwise it treats every fact as freshly started at the snapshot. That overstates the probability of still holding for facts whose hazard rises with age, such as fixed terms, and understates it for facts whose hazard falls with age, such as the pooled classes of Section 6.2. The paper's empirical gains are not in question here - they were measured - and in their application the age of a fact at the snapshot is generally unknown, which is the missing- $h$  problem of Section 6.3 in another form. The remark is that the discount carries an assumption that the paper, as read here, does not state, and that the assumption fails for the kind of scheduled fact the paper uses to illustrate what a duration is.

#### **6.5 What the derivations license**

Four statements follow and are stated flatly because they are derivations. A uniform exponential decay, read as validity, asserts one constant hazard for all records. Heterogeneous rates make the pooled hazard fall with age, so a single fitted decay is wrong in a known direction. Time since confirmation and tenure before confirmation carry evidence of opposite sign, and a store that keeps only one timestamp cannot use the second. For facts with schedules, the relevant survival is residual, not total.

The derivations fix the direction of each effect; its magnitude in agent memory turns on data that are not yet public. Nobody has published the distribution of change rates over the predicates that appear in personal-assistant memories, and the size of the gain from a rate-aware discount depends on that distribution.

## 7. The missing prediction step: silent change

### 7.1 Filtering has two steps, and memory systems implement one

Chao et al. (2026) frame conversational memory as latent state tracking, drawing on hidden Markov models and partially observable decision processes: each utterance is partial, noisy evidence about a user's underlying state, and a memory system should maintain a coherent belief about that state rather than cache snippets. The framing is apt, and it has a consequence the benchmark built on it does not test. A hidden Markov filter alternates two operations (Rabiner, 1989). The prediction step carries the belief forward through the transition model - what the state is likely to be now, given what it was and how much time has passed. The correction step conditions on a new observation. Between observations only the prediction step acts.

Read this way, a recency decay is a prediction step of the crudest kind: one transition kernel for every attribute, which lowers confidence without saying what the value is likely to have become. A predicate-level volatility prior is a better prediction step. And the conflict-handling machinery that most of the recent literature has built is a correction step, which does nothing until an observation arrives.

### 7.2 What the systems do

The designs examined here that retire records at all do so when, and only when, something newer contradicts them. (The decay-based designs of Section 3.1 retire nothing on grounds of validity.)

Zep's graph compares each new edge with semantically related existing edges, and when it finds a temporally overlapping contradiction it closes the old edge's validity interval at the start of the new one; the paper states that it "consistently prioritizes new information when determining edge invalidation" (Rasmussen et al., 2025). Mem0 lets a language model choose, for each extracted candidate fact, among adding it, updating an existing memory, deleting "memories contradicted by new information", or doing nothing; its graph variant marks conflicting relationships invalid rather than removing them (Chhikara et al., 2025). TEPA stores observations as keyed precedents and revokes an active precedent "when fresh evidence contradicts" it under the same key (Zhou et al., 2026). MemStrata retires a value in a bi-temporal ledger when a later value contradicts it under a deterministic subject-relation-object rule (Yadav, 2026). RD-Forget suppresses superseded values through same-slot replacement links while keeping them available for historical questions (Li and Li, 2026). HiCoMER keeps shared and per-member memories valid with a conflict updater and only then retrieves (Shi et al., 2026). TRACE, which treats a returning agent's memory as an admission decision, reconciles a departure checkpoint against the updates that arrived while the agent was

away (Xiong et al., 2026).

These are real advances. TEPA's result that keeping a stale record is worse than keeping nothing - append-only and last-write-wins memory both scored 0.210 under full reversal against 0.309 with no memory - is one of the clearest demonstrations in the literature that staleness costs accuracy (Zhou et al., 2026). MemStrata's measurement that plain retrieval served superseded values 15 to 40 percent of the time when forced to answer, and that cosine similarity separated a contradicted fact from a duplicated one at an AUROC of only 0.59, shows why the correction step cannot be left to embeddings (Yadav, 2026). But every one of them waits for the world to say that it changed.

### 7.3 What the benchmarks score

The benchmarks examined here share that structure. LongMemEval's knowledge-update ability is defined as recognising changes in a user's personal information and updating accordingly; its questions are built from histories that contain the change (Wu et al., 2025). STALE's defining failure is a conflict in which "a later observation invalidates an earlier memory without explicit negation" (Chao et al., 2026); the observation is implicit in form but present in the history. TEPA's drift conditions, TRACE's absence-period updates and MemStrata's evolving-knowledge sets likewise put the change in front of the system. MemoryAgentBench lists selective forgetting among four core competencies for memory agents (Hu et al., 2025), and LoCoMo builds very long conversations over temporal event graphs (Maharana et al., 2024); as described in their abstracts, neither withholds a change from the agent and scores whether it is anticipated. Sun and Zhang (2026) pair a confident memory claim with a contradicting observation in a grid world and find that, in their primary GPT-4o setting, an agent that took its memory at face value died over twice as frequently as it did with no memory at all; again the contradicting observation is supplied.

In none of these is there a record whose value has changed in the world while the history contains no trace of the change. That is the silent-change case, and it is the only case in which a volatility prior does work that a contradiction detector cannot. The claim is scoped: it covers the systems and benchmarks located for this paper, read through their published descriptions. A benchmark that withholds changes may exist and not have been found. None of the conversational-memory benchmarks examined here does it. The one located instrument that does is Nakayashiki's (2026) controlled study, taken up in Section 7.4: a constraint in the agent's store has been withdrawn by a later record that is "reachable only through" the memory's provenance path, and the agent must choose which paths to spend a small verification budget on. It is the silent-change case in miniature, but its own limitations list "Six-memory

stores, two scripted domains, one system prompt", and it has no conversational history and does not vary elapsed time or how volatile the withheld fact is.

#### 7.4 What crawling did that agents cannot copy directly

Web crawling faced the silent-change case from the start, because sources do not notify copies of their changes (Cho and Garcia-Molina, 2003b). The solution had three parts: estimate each page's change rate from repeated visits (Cho and Garcia-Molina, 2003a; Avrachenkov et al., 2020), schedule re-visits to maximise expected freshness under a bandwidth budget (Cho and Garcia-Molina, 2000), and decline to spend budget on content that will be stale before it is used (Olston and Pandey, 2008). All three depend on cheap, repeatable re-observation.

An agent's re-observation is neither cheap nor always available. For a fact about the user, re-observing means asking, which costs the user's attention and, if done often, the user's goodwill. For a fact about an external system, it means a tool call, which Chen et al. (2026) show can be made precise: their design follows a stored plan's links to the exact inputs it was derived from and asks each input's owner for its current version before a protected action. They report that a freshness-only executor acted on the stale plan in every one of 30 live workflows with a revision inserted after planning, while checking derivation inputs completed all 30. The verification channel there exists because the inputs have owners who can be queried.

Nakayashiki (2026) measures what happens when verification is possible but budgeted. Given six memories and a budget to inspect the provenance of two, sixteen models inspected the path behind a constraint that "read as settled" roughly once in every five episodes; after the constraint was withdrawn, their decisions stayed consistent with the stale version in 77.3, 74.7 and 74.7 percent of episodes in the original run, its replication and a domain held out from design, respectively. The record that most needed checking was the one that looked least in need of it. That is the allocation problem a volatility prior exists to solve, and the result shows that models do not solve it unaided.

### 8. What the reader does with temporal information

A validity estimate has to reach a consumer that changes its behaviour. The evidence on how language models use temporal information is consistent, and it points away from timestamps.

The sharpest result is Ahmed et al. (2026). Their benchmark holds 317 reversals, each checked against dated official sources, drawn from four domains (medicine, law, software, platform policy). On it, a superseded document retrieved alongside the question

turned a correct closed-book answer into a wrong one in 30 percent of cases for a Llama model and 37 percent for a Qwen model, with no instruction telling the model to defer to the document. For 50 source-verified reversals with explicit applicability boundaries, they then held a historical document fixed and changed only the evaluation date, so that the same evidence was valid in one prompt and superseded in the other. The applicability gap - the rate of giving the old answer while it was valid minus the rate after it had been superseded - was 0.04 to 0.14 across four models with dates alone. When the validity boundary was stated explicitly, Qwen-72B made all 50 required transitions in both prose and tabular formats, and Llama-70B made 47 of 50 with a prose boundary and 50 of 50 with a table (Figure 1b). The two smaller models moved much less. Causal interventions on the internal state at the evaluation date changed the answer while matched controls did not. The authors' summary is that models "can use temporal validity when it is made clear, but do not reliably infer and act on it from ordinary temporal cues."

The same shape appears in the agent setting. In the budgeted store of Nakayashiki (2026), appending "latest source record: day N" to every memory line - a content-free freshness cue - did not materially redirect verification, with a verdict the author labels inconclusive rather than null; a one-sentence rule telling the agent to prefer memories that state a limit on the tempting action did redirect it. In STALE, models often retrieved updated evidence and still failed to act on it; the best evaluated model reached 55.2 percent overall (Chao et al., 2026). Sun and He (2026) locate one cause in how responses are checked: an audit that starts from the draft checks what the response says, and a stale dependency is usually unsaid.

Parametric reading shows the same insensitivity. Zhao et al. (2024) report that pretrained models with a 2022 cutoff mostly answer time-sensitive questions with earlier knowledge, around 2019, and that aligning them to their latest knowledge improved performance by up to 62 percent even without mentioning time explicitly. Cheng et al. (2024) find that a model's effective knowledge cutoff differs from its reported one and differs across sub-resources. Kasai et al. (2022) observed that GPT-3 tended to return outdated answers when retrieved documents were insufficient. Retrieval components carry biases in both directions: An et al. (2026) report that re-rankers prefer older documents with richer semantic content even where those documents have been overtaken by events, while Prior et al. (2026) report that web search in legal QA shows a marked recency bias when a historical version of a statute governs the facts.

Two conclusions follow for the design of a validity-aware memory. The output of a volatility estimate should not be a timestamp placed in the prompt, because the evidence is that readers largely ignore timestamps. And

it should probably not be a scalar folded into a retrieval score either, because the reader never sees it. The format with evidence behind it is an explicit statement of applicability - "this held as of March; addresses of this user have changed twice in two years; treat as unconfirmed" - or an action taken before the reader is involved: verify, withhold, or ask. The author's earlier paper on trust values reached a parallel conclusion for source trust, that the defences which report guarantees use a gate rather than a graded score (Mahendrakar, 2026c). The analogy is noted, not relied on: the evidence here is about temporal cues specifically.

## 9. The strongest evidence that plain recency works

The case against this paper's premise deserves its best statement.

Recency signals help on the benchmarks that exist. Time-aware indexing and query expansion raised LongMemEval temporal-reasoning recall by an average of 11.3 percent when rounds were the stored unit and 6.8 percent when sessions were (Wu et al., 2025). TEPA matched a strong last-write-wins cache on a clean single-hop consolidation task, which its authors read as confirming that "current-key replacement is the decisive operation for single-hop fact consolidation" (Zhou et al., 2026). RoMem preserved performance on a largely non-temporal memory benchmark while improving temporal ones (Li et al., 2026). In Mansoor et al. (2026), a per-class half-life risk model (3.3 percent stale error against 14.9 percent for SemanticTTL) already captured much of the structure, and the remaining gap to the learned model was within classes; class-aware time-to-live baselines did not (13.5 to 23.1 percent).

Temporal deterioration is not universal. Agarwal and Nenkova (2022), studying tasks whose labels do not depend on time, found that depending on the task temporal model deterioration "is not necessarily a concern", while adaptation to newer data helped in every case. Luu et al. (2022) found stronger effects of temporal misalignment than previously reported across eight tasks; Lazaridou et al. (2021) found that language-model performance worsens steadily beyond the training period and that scale alone does not fix it. Liska et al. (2022) found that parametric updates helped most for questions about higher-frequency named entities. The picture is that time matters very unevenly, which is the premise here, but it also means that for many stored records elapsed time matters little either way.

And in the conflict case, newer usually is current. Where the store holds two records and the later one reports a change, preferring it is correct most of the time, and a volatility prior adds little.

None of these bears on the silent-change case. Each measures a setting in which either the change is observed or the label does not depend on time. The

benchmarks on which recency looks sufficient are the ones Section 7.3 showed supply the change. Recency works where the world announces itself; the question here is what to do where it does not.

## 10. A decision structure for reading a stored record

Algorithm 1 states the decision a validity-aware read would have to make. It is a specification of quantities and their order, derived from Sections 5 to 8, and is offered for scoring on the benchmark of Section 11.

Four features separate it from a recency term. The clock is confirmation, not retrieval, so reading a record does not refresh it (Section 3.2). Tenure enters with the opposite sign to elapsed time (Section 6.3). Scheduled predicates are handled by their end dates, not by a decay (Section 6.4). And the output is an action or an explicit statement, not a weight inside a ranking (Section 8).

Line 7 is where Section 5's difficulties live, and the algorithm does not solve them. It names three kinds of record-level evidence: the subject's own change history, which is noisy because it mixes corrections with changes; whether earlier rewrites of this record were corrections or changes, which is the distinction a store would have to record at write time; and dependencies on other attributes, which is STALE's propagated case. Line 10 needs a cost of error that depends on the query, which no memory benchmark located here supplies. Line 11 is the crawler's re-visit, and it is available only when someone can be asked.

## 11. What would settle the open parts

### 11.1 A silent-change benchmark

The missing instrument is a memory benchmark in which the world changes and the history does not say so. It would need six components.

The first is a set of predicates that appear in assistant memory - residence, employer, relationship status, dietary constraint, device, project status, travel location, preferences of several kinds - each with a change process. Where public histories exist, rates can be estimated from them, as Mousavi et al. (2024) and Chen et al. (2021) did with Wikidata validity intervals or Jang et al. (2022b) did with differences between successive snapshots; where they do not, the assumed process should be stated and results reported as conditional on it. For world facts, ground truth can be kept current by re-running knowledge-graph queries, as Meem et al. (2024) do for present-anchored questions.

The second is a set of user histories in which some values change at simulated times with no mention in the conversation, and others change with mention. The third is a verification action with a stated cost at query time - asking the user, calling a tool - with scoring that penalises both acting on a silently stale value and

**Algorithm 1.** Validity-aware use of one retrieved memory record

```

Input: record  $r = (\text{subject } s, \text{predicate } p, \text{value } v, t_{\text{conf}}, t_{\text{start}} \text{ or NONE})$ 
      query time  $t_{\text{now}}$ ; cost of acting on a stale value  $C_{\text{err}}(\text{query})$ 
      cost of verifying  $C_{\text{ver}}(p, s)$ ; verification channel  $V$  or NONE
1   $e \leftarrow t_{\text{now}} - t_{\text{conf}}$  # clock 1: elapsed since confirmation
2   $h \leftarrow t_{\text{conf}} - t_{\text{start}}$  if known # clock 2: tenure when confirmed
3  if  $p$  has a known schedule and its end precedes  $t_{\text{now}}$ :
4    return EXPIRED( $v$ ) # residual life, not a decay
5  prior  $\leftarrow$  rate classes ( $w_k, \lambda_k$ ) for predicate  $p$ 
6  if  $h$  known:  $w_k \leftarrow w_k * \exp(-\lambda_k * h)$ , renormalise
7  update  $w_k$  with record-level evidence: subject history,
8    corrections vs changes, dependent attributes
9   $q \leftarrow \sum_k w_k * \exp(-\lambda_k * e)$  #  $P(v \text{ still holds})$ 
10 if  $(1 - q) * C_{\text{err}} \leq C_{\text{ver}}$ : return USE( $v$ , horizon statement)
11 if  $V$  exists:  $v' \leftarrow V(s, p)$ ; write ( $s, p, v', t_{\text{now}}$ ); return USE( $v'$ )
12 if  $C_{\text{err}}$  is severe: return ABSTAIN_OR_ASK
13 return USE( $v$ , explicit caveat with  $t_{\text{conf}}$  and class)

```

verifying needlessly. The fourth is reporting by predicate class and by elapsed time, so that a uniform decay, a per-predicate prior and a record-level model can be compared on one axis. The fifth is a set of inversion pairs: an old record of a stable predicate beside a recent record of a volatile one, where the rate-aware and uniform models make opposite predictions (Section 6.2). The sixth is tenure, stated in some histories ("I have lived here for ten years") and absent from others, to measure whether the second clock is used.

One located study scores a silent change against a priced verification action, the core of the second and third components: Nakayashiki (2026) withholds a withdrawn constraint from a six-memory store and gives the agent a budget of two record inspections in its main runs. It does so in a single scripted store per domain, without a conversational history, tenure or inversion pairs, and without varying elapsed time or predicate volatility. Nothing in the literature located here scores the fifth or sixth component, or the second and third across predicates and elapsed times.

**11.2 Four narrower studies**

A calibration study of existing recency terms. Take any store with timestamps and a later ground truth for each record - LongMemEval and STALE both have the raw material - and plot the empirical fraction still valid against elapsed time, by predicate. If the curves separate by class, the uniform decay is miscalibrated in a measurable way; if they do not, the premise here is weaker for assistant memory than for world knowledge.

A reconciliation of the two volatility units. Run MuLan's relation-level labels and DynamicQA's per-fact edit counts on a common set of facts, with edits hand-classified as change or correction. Dey et al. (2026) show the disagreement is real; the missing step is attributing it.

A test of residual versus total survival. Re-run Zhang and Choi's (2023) calibration experiment with the discount replaced by a residual-life version for the subset of facts whose start dates are known, and report the difference for scheduled and unscheduled relations separately.

A format test for the reader. Ahmed et al. (2026) varied the evaluation date and the stated boundary. The agent-memory analogue would vary what the reader is told about a record - timestamp only, predicate class, estimated horizon, or an instruction to verify - and measure both the decision and whether a verification action is taken.

**12. The strongest case against this paper**

Four objections deserve an answer.

First: nobody claims recency is a validity estimate, so the paper attacks a reading no one holds. The paper agrees that the designers of the recency terms examined did not make that claim, and says so in Section 3.2. The reply is that the reading does not need a proponent to be in force. When a store holds one record and an agent must decide whether to use it, whatever temporal term decides whether the record reaches the model is functioning as a validity estimate, whatever its name. Section 3.3 gives three routes by which this happens.

Second: in personal-assistant memory the relevant horizons are short and the volatile facts are few, so a uniform decay is a harmless approximation. That may be true, and it has not been measured - which is the point. The only published calibration-relevant numbers come from world knowledge and web pages (Section 4), where the spread is two or more orders of magnitude. Whether assistant memories are more homogeneous is an empirical question the field has not yet asked; Section 11 specifies a benchmark that would answer it.

Third: modern long-context models can reason about time given the raw history, so explicit volatility modelling is unnecessary. The evidence in Section 8 runs the other way for the specific capability at issue: models largely ignore date cues when deciding whether evidence still applies, and act on explicit boundaries. They may improve. The claim here is about the published results, not about what future models will do.

Fourth: the derivations in Section 6 are textbook material, and a paper built on them adds little. The derivations are textbook; that is why they are stated

flatly. What is not in the textbooks is their application: that the tenure clock has the opposite sign to the one every agent-memory decay uses, that HTTP caching already encodes it while agent memories do not, and that the one located duration-based discount assumes a condition that fails for the kind of fact it uses to illustrate durations. If these points are obvious, they are nevertheless absent from the designs examined.

A fifth, weaker objection is that RoMem already does what this paper asks. It does part of it, and Section 4.4 credits it with the strongest evidence for the premise. What it does not do is estimate whether an uncontradicted record still holds or act on that estimate, and its rates are not calibrated as hazards by its authors' own account.

### 13. What is not known, and the quality of the evidence

The distribution of change rates over the predicates in personal-assistant memory is not known. Every rate cited in this paper was measured on world knowledge, web pages or synthetic scenarios.

Whether predicate-level priors, record-level histories, or their combination predict staleness best is not known. The one direct comparison of the two units, in parametric knowledge, found them producing opposite conclusions and could not declare either right (Dey et al., 2026). In web caching, class-level rules degraded more gracefully out of distribution than within-class refinement did (Mansoor et al., 2026); whether that transfers to memory is not known.

How often silent change happens relative to announced change in real agent use is not known. If most changes are announced, contradiction handling covers most of the harm and a volatility prior matters less.

Whether language models can supply calibrated volatility priors on demand is not known. Probes find mutability encoded in representations (Fierro et al., 2024), and fact-duration models trained with distant supervision beat simple baselines (Zhang and Choi, 2023), but neither result reports calibration of a hazard.

The quality of the evidence is uneven. The web-crawl literature is old, large-scale and measured on the thing it claims. The time-sensitive QA literature is large and consistent on the direction of the effect, but it measures model accuracy, which confounds change rate with popularity and training-data coverage. The agent-memory results are recent, mostly single papers, mostly preprints, often on synthetic scenarios, and several were posted within weeks of this paper. The two results that carry the most weight for the reader-format conclusion (Ahmed et al., 2026; Nakayashiki, 2026) are each one study, on 50 reversals and on one store design respectively.

### 14. Conclusion

A recency decay was designed to decide what an agent attends to and keeps. Read as a statement about truth, it asserts one constant hazard for every kind of fact. The evidence says that change rates differ by orders of magnitude by kind of fact, and one agent-memory system that learned a rate per relation lost two thirds of its retrieval quality when the rates were made equal. The evidence also says that "kind of fact" is an unstable unit: the two published ways of measuring volatility disagree, per-record histories mix change with correction, variation inside classes is large, and some records go stale because a different attribute changed. Standard survival arguments add that under heterogeneous rates, how long a value had held when it was confirmed is evidence for it, while time since confirmation is evidence against, and a single timestamp cannot carry both.

The systems and benchmarks examined here retire a record on grounds of validity only when something newer contradicts it. That leaves nearly unscored the one case in which a rate model does work that a contradiction detector cannot: a change the agent never hears about, which one controlled study located here scores in scripted six-memory stores. And the evidence on readers says that whatever estimate is formed should reach them as an explicit horizon or as an action, not as a date.

Age, in short, is evidence of staleness only through a rate, and the rate is a property of what the record says, qualified by what is known about this particular record - not of the clock. A benchmark that withholds changes to facts of differing volatility and prices verification would show how much that distinction is worth.

### References

- Adar, E., Teevan, J., Dumais, S. T., & Elsas, J. L. (2009). The web changes everything. *Proceedings of the Second ACM International Conference on Web Search and Data Mining (WSDM 2009)*, 282-291. doi:10.1145/1498759.1498837.
- Agarwal, O., & Nenkova, A. (2022). Temporal Effects on Pre-trained Models for Language Processing Tasks. *Transactions of the Association for Computational Linguistics*, 10, 904-921. doi:10.1162/tacl\_a\_00497.
- Ahmed, M. S., Galke Poech, L., & Rottger, R. (2026). Stale-Documents Poisoning: When Outdated Retrieval Overrides Correct Model Answers. *arXiv:2609.31342*. doi:10.48550/arXiv.2609.31342. [PREPRINT]
- Almqvist, A., & Jatowt, A. (2019). Towards Content Expiry Date Determination: Predicting Validity Periods of Sentences. *Advances in Information Retrieval (ECIR 2019)*, *Lecture Notes in Computer Science*, 86-101. doi:10.1007/978-3-030-15712-8\_6.
- An, S., Lee, H., Yuan, S., Chen, C. J., Hsieh, C.-J., Mohan, V., & Min, A. (2026). FRESKO: Benchmarking and Optimizing Re-rankers for Evolving Semantic Conflict in Retrieval-Augmented Generation. *arXiv:2604.14227*. doi:10.48550/arXiv.2604.14227. [PREPRINT]
- Anderson, J. R., & Schooler, L. J. (1991). Reflections of the Environment in Memory. *Psychological Science*, 2(6), 396-408. doi:10.1111/j.1467-9280.1991.tb00174.x.
- Avrachenkov, K., Patil, K., & Thoppe, G. (2020). Change Rate Estimation and Optimal Freshness in Web Page Crawling.

- VALUETOOLS 2020. arXiv:2004.02167. doi:10.48550/arXiv.2004.02167.
- Brewington, B. E., & Cybenko, G. (2000). How dynamic is the Web? *Computer Networks*, 33(1-6), 257-276. doi:10.1016/s1389-1286(00)00045-1.
- Chao, H., Bai, Y., Sheng, R., Li, T., & Sun, Y. (2026). STALE: Can LLM Agents Know When Their Memories Are No Longer Valid? arXiv:2605.06527. doi:10.48550/arXiv.2605.06527. [PREPRINT]
- Chen, E., Wang, S., & Brinton, C. G. (2026). Fresh Memory, Stale Plans: Derivation Currency for Distributed LLM-Agent Memory. arXiv:2609.03340. doi:10.48550/arXiv.2609.03340. [PREPRINT]
- Chen, W., Wang, X., & Wang, W. Y. (2021). A Dataset for Answering Time-Sensitive Questions. *NeurIPS 2021 Datasets and Benchmarks Track*. arXiv:2108.06314. doi:10.48550/arXiv.2108.06314.
- Cheng, J., Marone, M., Weller, O., Lawrie, D., Khashabi, D., & Van Durme, B. (2024). Dated Data: Tracing Knowledge Cutoffs in Large Language Models. arXiv:2403.12958. doi:10.48550/arXiv.2403.12958.
- Chhikara, P., Khant, D., Aryan, S., Singh, T., & Yadav, D. (2025). Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. arXiv:2504.19413. doi:10.48550/arXiv.2504.19413. [PREPRINT]
- Cho, J., & Garcia-Molina, H. (2000). Synchronizing a database to improve freshness. *Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data*, 117-128. doi:10.1145/342009.335391.
- Cho, J., & Garcia-Molina, H. (2003a). Estimating frequency of change. *ACM Transactions on Internet Technology*, 3(3), 256-290. doi:10.1145/857166.857170.
- Cho, J., & Garcia-Molina, H. (2003b). Effective page refresh policies for Web crawlers. *ACM Transactions on Database Systems*, 28(4), 390-426. doi:10.1145/958942.958945.
- Cox, D. R. (1972). Regression Models and Life-Tables. *Journal of the Royal Statistical Society Series B*, 34(2), 187-202. doi:10.1111/j.2517-6161.1972.tb00899.x.
- Dean, T., & Kanazawa, K. (1989). A model for reasoning about persistence and causation. *Computational Intelligence*, 5(2), 142-150. doi:10.1111/j.1467-8640.1989.tb00324.x.
- Dey, R., Ounis, I., McDonald, G., & Moshfeghi, Y. (2026). Temporal Fact Conflicts in LLMs: Reproducibility Insights from Unifying DYNAMICQA and MULAN. arXiv:2603.15892. doi:10.48550/arXiv.2603.15892. [PREPRINT]
- Dhingra, B., Cole, J. R., Eisenschlos, J. M., Gillick, D., Eisenstein, J., & Cohen, W. W. (2022). Time-Aware Language Models as Temporal Knowledge Bases. *Transactions of the Association for Computational Linguistics*, 10, 257-273. arXiv:2106.15110. doi:10.48550/arXiv.2106.15110.
- Dong, A., Chang, Y., Zheng, Z., Mishne, G., Bai, J., Zhang, R., Buchner, K., Liao, C., & Diaz, F. (2010). Towards recency ranking in web search. *Proceedings of the Third ACM International Conference on Web Search and Data Mining (WSDM 2010)*, 11-20. doi:10.1145/1718487.1718490.
- Dong, X. L., Berti-Equille, L., & Srivastava, D. (2009). Truth discovery and copying detection in a dynamic world. *Proceedings of the VLDB Endowment*, 2(1), 562-573. doi:10.14778/1687627.1687691.
- Efron, M., & Golovchinsky, G. (2011). Estimation methods for ranking recent information. *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 495-504. doi:10.1145/2009916.2009984.
- Fetterly, D., Manasse, M., Najork, M., & Wiener, J. (2003). A large-scale study of the evolution of web pages. *Proceedings of the Twelfth International Conference on World Wide Web (WWW 2003)*. doi:10.1145/775152.775246.
- Fielding, R., Nottingham, M., & Reschke, J. (Eds.) (2022). "HTTP Caching". RFC 9111, Internet Engineering Task Force. doi:10.17487/RFC9111.
- Fierro, C., Garneau, N., Bugliarello, E., Kementchedjheva, Y., & Sogaard, A. (2024). MuLan: A Study of Fact Mutability in Language Models. arXiv:2404.03036. doi:10.48550/arXiv.2404.03036.
- Hu, Y., Wang, Y., & McAuley, J. (2025). Evaluating Memory in LLM Agents via Incremental Multi-Turn Interactions. arXiv:2507.05257. doi:10.48550/arXiv.2507.05257. [PREPRINT]
- Jang, J., Ye, S., Yang, S., Shin, J., Han, J., Kim, G., Choi, S. J., & Seo, M. (2022a). Towards Continual Knowledge Learning of Language Models. *ICLR 2022*. arXiv:2110.03215. doi:10.48550/arXiv.2110.03215.
- Jang, J., Ye, S., Lee, C., Yang, S., Shin, J., Han, J., Kim, G., & Seo, M. (2022b). TemporalWiki: A Lifelong Benchmark for Training and Evaluating Ever-Evolving Language Models. *EMNLP 2022*. arXiv:2204.14211. doi:10.48550/arXiv.2204.14211.
- Kasai, J., Sakaguchi, K., Takahashi, Y., Le Bras, R., Asai, A., Yu, X., Radev, D., Smith, N. A., Choi, Y., & Inui, K. (2022). RealTime QA: What's the Answer Right Now? arXiv:2207.13332. doi:10.48550/arXiv.2207.13332.
- Lazaridou, A., Kuncoro, A., Gribovskaya, E., Agrawal, D., Liska, A., Terzi, T., Gimenez, M., de Masson d'Autume, C., Kocisky, T., Ruder, S., Yogatama, D., Cao, K., et al. (2021). Mind the Gap: Assessing Temporal Generalization in Neural Language Models. *NeurIPS 2021*. arXiv:2102.01951. doi:10.48550/arXiv.2102.01951.
- Leblay, J., & Chekol, M. W. (2018). Deriving Validity Time in Knowledge Graph. *Companion Proceedings of The Web Conference 2018*, 1771-1776. doi:10.1145/3184558.3191639.
- Li, W. W., Zhang, J., Yang, X. J., Ma, T., & Guo, Y. (2026). Time is Not a Label: Continuous Phase Rotation for Temporal Knowledge Graphs and Agentic Memory. *EMNLP 2026*. arXiv:2604.11544. doi:10.48550/arXiv.2604.11544.
- Li, X., & Croft, W. B. (2003). Time-based language models. *Proceedings of the Twelfth International Conference on Information and Knowledge Management (CIKM 2003)*, 469-475. doi:10.1145/956863.956951.
- Li, Y., & Li, Y. (2026). What Should an Agent Forget? Separating What Is Stored from What Is Used. arXiv:2609.10263. doi:10.48550/arXiv.2609.10263. [PREPRINT]
- Liska, A., Kocisky, T., Gribovskaya, E., Terzi, T., Sezener, E., Agrawal, D., de Masson d'Autume, C., Scholtes, T., Zaheer, M., Young, S., Gilsenan-McMahon, E., Austin, S., et al. (2022). StreamingQA: A Benchmark for Adaptation to New Knowledge over Time in Question Answering Models. arXiv:2205.11388. doi:10.48550/arXiv.2205.11388.
- Luu, K., Khashabi, D., Gururangan, S., Mandyam, K., & Smith, N. A. (2022). Time Waits for No One! Analysis and Challenges of Temporal Misalignment. *NAACL 2022*. arXiv:2111.07408. doi:10.48550/arXiv.2111.07408.
- Maharana, A., Lee, D.-H., Tulyakov, S., Bansal, M., Barbieri, F., & Fang, Y. (2024). Evaluating Very Long-Term Conversational Memory of LLM Agents. arXiv:2402.17753. doi:10.48550/arXiv.2402.17753.
- Mahendrakar, P. (2026a). Newer Is Not Truer: Three Presuppositions Behind the Recency Rule in Long-Term Agent Memory, and Why Retaining Both Records Relocates the Decision Rather Than Making It. *Life of Research*. doi:10.5281/zenodo.22182350.
- Mahendrakar, P. (2026b). Scored Before the Question Exists: What a Write-Time Importance Value in LLM Agent Memory Predicts, Why an Additive Term Is Not a Prior, and the Ablation the Canonical Architecture Did Not Run. *Life of Research*. doi:10.5281/zenodo.22868532.
- Mahendrakar, P. (2026c). A Trust Score Needs a Consumer: Four Places a Per-Source Trust Value Can Act on an LLM Agent, the Missing Test of a Graded Discount Against an Adaptive Attacker, and Why the Defences That Report Guarantees Use a Gate Instead of a Score. *Life of Research*. doi:10.5281/zenodo.23026299.
- Mansoor, M., Ahmad, T., & Yoon, Y.-C. (2026). Risk-Constrained Freshness-Aware Semantic Caching for Open-Web Retrieval-

- Augmented LLMs. arXiv:2607.04281. doi:10.48550/arXiv.2607.04281. [PREPRINT]
- Marjanovic, S. V., Yu, H., Atanasova, P., Maistro, M., Lioma, C., & Augenstein, I. (2024). DYNAMICQA: Tracing Internal Knowledge Conflicts in Language Models. Findings of EMNLP 2024. arXiv:2407.17023. doi:10.48550/arXiv.2407.17023.
- Meem, J. A., Rashid, M. S., Dong, Y., & Hristidis, V. (2024). PAT-Questions: A Self-Updating Benchmark for Present-Anchored Temporal Question-Answering. Findings of ACL 2024. arXiv:2402.11034. doi:10.48550/arXiv.2402.11034.
- Mockapetris, P. V. (1987). Domain names - implementation and specification. RFC 1035, Internet Engineering Task Force. doi:10.17487/RFC1035.
- Mousavi, S. M., Alghisi, S., & Riccardi, G. (2024). DyKnow: Dynamically Verifying Time-Sensitive Factual Knowledge in LLMs. arXiv:2404.08700. doi:10.48550/arXiv.2404.08700.
- Nakayashiki, K. (2026). When Stale Constraints Go Unchecked: Budgeted Verification Failures in Inherited Agent Memory. arXiv:2608.25553. doi:10.48550/arXiv.2608.25553. [PREPRINT]
- Olston, C., & Pandey, S. (2008). Recrawl scheduling based on information longevity. Proceedings of the 17th International Conference on World Wide Web (WWW 2008), 437-446. doi:10.1145/1367497.1367557.
- Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative Agents: Interactive Simulacra of Human Behavior. arXiv:2304.03442. doi:10.48550/arXiv.2304.03442.
- Park, Y., Yoon, C., Park, J., Lee, D., Jeong, M., & Kang, J. (2025). ChroKnowledge: Unveiling Chronological Knowledge of Language Models in Multiple Domains. ICLR 2025. arXiv:2410.09870. doi:10.48550/arXiv.2410.09870.
- Piryan, B., Abdallah, A., Mozafari, J., Anand, A., & Jatowt, A. (2026). It's High Time: A Survey of Temporal Question Answering. Proceedings of ACL 2026. arXiv:2505.20243. doi:10.48550/arXiv.2505.20243.
- Prior, M., Schultz, A., & Grabmair, M. (2026). Asking For An Old Friend: Diagnosing and Mitigating Temporal Failure Modes in LLM-based Statutory Question Answering. ICAIL 2026. arXiv:2605.23497. doi:10.48550/arXiv.2605.23497.
- Proshan, F. (1963). Theoretical Explanation of Observed Decreasing Failure Rate. *Technometrics*, 5(3), 375-383. doi:10.1080/00401706.1963.10490105.
- Rabiner, L. R. (1989). A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2), 257-286. doi:10.1109/5.18626.
- Rasmussen, P., Paliychuk, P., Beauvais, T., Ryan, J., & Chalef, D. (2025). Zep: A Temporal Knowledge Graph Architecture for Agent Memory. arXiv:2501.13956. doi:10.48550/arXiv.2501.13956. [PREPRINT]
- Shi, Y., Yao, R., Li, A., Wu, Y., Jiang, Z., & Liu, X. (2026). Not All Memories Are Equal: Hierarchical Collaborative Memory for Validity-Aware Retrieval in LLM Agents. arXiv:2609.30289. doi:10.48550/arXiv.2609.30289. [PREPRINT]
- Sun, H., & He, L. (2026). When Memory Updates but Behavior Does Not: Repairing Implicit Stale Dependencies in Personalized Agent Responses. arXiv:2608.01619. doi:10.48550/arXiv.2608.01619. [PREPRINT]
- Sun, Y., & Zhang, Y. (2026). When Memory Lies: An Empirical Study of Spatial Memory Staleness in VLM Agents. arXiv:2608.04574. doi:10.48550/arXiv.2608.04574. [PREPRINT]
- Vaupel, J. W., Manton, K. G., & Stallard, E. (1979). The impact of heterogeneity in individual frailty on the dynamics of mortality. *Demography*, 16(3), 439-454. doi:10.2307/2061224.
- Vu, T., Iyyer, M., Wang, X., Constant, N., Wei, J., Wei, J., Tar, C., Sung, Y.-H., Zhou, D., Le, Q., & Luong, T. (2023). FreshLLMs: Refreshing Large Language Models with Search Engine Augmentation. arXiv:2310.03214. doi:10.48550/arXiv.2310.03214.
- Wei, L., Peng, X., Dong, X., Xie, N., & Wang, B. (2026). FadeMem: Biologically-Inspired Forgetting for Efficient Agent Memory. arXiv:2601.18642. doi:10.48550/arXiv.2601.18642. [PREPRINT]
- Wenzel, G., & Jatowt, A. (2024). Temporal Validity Change Prediction. arXiv:2401.00779. doi:10.48550/arXiv.2401.00779.
- Wu, D., Wang, H., Yu, W., Zhang, Y., Chang, K.-W., & Yu, D. (2025). LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory. ICLR 2025. arXiv:2410.10813. doi:10.48550/arXiv.2410.10813.
- Xiong, W., Zhang, S., Gu, S., Tang, B., Li, Z., Xiong, F., Wen, Y., & Wen, M. (2026). TRACE: Governing Memory Validity in Evolving Multi-Agent Systems. arXiv:2609.33517. doi:10.48550/arXiv.2609.33517. [PREPRINT]
- Yadav, N. (2026). Temporal Validity in Retrieval Memory: Eliminating Stale-Fact Errors for AI Agents over Evolving Knowledge. arXiv:2606.26511. doi:10.48550/arXiv.2606.26511. [PREPRINT]
- Zhang, M. J. Q., & Choi, E. (2021). SituatedQA: Incorporating Extra-Linguistic Contexts into QA. EMNLP 2021. arXiv:2109.06157. doi:10.48550/arXiv.2109.06157.
- Zhang, M. J. Q., & Choi, E. (2023). Mitigating Temporal Misalignment by Discarding Outdated Facts. EMNLP 2023. arXiv:2305.14824. doi:10.48550/arXiv.2305.14824.
- Zhao, B., Brumbaugh, Z., Wang, Y., Hajishirzi, H., & Smith, N. A. (2024). Set the Clock: Temporal Alignment of Pretrained Language Models. Findings of ACL 2024. arXiv:2402.16797. doi:10.48550/arXiv.2402.16797.
- Zhong, W., Guo, L., Gao, Q., Ye, H., & Wang, Y. (2023). MemoryBank: Enhancing Large Language Models with Long-Term Memory. arXiv:2305.10250. doi:10.48550/arXiv.2305.10250.
- Zhou, B., Khashabi, D., Ning, Q., & Roth, D. (2019). 'Going on a vacation' takes longer than 'Going for a walk': A Study of Temporal Commonsense Understanding. EMNLP 2019. arXiv:1909.03065. doi:10.48550/arXiv.1909.03065.
- Zhou, B., Ning, Q., Khashabi, D., & Roth, D. (2020). Temporal Common Sense Acquisition with Minimal Supervision. ACL 2020. arXiv:2005.04304. doi:10.48550/arXiv.2005.04304.
- Zhou, Y., Ouyang, Y., Zheng, K., & Xiang, S. (2026). TEPA: Revoking Stale Memories for Conflict-Robust Language Agents. arXiv:2608.07429. doi:10.48550/arXiv.2608.07429. [PREPRINT]