

Time is Not a Label: Continuous Phase Rotation for Temporal Knowledge Graphs and Agentic Memory

Weixian Waylon Li*
University of Edinburgh
waylon.li@ed.ac.uk

Jiaxin Zhang†
LIGHTSPEED
jiaxijzhang@global.tencent.com

Xianan Jim Yang
University of St Andrews
xy60@st-andrews.ac.uk

Tiejun Ma†
University of Edinburgh
tiejun.ma@ed.ac.uk

Yiwen Guo†
Independent Researcher
guoyiwen89@gmail.com

Abstract

Structured memory representations such as knowledge graphs are central to autonomous agents and other long-lived systems. However, most existing approaches model time as discrete metadata, either sorting by recency (burying old-yet-permanent knowledge), simply overwriting outdated facts, or requiring an expensive LLM call at every ingestion step, leaving them unable to distinguish persistent facts from evolving ones. To address this, we introduce ROMEM¹, a drop-in temporal knowledge graph module for structured memory systems, applicable to agentic memory and beyond. A pretrained *Semantic Speed Gate* maps each relation’s text embedding to a volatility score, learning from data that evolving relations (e.g., “president of”) should rotate fast while persistent ones (e.g., “born in”) should remain stable. Combined with continuous phase rotation, this enables *geometric shadowing*: obsolete facts are rotated out of phase in complex vector space, so temporally correct facts naturally outrank contradictions without deletion. On temporal knowledge graph completion, ROMEM achieves state-of-the-art results on ICEWS05-15 (72.6 MRR). Applied to agentic memory, it delivers 2~3× MRR and answer accuracy on temporal reasoning (MultiTQ), dominates hybrid benchmark (LoCoMo), preserves static memory with zero degradation (DMR-MS), and generalises zero-shot to unseen financial domains (FinTMMBench).

1 Introduction

Structured memory representations such as knowledge graphs have become widely adopted as the long-term memory substrate for agentic systems (Pan et al., 2024b; Chhikara et al., 2025; Gutierrez

et al., 2024; Gutiérrez et al., 2025; Rasmussen et al., 2025; Huang et al., 2026; Jiang et al., 2026), providing unbounded, structured, and verifiable memory that decouples storage from the LLM.

However, a fundamental challenge is that, most graph-based systems model time as *discrete metadata*, a timestamp that cannot encode whether a relation is permanent or transient (Tang and Guo, 2026). The real world is dynamic (Cai et al., 2023): executive boards shift, borders change, and markets fluctuate. When temporal conflicts arise (e.g., “Obama is president” vs. “Biden is president”), current systems resort to three workarounds: (i) **destructive overwriting**, which erases historical context (Xu et al., 2025; Gutierrez et al., 2024); (ii) **LLM arbitration**, which requires a language model call at every ingestion step to predict symbolic UPDATE/DELETE commands (Chhikara et al., 2025; Rasmussen et al., 2025; Yan et al., 2025); or (iii) **recency sorting**, which ranks facts by timestamp to surface the latest version. Each has notable limitations. Destructive overwriting permanently loses historical context. LLM arbitration may suit short-term conversational memory, but becomes infeasible when scaling to long-term memory with millions of facts. Recency sorting, the most common workaround, appears to work until it silently buries static knowledge: a recency-based system ranks the decades-old fact (Obama, born_in, Hawaii) below fresher but irrelevant entries. Disabling recency bias leaves temporal conflicts unresolved, confusing the downstream LLM (Liu et al., 2024). We call this the *static-dynamic dilemma*: discrete metadata treats all relations identically and cannot resolve temporal conflicts without sacrificing static knowledge.

To this end, we introduce ROMEM, a temporal reasoning module for graph-based agentic memory that internalises time as a continuous geometric operator. Rather than building a new memory system, ROMEM provides a temporal engine for

*This work was done during an internship of Weixian Waylon Li at LIGHTSPEED (UK).

†Corresponding authors.

¹Codes available at: <https://github.com/Tencent/RoMem>

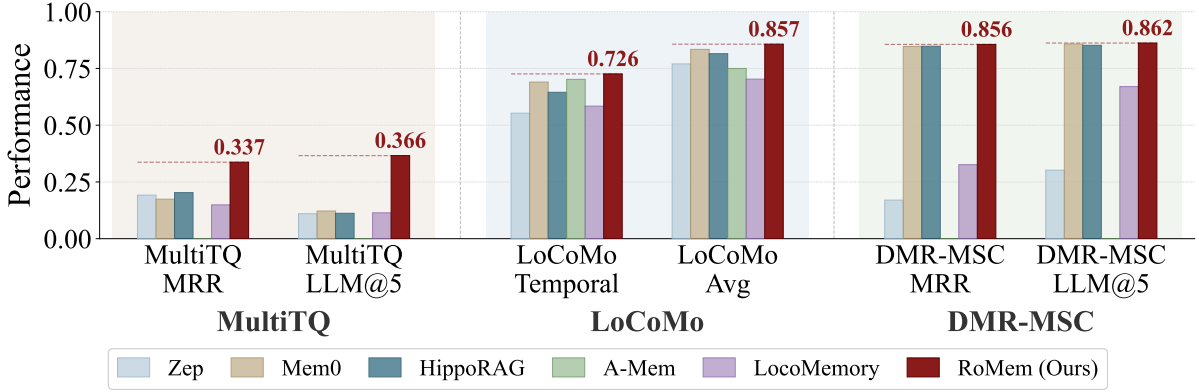


Figure 1: **Performance overview.** ROMEM adds temporal reranking to the same HippoRAG graph-construction and semantic-retrieval pipeline. It improves temporal and hybrid retrieval while preserving performance on largely non-temporal memory. Table 2 reports results for both the closed- and open-source configurations.

the knowledge graph component: it learns to distinguish static from dynamic relations zero-shot and resolves conflicts through geometry rather than database operations. We achieve this through two mechanisms: (1) **Continuous Geometric Shadowing**, which models time as a functional phase shift in complex vector space, rotating dynamic facts out of alignment as they become obsolete while keeping static facts locked in phase; and (2) a **Semantic Speed Gate** that estimates relational volatility from text embeddings, outputting a per-relation scalar $\alpha_r \in (0, 1)$ that controls rotation speed, so static relations ($\alpha_r \approx 0$) remain stable while dynamic ones ($\alpha_r \approx 1$) rotate to shadow obsolete facts. The memory remains strictly append-only, yet the LLM receives a clean, unambiguous context window driven entirely by geometric proximity.

Our main contributions are as follows:

- We formalise the *static-dynamic dilemma* in graph-based agentic memory, showing that discrete timestamp metadata treats all relations identically, preventing temporal conflict resolution without sacrificing static knowledge.
- We formulate temporal conflict resolution as continuous geometric shadowing in complex vector space, replacing destructive database updates and per-ingestion LLM calls with an append-only architecture.
- We introduce a Semantic Speed Gate that addresses this dilemma by learning relational volatility from text embeddings, generalising zero-shot to unseen relations and domains without manual annotation.

- We demonstrate that ROMEM achieves SOTA temporal knowledge graph completion on ICEWS05-15 (72.6 MRR) and, applied to agentic memory, delivers $2\sim 3\times$ MRR and accuracy on temporal reasoning (MultiTQ), dominates hybrid tasks (LoCoMo), preserves static memory (DMR-MSC), and generalises zero-shot to unseen financial domains (FinTMMBench). The performance overview is in Figure 1.

2 Related Work

Agentic Memory Paradigms. Graph-based memory is widely adopted for long-term agent knowledge, with frameworks such as Mem0 (Chhikara et al., 2025), HippoRAG (Gutiérrez et al., 2024; Gutiérrez et al., 2025), Zep (Rasmussen et al., 2025), DialogGSR (Park et al., 2024), LicoMemory (Huang et al., 2026), and DescGraph (Hu et al., 2026a) offering scalable, controllable retrieval. Parametric approaches (Yao et al., 2024; Zhang et al., 2025a) require costly re-training and offer less transparent retrieval (Zhang et al., 2025b; Hu et al., 2026b). Context engineering strategies, including compression (Ye et al., 2025), dynamic context management (Yu et al., 2025; Zhou et al., 2025b; Yan et al., 2025; Salama et al., 2025; Li et al., 2026b; Yan et al., 2026), and reflective evolution (Liang et al., 2024; Zhou et al., 2025a), are typically limited to short-term conversational state.

Temporal Gap in Memory. Existing memory systems lack a native mechanism for managing changing facts. Most treat memory as a static snapshot and resort to three workarounds: (i) destruc-

tive overwriting that permanently erases historical context (Xu et al., 2025; Gutierrez et al., 2024; Yu et al., 2025); (ii) LLM-driven arbitration that predicts UPDATE or DELETE actions during ingestion (Chhikara et al., 2025; Rasmussen et al., 2025; Yan et al., 2025); or (iii) recency-based metadata sorting, which inadvertently buries old-yet-permanent facts (Jiang et al., 2026). LLM arbitration and ROMEM operate at different stages. The former updates memory during ingestion, whereas ROMEM retains the full history and ranks facts at query time. Notably, none of these approaches distinguish *relational volatility*: they apply the same temporal policy to static facts (“born in”) and dynamic ones (“president of”).

Temporal Rotation and Semantic Transfer. TKG embedding methods such as TeRo (Xu et al., 2020) and ChronoR (Sadeghian et al., 2021) already represent time through rotation; temporal rotation itself is therefore not our contribution. These methods learn a separate representation for each timestamp at a fixed resolution. ROMEM instead defines time through the relation-conditioned function $\theta_r(\tau)$, so it can score an unseen timestamp without learning a new lookup embedding. We use this continuous operator to resolve conflicting facts while retaining the full history in agentic memory. Relation semantics have also been used for zero-shot transfer. For example, zrLLM (Ding et al., 2024) constructs relation representations with an LLM for zero-shot TKG forecasting. In contrast, our Semantic Speed Gate maps relation text to a scalar volatility that controls rotation speed. The entity and relation embeddings and the global temporal spectrum are then learned on the target graph.

3 Methodology: ROMEM

ROMEM (Figure 2) models temporal conflict resolution as a geometric physical law within the knowledge graph embedding space, replacing discrete memory management with continuous phase rotation. The memory module is strictly *append-only*: contradictions co-exist in memory and are resolved at query time through *geometric shadowing*, where the temporally aligned fact naturally outranks obsolete ones via phase proximity. Because time is a continuous function rather than a discrete label, the system natively supports historical retrieval and zero-shot evaluation of unseen dates.

3.1 Problem Setting and Memory Design

We process a stream of textual episodes $\{d_i\}_{i=1}^N$ to answer queries q . Each episode yields relational facts via temporal Open Information Extraction (OpenIE): $f = (h, r, t)$ where $h, t \in \mathcal{E}$, $r \in \mathcal{R}$, with a valid time $t_{\text{happen}}(f)$ extracted from text and an observation time $t_{\text{obs}}(f)$ at ingestion. If no valid time is present, t_{happen} remains unknown. We maintain an append-only memory state $\mathcal{M}$ where contradictions co-exist: $m = (f, t_{\text{happen}}(f), t_{\text{obs}}(f), \text{src})$, where $\text{src} \in \{d_i\}_{i=1}^N$. We store dense embeddings for passages and entities using a text encoder $\phi(\cdot)$ and build a heterogeneous graph $G = (V, E)$ induced by facts. Extraction prompts are provided in Appendix B.

3.2 Core Insight: Why Geometry Solves Temporal Conflicts

The most straightforward approach to temporal memory is to store a timestamp with each fact and sort by recency at retrieval time. However, as discussed in §1, this metadata-based approach faces the *static-dynamic dilemma*: it treats all relations identically, unable to distinguish “born in” (static) from “president of” (dynamic). To resolve this dilemma, we employ a Temporal Knowledge Graph Embedding (TKGE) that internalises time as a continuous geometric operator rather than a discrete metadata field. ROMEM exploits the inductive power of TKG representational learning (Cai et al., 2023) to resolve temporal conflicts natively in vector space.

Our central idea is to model time as a continuous geometric rotation, aligning with evidence from cognitive neuroscience that the mammalian hippocampus encodes time as continuous geometric trajectories rather than discrete timestamps (Eichenbaum, 2014; Howard et al., 2014). Consider a “clock hand” analogy: the entity embedding for “Donald Trump” at $\tau = 2025$ is rotated to a phase angle aligned with the “President” relation (pointing to 12 o’clock). As time flows to $\tau = 2010$, the vector continuously rotates away from this alignment, reducing the retrieval score for “Donald Trump” while simultaneously aligning with “Barack Obama”. The most temporally relevant fact naturally *shadows* obsolete ones through geometric proximity, without deletion. Crucially, the Semantic Speed Gate (§3.4) controls this rotation per-relation: static facts do not rotate (and therefore are never buried by recency), while dynamic

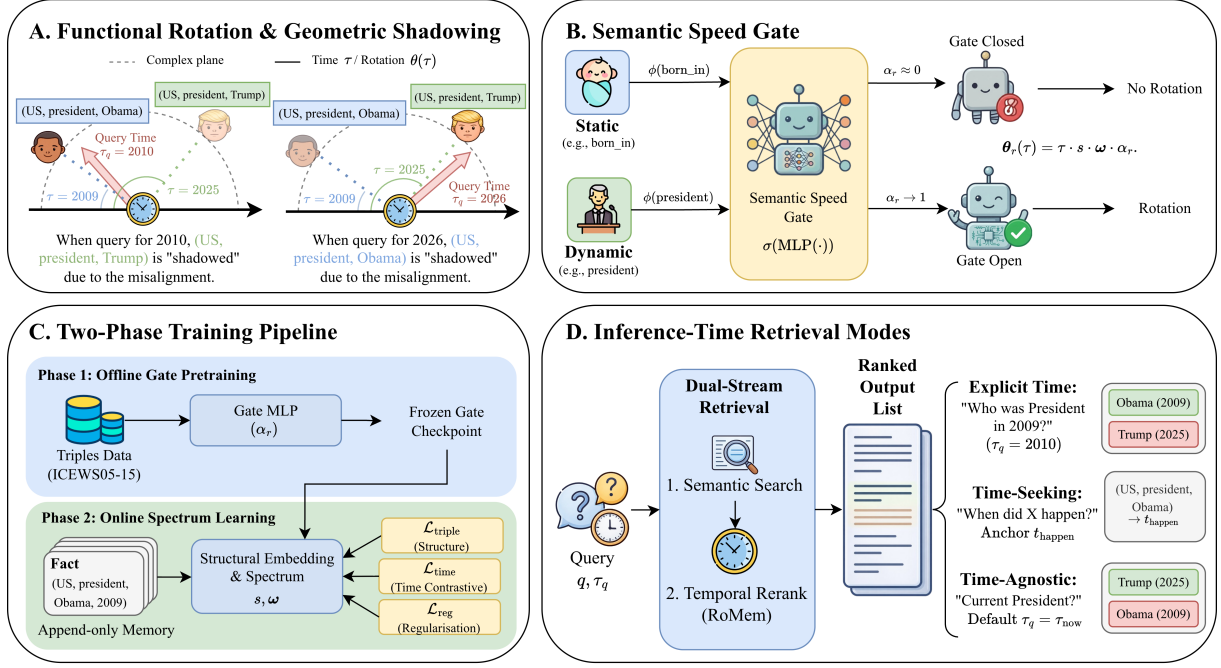


Figure 2: **Overview of the ROMEM Architecture.** The framework consists of four stages: (A) *Functional Rotation* (§3.3) applies geometric phase shifts to obsolete facts; (B) *Semantic Speed Gate* (§3.4) determines relational volatility from text embeddings; (C) *Two-Phase Training* (§3.5) pretrains the gate and learns the temporal spectrum; (D) *Inference-Time Retrieval* (§3.6) resolves contradictions via geometric shadowing.

facts rotate rapidly (resolving temporal conflicts geometrically).

Beyond the metadata-based approach, this design also overcomes two limitations of prior *temporal embedding* methods. First, **additive models** (e.g., T-TransE (Leblay and Chekol, 2018), HyTE (Dasgupta et al., 2018)) treat time as a linear bias added to structural embeddings. This suffers from *additive decoupling*: a strong structural affinity (e.g., for popular entities) can overpower the temporal penalty so that anachronistic facts are retrieved based on popularity. Our multiplicative rotation couples temporal alignment with structural affinity, helping the model demote facts that are structurally plausible but temporally mismatched. Second, **discrete rotation models** (e.g., ChronoR (Sadeghian et al., 2021), TeRo (Xu et al., 2020)) learn a separate embedding vector τ_t for every observed timestamp. Lacking a continuous functional bridge, this design leaves blind spots between observed timestamps. Our functional definition $\theta(\tau)$ instead treats time as a continuous geometric variable, allowing the model to score timestamps that were not observed during training. Appendix H.4 proves monotonic interpolation under a sufficient condition.

3.3 Functional Rotation Mechanism

We embed entities and relations in $\mathbb{R}^{2d}$, interpreted as complex vectors in $\mathbb{C}^d$. A scalar time τ acts as a rotation operator in the unitary group $U(1)^d$ via the operator $\text{Rot}(\mathbf{x}, \boldsymbol{\theta})$, which applies an element-wise phase shift $\mathbf{x} \odot e^{i\boldsymbol{\theta}}$. We define the relation-specific rotation angle as:

$$\boldsymbol{\theta}_r(\tau) = s \cdot \alpha_r \cdot \tau \cdot \boldsymbol{\omega}, \quad (1)$$

where $s \in \mathbb{R}^+$ is a global time scale parameter. We initialise s with a day-level prior ($s_0 = 1/86400$) and learn it jointly with the target graph. $\alpha_r \in (0, 1)$ is the semantic speed gate (§3.4), τ is the continuous timestamp, and $\boldsymbol{\omega} \in \mathbb{R}^{k \times d}$ is a learnable inverse-frequency tensor. Under a flattened index i , its elements are initialised as $\omega_i = b^{-i/(kd)}$, where the base b is initialised at 10,000 and learned during training.

We build upon the multi-component bilinear architecture of ChronoR (Sadeghian et al., 2021), replacing its discrete timestamp lookup with our functional time definition. Each entity has k complex components (we use $k=3$, following ChronoR), each in $\mathbb{C}^d$, with relation embeddings $\mathbf{w}_r, \hat{\mathbf{w}}_r \in \mathbb{R}^{k \times 2d}$ for forward and inverse semantics, respectively. The scoring function is:

$$\mathbf{v}_r^c(\mathbf{e}, \tau) = \text{Rot}(\mathbf{e}^c, \boldsymbol{\theta}_r^c(\tau)) \quad (2)$$

$$\tilde{\mathbf{v}}_r^c(\mathbf{e}, \tau) = \mathbf{v}_r^c(\mathbf{e}, \tau) \odot \mathbf{w}_r^c \odot \hat{\mathbf{w}}_r^c \quad (3)$$

$$s_{\text{kge}}((h, r, t) \mid \tau) = \sum_{c=1}^k \langle \tilde{\mathbf{v}}_r^c(\mathbf{e}_h, \tau), \mathbf{v}_r^c(\mathbf{e}_t, \tau) \rangle \quad (4)$$

Since unitary rotation preserves the vector modulus and only shifts the phase, an invalid timestamp rotates the fact out of alignment rather than merely penalising its magnitude. The efficient 1-vs- N retrieval reformulation and a simplified DistMult variant are detailed in Appendix A.

3.4 Semantic Speed Gate

A fundamental challenge in applying KGE to OpenIE is relational diversity: OpenIE yields thousands of surface forms (e.g., “married to”, “wedded to”, “spouse of”) for identical relations. Methods that learn a fixed parameter per string cannot generalise across linguistic variations. Moreover, relations have distinct temporal natures: “born in” is static while “visiting” is dynamic. We introduce a Semantic Speed Gate that derives rotation velocity from the relation’s text embedding $\phi(r)$: $\alpha_r = \sigma(\text{MLP}(\phi(r))) \in (0, 1)$. This achieves *zero-shot temporal transfer*: if the model learns that “married” implies stability ($\alpha \approx 0$), it automatically stabilises unseen relations like “wedded” because their embeddings lie close in semantic space.

The model does not use static/dynamic relation labels. Instead, it learns from temporal transitions. For **dynamic relations** where the tail entity changes over time (e.g., `president_of`), changed counterparts provide a high-volatility signal and drive $\alpha_r \rightarrow 1$. For **persistent slots**, unchanged counterparts provide a low-volatility signal and drive $\alpha_r \rightarrow 0$. This functions as a “temporal clutch”: static facts ($\alpha_r \rightarrow 0$) are locked in alignment, while dynamic facts ($\alpha_r \rightarrow 1$) rotate to shadow obsolete contradictions.

3.5 Two-Phase Training

A central design challenge is decoupling the semantic gate α_r from the global time spectrum (s, ω) . Joint training on a single dataset causes two failure modes: (i) sparse datasets lack sufficient competing facts to provide a learning signal for α_r , causing gate collapse; and (ii) temporal discrimination objectives (Equation (7)) treat alternative timestamps as negative samples, which incorrectly penalises the infinite validity of static relations and forces α_r away from the desired zero state. We address these

issues with a two-phase training procedure since α_r depends exclusively on relation text embeddings.

Phase 1: Offline Gate Pretraining. We construct a self-supervised dataset of temporal transitions from ICEWS05-15 (García-Durán et al., 2018). ICEWS05-15 is event-heavy, but the gate does not rely on manually labelled static relations. For each relational slot (h, r) , an unchanged counterpart provides a low-volatility observation, while a changed counterpart provides a high-volatility observation. We filter non-functional slots whose ratio of unique counterparts to observations exceeds a threshold. The gate MLP is trained with a rotation-based BCE objective:

$$\theta_i = \alpha_{r_i} \cdot \lambda \cdot \Delta t_i, \quad p_{\text{change}}(\theta_i) = 1 - e^{-\theta_i}, \quad (5)$$

$$\mathcal{L}_{\text{gate}} = \text{BCE}(y_i, p_{\text{change}}(\theta_i)) \quad (6)$$

where Δt_i is the time gap between adjacent observations, $y_i \in \{0, 1\}$ indicates entity change. After pretraining, only the MLP weights are retained.

Phase 2: Online Spectrum Learning. We load the pretrained gate and keep it frozen. We then learn the global spectrum (s, ω) and the entity and relation embeddings on the target graph using $\mathcal{L} = \mathcal{L}_{\text{triple}} + \lambda_t \mathcal{L}_{\text{time}} + \mathcal{L}_{\text{reg}}$. This phase does not require target-domain volatility labels or fine-tuning the gate, but it does require training on the target graph. The structural loss $\mathcal{L}_{\text{triple}}$ uses 1-vs-all cross-entropy, while the time-contrastive loss $\mathcal{L}_{\text{time}}$ uses a listwise objective:

$$\begin{aligned} \mathcal{L}_{\text{time}} &= - \sum_{j=0}^J p_j^* \log p_j, \\ p_j &= \text{softmax}(\mathbf{z})_j, \quad z_0 = s(f|\tau), \\ z_j &= s(f|\tilde{\tau}_j), \quad j \in \{1, \dots, J\} \end{aligned} \quad (7)$$

where p_j^* uses a Gaussian kernel to softly prefer timestamps close to the validity center. Detailed loss formulations, regularisation, and negative sampling strategies are provided in Appendix A.

3.6 Inference-Time Retrieval

Geometric Shadowing. Training aligns each fact with its observed validity time. At query time τ_q , its phase changes from the observed anchor t_{happen} by $\Delta \theta_r = s \alpha_r (\tau_q - t_{\text{happen}}) \omega$. For a current-information query ($\tau_q \approx \tau_{\text{now}}$), a recently valid fact is expected to remain close to its learned alignment, whereas an older conflicting fact is more

likely to rotate out of phase. Thus, the new fact naturally shadows the old one without explicit deletion. Similarly, setting τ_q to a past date restores the historical fact’s alignment while rotating modern facts out of focus. Appendix H.4 analyses this behaviour in a simplified pairwise setting, and Appendix I provides a scoring trace for ICEWS05-15 facts.

Dual-Stream Retrieval. We build upon HippoRAG’s (Gutiérrez et al., 2025) retrieval pipeline, which computes a semantic score S_{sem} by combining dense passage similarity with Personalised PageRank over the knowledge graph. We then apply the TKGE scoring function $S_{\text{kge}} = s_{\text{kge}}((h, r, t) \mid \tau)$ from §3.3 as a temporal re-ranker. To prevent “right time, wrong topic” boosts, we use multiplicative gating with strength $\alpha_g \geq 0$ (set to 0.3 in all agentic-memory experiments): $S_{\text{final}} = S_{\text{sem}} \cdot (1 + \alpha_g \cdot S_{\text{kge}})$, so temporal signals only amplify facts that are already semantically plausible. We use the same candidate generator as HippoRAG. As a result, ROMEM cannot recover a fact that is missing from the initial candidate pool, and the direct comparison with HippoRAG isolates the effect of temporal reranking.

Query-Time Modes. After the query time and intent have been resolved, we support three retrieval modes: (1) **Explicit Time** (τ_q present), which scores candidates at a specific timestamp to strictly enforce temporal validity; (2) **Time-Seeking** (e.g., “When did X happen?”), which evaluates each candidate against its own stored t_{happen} to verify internal validity without an external τ_q ; and (3) **Time-Agnostic**, which defaults to τ_{now} , leveraging geometric shadowing to prioritise fresher dynamic facts. The temporal extractor handles explicit dates and simple relative expressions. References such as “then” must first be resolved from the dialogue history before the resulting time constraint is passed to ROMEM.

4 Experiments

We evaluate ROMEM through three research questions: **(RQ1)** Does the transition from discrete timestamp projections to functional temporal modelling maintain or improve performance on standard TKGE benchmarks? (§4.2); **(RQ2)** Can ROMEM outperform existing agentic memory baselines on temporal reasoning tasks while maintaining robustness on non-temporal retrieval? (§4.3);

and **(RQ3)** Can ROMEM generalise to unseen domain-specific relations? (§4.4).

4.1 Experimental Setup

Datasets. We evaluate on a diverse set of benchmarks categorised by our research questions. For RQ1, we use ICEWS05-15 (García-Durán et al., 2018). For RQ2, we evaluate agentic memory across three levels of temporal complexity: (1) **Heavy Temporal:** MultiTQ (Chen et al., 2023) focuses on temporal reasoning and conflicting facts; (2) **Hybrid:** LoCoMo (Maharana et al., 2024) combines temporal updates with general memory questions; and (3) **Largely Non-Temporal:** DMR-MS (Packer et al., 2024) tests whether temporal reranking preserves performance on conversational memory with little temporal conflict. Finally, for RQ3, we use FinTMMBench (Zhu et al., 2025). Full dataset statistics are provided in Appendix C.

Metrics and Baselines. For retrieval, we report Mean Reciprocal Rank (MRR), Hits@ k , and Recall@ k . We report answer accuracy (Acc@ k) using a deterministic cascading verifier for MultiTQ and an LLM judge for DMR-MS and FinTMMBench. For LoCoMo, Appendix F reports the standard partial-match F1 in addition to retrieval recall. For TKG completion on ICEWS05-15, we compare with non-rotation baselines: DistMult (Yang et al., 2015), DE-Simple (Goel et al., 2020), TCompLex (Lacroix et al., 2020), TLT-KGE (Zhang et al., 2022), HGE (Pan et al., 2024a), and TimeGate (Shen et al., 2025), and rotation-based methods, TeRo (Xu et al., 2020), ChronoR (Sadeghian et al., 2021), RotateQVS (Chen et al., 2022), TeAST (Li et al., 2023), TCompoundE (Ying et al., 2024), and 3DG-TE (Li et al., 2025). For agentic memory benchmarks, we compare against recent graph-based agentic memory systems, including Mem0 (Chhikara et al., 2025), Zep (Rasmussen et al., 2025), LicoMemory (Huang et al., 2026) and HippoRAG (Gutiérrez et al., 2024; Gutiérrez et al., 2025), as well as a widely used non-graph method, A-Mem (Xu et al., 2025).

Detailed metric definitions, answer verification procedures, implementation configurations, and TKGE hyperparameters are provided in Appendices E, G.1, and G.2, respectively.

Table 1: Results on ICEWS05-15. Baseline results are taken from Li et al. (2025) and Shen et al. (2025). Best results are in **bold**. Green cells indicate results where ROMEM improves over its backbones (DistMult and ChronoR).

Method	MRR	Hit@1	Hit@3	Hit@10
Non-Rotation Based				
DistMult (2015)	45.6	33.7	-	69.1
DE-Simple (2020)	51.3	39.2	57.8	74.8
TComplEx (2020)	66.5	58.3	71.6	81.1
TLT-KGE (2022)	68.6	60.7	73.5	83.1
HGE (2024a)	68.8	60.8	74.0	83.5
TimeGate (2025)	69.2	61.3	74.5	83.7
Rotation Based				
TeRo (2020)	58.6	46.9	66.8	79.5
ChronoR (2021)	68.4	61.1	73.0	82.1
RotateQVS (2022)	63.3	52.9	70.9	81.3
TeAST (2023)	68.3	60.4	73.2	82.9
TCompoundE (2024)	69.2	61.2	74.3	83.7
3DG-TE (2025)	69.4	61.4	74.7	84.1
ROMEM (Ours)				
ROMEM-DistMult	62.1	54.2	66.3	77.2
ROMEM-ChronoR	72.6	66.8	75.9	83.7

We use $k = 3$ complex components for ChronoR and ROMEM-ChronoR, following Sadeghian et al. (2021).

4.2 Verification of Functional Temporal Modelling (RQ1)

We first test the complete model on temporal knowledge graph completion. This experiment shows whether the functional time operator, Semantic Speed Gate, and time-contrastive loss preserve the link-prediction performance of the DistMult and ChronoR backbones.

As shown in Table 1, ROMEM-ChronoR achieves an MRR of 72.6, outperforming the vanilla ChronoR (68.4) on ICEWS05-15. Similarly, our DistMult-based variant, ROMEM-DistMult, shows a substantial performance improvement (62.1 MRR) compared to the static DistMult baseline (45.6 MRR). Among the listed methods, ROMEM-ChronoR obtains the best MRR (72.6), Hit@1 (66.8), and Hit@3 (75.9); its Hit@10 is 83.7 compared with 84.1 for 3DG-TE.

Together, these results show that the complete model improves filtered link prediction over both backbones. We isolate the three components on MultiTQ in Table 3.

4.3 Performance on Episodic and Temporal Memory Tasks (RQ2)

To answer RQ2, we evaluate whether ROMEM can resolve complex temporal conflicts in agentic memory without degrading foundational, non-temporal

retrieval capabilities. We evaluate all agentic memory methods under two implementation configurations: a closed-source API setup (GPT-5-mini with text-embedding-3-small) and an open-source setup (LLaMA-3.1-70B with BGE-M3) to serve as a robustness check. Because ROMEM adds only temporal reranking to HippoRAG, the direct comparison between the two systems isolates the effect of our method. Comparisons with Mem0, Zep, A-Mem, and LicoMemory provide broader system-level context.

Temporal Reasoning (MultiTQ). MultiTQ directly tests temporal reasoning over changing and conflicting facts. With GPT-5-mini for graph construction, adding ROMEM to HippoRAG improves MRR from 0.203 to 0.337 and Acc@5 from 0.112 to 0.366 (Table 2a). The open-source configuration shows similar gains: MRR increases from 0.236 to 0.316 and Acc@5 from 0.122 to 0.312. The improvement therefore holds across both graph-construction and embedding configurations.

Table 3 serves as an ablation on MultiTQ. Removing the Semantic Speed Gate lowers MRR from 0.3371 to 0.1044, showing that one shared rotation speed is unsuitable for both persistent and evolving relations. Replacing functional time with learned timestamp embeddings lowers MRR further to 0.0452. Removing the time-contrastive loss has a smaller effect: MRR decreases from 0.3371 to 0.3254 and Acc@5 from 0.366 to 0.340. Functional time and semantic gating therefore provide the main gains, while the time-contrastive loss provides an additional improvement.

Hybrid Retrieval (LoCoMo). LoCoMo contains single-hop, multi-hop, open-domain, and temporal questions. ROMEM improves the average Recall@10 of HippoRAG from 0.815 to 0.857 with GPT-5-mini and from 0.830 to 0.838 with the open-source configuration (Table 2b). In the GPT-5-mini configuration, the largest gain is on temporal questions, where Recall@10 increases from 0.645 to 0.726. These results measure retrieval quality. Appendix F reports answer-level partial-match F1, where ROMEM matches HippoRAG at 0.457 in the GPT-5-mini configuration and improves by 0.001 in the open-source configuration. Thus, the retrieval gains do not lead to a comparable improvement in answer F1 with the current reader.

Preservation of General Memory (DMR-MSR). DMR-MSR tests conversational memory in a set-

²A-Mem is excluded from MultiTQ because it lacks native support for massive structured triple ingestion (~11K).

Table 2: Comprehensive evaluation of ROMEM. (a) **MultiTQ**: temporal reasoning. (b) **LoCoMo**: hybrid retrieval (Recall@10; answer F1 is in Appendix F). (c) **DMR-MSC**: largely non-temporal memory. (d) **FinTMMBench**: transfer to an unseen domain. Implementation denotes the graph-construction LLM and embedding model. Best results are in **bold**. **Green cells** mark the direct comparison with HippoRAG under the same graph-construction and semantic-retrieval pipeline.

(a) MultiTQ (RQ2, Heavy Temporal)						(b) LoCoMo (RQ2, Hybrid Tasks)					
Method	MRR	Hit@3	Hit@10	Acc@5	Acc@10	Method	Single Hop	Multi Hop	Open Domain	Temporal Reason	Average
GPT-5-mini + text-embedding-3-small						GPT-5-mini + text-embedding-3-small					
Zep	0.192	0.208	0.310	0.110	0.118	Zep	0.557	0.861	0.831	0.553	0.770
Mem0	0.174	0.190	0.282	0.122	0.122	Mem0	0.740	0.832	0.883	0.690	0.834
A-Mem ²	—	—	—	—	—	A-Mem	0.740	0.846	0.860	0.691	0.825
LicoMem.	0.149	0.160	0.292	0.114	0.128	LicoMem.	0.727	0.856	0.848	0.661	0.816
HippoRAG	0.203	0.232	0.348	0.112	0.102	HippoRAG	0.711	0.837	0.862	0.645	0.815
ROMEM	0.337	0.384	0.502	0.366	0.392	ROMEM	0.768	0.850	0.904	0.726	0.857
LLaMA-3.1-70B + BGE-M3						LLaMA-3.1-70B + BGE-M3					
Zep	0.217	0.252	0.370	0.098	0.116	Zep	0.557	0.861	0.831	0.553	0.770
Mem0	0.228	0.264	0.356	0.120	0.114	Mem0	0.746	0.860	0.875	0.737	0.839
A-Mem ²	—	—	—	—	—	A-Mem	0.658	0.776	0.777	0.702	0.750
LicoMem.	0.159	0.182	0.304	0.114	0.120	LicoMem.	0.605	0.768	0.725	0.584	0.703
HippoRAG	0.236	0.266	0.354	0.122	0.116	HippoRAG	0.717	0.852	0.870	0.732	0.830
ROMEM	0.316	0.342	0.440	0.312	0.338	ROMEM	0.759	0.824	0.879	0.759	0.838
(c) DMR-MSC (RQ2, Largely Non-Temporal)						(d) FinTMMBench (RQ3)					
Method	MRR	Hit@1	Hit@3	Acc@5	Acc@10	Method	MRR	R@5	R@10	Acc@5	Acc@10
GPT-5-mini + text-embedding-3-small						GPT-5-mini + text-embedding-3-small					
Zep	0.170	0.110	0.180	0.302	0.376	Zep	0.703	0.644	0.759	0.480	0.520
Mem0	0.847	0.766	0.926	0.858	0.848	Mem0	0.691	0.645	0.768	0.550	0.610
A-Mem	0.825	0.732	0.912	0.848	0.856	A-Mem	0.716	0.647	0.796	0.540	0.640
LicoMem.	0.326	0.224	0.372	0.670	0.728	LicoMem.	0.488	0.480	0.609	0.480	0.590
HippoRAG	0.848	0.768	0.926	0.852	0.850	HippoRAG	0.690	0.645	0.768	0.550	0.650
ROMEM	0.856	0.774	0.934	0.862	0.858	ROMEM	0.728	0.673	0.779	0.580	0.650
LLaMA-3.1-70B + BGE-M3						LLaMA-3.1-70B + BGE-M3					
Zep	0.333	0.232	0.394	0.384	0.428	Zep	0.515	0.510	0.591	0.430	0.450
Mem0	0.821	0.714	0.926	0.758	0.770	Mem0	0.718	0.647	0.765	0.570	0.610
A-Mem	0.823	0.732	0.902	0.728	0.738	A-Mem	0.650	0.631	0.742	0.520	0.590
LicoMem.	0.202	0.138	0.228	0.258	0.338	LicoMem.	0.554	0.559	0.662	0.460	0.520
HippoRAG	0.818	0.718	0.912	0.768	0.776	HippoRAG	0.724	0.680	0.766	0.610	0.610
ROMEM	0.847	0.760	0.930	0.774	0.786	ROMEM	0.726	0.707	0.793	0.620	0.650

Table 3: Single-factor ablation on the fixed MultiTQ subset (selected metrics; full results in Appendix F.1). “No gate” uses one speed for all relations; “discrete time” uses learned timestamp lookups.

Variant	MRR	H@3	Acc@5	Acc@10
Full ROMEM	0.3371	0.384	0.366	0.392
w/o time contrastive	0.3254	0.368	0.340	0.390
w/o semantic gate	0.1044	0.116	0.100	0.108
Discrete time	0.0452	0.046	0.040	0.040

ting with little explicit temporal conflict. As shown in Table 2c, ROMEM achieves an MRR of 0.856 and an LLM@5 Accuracy of 0.862 under the GPT-5-mini setup, slightly improving upon the baseline

HippoRAG performance of 0.848 and 0.852, respectively. We observe similar gains in the LLaMA-3.1-70B implementation. These results show that the temporal reranker preserves performance on this largely non-temporal benchmark. The ranking of the methods is unchanged when we use the stricter conflict-aware judge described in Appendix F.

Conclusion for RQ2. Across both configurations, ROMEM improves temporal retrieval over the same HippoRAG candidate pool and preserves performance when temporal conflicts are largely absent.

4.4 Domain Generalisation (RQ3)

To answer RQ3, we evaluate ROMEM on FinTMMBench to test zero-shot generalisation in high-volatility financial contexts (Li and Ma, 2025; Li et al., 2026a). As shown in Table 2d, ROMEM achieves 0.728 MRR and 0.580 Acc@5 with GPT-5-mini, compared with 0.690 and 0.550 for HippoRAG. In the open-source configuration, ROMEM improves retrieval recall, while MRR increases slightly from 0.724 to 0.726. The frozen gate therefore transfers from geopolitical event relations to the unseen financial relations evaluated here. ROMEM also remains the top-ranked method under the stricter conflict-aware judge (Appendix F).

5 Conclusion

We identified two limitations in how graph-based memory systems handle time. First, discrete timestamp metadata applies the same policy to static and dynamic relations, so recency sorting can bury static facts. Second, existing conflict-management strategies either overwrite history or require an LLM at every ingestion step, which becomes costly at scale. ROMEM addresses these limitations by representing time as a continuous phase rotation in the KG embedding space. Its pretrained Semantic Speed Gate maps relation text embeddings to rotation speeds, keeping persistent relations aligned while rotating outdated facts out of phase. Because old facts remain stored, the memory is append-only and can answer both current and historical queries. Empirically, ROMEM achieves state-of-the-art TKGE results and large gains over the corresponding HippoRAG pipeline on temporal memory, while preserving performance on largely non-temporal memory and transferring the frozen gate to an unseen financial domain.

Limitations

Our work has three main limitations. First, the Semantic Speed Gate is pretrained on ICEWS05-15, a political events dataset. While it generalises zero-shot to unseen domains in our experiments, incorporating additional domain-specific temporal data during pretraining, which is efficient due to the lightweight MLP architecture, could further improve calibration for specialised settings.

Second, our evaluation is conducted exclusively on English-language benchmarks. The effectiveness of the text-embedding-based gate on multilin-

gual or low-resource settings remains to be validated.

Additionally, the current system operates on textual relational facts and does not handle multimodal inputs such as images, or audio, which are increasingly relevant in real-world agentic applications. Extending ROMEM to multimodal temporal reasoning is a promising direction for future work.

Acknowledgements

AI assistants were used only to polish author-written text and to help debug code. All suggestions were reviewed and validated by the authors, who take full responsibility for the paper and code.

References

- Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. 2023. [Temporal knowledge graph completion: a survey](#). In *Proceedings of the Thirty-Second International Joint Conference on Artificial Intelligence, IJCAI '23*.
- Jianlyu Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024. [M3-embedding: Multi-linguality, multi-functionality, multi-granularity text embeddings through self-knowledge distillation](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 2318–2335, Bangkok, Thailand. Association for Computational Linguistics.
- Kai Chen, Ye Wang, Yitong Li, and Aiping Li. 2022. [RotateQVS: Representing temporal information as rotations in quaternion vector space for temporal knowledge graph completion](#). In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5843–5857, Dublin, Ireland. Association for Computational Linguistics.
- Ziyang Chen, Jinzhi Liao, and Xiang Zhao. 2023. [Multi-granularity temporal question answering over knowledge graphs](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11378–11392, Toronto, Canada. Association for Computational Linguistics.
- Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. [Mem0: Building production-ready ai agents with scalable long-term memory](#).
- Shib Sankar Dasgupta, Swayambhu Nath Ray, and Partha Talukdar. 2018. [HyTE: Hyperplane-based temporally aware knowledge graph embedding](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2001–2011, Brussels, Belgium. Association for Computational Linguistics.

- Zifeng Ding, Heling Cai, Jingpei Wu, Yunpu Ma, Ruotong Liao, Bo Xiong, and Volker Tresp. 2024. [zrLLM: Zero-shot relational learning on temporal knowledge graphs with large language models](#). In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1877–1895, Mexico City, Mexico. Association for Computational Linguistics.
- Howard Eichenbaum. 2014. [Time cells in the hippocampus: a new dimension for mapping memories](#). *Nature Reviews Neuroscience*, 15(11):732–744.
- Alberto García-Durán, Sebastijan Dumančić, and Mathias Niepert. 2018. [Learning sequence encoders for temporal knowledge graph completion](#). In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4816–4821, Brussels, Belgium. Association for Computational Linguistics.
- Rishab Goel, Seyed Mehran Kazemi, Marcus Brubaker, and Pascal Poupart. 2020. [Diachronic embedding for temporal knowledge graph completion](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):3988–3995.
- Bernal Jimenez Gutierrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. [Hipporag: Neurobiologically inspired long-term memory for large language models](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- Bernal Jiménez Gutiérrez, Yiheng Shu, Weijian Qi, Sizhe Zhou, and Yu Su. 2025. [From RAG to memory: Non-parametric continual learning for large language models](#). In *Forty-second International Conference on Machine Learning*.
- Marc W Howard, Christopher J. MacDonald, Zoran Tiganj, Karthik H. Shankar, Qian Du, Michael E. Hasselmo, and Howard Eichenbaum. 2014. A unified mathematical framework for coding time, space, and sequences in the hippocampal region. *The Journal of Neuroscience*, 34:4692 – 4707.
- Sen Hu, Yuxiang Wei, Jiaxin Ran, Zhiyuan Yao, Xueran Han, Huacan Wang, Ronghao Chen, and Lei Zou. 2026a. [Does memory need graphs? a unified framework and empirical analysis for long-term dialog memory](#). *Preprint*, arXiv:2601.01280.
- 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, and 28 others. 2026b. [Memory in the age of ai agents](#). *Preprint*, arXiv:2512.13564.
- Zhengjun Huang, Zhoujin Tian, Qintian Guo, Fangyuan Zhang, Yingli Zhou, Di Jiang, Zeyang Xie, and Xiaofang Zhou. 2026. [Licomemory: Lightweight and cognitive agentic memory for efficient long-term reasoning](#). *Preprint*, arXiv:2511.01448.
- Dongming Jiang, Yi Li, Guanpeng Li, and Bingzhe Li. 2026. [Magma: A multi-graph based agentic memory architecture for ai agents](#). *Preprint*, arXiv:2601.03236.
- Timothée Lacroix, Guillaume Obozinski, and Nicolas Usunier. 2020. [Tensor decompositions for temporal knowledge base completion](#). In *International Conference on Learning Representations*.
- Julien Leblay and Melisachew Wudage Chekol. 2018. [Deriving validity time in knowledge graph](#). In *Companion Proceedings of the The Web Conference 2018, WWW ’18*, pages 1771–1776, Republic and Canton of Geneva, CHE. International World Wide Web Conferences Steering Committee.
- Jiang Li, Xiangdong Su, and Guanglai Gao. 2023. [TeAST: Temporal knowledge graph embedding via archimedean spiral timeline](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15460–15474, Toronto, Canada. Association for Computational Linguistics.
- Jiang Li, Xiangdong Su, and Guanglai Gao. 2025. [Leveraging 3D Gaussian for temporal knowledge graph embedding](#). In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 7852–7865, Suzhou, China. Association for Computational Linguistics.
- Weixian Waylon Li, Hyeonjun Kim, Mihai Cucuringu, and Tiejun Ma. 2026a. [Can llm-based financial investing strategies outperform the market in long run?](#) *Preprint*, arXiv:2505.07078.
- Weixian Waylon Li and Tiejun Ma. 2025. [Learn to rank risky investors: A case study of predicting retail traders’ behaviour and profitability](#). *ACM Trans. Inf. Syst.*, 44(1).
- Weixian Waylon Li, Yuchen Niu, Yongxin Yang, Keshuang Li, Tiejun Ma, and Shay B Cohen. 2026b. [Spectral attention steering for prompt highlighting](#). In *The Fourteenth International Conference on Learning Representations*.
- Xuechen Liang, Yangfan He, Yinghui Xia, Xinyuan Song, Jianhui Wang, Meiling Tao, Li Sun, Xinhang Yuan, Jiayi Su, Keqin Li, Jiaqi Chen, Jinsong Yang, Siyuan Chen, and Tianyu Shi. 2024. [Self-evolving agents with reflective and memory-augmented abilities](#).
- Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024. [Lost in the middle: How language models use long contexts](#). *Transactions of the Association for Computational Linguistics*, 12:157–173.

- Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. [Evaluating very long-term conversational memory of LLM agents](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, Bangkok, Thailand. Association for Computational Linguistics.
- Meta AI. 2024. [Introducing meta llama 3: The most capable openly available llm to date](#).
- Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. [Memgpt: Towards llms as operating systems](#). *Preprint*, arXiv:2310.08560.
- Jiaxin Pan, Mojtaba Nayyeri, Yinan Li, and Steffen Staab. 2024a. [Hge: Embedding temporal knowledge graphs in a product space of heterogeneous geometric subspaces](#). *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(8):8913–8920.
- Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiaapu Wang, and Xindong Wu. 2024b. [Unifying large language models and knowledge graphs: A roadmap](#). *IEEE Transactions on Knowledge and Data Engineering*, 36(7):3580–3599.
- Jinyoung Park, Minseok Joo, Joo-Kyung Kim, and Hyunwoo J. Kim. 2024. [Generative subgraph retrieval for knowledge graph-grounded dialog generation](#). In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 21167–21182, Miami, Florida, USA. Association for Computational Linguistics.
- Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. [Zep: A temporal knowledge graph architecture for agent memory](#).
- Ali Sadeghian, Mohammadreza Armandpour, Anthony Colas, and Daisy Zhe Wang. 2021. [Chronor: Rotation based temporal knowledge graph embedding](#). In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 6471–6479.
- Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. 2025. [MemInsight: Autonomous memory augmentation for LLM agents](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33124–33140, Suzhou, China. Association for Computational Linguistics.
- Jinqing Shen, Chengjin Xu, Yingqi Liu, Xuhui Jiang, Jiaming Li, Zhenxin Huang, Jens Lehmann, and Xuesong Chen. 2025. [Learning temporal knowledge graphs via time-sensitive graph attention](#). *IEEE Access*, 13:178517–178526.
- Zhiqing Sun, Zhi-Hong Deng, Jian-Yun Nie, and Jian Tang. 2019. [Rotate: Knowledge graph embedding by relational rotation in complex space](#). In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.
- Xingyu Tan, Xiaoyang Wang, Qing Liu, Xiwei Xu, Xin Yuan, Liming Zhu, and Wenjie Zhang. 2026. [Memotime: Memory-augmented temporal knowledge graph enhanced large language model reasoning](#). *Preprint*, arXiv:2510.13614.
- Bohan Tang and Yiwen Guo. 2026. [Memoryforge: Synthesize lifelong memory for human-like llm agents](#). *Preprint*, arXiv:2608.00007.
- Chengjin Xu, Mojtaba Nayyeri, Fouad Alkhoury, Hamed Shariat Yazdi, and Jens Lehmann. 2020. [TeRo: A time-aware knowledge graph embedding via temporal rotation](#). In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1583–1593, Barcelona, Spain (Online). International Committee on Computational Linguistics.
- Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. [A-mem: Agentic memory for llm agents](#).
- Ruiyi Yan, Zhuoyuan Mao, and Yiwen Guo. 2026. [Memdefrag: Latent memory defragmentation for large language models](#). *Preprint*, arXiv:2607.05969.
- Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Kristian Kersting, Jeff Z. Pan, Hinrich Schütze, Volker Tresp, and Yunpu Ma. 2025. [Memory-r1: Enhancing large language model agents to manage and utilize memories via reinforcement learning](#).
- Bishan Yang, Wen-tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. 2015. [Embedding entities and relations for learning and inference in knowledge bases](#). In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.
- Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh R N, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, Ran Xu, Phil L Mui, Huan Wang, Caiming Xiong, and Silvio Savarese. 2024. [Retroformer: Retrospective large language agents with policy gradient optimization](#). In *The Twelfth International Conference on Learning Representations*.
- Rui Ye, Zhongwang Zhang, Kuan Li, Huifeng Yin, Zhengwei Tao, Yida Zhao, Liangcai Su, Liwen Zhang, Zile Qiao, Xinyu Wang, Pengjun Xie, Fei Huang, Siheng Chen, Jingren Zhou, and Yong Jiang. 2025. [Agentfold: Long-horizon web agents with proactive context management](#).
- Rui Ying, Mengting Hu, Jianfeng Wu, Yalan Xie, Xiaoyi Liu, Zhunheng Wang, Ming Jiang, Hang Gao, Linlin Zhang, and Renhong Cheng. 2024. [Simple but effective compound geometric operations for temporal knowledge graph completion](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11074–11086, Bangkok, Thailand. Association for Computational Linguistics.

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. 2025. [Memagent: Reshaping long-context llm with multi-conv rl-based memory agent](#).

Fuwei Zhang, Zhao Zhang, Xiang Ao, Fuzhen Zhuang, Yongjun Xu, and Qing He. 2022. [Along the time: Timeline-traced embedding for temporal knowledge graph completion](#). In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management, CIKM '22*, pages 2529–2538, New York, NY, USA. Association for Computing Machinery.

Kai Zhang, Xiangchao Chen, Bo Liu, Tianci Xue, Zeyi Liao, Zhihan Liu, Xiyao Wang, Yuting Ning, Zhaorun Chen, Xiaohan Fu, Jian Xie, Yuxuan Sun, Boyu Gou, Qi Qi, Zihang Meng, Jianwei Yang, Ning Zhang, Xian Li, Ashish Shah, and 11 others. 2025a. [Agent learning via early experience](#). *Preprint*, arXiv:2510.08558.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2025b. [A survey on the memory mechanism of large language model-based agents](#). *ACM Trans. Inf. Syst.*, 43(6).

Huichi Zhou, Yihang Chen, Siyuan Guo, Xue Yan, Kin Hei Lee, Zihan Wang, Ka Yiu Lee, Guchun Zhang, Kun Shao, Linyi Yang, and Jun Wang. 2025a. [Memento: Fine-tuning llm agents without fine-tuning llms](#).

Zijian Zhou, Ao Qu, Zhaoxuan Wu, Sunghwan Kim, Alok Prakash, Daniela Rus, Jinhua Zhao, Bryan Kian Hsiang Low, and Paul Pu Liang. 2025b. [Mem1: Learning to synergize memory and reasoning for efficient long-horizon agents](#).

Fengbin Zhu, Junfeng Li, Liangming Pan, Wenjie Wang, Fuli Feng, Chao Wang, Huanbo Luan, and Tat-Seng Chua. 2025. [Towards temporal-aware multi-modal retrieval augmented generation in finance](#). In *Proceedings of the 33rd ACM International Conference on Multimedia, MM '25*, pages 6289–6297, New York, NY, USA. Association for Computing Machinery.

A Scoring, Training, and Gate Formulations

A.1 Scoring Variants

Efficient 1-vs- N Retrieval. In practice, the trace-diagonal sum across k components reduces to a flat element-wise product and summation over the $k \times 2d$ real dimensions. An important consequence of this structure is the *unrotation trick*: to score a query $(h, r, ?)$ against all N entities simultaneously, we form $\mathbf{q} = \text{Rot}(\mathbf{e}_h, \boldsymbol{\theta}) \odot \mathbf{w}_r \odot \hat{\mathbf{w}}_r$ and then *unrotate* by $-\boldsymbol{\theta}$, yielding a vector in the same

space as the raw entity embeddings. A single matrix multiplication $\text{Rot}(\mathbf{q}, -\boldsymbol{\theta}) \cdot \mathbf{E}^\top$ then produces scores for all entities without materialising N separate rotations, enabling efficient 1-vs-all training. A formal proof with complexity analysis is provided in §H.

DistMult Variant. As a simplified variant, setting $k=1$ and removing the inverse relation table recovers a time-conditioned DistMult (Yang et al., 2015) backbone. This variant uses self-adversarial negative sampling instead of the 1-vs-all cross-entropy loss: $\mathcal{L}_{\text{triple}} = -\log \sigma(s^+) - \sum_k w_k \log \sigma(-s_k^-)$, where $w_k = \text{softmax}(s_k^-/T)$ are self-adversarial weights (Sun et al., 2019). For regularisation, we use global L3 on the full embedding tables instead of per-batch N3.

A.2 Training Objectives

Structural Triple Loss. With the ChronoR backbone, we use a 1-vs-all cross-entropy loss exploiting the unrotation trick (§A.1). The unrotated query $\mathbf{q}_t(\tau) = \text{Rot}(\text{Rot}(\mathbf{e}_h, \boldsymbol{\theta}_r(\tau)) \odot \mathbf{w}_r \odot \hat{\mathbf{w}}_r, -\boldsymbol{\theta}_r(\tau))$ lives in the same space as the raw entity embeddings, enabling a single matrix multiplication $\mathbf{q}_t(\tau) \cdot \mathbf{E}^\top$ to score all entities. Crucially, $\mathbf{q}_t(\tau)$ remains τ -dependent because the relation weights break the symmetry between the forward and inverse rotations; consequently, the same (h, r) pair produces distinct query vectors at different timestamps, allowing the structural loss alone to learn temporal discrimination:

$$\mathcal{L}_{\text{triple}} = \frac{1}{2} (\text{CE}_{\text{tail}}(\mathbf{q}_t(\tau) \cdot \mathbf{E}^\top, t) + \text{CE}_{\text{head}}(\mathbf{q}_h(\tau) \cdot \mathbf{E}^\top, h)) \quad (8)$$

where CE is the standard cross-entropy over the full entity vocabulary and $\mathbf{q}_h(\tau)$ is the symmetric head-prediction query. This loss is computed with the gate α_r detached, so gradients update only the entity, relation, and time parameters.

Conflict-Aware Negative Sampling. We enhance both training variants with *conflict-aware negative sampling*: rather than sampling random entities, we prioritise sampling competing tails from the same (h, r) group when available. This forces the model to discriminate between mutually exclusive facts (e.g., Obama born in *Hawaii* vs. *Kenya*) rather than easy negatives.

Regularisation. We apply backbone-specific embedding regularisation. For ChronoR, we use per-batch N3 regularisation (fourth-power penalty):

$\mathcal{L}_{\text{reg}} = \lambda_r (\|\mathbf{e}_h\|_4^4 + \|\mathbf{w}_r\|_4^4 + \|\hat{\mathbf{w}}_r\|_4^4 + \|\mathbf{e}_t\|_4^4) / B$, where B is the batch size. For the DistMult variant, we use global L3 regularisation on the full embedding tables. Additionally, a gate regularisation term $\mathcal{L}_{\text{gate_reg}} = \bar{\alpha}_r^{\text{non-comp}}$ softly encourages $\alpha_r \rightarrow 0$ for non-competing relation slots, reinforcing the temporal clutch’s static behaviour where no temporal discrimination is needed.

Time-Contrastive Loss. The listwise time-contrastive loss $\mathcal{L}_{\text{time}}$ (Equation 7 in §3.5) uses a Gaussian target kernel whose width σ follows a cosine curriculum from $\sigma_{\text{start}} = 0.5$ yr to $\sigma_{\text{end}} = 0.02$ yr over the configured decay epochs (Table 9), progressively sharpening temporal discrimination. Negative times are sampled preferentially from the same (h, r) slot history when available, with jitter (± 0.02 years) and one forced far negative (± 365 days). A minimum-gap curriculum decays from 90 to 3 days over 60 epochs to avoid trivially easy negatives early in training.

A.3 Semantic Speed Gate Pretraining

The semantic speed gate $\alpha_r = \text{MLP}(\phi(r))$ is pre-trained on self-supervised transition observations mined from ICEWS05-15 before online TKGE training begins.

Stage 1: Transition Mining. We group all temporal triples by (h, r) slot and identify consecutive temporal observations. For each pair of adjacent observations at times t_i and t_{i+1} , we record a binary changed label indicating whether the tail entity changed. Filtering criteria: minimum 3 observations per slot, maximum functional ratio 0.5, and up to 256 pairs per slot.

Stage 2: Gate Training. The gate MLP is trained with the rotation-based BCE objective defined in Equation 5 (§3.5), with p_{change} clamped to avoid numerical overflow. Training uses 100 epochs with learning rate 5×10^{-4} , embedding batch size 64, and class-weighted loss with auto-computed weights. The pretrained checkpoint is loaded and frozen at the start of online TKGE training.

B Prompts

B.1 Answer Generation

Answer LLM. All downstream answer generation uses GPT-5.2 with temperature=0 and JSON response format $\{"answer": "<text>"\}$. The answer LLM is always routed to the OpenAI API, in-

dependent of whether the graph construction LLM is hosted locally via vLLM.

System prompts. The system prompt varies by benchmark:

- **MultiTQ, DMR-MSc:** “You answer questions using only provided facts.”
- **LoCoMo:** “You answer using only the provided context.”
- **FinTMMBench:** “You are a financial analyst assistant. Use only the provided financial data to answer the question. Be concise and precise.”

Box I and Box II show the user prompt templates for MultiTQ and FinTMMBench, respectively.

Box I: Answer Generation Prompt (MultiTQ / DMR-MSc)

Question: {question}

Supporting facts:
{bullet_list_of_retrieved_documents}

Provide a concise answer grounded in the facts.
Respond with JSON $\{"answer": "<text>"\}$.

Box II: Answer Generation Prompt (FinTMMBench)

Question: {question}

Financial data:
{bullet_list_of_retrieved_documents}

Provide a short answer grounded in the data.
Respond with JSON $\{"answer": "<text>"\}$.

LoCoMo answer prompts. For LoCoMo, we follow the original benchmark protocol (Maharana et al., 2024):

- **Categories 1-4** (open-ended): “Based on the above context, write an answer in the form of a short phrase for the following question. Answer with exact words from the context whenever possible. Question: {q} Short answer:”
- **Category 5** (adversarial): “Based on the above context, answer the following question. Question: {q} Short answer:”

- **Multiple choice:** "Based on the above context, choose the best answer from the options below. Respond with the exact choice text. Question: {q} Choices: {choices} Answer:"

All LoCoMo responses are appended with: Respond with JSON {"answer": "<text>"}

B.2 Information Extraction

ROMEM uses three LLM-based extraction stages during knowledge graph construction: named entity recognition (Appendix B.2.1), temporal triple extraction (Appendix B.2.2), and query-time entity and time extraction (Appendix B.2.3, Appendix B.2.4).

B.2.1 Named Entity Recognition (NER)

Box III shows the one-shot NER prompt used to extract entities from each passage during graph construction.

Box III: Named Entity Recognition Prompt (One-Shot)

System:

Your task is to extract named entities from the given paragraph. Respond with a JSON list of entities.

Rules:

- Only include real-world entities, people, organizations, locations, products, and named events.
- Do not include time expressions, dates, durations, or clock times (these are handled separately).
- Do not include numbers that are only quantities or durations (e.g., "5 years").

One-shot example:

User: "Radio City is India's first private FM radio station and was started on 3 July 2001\ldots"

Assistant: {"named_entities": ["Radio City", "India", "Hindi", "English", "PlanetRadiocity.com"]}

B.2.2 Temporal Triple Extraction (OpenIE)

Box IV shows the system prompt for NER-conditioned triple extraction with temporal metadata. This prompt is paired with three few-shot examples covering standard extraction, relative time resolution, and duration inference.

Box IV: Temporal Triple Extraction System Prompt

Your task is to construct an RDF graph from the given passages and named entity lists. Respond with JSON where each triple entry also carries timing metadata.

Requirements:

- Represent each triple as an object with fields: head, relation, tail, text_time, observed_time.
- text_time must be a normalized date string in YYYY-MM-DD, or "" if no time is mentioned.
- If only a month or year is mentioned, use the first day of that month or year.
- If the year is missing, use the year from observed_time.
- Resolve relative expressions using the closest explicit date in the passage. If none exists, use observed_time.
- observed_time is always set to the provided observed_time string.
- Do not infer a time if the passage does not contain a temporal expression.
- Do NOT use time expressions as head or tail entities; keep time only in text_time.
- Each triple should contain at least one, but preferably two, of the named entities in the list.
- Resolve pronouns to their specific names.
- Extract all factual relations, including actions, states, plans, and events.
- Do not omit facts; prioritize completeness over brevity.
- Duration handling: if the passage states a duration such as "for 5 years" and an explicit reference date is present, infer the start date and put it in text_time.

Output format:

```
{
  "triples": [
    {
      "head": "X",
      "relation": "Y",
      "tail": "Z",
      "text_time": "YYYY-MM-DD",
      "observed_time": "2024-01-01T00:00:00Z"
    }
  ]
}
```

B.2.3 Query-Time Entity Extraction

At query time, we extract named entities from the question to initialise graph traversal.

Box V: Query NER Prompt (One-Shot)

System:

"You're a very effective entity extraction system."

User:

Please extract all named entities that are important for solving the questions below. Place the named entities in json format. Question: Which magazine was started first Arthur's Magazine or First for Women?

Assistant:

```
{"named_entities": ["First for Women", "Arthur's Magazine"]}
```

B.2.4 Time Extraction

At query time, we extract temporal constraints and ordering intent from the question.

Box VI: Time Extraction Prompt**System:**

"You extract the time constraint from a query. Return a single line only."

User:

Given the query, output the time constraint, temporal ordering intent, and whether the query asks for a time.

Return exactly one line in this format:
time=YYYY-MM-DD; ordering=earliest|latest|none;
time_request=yes|no

Rules:

- If a time constraint exists, return it as YYYY-MM-DD.
- If the query only specifies a month or year, use the first day of that month or year.
- If the query omits the year, use the year from the reference date.
- Resolve relative expressions using the reference date.
- If no time constraint exists, return time=NONE.
- ordering=earliest for queries like "earliest/first/oldest".
- ordering=latest for queries like "latest/most recent/last time".
- ordering=none otherwise.
- time_request=yes if the query asks for a time as the answer (e.g., when/what year/which year/what date).
- time_request=no if the query only uses time as a constraint or does not ask for time.

Reference date (UTC): {reference_date}

Query: {query}

C Benchmark Datasets

Table 4 summarises the key statistics for each dataset. Detailed descriptions follow.

ICEWS05-15 (García-Durán et al., 2018). The Integrated Crisis Early Warning System (ICEWS) dataset contains geopolitical event triples spanning from 2005 to 2015. Each triple takes the form (head, relation, tail, date) with dates in YYYY-MM-DD format. The standard split comprises

368,962 training, 46,275 validation, and 46,092 test triples. We use this dataset exclusively for RQ1 to verify that our functional temporal modelling preserves standard TKGE accuracy.

MultiTQ (Chen et al., 2023). The full benchmark builds on the ICEWS05-15 KG, which contains 461,329 temporal quadruples at 4,017 timestamps. Its test set contains 54,584 questions at day, month, and year resolution. Our end-to-end agentic-memory evaluation uses a fixed set of questions: the first 500 entries in the released test.json (IDs 3000000-3000499). We selected this set before running any method and did not filter questions based on their results. We process only the corresponding time snapshots, yielding 11,074 facts ingested incrementally per snapshot. Evaluating all 54,584 questions across ten method/configuration combinations with both top-5 and top-10 contexts would require at least 1,091,680 answer-generation calls, excluding framework-specific graph-construction calls. We therefore use the fixed 500-question set for the end-to-end evaluation. As shown in Table 5, the question- and answer-type distributions differ from those of the full test set by at most 2.4 percentage points.

LoCoMo (Maharana et al., 2024). Long-Context Conversational Memory benchmark consisting of 10 synthetic multi-session conversations with 1,986 question-answer pairs. Questions are divided into five categories: (1) *Single-Hop*: direct fact retrieval; (2) *Multi-Hop*: multi-step reasoning; (3) *Temporal Reasoning*: time-sensitive queries requiring date resolution; (4) *Open Domain*: general knowledge queries; and (5) *Adversarial*: queries about information not present in the conversations. Each question is accompanied by evidence references (e.g., D1 : 3) pointing to specific conversation segments.

DMR-MS (Packer et al., 2024). The Dynamic Memory Retrieval Multi-Session Chat dataset contains 500 multi-session dialogues with self-instructed question-answer pairs. Each example includes persona statements, dialog turns with speaker identities, and temporal context via time_back annotations (e.g., "14 days"). We use this largely non-temporal conversational benchmark to check that temporal reranking does not reduce standard memory retrieval performance.

FinTMMBench (Zhu et al., 2025). Financial Temporal Multi-Modal Benchmark containing

Table 4: Summary statistics of evaluation datasets.

Dataset	RQ	Facts/Docs	Queries	Task Type
ICEWS05-15	RQ1	461,329	46,092	TKG completion
MultiTQ	RQ2	11,074	500	Temporal KGQA
LoCoMo	RQ2	10 conv.	1,986	Conv. memory QA
DMR-MSC	RQ2	500 dial.	500	Dialogue memory QA
FinTMMBench	RQ3	908 docs [†]	100 [†]	Financial temporal QA

[†]Stratified sample: 100 questions with a corpus of 908 documents (227 gold sources + 681 randomly sampled non-gold documents).

Table 5: Distribution of the full MultiTQ test set and the fixed 500-question evaluation subset.

Category	Full test set	Fixed subset
equal	17,311 (31.71%)	155 (31.00%)
before_after	11,073 (20.29%)	102 (20.40%)
first_last	10,480 (19.20%)	108 (21.60%)
equal_multi	3,207 (5.88%)	23 (4.60%)
after_first	6,266 (11.48%)	49 (9.80%)
before_last	6,247 (11.44%)	63 (12.60%)
Entity answer	38,700 (70.90%)	345 (69.00%)
Time answer	15,884 (29.10%)	155 (31.00%)

5,676 question-answer pairs over NASDAQ-100 companies. The corpus comprises 162,311 documents across four modalities: *News* (3,143 articles), *FinancialTable* (35,038 indicator records), *Stock-Price* (124,130 price records), and *Chart* (vision-based, excluded from our text-only evaluation). Each question references specific date ranges (e.g., “from 2022-06-27 to 2022-09-23”) and gold source document UUIDs for provenance evaluation. Question subtasks include *Extraction*, *Calculation*, *Sentiment*, and *Trend* analysis. To enable feasible evaluation, we use a stratified sample of 100 questions and a reduced corpus of 908 documents, comprising 227 gold sources and 681 randomly sampled non-gold documents.

D Licenses and Terms of Use

The following terms are those stated in the official upstream releases at the time of access. “Not specified” means that we found no explicit license in the official repository or dataset card; it should not be interpreted as permission for unrestricted reuse. We use all third-party resources only for the research reported here. Our release does not relicense these resources, and users should obtain them from the linked upstream sources and comply with their terms. A repository license also does not override any separate rights in third-party source material contained in a dataset.

Datasets.

- **ICEWS05-15:** The source ICEWS Coded Event Data is available for research and educational use only; commercial use is prohibited. We use the derived ICEWS05-15 split for gate pretraining and RQ1 evaluation.
- **MultiTQ:** Not specified. Its ICEWS-based knowledge graph remains subject to the ICEWS terms above.
- **LoCoMo:** Creative Commons Attribution–NonCommercial 4.0 International (CC BY-NC 4.0).
- **DMR-MSC:** Apache License 2.0, as stated by the official MSC-Self-Instruct dataset card.
- **FinTMMBench:** MIT License for the official repository release; rights in any underlying third-party source material remain with the respective providers.

Agentic-memory baselines.

- **Mem0:** Apache License 2.0.
- **Zep/Graphiti:** Apache License 2.0. We evaluate Graphiti, the open-source temporal graph engine associated with Zep, rather than the managed Zep service.
- **A-Mem:** MIT License.
- **LiCoMemory:** Not specified. We therefore do not treat its public repository as permission to redistribute its source code.
- **HippoRAG:** MIT License.

The TKGE comparison values in Table 1 are taken from the cited papers, as stated in its caption, rather than from redistributed third-party implementations. Model and API access (including GPT-5-mini, text-embedding-3-small, LLaMA-3.1-70B, and BGE-M3) remains subject to the corresponding providers’ terms.

E Evaluation Metrics and Answer Verification

Temporal KG Completion (RQ1). For ICEWS05-15, we report Mean Reciprocal Rank (MRR) and Hits@ k for $k \in \{1, 3, 10\}$, following the standard filtered setting:

$$\text{MRR} = \frac{1}{|Q|} \sum_{q \in Q} \frac{1}{\text{rank}(q)} \quad (9)$$

$$\text{Hits}@k = \frac{1}{|Q|} \sum_{q \in Q} \mathbb{I}[\text{rank}(q) \leq k] \quad (10)$$

Agentic Memory Retrieval (RQ2, RQ3). For MultiTQ and DMR-MSc, we report MRR and Hits@ k ($k \in \{1, 3, 10\}$) computed over retrieved facts, where each query has a single gold answer:

$$\text{Hits}@k = \mathbb{I}[\exists i \leq k : \text{doc}_i \in \mathcal{G}] \quad (11)$$

For LoCoMo, the main table reports Recall@10 per question category, computed as the fraction of gold evidence passages found in the top-10 retrieved documents:

$$\text{Recall}@k = \frac{|\{d \in \text{top-}k : d \in \mathcal{G}\}|}{|\mathcal{G}|} \quad (12)$$

For FinTMMBench, we report Recall@ k for $k \in \{1, 3, 5, 10\}$ and MRR, computed against gold source document UUIDs. Appendix F additionally reports aggregate LoCoMo partial-match F1.

Answer Quality. We evaluate downstream answer quality using LLM@ k Accuracy, where k denotes the number of retrieved documents provided to the answer LLM. For DMR-MSc and FinTMMBench, a generated answer is scored as correct via the two-stage LLM judge pipeline described in Appendix E.1. For MultiTQ, we instead use a rule-based cascading verifier (Appendix E.2) following the original benchmark protocol. We report LLM@ k Accuracy at context sizes $k \in \{5, 10\}$. Appendix F re-evaluates the saved DMR-MSc and FinTMMBench outputs with a stricter conflict-aware judge and reports a manual audit of unresolved alternatives.

E.1 LLM Judge

We use a two-stage answer evaluation pipeline. **Stage 1** (fast path): if the lowercased, whitespace-normalised gold answer is a substring of the generated answer, the answer is immediately labelled

CORRECT. **Stage 2** (LLM fallback): for non-matching answers, we invoke GPT-5.2 as a judge using the prompt shown in Box VII. The judge is called with temperature=0 and JSON response format {"label": "CORRECT"|"WRONG"}.

Box VII: LLM Judge Prompt

Your task is to label an answer to a question as 'CORRECT' or 'WRONG'. You will be given:

- (1) a question (posed by one user to another user),
- (2) a 'gold' (ground truth) answer,
- (3) a generated answer

which you will score as CORRECT/WRONG.

The point of the question is to ask about something one user should know about the other user based on their prior conversations. The gold answer will usually be a concise and short answer that includes the referenced topic, for example:

Question: Do you remember what I got the last time I went to Hawaii?
Gold answer: A shell necklace

The generated answer might be much longer, but you should be generous with your grading - as long as it touches on the same topic as the gold answer, it should be counted as CORRECT.

For time related questions, the gold answer will be a specific date, month, year, etc. The generated answer might be much longer or use relative time references (like "last Tuesday" or "next month"), but you should be generous with your grading - as long as it refers to the same date or time period as the gold answer, it should be counted as CORRECT. Even if the format differs (e.g., "May 7th" vs "7 May"), consider it CORRECT if it's the same date.

Now it's time for the real question:

Question: {question}
Gold answer: {gold_answer}
Generated answer: {generated_answer}

First, provide a short (one sentence) explanation of your reasoning, then finish with CORRECT or WRONG. Do NOT include both CORRECT and WRONG in your response, or it will break the evaluation script.

Just return the label CORRECT or WRONG in a json format with the key as "label".

E.2 MultiTQ Answer Verifier

For MultiTQ, answer correctness is determined by a cascading multi-strategy rule-based verifier (rather than the LLM judge used for other bench-

marks), adopted from Tan et al. (2026) to handle the diverse answer formats in temporal KGQA (entities, dates at varying granularities, and multi-part answers). The strategies are applied in order; the first match determines the verdict:

1. **Exact match:** normalised entity string equality.
2. **Containment:** bidirectional substring check ($\text{gold} \subseteq \text{prediction}$ or $\text{prediction} \subseteq \text{gold}$).
3. **Advanced normalisation:** strip prefixes (e.g., “The”), brackets, and punctuation, then substring match.
4. **Time format matching:** year-month level matching with month-name support (e.g., “January 2013” $\approx$ “2013-01”).
5. **Multi-answer:** comma-separated answer parts are matched individually.
6. **Semantic overlap:** word overlap $> 50\%$ between prediction and gold answer tokens.
7. **Loose match:** remove all spaces and underscores, then substring match.

F Additional Evaluation Analyses

F.1 Component Ablation Details

Table 6 reports all metrics for the MultiTQ ablation summarised in Table 3. Every variant uses the same fixed set of 500 questions, the complete graph of 461,329 facts, cached embeddings, ChronoR scoring dimension, 200 training epochs, reader, verifier, and optimisation settings.

F.2 LoCoMo Answer-Level F1

Table 7 reports partial-match F1 over all 1,986 LoCoMo questions, using the same reader outputs as the retrieval evaluation. Although ROMEM improves Recall@10 in Table 2b, it matches HippoRAG at 0.457 F1 in the OpenAI configuration and improves F1 by only 0.001 in the server configuration. The original LoCoMo study reports a similar gap between evidence recall and answer F1, which it attributes to the reader’s difficulty in using a larger or noisier context (Maharana et al., 2024).

F.3 Conflict-Aware Judge Robustness

We use the answer-level judge only for DMR-MSD and FinTMMBench. ICEWS05-15, MultiTQ retrieval, MultiTQ answer accuracy, and LoCoMo

retrieval use deterministic or gold-evidence metrics. To test whether the wording of the original judge accepts unresolved contradictions, we evaluated the same saved answers with a stricter prompt. We did not rerun retrieval or answer generation. The stricter judge requires the answer to state the gold fact and marks unresolved conflicting alternatives as incorrect.

As shown in Table 8, ROMEM remains the top-ranked method on both benchmarks. The original and strict judges agree on 98.8% of the 500 DMR-MSD answers and 98.0% of the 100 FinTMMBench answers. We also inspected answers that contained conflicting candidates. Three ROMEM answers on DMR-MSD contained unresolved alternatives, and the original judge accepted two of them (0.4% of the evaluation set). We found no such case on FinTMMBench.

G Implementation Configurations and Hyperparameters

G.1 Configurations

We evaluate all agentic memory benchmarks under two implementation configurations:

1. **OpenAI:** GPT-5-mini for graph construction (NER + OpenIE) and text-embedding-3-small for embedding. This configuration tests performance with API-based models.
2. **Server:** LLaMA-3.1-70B-Instruct (Meta AI, 2024) served via vLLM for graph construction and BAAI/BGE-M3 (Chen et al., 2024) for embedding. This configuration demonstrates open-source reproducibility.

In both configurations, the **answer LLM** and **LLM judge** always use GPT-5.2 via the OpenAI API to ensure fair comparison across all baselines.

Baseline systems. We compare against:

- **Mem0** (Chhikara et al., 2025): FAISS-based vector memory with per-document embedding and search.
- **Zep** (Rasmussen et al., 2025): Temporal knowledge graph with Neo4j backend and entity extraction.
- **HippoRAG** (Gutierrez et al., 2024; Gutiérrez et al., 2025): Knowledge graph-augmented RAG with Personalised PageRank retrieval and Neo4j backend. ROMEM builds upon HippoRAG’s graph construction pipeline.

Table 6: Full single-factor ablation on MultiTQ. “No gate” uses one temporal speed for every relation; “discrete time” replaces functional time with learned timestamp lookups.

Variant	MRR	H@1	H@3	H@10	Acc@5	Acc@10
Full ROMEM	0.3371	0.252	0.384	0.502	0.366	0.392
w/o time contrastive	0.3254	0.238	0.368	0.504	0.340	0.390
w/o semantic gate	0.1044	0.084	0.116	0.142	0.100	0.108
Discrete time	0.0452	0.042	0.046	0.048	0.040	0.040

Table 7: Aggregate LoCoMo partial-match F1. “OpenAI” and “Server” refer to the graph-construction and embedding configurations; the answer reader is fixed.

Method	OpenAI F1	Server F1
Zep	0.343	0.320
Mem0	0.457	0.444
A-Mem	0.459	0.382
LicoMemory	0.346	0.306
HippoRAG	0.457	0.446
ROMEM	0.457	0.447

All baselines use the same answer LLM, LLM judge, and evaluation metrics for fair comparison. We use the same temporal configuration for every target benchmark and do not tune hyperparameters separately for each dataset. The Semantic Speed Gate remains frozen; training on the target graph updates only the entity and relation embeddings and the global temporal-spectrum parameters.

G.2 TKGE Hyperparameters

Table 9 reports the full TKGE hyperparameter configuration used across all experiments.

H Theoretical Analysis

H.1 Overview

While transitioning from a discrete timestamp dictionary to a continuous functional rotation resolves the granularity and extrapolation limitations of traditional temporal models, it introduces distinct algebraic and computational challenges. This appendix section provides the rigorous theoretical justification for our framework, structured around three core aspects:

- **Complex vs. Real Representation (H.2):** We clarify the relationship between the complex-space formulation in $\mathbb{C}^d$ and its real-space equivalent in $\mathbb{R}^{2d}$, demonstrating why the complex unitary group offers concrete structural advantages for temporal knowledge graphs.
- **Retrieval Reformulation (H.3):** We prove that the orthogonal structure of the rotation operator

allows the continuous temporal transformation to be isolated entirely on the query side, thereby preserving strict compatibility with static vector search indices.

- **Temporal Interpolation (H.4):** We analyse a simplified pairwise setting and give a sufficient condition for a unique, monotonic crossover between historical anchors.

H.2 Complex vs. Real Representation

Our methodology embeds entities in $\mathbb{R}^{2d}$ and interprets them as complex vectors in $\mathbb{C}^d$. These two views are algebraically equivalent: a rotation by angle θ in $\mathbb{C}^d$ corresponds to applying a block-diagonal orthogonal matrix $R_\theta \in \mathbb{R}^{2d \times 2d}$ composed of d independent 2×2 rotation blocks.

The complex formulation is not merely notational convenience. As established by RotatE (Sun et al., 2019) and ChronoR (Sadeghian et al., 2021), expressing relational transformations as element-wise rotations (Hadamard products) in the unitary group $U(1)^d$ naturally captures critical graph structures such as symmetry, antisymmetry, inversion, and composition, while reducing transformation cost from $\mathcal{O}(d^2)$ (general matrix multiplication) to $\mathcal{O}(d)$ (element-wise operations). Our analysis below establishes that these algebraic benefits seamlessly extend to continuous time dynamics.

H.3 Retrieval Reformulation and Static Index Compatibility

In the temporal knowledge graph retrieval setting, evaluating a query $(h, r, ?)$ against all candidate tail entities $t \in \mathcal{E}$ under continuous time introduces a severe scalability bottleneck. Naively applying the time-dependent rotation to the entire candidate vocabulary dictates an $\mathcal{O}(N \cdot d)$ dynamic transformation at query time, which fundamentally breaks compatibility with prebuilt static vector search indices and renders large-scale querying computationally intractable.

We ask whether continuous temporal queries can be evaluated without changing the candidate index.

Table 8: Acc@5 under the original and stricter conflict-aware answer judges. All saved answers are identical across judge protocols.

Method	DMR-MSC original	DMR-MSC strict	FinTMMBench original	FinTMMBench strict
Zep	0.302	0.278	0.480	0.470
Mem0	0.858	0.834	0.550	0.540
A-Mem	0.848	0.830	0.540	0.540
LicoMemory	0.670	0.604	0.480	0.480
HippoRAG	0.852	0.832	0.550	0.550
RoMEM	0.862	0.850	0.580	0.560

Table 9: TKGE hyperparameters. Parameters above the mid-rule are core KGE settings; below are temporal extension parameters.

Parameter	Value	Description
temporal_backbone	chronor	ChronoR rotation backbone
chronor_k	3	Number of rotation sub-spaces
gamma	200.0	Embedding range: $(\gamma + 2)/\text{dim}$
adversarial_temperature	1.0	Self-adversarial sampling temperature
regularization_weight	10^{-5}	N3 per-batch regularization (ChronoR)
steps_per_update	500	Training epochs per update cycle
num_conflict_negatives	1	Tails from same (h, r) group
time_source	happen	Use text_time for temporal scoring
time_loss_type	listwise	Distribution-matching time loss
time_contrastive_weight	0.5	Time-contrastive loss weight (λ_{time})
num_time_negatives	8	Negative time samples per fact
time_sigma_years	0.25	Gaussian kernel σ (years)
time_sigma_years_start	0.5	Curriculum start σ
time_sigma_years_end	0.02	Curriculum end σ
time_neg_jitter_years	0.02	Temporal jitter for negatives
time_neg_far_days	365	Far negative offset (days)
time_neg_min_days_start	90	Curriculum start min-gap (days)
time_neg_min_days_end	3	Curriculum end min-gap (days)
time_neg_min_days_decay	60	Min-gap curriculum decay (epochs)
tkge_weight (α_g)	0.3	Temporal reranking strength

We use the real-space implementation from §3.3, where $R_{\theta_r(\tau)} \in \mathbb{R}^{2d \times 2d}$ is the block-diagonal orthogonal matrix corresponding to $\text{Rot}(\cdot, \theta_r(\tau))$, and $\mathbf{W}_r = \text{diag}(\mathbf{w}_r \odot \hat{\mathbf{w}}_r) \in \mathbb{R}^{2d \times 2d}$ is the relation-specific diagonal scaling matrix. For clarity, we omit the component index c below. The same reformulation applies to each of the k components before their scores are summed. It moves the temporal transformation to the query side, allowing the candidate index to remain unchanged and compatible with Maximum Inner Product Search (MIPS).

Proposition 1 (Query-Side Reformulation of Temporal Retrieval). *Let $\mathbf{C} \in \mathbb{R}^{N \times 2d}$ be the static matrix of candidate embeddings in the real-space implementation, where the t -th row of $\mathbf{C}$ is $\mathbf{e}_t^\top$. For a continuous timestamp τ , suppose the score against candidate t is defined by the standard Euclidean inner product:*

$$s_{\text{kge}}((h, r, t) \mid \tau) = \langle \mathbf{W}_r R_{\theta_r(\tau)} \mathbf{e}_h, R_{\theta_r(\tau)} \mathbf{e}_t \rangle$$

where $R_{\theta_r(\tau)} \in \mathbb{R}^{2d \times 2d}$ is a block-diagonal orthogonal matrix representing the phase shift, and $\mathbf{W}_r \in \mathbb{R}^{2d \times 2d}$ is the diagonal relation-specific scaling operator. Then there exists a candidate-independent query vector $\mathbf{q}(\tau) \in \mathbb{R}^{2d}$ defined as:

$$\mathbf{q}(\tau) = R_{-\theta_r(\tau)} \mathbf{W}_r R_{\theta_r(\tau)} \mathbf{e}_h$$

such that the score reduces to:

$$s_{\text{kge}}((h, r, t) \mid \tau) = \langle \mathbf{q}(\tau), \mathbf{e}_t \rangle$$

for all candidates t . Consequently, the exact 1-vs- N retrieval reduces to an $\mathcal{O}(d)$ query-side pre-processing step followed by an $\mathcal{O}(N \cdot d)$ static inner-product search over $\mathbf{C}$.

Proof. In a direct candidate-side implementation, evaluating the score requires applying the time-dependent rotation $R_{\theta_r(\tau)}$ to every candidate vector $\mathbf{e}_t$ at query time. Although this can be computed in a streaming fashion in $\mathcal{O}(N \cdot d)$ time, this query-time temporal transformation fundamentally

prevents the use of any prebuilt static inner-product index.

However, since the rotation matrix $R_{\theta_r(\tau)}$ is orthogonal, its transpose satisfies $R_{\theta_r(\tau)}^\top = R_{-\theta_r(\tau)}$. Utilizing the adjoint property of the Euclidean inner product, we can strictly transfer the rotation from the candidate vector back to the query side:

$$s_{\text{kge}}((h, r, t) \mid \tau) = \langle R_{-\theta_r(\tau)}(\mathbf{W}_r R_{\theta_r(\tau)} \mathbf{e}_h), \mathbf{e}_t \rangle$$

By substituting the definition of the unrotated query vector $\mathbf{q}(\tau)$, the scoring function trivially simplifies to $s_{\text{kge}}((h, r, t) \mid \tau) = \langle \mathbf{q}(\tau), \mathbf{e}_t \rangle$. We note that the temporal dependence remains non-trivial whenever the relation-specific operator $\mathbf{W}_r$ does not commute with $R_{\theta_r(\tau)}$.

Complexity Analysis. Constructing the query vector $\mathbf{q}(\tau)$ requires $\mathcal{O}(d)$ trigonometric evaluations and $\mathcal{O}(d)$ structured linear operations, since $R_{\theta_r(\tau)}$ is block-diagonal and $\mathbf{W}_r$ is a diagonal matrix. To evaluate the query against all N candidates simultaneously, we compute the full score vector $\mathbf{s}(\tau) \in \mathbb{R}^N$, where each t -th element strictly corresponds to the individual scalar score $s_{\text{kge}}((h, r, t) \mid \tau)$. This is achieved via a single matrix-vector multiplication:

$$\mathbf{s}(\tau) = \mathbf{C} \mathbf{q}(\tau)$$

which requires $\mathcal{O}(N \cdot d)$ arithmetic operations. The additional query-time workspace is $\mathcal{O}(d)$, and the retrieval stage requires no candidate-side temporal transformation. This proves the proposition. $\square$

Remark: Because the final formulation reduces retrieval to a standard inner-product evaluation over a static candidate matrix $\mathbf{C}$, the model is directly compatible with exact or approximate inner-product search libraries (e.g., FAISS) without requiring index rebuilds across queries.

H.4 A Stylised Analysis of Pairwise Temporal Interpolation

A fundamental advantage of formulating time as a continuous rotation operator lies in its structural capacity to interpolate knowledge between historical observations. While exact global retrieval over an entire knowledge graph entails complex multi-dimensional phase interference, we can rigorously demonstrate the model’s interpolation mechanics by analysing a stylised pairwise regime.

Trigonometric Isomorphism of the Scoring Function. Before stating the formal proposition, we establish the algebraic bridge between the

global inner-product scoring formulation (Equation (4)) and its corresponding trigonometric expansion. Since the total score s_{kge} is a linear sum of independent dimensional contributions ($s_{\text{kge}} = \sum_j s_j$), we can isolate the temporal dynamics within a single complex dimension. For each 2D rotational subspace j , the partial score evaluates a bilinear form:

$$s_j(\tau) = \mathbf{h}_j^\top R_{\theta_j(\tau)}^\top \mathbf{W}_j R_{\theta_j(\tau)} \mathbf{t}_j$$

where $\mathbf{h}_j, \mathbf{t}_j \in \mathbb{R}^2$ are the static entity vectors, $\mathbf{W}_j = \text{diag}(w_{j,1}, w_{j,2})$ is the relation weight matrix, and $\theta_j(\tau)$ is the j -th scalar component of the relation-specific rotation vector $\theta_r(\tau)$ (defined in Equation (1)). Because the relation weights typically break rotational symmetry ($w_{j,1} \neq w_{j,2}$), expanding $R_{\theta_j(\tau)}^\top \mathbf{W}_j R_{\theta_j(\tau)}$ via double-angle identities yields a linear combination of trigonometric functions:

$$s_j(\tau) = C_j + A_j \cos(2\theta_j(\tau)) + B_j \sin(2\theta_j(\tau))$$

where C_j, A_j, B_j are time-independent constants determined by the entities and relation weights. By applying the harmonic addition theorem, this combination can be exactly re-parameterised as a phase-shifted cosine wave:

$$s_j(\tau) = C_j + \gamma_j \cos(2\theta_j(\tau) - \phi_j)$$

with amplitude $\gamma_j = \sqrt{A_j^2 + B_j^2} > 0$ and phase shift $\phi_j = \text{atan2}(B_j, A_j)$. Since the rotation angle $\theta_j(\tau)$ is linearly proportional to time τ (Equation (1)), summing these independent subspaces across all dimensions reduces the exact relational inner product to a multi-frequency cosine expansion. This form motivates the simplified scoring model below.

In this section, we establish a sufficient condition under which the temporal competition between two consecutive facts exhibits a smooth, monotonic crossover, thereby yielding a deterministic decision boundary for unobserved intermediate timestamps.

Proposition 2 (Sufficient Condition for Monotone Pairwise Crossover). *Consider a temporal query (h, r) with two mutually exclusive facts, A and B , observed at consecutive timestamps T and $T + t$ ($t > 0$). Assume that over the interpolation interval $[T, T + t]$, the expansion of the inner-product scoring function (Equation (4)) for each candidate*

$c \in \{A, B\}$ is governed by a stylised synchronised cosine expansion:

$$s_c(\tau) = \sum_{j=1}^d \gamma_{c,j} \cos(\tilde{\omega}_j(\tau - \tau_c)), \quad \gamma_{c,j} > 0$$

where $\tau_A = T$ and $\tau_B = T + t$ denote the local phase-alignment peaks. Consistent with the double-angle expansion above and Equation (1), $\tilde{\omega}_j = 2s \cdot \alpha_r \cdot \omega_j$ represents the effective angular velocity, which incorporates the global time scale s , inverse frequency ω_j , and relation-specific semantic speed gate α_r .

Assume the model accurately reconstructs these historical anchors such that $s_A(T) > s_B(T)$ and $s_A(T + t) < s_B(T + t)$. Provided the effective angular velocities satisfy the strict half-period bound $\tilde{\omega}_j \in (0, \frac{\pi}{t}]$ for all j , the pairwise confidence gap $\Delta s(\tau) = s_A(\tau) - s_B(\tau)$ is strictly monotonically decreasing on the open interval $(T, T + t)$.

Consequently, there exists a unique crossover timestamp $\tau^* \in (T, T + t)$ satisfying $s_A(\tau^*) = s_B(\tau^*)$. For any intermediate time $\tau \in (T, T + t)$, the model strictly prefers A when $\tau < \tau^*$, strictly prefers B when $\tau > \tau^*$, and yields an exact pairwise tie at τ^* .

Proof. By the assumption of anchor correctness, the relative confidence at the boundaries satisfies $\Delta s(T) > 0$ and $\Delta s(T + t) < 0$.

To analyse the transition mechanics, we differentiate the confidence gap with respect to continuous time τ :

$$\frac{d}{d\tau} \Delta s(\tau) = s'_A(\tau) - s'_B(\tau)$$

The derivatives of the stylised scoring functions are given by:

$$s'_A(\tau) = - \sum_{j=1}^d \gamma_{A,j} \tilde{\omega}_j \sin(\tilde{\omega}_j(\tau - T))$$

$$s'_B(\tau) = - \sum_{j=1}^d \gamma_{B,j} \tilde{\omega}_j \sin(\tilde{\omega}_j(\tau - (T + t)))$$

For any intermediate time $\tau = T + dt$ with $dt \in (0, t)$, the temporal displacement for candidate A is dt . Given $\tilde{\omega}_j \in (0, \frac{\pi}{t}]$, the phase argument $\tilde{\omega}_j dt$ strictly resides in $(0, \pi)$. In this interval, the sine function is strictly positive. Since $\gamma_{A,j} > 0$ and $\tilde{\omega}_j > 0$, it strictly follows that $s'_A(\tau) < 0$.

Conversely, the temporal displacement for candidate B is $dt - t < 0$. Under the same frequency

bound, the phase argument $\tilde{\omega}_j(dt - t)$ strictly resides in $(-\pi, 0)$, where the sine function is strictly negative. This renders $s'_B(\tau) > 0$.

Therefore, the derivative of the pairwise gap is strictly negative across the entire open interval:

$$\frac{d}{d\tau} \Delta s(\tau) < 0 \quad \forall \tau \in (T, T + t)$$

This strict monotonicity, coupled with the continuity of the scoring functions on $[T, T + t]$ and the boundary conditions, guarantees via the Intermediate Value Theorem (IVT) the existence of exactly one root τ^* in $(T, T + t)$ where $\Delta s(\tau^*) = 0$. $\square$

Remark: The proposition gives a sufficient condition for pairwise interpolation; it does not guarantee a single crossover for the complete learned spectrum or the full candidate set. Under the half-period bound, the two candidates cross once between consecutive anchors. The crossover τ^* need not occur at the midpoint $T + t/2$, because its position depends on the amplitudes $\gamma_{A,j}$ and $\gamma_{B,j}$. Higher frequencies may violate the bound and produce local oscillations, as shown in Appendix I.

I Qualitative Analysis: Geometric Shadowing in Action

To make the geometric shadowing mechanism concrete, we present scoring traces from a controlled experiment. We select a small subset of real temporal triples from ICEWS05-15 (García-Durán et al., 2018), including 4 (Obama, Consult, Blair) facts timestamped between June 2007 and April 2008, and 6 (Obama, Consult, Xi Jinping) facts timestamped between June 2013 and September 2015, along with static facts (born in) and auxiliary relation slots (Make a visit, Express intent to cooperate). These additional facts are included to provide sufficient training signal for the shared entity embeddings and frequency spectrum while training on the two competing facts alone leaves the model severely underconstrained, producing degenerate oscillations. We train a ROMEM-ChronoR model on this subset with the bundled pre-trained gate and sweep the query timestamp τ_q to observe how candidate scores evolve continuously.

I.1 Dynamic Relation: Score Crossover

Consider the relational slot (Barack Obama, Consult, ?) with two competing tail entities: Tony Blair (4 observations, 2007-06 to 2008-04)

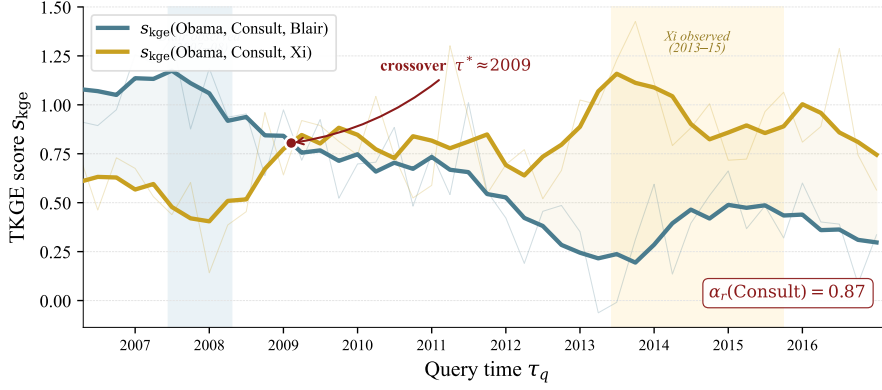


Figure 3: Scores for the competing slot (Obama, Consult, ?). The light curves are the raw quarterly scores used for retrieval, and the bold curves are 5-quarter rolling averages shown only to make the overall trend easier to see. The blue and yellow regions mark the observation periods for Blair (2007-2008) and Xi (2013-2015), respectively. The red dot marks where the smoothed curves cross. The gate assigns a high rotation speed to “Consult” ($\alpha_r = 0.87$).

and *Xi Jinping* (6 observations, 2013-06 to 2015-09). The pretrained Semantic Speed Gate assigns $\alpha_r(\text{Consult}) = 0.87$, which is consistent with the rapidly changing participants of this relation.

Figure 3 shows the TKGE scores s_{kge} as the query time τ_q moves from 2007 to 2016. Blair has the higher smoothed score at the beginning of this period, but his score gradually falls as Xi’s rises. The smoothed curves cross around 2009, after which Xi *shadows* Blair. Proposition 2 guarantees a unique crossover only when its half-period condition holds for every frequency component.

Crucially, neither fact is deleted: both remain in the append-only memory, and the rotation operator continuously modulates their alignment with the query time.

On raw oscillations and multiple crossings. The raw quarterly scores (light curves in Figure 3) oscillate and cross more than once. Thus, not every learned frequency in this example satisfies the sufficient condition in Proposition 2. The proposition guarantees monotonicity only when all frequencies satisfy the half-period bound $\tilde{\omega}_j \leq \pi/t$. Two factors explain these fluctuations. First, the model learns a spectrum of kd frequency components, and higher-frequency components (those with period shorter than the ~ 5 -year observation gap) violate this bound, producing local oscillations. Second, entity embeddings are shared across all relation slots. Obama’s embedding is jointly trained on Consult, Make a visit, and born in facts, so temporal dynamics from other slots introduce additional phase interference into the Consult scores.

For instance, Blair’s raw score briefly rises around 2010-2011 before resuming its decline, and Xi’s score dips near 2015 before recovering.

The 5-quarter rolling average summarises the overall trend and produces a single crossover of the form analysed in Proposition 2. The raw scores show that the proposition is not a guarantee for the complete learned model. The rolling average is used only in this figure, not during retrieval.

I.2 Gate Inspection: Learned Relational Volatility

Table 10 reports α_r for representative relations that either appeared during ICEWS05-15 pretraining (*seen*) or were evaluated only after pretraining (*unseen*). The values follow the expected volatility ordering without manual static/dynamic labels. These examples are qualitative and do not measure calibration over all relations.

A notable property of this result is that ICEWS05-15 contains *few semantically static relations*. All 251 relation types describe political events (consulting, visiting, threatening, etc.), lacking permanent properties like “born in” or “species”. Across 461K facts spanning 4,017 timestamps, the average (h, r) slot has 3.08 distinct tail entities, which means every relation exhibits temporal variation.³ Yet the gate still learns a meaningful volatility gradient within this event-driven spectrum: episodic interactions like “Consult” (0.87)

³57 of 251 relations technically have single-tail slots, but these are all rare event types (≤ 42 facts each) that appear static mostly due to data sparsity, not semantic permanence (e.g., “Attempt to assassinate,” “Demand mediation”).

Table 10: Pretrained semantic speed gate values. Higher α_r = faster rotation. *Seen*: appeared in ICEWS05-15 during pretraining; *Unseen*: zero-shot via text embedding similarity.

Relation	α_r	Category
<i>Seen during gate pretraining (ICEWS05-15)</i>		
Consult	0.87	Dynamic
Host a visit	0.86	Dynamic
Engage in negotiation	0.63	Dynamic
Sign formal agreement	0.53	Dynamic
Cooperate economically	0.16	Low
Cooperate militarily	0.09	Low
<i>Unseen (zero-shot via text embeddings)</i>		
met with	0.71	Dynamic
visited	0.64	Dynamic
negotiated with	0.62	Dynamic
CEO of	0.44	Moderate
capital of	0.36	Moderate
species	0.22	Static
citizen of	0.17	Static

and “Host a visit” (0.86) receive high α_r , while sustained state-level conditions like “Cooperate militarily” (0.09) and “Cooperate economically” (0.16) receive lower values.

The unseen examples show that the gate transfers beyond the relation strings observed during pretraining. Episodic predicates receive higher values (“met with” at 0.71 and “visited” at 0.64), while persistent properties receive lower values (“citizen of” at 0.17 and “species” at 0.22). This transfer comes from using text embeddings rather than relation IDs. We evaluate only the relations and domains reported here, so these examples do not establish that the gate is calibrated for arbitrary predicates.

The DMR-MSC results in Table 2c provide a separate preservation check: temporal reranking does not reduce performance on this largely non-temporal conversational benchmark.