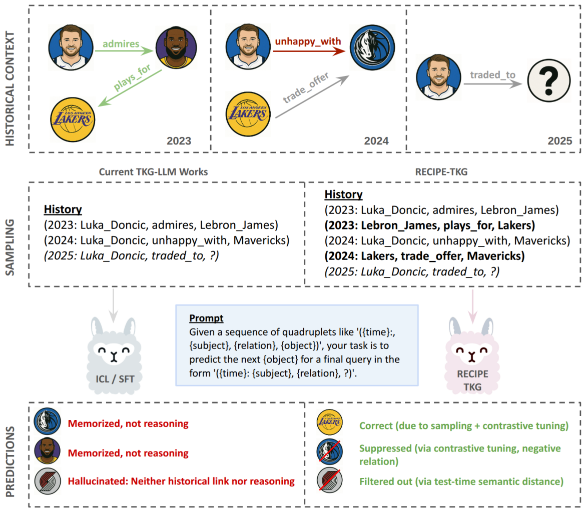
RECIPE-TKG: From Sparse History to Structured Reasoning for LLM-based Temporal Knowledge Graph Completion

Anonymous ACL submission

Abstract

Temporal Knowledge Graphs (TKGs) represent dynamic facts as timestamped relations between entities. While Large Language Models (LLMs) show promise for TKG completion, current approaches typically apply generic pipelines (neighborhood sampling, supervised fine-tuning, uncalibrated inference) without task-specific adaptation to temporal relational reasoning. Through systematic analysis under unified evaluation, we reveal three key failure modes: (1) retrieval strategies miss multi-hop dependencies when target entities are not directly observed in history, (2) standard fine-tuning reinforces memorization over relational generalization, and (3) uncalibrated generation produces contextually implausible entities. We present RECIPE-TKG, a parameter-efficient framework that addresses each limitation through principled, task-specific design: rule-based multi-hop sampling for structural grounding, contrastive fine-tuning to shape relational compatibility, and test-time semantic filtering for contextual alignment. Experiments on four benchmarks show that RECIPE-TKG outperforms prior LLM-based methods across input regimes, achieving up to 22.4% relative improvement in Hits@10, with particularly strong gains when historical evidence is sparse or indirect.


HISTORICAL CONTEXT
admires
plays_for
2023
unhappy_with
trade_offer
2024
traded_to
?
2025
Current TKG-LLM Works
RECIPE-TKG
SAMPLING
ICL / SFT
RECIPE TKG
PREDICTIONS
Memorized, not reasoning
Memorized, not reasoning
Hallucinated: Neither historical link nor reasoning
Correct (due to sampling + contrastive tuning)
Suppressed (via contrastive tuning, negative relation)
Filtered out (via test-time semantic distance)

History
(2023: Luka_Doncic, admires, Lebron_James)
(2024: Luka_Doncic, unhappy_with, Mavericks)
(2025: Luka_Doncic, traded_to, ?)

History
(2023: Luka_Doncic, admires, Lebron_James)
(2023: Lebron_James, plays_for, Lakers)
(2024: Luka_Doncic, unhappy_with, Mavericks)
(2024: Lakers, trade_offer, Mavericks)
(2025: Luka_Doncic, traded_to, ?)

Prompt
Given a sequence of quadruplets like '({time}:, {subject}, {relation}, {object})', your task is to predict the next {object} for a final query in the form '({time}: {subject}, {relation}, ?)'.

Figure 1: Example of LLM-based TKG reasoning. Pipelines that rely on one-hop context tend to prefer locally frequent or lexically similar entities, yielding off-context outputs. RECIPE-TKG augments history with structurally and temporally richer facts and applies semantic checking, producing more plausible predictions.

1 Introduction

Temporal Knowledge Graphs (TKGs) are widely used to represent dynamic, real-world knowledge across domains such as news (Boschee et al., 2015; Leetaru and Schrodt, 2013), biomedicine (Chaturvedi, 2024), and finance (Dukkipati et al., 2025). They capture facts as time-stamped relational tuples (`subject`, `relation`, `object`, `timestamp`), modeling how interactions evolve over time (Tresp et al., 2015). A core task in this setting is TKG completion, which involves predicting missing or future links based on observed temporal interactions. This task requires reasoning over both relational and temporal structure, with downstream applications in forecasting and decision support (Trivedi et al., 2017; Jin et al., 2020).

The rise of Large Language Models (LLMs) has sparked interest in using pretrained generative models for TKG completion, driven by their generalization capability and emergent reasoning skills (Liao et al., 2024; Luo et al., 2024; Lee et al., 2023). While LLM reasoning is often benchmarked on math or logic-based tasks (Lewkowycz et al., 2022; Wang et al., 2025), TKG completion provides a complementary testbed that emphasizes two challenges: (1) integrating temporal and structural signals beyond one-hop evidence, and (2) generalizing when history is *sparse* or the answer is *non-historical* (absent from retrieved context and typi-

The code is available at this anonymous repository.

cally reachable only via multi-hop paths).

Recent prompting-based and fine-tuned LLM methods (Lee et al., 2023; Liao et al., 2024; Luo et al., 2024; Xia et al., 2024a) report promising results. However, **these approaches typically adapt LLMs through generic pipelines** borrowed from other domains: shallow neighborhood sampling for context retrieval, standard supervised fine-tuning objectives, and uncalibrated inference at test time. This overlooks the unique characteristics of temporal relational reasoning, where answers often require multi-hop inference over time-evolving structure rather than surface pattern matching. As illustrated in Figure 1, models frequently favor entities that are lexically similar or locally frequent in the input even when the graph structure supports better completions.

Under a unified evaluation (Section 2), our analysis reveals three recurrent limitations. (1) *Shallow retrieval* misses multi-hop, time-aligned evidence, which is crucial when the gold entity is not observed in history. (2) *Standard fine-tuning* primarily rewards token correctness and reinforces memorization rather than relational compatibility, with sharp drops on queries that require generalization beyond observed patterns. (3) *Uncalibrated inference* produces contextually implausible entities, often deviating from history without improving accuracy. These limitations indicate that task-specific design is necessary for effective LLM-based TKG completion.

We present RECIPE-TKG, a parameter-efficient framework that addresses each limitation with a principled, stage-wise design. (1) **Rule-Based Multi-Hop (RBMH) sampling** enriches the retrieved history with structurally diverse, temporally aligned facts to improve multi-hop reachability. (2) **Contrastive Fine-Tuning (CFT)** augments next-token prediction with a relational compatibility objective over LoRA adapters, encouraging discrimination among plausible candidates rather than memorization of token patterns. (3) **Test-time semantic filtering** verifies contextual alignment during inference and refines low-alignment outputs, reducing off-context predictions without additional training.

Across four benchmarks, RECIPE-TKG improves accuracy and plausibility, with especially strong gains in short-history and non-historical settings, and achieves up to **22.4%** relative improvement in Hits@10 over prior LLM-based methods.

Contributions.

- We standardize evaluation to separate the effects of sampling, training, and inference, clarifying where reported gains originate.
- We provide a systematic characterization of failure modes in LLM-based TKG completion, centered on retrieval depth, supervision signal, and inference calibration.
- We introduce **RECIPE-TKG**, a task-specific, parameter-efficient framework whose stages directly target these limitations, yielding consistent gains across datasets and input regimes.

2 Unified Analysis of Failure Modes in LLM-based TKG Completion

Despite recent progress, LLMs adapted to TKG completion often default to surface patterns and fail when structural or temporal cues are indirect or require multi-hop reasoning. To guide design choices, we conduct a controlled re-evaluation of recent approaches (Lee et al., 2023; Liao et al., 2024) under a unified setup.

2.1 Grounding Predictions in Historical Context

Definition 2.1

A query's history is **sparse** when the retrieved context contains few and/or low-diversity facts. A prediction is **non-historical** if the gold entity does not appear in the retrieved history prior to the query time. The notions overlap but are not identical.

Figure 2(a–c) provides three empirical facts that drive our design. **(1) History length matters:** Hits@10 is below 0.3 with only one retrieved fact and exceeds 0.5 with 20–50 facts, and this trend holds for both ICL and SFT (Fig. 2a). **(2) Structure, not just tokens:** over 25% of targets require multi-hop reachability and about 4% are unreachable with shallow sampling (Fig. 2b), indicating that merely adding more one-hop facts is insufficient. **(3) Non-historical collapse:** while LLMs achieve 80–83% Hits@10 on historical cases, accuracy falls below 5% when the gold entity is unseen in history (Fig. 2c), revealing a reliance on lexical overlap and pattern recall.

Taken together, these results point to *depth and temporal alignment in history sampling*, rather than

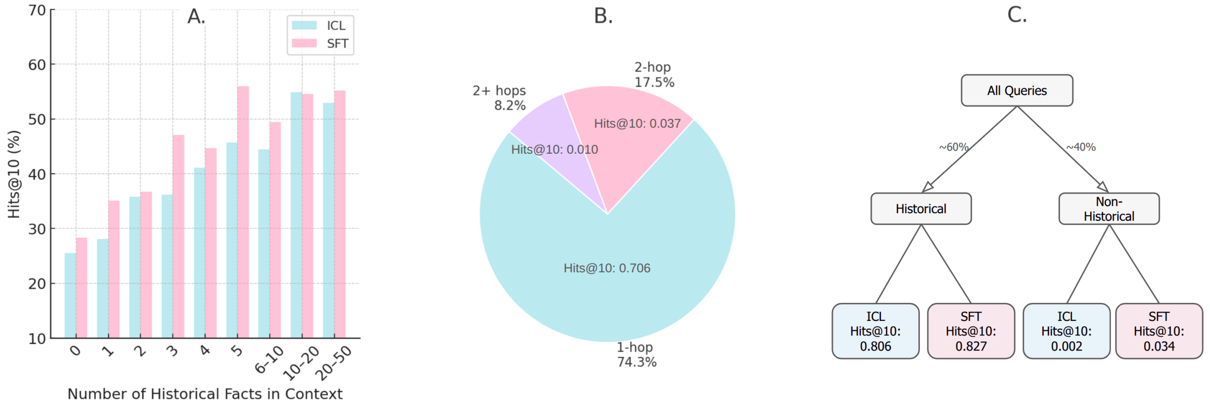


Figure 2: **Failures under short history and non-historical answers.** *Dataset: ICEWS14. Model: LLaMA-2-7B.* (a) Hits@10 vs. number of retrieved facts (history length): longer histories support better reasoning. (b) Share of queries by minimum hop distance from subject to gold entity: over 25% require multi-hop reachability. (c) Hits@10 by historical (gold seen in retrieved history) vs. non-historical (gold unseen) splits for ICL and SFT, showing a sharp drop in the latter.

longer but shallow context, as the primary driver of accuracy. They motivate a mitigation that (i) recovers multi-hop, time-aligned evidence and (ii) trains for *relational compatibility* beyond memorized associations. We operationalize this in Section 4 via multi-hop, graph-aware history sampling and CFT that encourages such compatibility.

2.2 Limitations of Supervised Fine-Tuning

Table 1: Re-evaluation of ICL and SFT using consistent decoding and evaluation. Gains largely stem from evaluation setup and history sampling; the marginal effect of SFT is smaller under a unified setup.

Method	Hits@1	Hits@3	Hits@10
Reported in GenTKG			
ICL (naive sampling + basic eval)	0.258	0.430	0.510
+ Fine-Tuning (TLR sampling + eval)	0.369	0.480	0.535
Re-evaluated under consistent setup			
ICL (naive sampling) + GenTKG eval	0.344	0.464	0.523
ICL (TLR sampling) + GenTKG eval	0.351	0.473	0.527
SFT (TLR sampling) + GenTKG eval	0.364	0.476	0.532

Supervised fine-tuning (SFT) is widely used to adapt LLMs to TKG tasks, and prior work such as GenTKG (Liao et al., 2024) reports notable improvements over prompting-based strategies (Lee et al., 2023). However, our re-evaluation under controlled conditions shows that much of this improvement originates not from fine-tuning itself, but from differences in sampling strategies and evaluation setups.

Evaluation Frameworks Explain Much of the Gap. LLMs produce open-ended text that requires careful postprocessing to extract valid entity predictions. While Lee et al. (2023) uses a basic evaluation setup, GenTKG applies a more refined pipeline with canonicalization and output filtering, making direct comparisons misleading.

To disentangle these effects, we re-evaluate prompting-based strategies and fine-tuned models with different sampling and evaluation pipelines under a unified framework. We compare naive sampling used in Lee et al. (2023) and TLR sampling (Liao et al., 2024), and two evaluation settings (basic eval and GenTKG eval (Liao et al., 2024)). As shown in Table 1, replacing the evaluation code alone increases Hits@1 from 25.8% to 34.4%. TLR sampling provides a modest improvement (35.1%) compared to one-hop sampling, while fine-tuning adds only a small additional gain (36.4%). This suggests that a large portion of the reported gain stems from implementation choices, not from the model's improved reasoning capabilities.

Fine-tuning alone does not fix generalization. As established in Section 2.1, both ICL and fine-tuned models struggle with non-historical predictions, where the correct answer does not appear in the retrieved history. These failures persist across a range of input sizes and are especially severe when the gold entity requires multi-hop reasoning, which is not supported by current sampling methods. Fine-tuning improves memorization of patterns seen during training but does not provide the relational inductive bias needed to reason about unseen or indirectly connected entities.

Motivating contrastive fine-tuning. We therefore supplement next-token prediction with a contrastive objective that explicitly separates plausible from implausible candidates conditioned on relations, encouraging compatibility-driven discrimination under sparse or indirect evidence (Section 4).

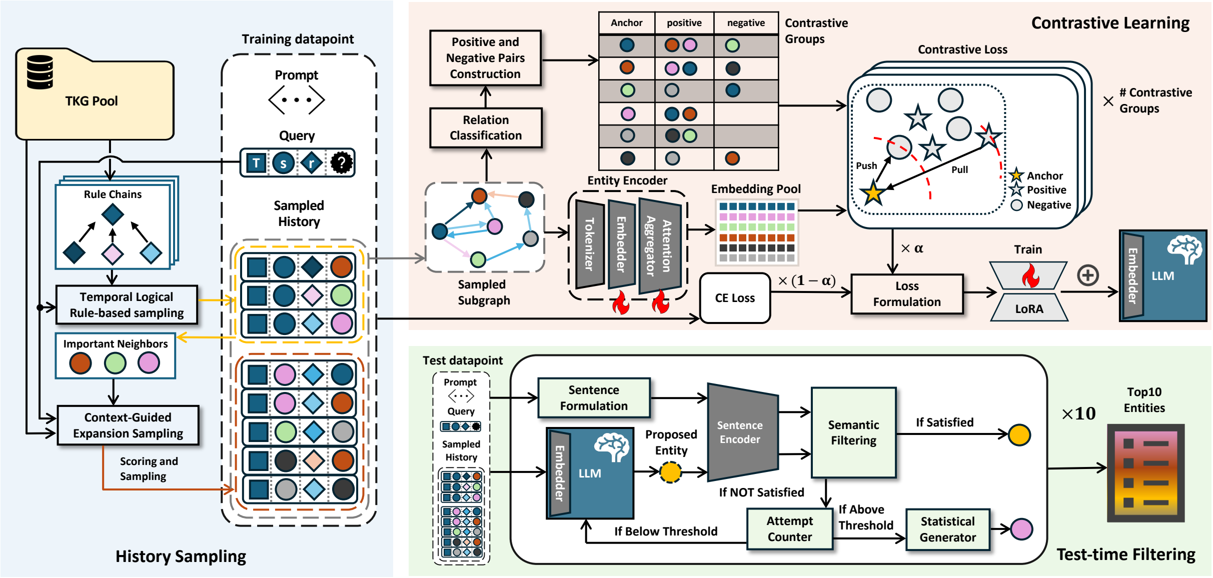


Figure 3: **Overview of RECIPE-TKG.** RECIPE-TKG follows a three-stage framework: (1) **History Sampling**, which retrieves query-relevant facts via a two-phase strategy combining rule-based retrieval and context-guided expansion; (2) **Contrastive Learning**, which jointly optimizes entity embeddings using contrastive and cross-entropy losses. Positive/negative pairs are sampled from the subgraph, and embeddings are generated via a learnable encoder; (3) **Test-time Filtering**, where predicted entities are iteratively verified by a semantic filter. Unsatisfactory outputs are refined using a statistical generator until confident predictions are obtained.

3 Preliminaries

Problem Formulation. A Temporal Knowledge Graph is a collection of time-stamped facts represented as quadruples (s, p, o, t), where s and o are subject and object entities, p is a relation, and t denotes the timestamp of the event. Formally, a TKG is denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{R}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V}$ is the set of entities, $\mathcal{R}$ the relations, $\mathcal{E}$ the event facts, and $\mathcal{T}$ the time indices. Each time step t defines a historical snapshot $\mathcal{G}_t \subseteq \mathcal{E}$. The forecasting task involves predicting a missing entity in a future quadruple. Given a query of the form $(s, p, ?, t)$ or $(?, p, o, t)$ and a set of historical snapshots $\{\mathcal{G}_1, \ldots, \mathcal{G}_{t-1}\}$, the model must return the most plausible entity that completes the query at time t.

Low-Rank Adaptation (LoRA) To reduce the number of trainable parameters, we adopt LoRA (Hu et al., 2022), which re-parameterizes the weight update as % %

$$\hat{h}(x) = W_0 x + ABx, \tag{1}$$

where W_0 is a frozen pretrained weight and A, B are trainable low-rank matrices.

4 Method

In this section, we present RECIPE-TKG, a three-stage LLM-based lightweight (see Appendix B) framework for temporal knowledge forecasting. The complete framework is illustrated in Figure 3.

4.1 RBMH: Rule-Based Multi-Hop History Sampling

The first stage of RECIPE-TKG focuses on retrieving a compact yet informative history from the temporal knowledge graph $\mathcal{G}$. For a given query $(s_q, p_q, ?, T)$, we aim to retrieve historical facts $\{(s, p, o, t) \in \mathcal{G} \mid t < T\}$ that are temporally valid and structurally relevant. Our sampling process combines rule-based retrieval with context-guided expansion to provide richer support for reasoning, particularly in sparse or non-historical settings.

Stage 1: Temporal Logical Rule-based Sampling. We begin by retrieving subject-aligned 1-hop facts using a rule-based procedure adapted from TLR (Liao et al., 2024), which learns relational rules of the form $p_q \Leftarrow \{p_{b_1}, \ldots, p_{b_k}\}$ through 1-step temporal random walks, capturing event regularities. We retrieve historical quadruples (s, p, o, t) such that $s = s_q$ and p appears in the rule body for the query relation p_q. See Appendix A.1 for the details.

However, this 1-hop retrieval cannot reach facts involving semantically relevant but structurally distant entities. Due to the fixed number of learned rules, this stage often retrieves fewer than N quadruples, the maximum the LLM can handle. This motivates a second stage to expand context with more diverse and informative facts.

Stage 2: Context-guided Multi-hop Expansion We then samples additional historical facts from $\mathcal{G}$. The candidate pool includes any quadruples not retrieved in Stage 1 whose subjects differ from s_q.

This stage is designed to support multi-hop reasoning by identifying facts that may not directly connect to the query subject but are structurally and semantically relevant. Each candidate (s, p, o, t) is assigned a composite weight:

$$w = w_n \cdot w_f \cdot (w_t + w_c + w_{cp}), \qquad (2)$$

where w_n downweights unreachable or distant nodes, w_f penalizes high-frequency triples, w_t prioritizes temporal recency, w_c favors co-occurrence with the query subject or relation, and w_{cp} reinforces connectivity with the initial TLR context.

To sample from candidate pool, We first select the top $10 \times M$ candidates by score to form a reduced pool, where M is the context window budget. From this pool, we sample M quadruples with probabilities proportional to their weights. This soft filtering strategy preserves diversity while prioritizing high-quality candidates, avoiding overreliance on only the highest-scoring facts. Our two-stage RBMH sampling method supports reasoning beyond immediate neighbors and avoids overfitting to shallow or overly common facts. The overall design motivation, formal definitions, hyperparameters and algorithms are provided in Appendix A.2.

4.2 Contrastive Fine-Tuning for Structured Reasoning

To improve generalization beyond memorized entity associations, we introduce a contrastive fine-tuning objective that supplements the standard next-token prediction loss, helping to disambiguate plausible from implausible predictions, especially when historical context is sparse or indirect.

Relation-Guided Contrastive Pair Construction. Our design is guided by the international relations principle, *The enemy of my enemy is my friend*, which reflects relational patterns common in geopolitical TKGs and motivates how we position entities in embedding space. Inspired by this structure, we first categorize relations into **positive**, **negative**, and **neutral** types using `GPT-4o`, minimizing the inclusion of neutral cases (see Appendix C.1). Given a sampled subgraph (Figure 3), we treat each unique entity as an anchor and examine its 1-hop neighbors. A neighbor is assigned as a *positive* sample if it connects via a positive relation, or a *negative* sample if it connects via a negative relation. If both types of edges exist, the neighbor is excluded to avoid contradiction. Neutral relations are ignored. This process forms contrastive groups that are used to calculate the contrastive loss.

Entity Embedding Encoding. Since an entity typically spans multiple tokens, we adopt a multi-stage process to compute its representation. First, the entity string is tokenized. Each resulting token is then passed through the model's embedding layer (embedder), which produces an embedding vector. These token embeddings $\{h_1, h_2, \ldots, h_k\}$ are subsequently aggregated into a single entity-level embedding e using a trainable **attention aggregator**.

The final embedding is a weighted sum:

$$e = \sum_{j=1}^{k} \lambda_j h_j, \qquad (3)$$

where λ_j are attention weights satisfying $\sum_j \lambda_j = 1$. Both the embedding layer and the aggregator are learnable modules, jointly optimized during fine-tuning.

Training Objective. The overall loss function is defined as:

$$\mathcal{L} = \alpha \cdot \mathcal{L}_{\text{contrastive}} + (1 - \alpha) \cdot \mathcal{L}_{\text{ce}}(o, o_p), \qquad (4)$$

where $\mathcal{L}_{\text{ce}}$ denotes the cross-entropy loss between the predicted token o_p and the ground truth o, $\mathcal{L}_{\text{contrastive}}$ represents the contrastive loss, and $\alpha \in [0, 1]$ is a balancing hyperparameter.

The contrastive loss is formulated as:

$$\mathcal{L}_{\text{contrastive}} = \frac{1}{N_c} \sum_{i=1}^{N_c} \max \Big(0,$$
$$\|a_i - pos_i\|^2 - \|a_i - neg_i\|^2 + m\Big) \qquad (5)$$

where N_c is the number of contrastive groups, and a_i denotes the embedding of the anchor entity. For each group, pos_i is the hardest positive, defined as the farthest positive entity from the anchor in the embedding space, while neg_i is the closest negative. This formulation emphasizes challenging examples and enforces a margin m to improve the separation between positive and negative pairs.

This training objective encourages the model to pull the most distant positive samples closer to the anchor and push the nearest negatives farther away. This dynamic adjustment refines the semantic structure of the latent space, enabling better entity discrimination and improving downstream reasoning performance. More details can be found at Appendix C.

4.3 Similarity-Based Test-Time Filtering

Recent work shows that language models can improve inference without parameter updates by using lightweight test-time strategies (Snell et al., 2024; Ji et al., 2025). Building on this idea, we introduce a semantic similarity-based filtering method to reduce hallucinations by removing predictions misaligned with the input context.

Our filtering approach is motivated by two empirical observations:

1. Models often generate non-historical entities that have low semantic alignment with the input context, especially in sparse settings despite higher similarity scores correlating with correctness (Figure 4).

2. In many cases, the ground truth entity already appears in the historical context $\mathcal{H}$, yet the model produces a non-historical prediction that yields negligible gain in accuracy.

These patterns suggest that enforcing semantic consistency and reconsidering salient entities from the input can correct many low-quality predictions. Rather than rejecting or reranking predictions with fixed rules, we apply an adaptive refinement strategy grounded in semantic similarity.

Semantic Consistency Verification. We embed the generated prediction p and the input context c using a sentence transformer model to compute a similarity score:

$$\phi(p, c) = \text{cos-sim}(E(p), E(c)) \tag{6}$$

$$E(x) = \text{SentenceTransformer}(x) \in \mathbb{R}^d \tag{7}$$

where $E(\cdot)$ denotes the output vector of a pretrained transformer model. We use this similarity as a proxy for contextual alignment. A prediction is accepted if its similarity score exceeds a learned threshold τ, or if it already appears in the retrieved history $\mathcal{H}$. Otherwise, we regenerate until a satisfactory prediction is found, or fall back to history-aware scoring.

This process is formalized as:

$$p' = \begin{cases} p & \text{if } p \in \mathcal{H} \text{ or } \phi(p, c) \geq \tau \\ \text{regenerate}(p) & \text{if } \phi(p, c) < \tau \\ \arg\max_{h \in \mathcal{H}} \psi(h) & \text{after } k \text{ attempts} \end{cases} \tag{8}$$

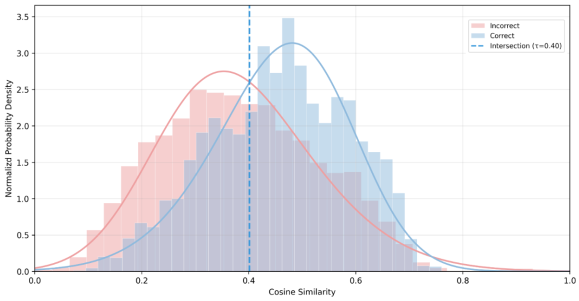


Figure 4: Distribution of semantic similarity values for correctly and incorrectly classified samples to the input context.

Figure 3 illustrates how filtering interacts with the generation process to improve robustness.

Historical Relevance Fallback. If repeated generations yield unsatisfactory predictions, we fall back to the historical candidates $\mathcal{H}$. Each candidate $h \in \mathcal{H}$ is scored by:

$$\psi(h) = \beta \cdot f(h) + (1 - \beta) \cdot r(h) \tag{9}$$

where $f(h)$ is the frequency of h in the input history and $r(h)$ captures recency. This mechanism biases the selection toward historically grounded entities when semantic alignment fails.

Threshold Selection. The threshold τ is optimized to best separate correct and incorrect predictions based on empirical distributions of $\phi(p, c)$. We describe the optimization objective and quantitative justification in Appendix D, along with implementation details and discuss its generalizability in Appendix E.

5 Experiments

5.1 Experimental Setup

Proposed method. We refer to our full method as RECIPE-TKG, which combines rule-based multi-hop history sampling (*RBMH Sampling*), contrastive fine-tuning denoted as *CFT*, and *Test-time Filtering*.

Language Models. Our primary experiments are conducted on `LLaMA-2-7B` (Touvron et al., 2023), a widely used open-source model in LLM-based TKG completion research (Liao et al., 2024; Luo et al., 2024). To ensure modern relevance, we also evaluate `LLaMA-3-8B` (Meta AI, 2024). Prompts and implementation details are provided in Appendix C.2 and C.4

Table 2: **Temporal link prediction results on temporal-aware filtered Hits@1/3/10.** LLM-based models are implemented based on `LLaMA2-7B`. Best results for each metric are highlighted in **bold**, and the best results among LLM-based models are underlined. The last row shows the relative improvement (Δ) of RECIPE-TKG over the best-performing LLM-based baseline.

	Datasets	ICEWS14			ICEWS18			GDELT			YAGO		
	Models	Hits@1	Hits@3	Hits@10	Hits@1	Hits@3	Hits@10	Hits@1	Hits@3	Hits@10	Hits@1	Hits@3	Hits@10
Embedding-based	RE-NET (Jin et al., 2020)	0.260	0.401	0.548	0.165	0.297	0.447	**0.117**	0.202	0.333	-	-	-
	RE-GCN (Li et al., 2021)	0.313	0.473	0.626	0.223	0.367	**0.525**	0.084	0.171	0.299	0.468	0.607	0.729
	xERTE (Han et al., 2020)	0.330	0.454	0.570	0.209	0.335	0.462	0.085	0.159	0.265	0.561	0.726	0.789
	TANGO (Han et al., 2021)	0.272	0.408	0.550	0.191	0.318	0.462	0.094	0.189	0.322	0.566	0.651	0.718
	Timetraveler (Sun et al., 2021)	0.319	0.454	0.575	0.212	0.325	0.439	0.112	0.186	0.285	0.604	0.770	0.831
Rule-based	TLogic (Liu et al., 2022)	0.332	0.476	0.602	0.204	0.336	0.480	0.113	**0.212**	**0.351**	0.638	0.650	0.660
LLM-based	CoH (Xia et al., 2024b)	0.242	0.397	0.512	0.168	0.282	0.427	-	-	-	-	-	-
	PPT (Xu et al., 2023)	0.289	0.425	0.570	0.169	0.306	0.454	-	-	-	-	-	-
	HFL (Xu et al., 2025)	0.277	0.427	0.573	0.178	0.304	0.455	-	-	-	-	-	-
	ICL (Lee et al., 2023)	0.344	0.464	0.523	0.164	0.302	0.382	0.090	0.172	0.242	0.738	0.807	0.823
	GenTKG (Liao et al., 2024)	0.364	0.476	0.532	0.200	0.329	0.395	0.099	0.193	0.280	0.746	0.804	0.821
	RECIPE-TKG	**0.393**	**0.526**	**0.651**	**0.224**	**0.369**	0.516	0.095	0.192	0.327	**0.811**	**0.880**	**0.930**
	Δ	8.0%	10.5%	22.4%	12.0%	12.2%	13.4%	-4.0%	-0.5%	16.8%	8.7%	9.0%	13.0%

Datasets. We evaluate RECIPE-TKG on four commonly adopted benchmark datasets: ICEWS14 and ICEWS18, both derived from the ICEWS project (Boschee et al., 2015), GDELT (Leetaru and Schrodt, 2013), and YAGO (Mahdisoltani et al., 2013). Detailed dataset statistics are provided in Appendix H.

Evaluation Metrics. We choose temporal-aware filtered Hits@1/3/10 as our evaluation metrics, following prior work (Gastinger et al., 2023).

Baselines. We compare RECIPE-TKG against three categories of methods. **Embedding-based methods** include RE-NET (Jin et al., 2020), RE-GCN (Li et al., 2021), xERTE (Han et al., 2020), TANGO (Han et al., 2021), and TimeTraveler (Sun et al., 2021). **Rule-based method** includes TLogic (Liu et al., 2022). **LLM-based methods** include ICL (Lee et al., 2023), GenTKG (Liao et al., 2024), PPT (Xu et al., 2023), CoH (Xia et al., 2024b), and HFL (Xu et al., 2025). Additional information about baselines are in Appendix G.

5.2 Main Results

Results in Table 2 show that RECIPE-TKG consistently performs well across four benchmarks, surpassing both embedding-based and LLM-based baselines on nearly all evaluation metrics. On ICEWS14 and YAGO, RECIPE-TKG establishes new state-of-the-art results among LLM-based methods, achieving up to **22.4%** relative improvement in Hits@10. On ICEWS18, it exceeds the best LLM-based baseline on all three metrics and is competitive with RE-GCN, the strongest embedding-based model. For GDELT, which is known to be noisy and dominated by repetitive event patterns (Trivedi et al., 2017; Li et al., 2021) with fine-grained 15-minute timestamps that favor symbolic rule chaining, frequent rules can be mined reliably and simple chains often suffice, explaining TLogic's advantage (Liu et al., 2022); nevertheless, RECIPE-TKG attains the highest Hits@10 (0.327) among LLM-based models and remains competitive on Hits@1 and Hits@3. These results highlight the effectiveness of RECIPE-TKG and position LLM-based methods as strong candidates for foundation models in TKG completion.

Table 3: **Ablation study on ICEWS14 with `LLaMA2-7B`.** Comparison of training paradigms across different history sampling strategies. The bold results show the original combinations of components in prior works and our method.

	ICL			SFT			CFT		
	H@1	H@3	H@10	H@1	H@3	H@10	H@1	H@3	H@10
Lee et al. (2023)	**0.344**	**0.464**	**0.523**	0.360	0.469	0.530	0.363	0.479	0.529
TLR (Liao et al., 2024)	0.351	0.473	0.527	**0.364**	**0.476**	**0.532**	0.367	0.476	0.532
RBMH	0.364	0.500	0.572	0.389	0.519	0.582	**0.392**	**0.521**	**0.580**

6 Analysis

6.1 Ablation Study

We conducted ablation studies to evaluate key components of our framework against prior works. We compare three sampling methods (Lee et al. (2023), TLR (Liao et al., 2024), and our *RBMH Sampling*) and three training paradigms (in-context learning, supervised fine-tuning, and contrastive fine-tuning) on ICEWS14 using `LLaMA2-7B`. As shown in Table 3, bold results indicate original combinations from prior works and RECIPE-TKG w/o filtering. The results show that *RBMH Sampling* consistently improves performance across all training paradigms by retrieving structurally diverse and se-

Table 4: Effect of removing RECIPE-TKG components.

SETTINGS	Hits@1	Hits@3	Hits@10
RECIPE-TKG w/o CFT	0.364	0.501	0.643
RECIPE-TKG w/o RBMH Sampling	0.364	0.483	0.581
RECIPE-TKG w/o Filtering	0.392	0.521	0.580
RECIPE-TKG	0.393	0.526	0.651

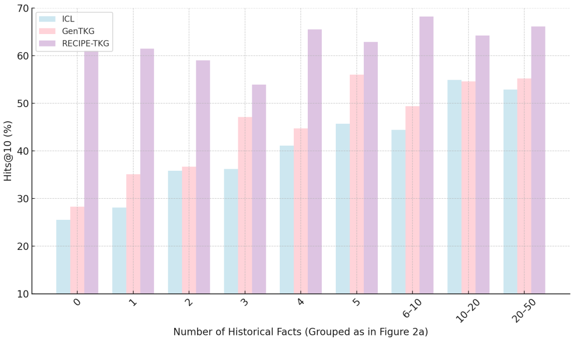


Figure 5: Hits@10 grouped by number of historical facts. RECIPE-TKG consistently outperforms ICL and GenTKG across all history lengths, with particularly strong improvements when the input history is sparse.

mantically relevant context. While *CFT* performs comparably to SFT with the same sampling strategy, it shows clear advantages when historical context is sparse. As discussed in Appendix I.1, contrastive models generate predictions semantically closer to the ground truth, even when exact matches aren't possible, promoting structure-aware generalization beyond surface-level accuracy, especially in sparse settings where lexical cues are insufficient.

Table 4 provides additional insights into the effects of each of the three components, especially *test-time filtering*. When comparing the CFT-RBMH setting with and without *Test-time Filtering*, we observe a substantial boost in Hits@10 from 0.580 to 0.651, underscoring the effectiveness of our test-time refinement mechanism. Notably, combining *test-time filtering* with *RBMH Sampling* and *Test-time Filtering* (**RECIPE-TKG**) yields the best performance across all metrics.

6.2 Performance Gains Across Input Regimes

To evaluate how historical input affects model performance, we group queries by the number of retrieved facts and compare Hits@10 across methods. These bins align with Figure 2(a), allowing direct comparison with prior failure patterns. As shown in Figure 5, RECIPE-TKG outperforms both ICL and GenTKG across all groups, with especially large gains in the low-history regime.

Table 5: Comparison between `LLaMA2-7B` and `LLaMA3-8B` on ICEWS14.

Model	LLaMA2-7B			LLaMA3-8B		
	hit@1	hit@3	hit@10	hit@1	hit@3	hit@10
ICL	0.344	0.464	0.523	0.351	0.484	0.578
RECIPE-TKG	0.393	0.526	0.651	0.367	0.529	0.658

Two key insights emerge. First, prior failures on short-history queries were not due to intrinsic difficulty, but rather to shallow retrieval. Since all methods are evaluated on the same query set, the strong gains from RECIPE-TKG (reaching over 60% Hits@10 for history length 0 to 2) indicate that even sparse queries can be completed accurately when provided with deeper, multi-hop context. This validates the effectiveness of *RBMH Sampling* in recovering structurally and temporally relevant support.

Second, RECIPE-TKG continues to outperform baselines even with longer histories (10–50 facts), where other methods begin to plateau. This sustained advantage reflects the contributions of *CFT* and *Test-time Filtering*, which improve generalization and reduce hallucinations.

Overall, these results show that RECIPE-TKG not only addresses the limitations of shallow context but also improves reasoning and prediction quality across a wide range of query types.

6.3 Case Study: Performance of Llama3-8b

As shown in Table 5, `LLaMA3-8B` performs comparably to `LLaMA2-7B`, supporting our choice of the latter for most experiments. Moreover, this choice of base model enables a fair comparison with prior work using fine-tuned models. Under both backbones, RECIPE-TKG consistently outperforms ICL, demonstrating its robustness and generalizability across different LLMs.

Lightweight design. We update only $\sim 0.81\%$ of `LLaMA-2-7B` (54.3M params), mine rules in <20s per dataset, and incur a 16.6% inference overhead from filtering (details in Appendix B).

7 Conclusion

We introduced RECIPE-TKG, a framework for LLM-based temporal knowledge graph forecasting that combines multi-hop sampling, contrastive fine-tuning, and semantic filtering. It delivers consistent accuracy gains, especially under sparse or indirect evidence in practice. By aligning context with relational structure and refining inference, RECIPE-TKG improves reasoning without large-scale retraining, validating a modular, temporally grounded design overall.

Limitations

Although RECIPE-TKG adopts a structured three-stage framework, it is still built on clean, fully observed temporal knowledge graphs, which may not reflect real-world scenarios. The rule mining step requires offline learning before sampling, and must be repeated if the TKG changes. Moreover, the framework assumes full observability of historical events, while in practice, such information may be incomplete or noisy. Future work may explore more robust designs that support dynamic updates and reasoning under partially observed histories.

License and Ethics

All datasets used in this study are publicly available and licensed for academic research. Specifically, ICEWS14, ICEWS18, GDELT, and YAGO have been widely adopted in prior work on temporal knowledge graphs. No personally identifiable information (PII) or sensitive content is present in any of the datasets.

We use LLaMA-2 and LLaMA-3 models under Meta's official research license, and all model adaptations are conducted in compliance with their intended use for academic and non-commercial research. The training and evaluation procedures are entirely conducted on benchmark data, and no human subjects are involved.

We adhere to the ethical guidelines set forth by the ACL Code of Ethics, including transparency, reproducibility, and the responsible use of language models. Our work poses minimal risk of harm and does not involve content generation, human annotation, or interaction with real users.

References

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. ICEWS Coded Event Data.

Rochana Chaturvedi. 2024. Temporal knowledge graph extraction and modeling across multiple documents for health risk prediction. In *Companion Proceedings of the ACM Web Conference 2024*, pages 1182–1185.

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. *arXiv preprint arXiv:2203.07993*.

Ambedkar Dukkipati, Kawin Mayilvaghanan, Naveen Kumar Pallekonda, Sai Prakash Hadnoor, and Ranga Shaarad Ayyagari. 2025. Predictive ai with external knowledge infusion for stocks. *arXiv preprint arXiv:2504.20058*.

Julia Gastinger, Timo Sztyler, Lokesh Sharma, Anett Schuelke, and Heiner Stuckenschmidt. 2023. Comparing apples and oranges? on the evaluation of methods for temporal knowledge graph forecasting. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 533–549. Springer.

Zhen Han, Peng Chen, Yunpu Ma, and Volker Tresp. 2020. Explainable subgraph reasoning for forecasting on temporal knowledge graphs. In *International Conference on Learning Representations*.

Zhen Han, Zifeng Ding, Yunpu Ma, Yujia Gu, and Volker Tresp. 2021. Learning neural ordinary equations for forecasting future links on temporal knowledge graphs. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8352–8364.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*.

Yixin Ji, Juntao Li, Hai Ye, Kaixin Wu, Kai Yao, Jia Xu, Linjian Mo, and Min Zhang. 2025. Test-time compute: from system-1 thinking to system-2 thinking. *Preprint*, arXiv:2501.02497.

Woojeong Jin, Meng Qu, Xisen Jin, and Xiang Ren. 2020. Recurrent event network: Autoregressive structure inference over temporal knowledge graphs. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6669–6683.

Dong-Ho Lee, Kian Ahrabian, Woojeong Jin, Fred Morstatter, and Jay Pujara. 2023. Temporal knowledge graph forecasting without knowledge using in-context learning. *Preprint*, arXiv:2305.10613.

Kalev Leetaru and Philip A Schrodt. 2013. Gdelt: Global data on events, location, and tone, 1979–2012. In *ISA annual convention*, volume 2, pages 1–49. Citeseer.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. 2022. Solving quantitative reasoning problems with language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, Online. Association for Computational Linguistics.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Huawei Shen, Yuanzhuo Wang, and Xueqi Cheng. 2021. Temporal knowledge graph reasoning based on evolutional representation learning. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 408–417.

Ruotong Liao, Xu Jia, Yangzhe Li, Yunpu Ma, and Volker Tresp. 2024. Gentkg: Generative forecasting on temporal knowledge graph with large language models. *Preprint*, arXiv:2310.07793.

Yushan Liu, Yunpu Ma, Marcel Hildebrandt, Mitchell Joblin, and Volker Tresp. 2022. Tlogic: Temporal logical rules for explainable link forecasting on temporal knowledge graphs. In *Proceedings of the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI)*, pages 4120–4127.

Ruilin Luo, Tianle Gu, Haoling Li, Junzhe Li, Zicheng Lin, Jiayi Li, and Yujiu Yang. 2024. Chain of history: Learning and forecasting with llms for temporal knowledge graph completion. *Preprint*, arXiv:2401.06072.

Farzaneh Mahdisoltani, Joanna Biega, and Fabian M Suchanek. 2013. Yago3: A knowledge base from multilingual wikipedias. In *CIDR*.

Shreyas Mangrulkar and 1 others. 2022. Peft: parameter-efficient fine-tuning. `https://github.com/huggingface/peft`. GitHub repository, accessed May 2025.

Johannes Messner, Ralph Abboud, and Ismail Ilkan Ceylan. 2022. Temporal knowledge graph completion using box embeddings. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pages 7779–7787.

Meta AI. 2024. Meta llama 3: Open foundation and fine-tuned chat models. `https://ai.meta.com/blog/meta-llama-3/`. Accessed: 2025-05-16.

Michael Schlichtkrull, Thomas N Kipf, Peter Bloem, Rianne Van Den Berg, Ivan Titov, and Max Welling. 2018. Modeling relational data with graph convolutional networks. In *European semantic web conference*, pages 593–607. Springer.

Sentence-Transformers. all-mpnet-base-v2. https://huggingface.co/sentence-transformers/all-mpnet-base-v2. Accessed: 2025-05-19.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Kaitao Song, Xu Tan, Tao Qin, Jianfeng Lu, and Tie-Yan Liu. 2020. Mpnet: Masked and permuted pre-training for language understanding. In *Advances in Neural Information Processing Systems*, volume 33, pages 16857–16867.

Haohai Sun, Jialun Zhong, Yunpu Ma, Zhen Han, and Kun He. 2021. Timetraveler: Reinforcement learning for temporal knowledge graph forecasting. *arXiv preprint arXiv:2109.04101*.

Hugo Touvron, Louis Martin, Kevin Stone, Abdullah Al-Dujaili, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, and 1 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Volker Tresp, Cristóbal Esteban, Yinchong Yang, Stephan Baier, and Denis Krompaß. 2015. Learning with memory embeddings. *arXiv preprint arXiv:1511.07972*.

Rakshit Trivedi, Hanjun Dai, Yichen Wang, and Le Song. 2017. Know-evolve: Deep temporal reasoning for dynamic knowledge graphs. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3462–3471. PMLR.

Shangshang Wang, Julian Asilis, Ömer Faruk Akgül, Enes Burak Bilgin, Ollie Liu, and Willie Neiswanger. 2025. Tina: Tiny reasoning models via lora. *arXiv preprint arXiv:2504.15777*.

Yuwei Xia, Ding Wang, Qiang Liu, Liang Wang, Shu Wu, and Xiao-Yu Zhang. 2024a. Chain-of-history reasoning for temporal knowledge graph forecasting. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 16144–16159, Bangkok, Thailand. Association for Computational Linguistics.

Yuwei Xia, Ding Wang, Qiang Liu, Liang Wang, Shu Wu, and Xiao-Yu Zhang. 2024b. Chain-of-history reasoning for temporal knowledge graph forecasting. In *Findings of the Association for Computational Linguistics: ACL 2024*.

Wenjie Xu, Ben Liu, Miao Peng, Xu Jia, and Min Peng. 2023. Pre-trained language model with prompts for temporal knowledge graph completion. *arXiv preprint arXiv:2305.07912*.

Wenjie Xu, Ben Liu, Miao Peng, Zihao Jiang, Xu Jia, Kai Liu, Lei Liu, and Min Peng. 2025. Historical facts learning from long-short terms with language model for temporal knowledge graph reasoning. *Information Processing & Management*, 62(3):104047.

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. *arXiv preprint arXiv:2408.06603*.

A Rule-Based Multi-Hop History Sampling Details

A.1 TLR Algorithm

Algorithm 1 shows the TLR retrieval procedure used in our framework, reproduced from (Liao et al., 2024).

Algorithm 1 TLR Retrieval

Input: Temporal knowledge graph $\mathcal{G}$, query $(s_q, r_q, ?, T)$, learned rules $\mathcal{TR}$

Output: A set of retrieved facts $\mathcal{G}_{s_q}(s_q, r_q, T)$

```
$\mathcal{G}_{s_q}(s_q, r_q, T) \leftarrow \emptyset$
for $fact \leftarrow (s_q, r_q, o, t < T)) \in \mathcal{G}$ do
    $\mathcal{G}_{s_q}(s_q, r_q, T) \leftarrow \mathcal{G}_{s_q}(s_q, r_q, T) \cup fact$
end for
for top k rules $w.r.t.\ r_q \leftarrow r_b \in \mathcal{TR}$ do
    Get a list $r_b \leftarrow \{r_{b_1}, r_{b_2}, \cdots, r_{b_k}\}$
end for
for $fact \leftarrow (s_q, r \in r_b, o, t < T) \in \mathcal{G}$ do
    $\mathcal{G}_{s_q}(s_q, r_q, T) \leftarrow \mathcal{G}_{s_q}(s_q, r_q, T) \cup fact$
end for
return $\mathcal{G}_{s_q}(s_q, r_q, T)$
```

A.2 Context-guided Multi-hop Expansion Details

A.2.1 Weight Formulation Discussion

We adopt a multiplicative combination of the weight components rather than a simple sum to for two reasons. First, the neighbor weight w_n acts as a hard constraint: it equals zero if the subject or object of a candidate quadruple is not reachable from the query, effectively filtering out irrelevant facts. Second, the frequency weight w_f is designed to down-weight commonly repeated triples while preserving their relative order. This logarithmic scaling ensures that rare but structurally relevant facts are not overshadowed. Together, the multiplicative form enables a soft prioritization across dimensions while preserving hard structural constraints.

A.2.2 Weight Component

The five weight components of equation 2 are defined as follows:

Neighbor weight w_n ensures that structurally closer quadruples receive higher scores:

$$w_n = \exp\left(-\gamma_1 \cdot (\text{hop}_s + \text{hop}_o - 1)\right),$$

where hop_s and hop_o denote the shortest hop distances from the subject and object to the query subject. The weight decays exponentially with increasing distance, and vanishes to zero when either hop_s or hop_o is infinite, corresponding to cases where the entity is not reachable from the query subject in the graph. Importantly, all structural statistics (e.g., hop distance, co-occurrence counts, and context connectivity) are computed over the subgraph excluding quadruples with timestamps after the query time T.

Frequency weight w_f reduces the dominance of frequent triples (history quadruples excluding timestamp):

$$w_f = \frac{1}{\gamma_2 \cdot \log(n_{spo}) + 1},$$

where n_{spo} is the count of the subject-predicate-object triple. This logarithmic form discourages over-sampling of repetitive patterns while maintaining frequency order.

More precisely, for any two triples with frequency counts $n_1 < n_2$, the corresponding weights satisfy:

$$w(n_1) > w(n_2), \quad \text{and} \quad \frac{w(n_1)}{w(n_2)} = \frac{\log(n_2) + 1}{\log(n_1) + 1},$$

assuming all other components of the weight function are equal. This shows that the multiplicative formulation preserves the relative ranking induced by frequency, while still suppressing the absolute dominance of highly frequent triples.

Time weight w_t favors temporally recent events:

$$w_t = \exp\left(-\gamma_3 \cdot \frac{T - t}{\delta}\right),$$

where T is the timestamp of the query, t is the timestamp of the event quadruple (with $T > t$), δ is the time granularity (e.g., $\delta = 24$ in ICEWS14), and γ_3 controls the decay rate.

Connection weight w_c promotes inclusion of frequently co-occurring entity pairs:

$$w_c = \frac{\log(1 + \gamma_4 \cdot n_{so})}{1 + \log(1 + \gamma_4 \cdot n_{so})},$$

where n_{so} is the co-occurrence count of the subject-object pair prior to T, and γ_4 is a smoothing parameter. This bounded function emphasizes structural relevance while limiting hub bias.

Contextual priority weight w_{cp} encourages sampling quadruples that remain connected to the initial TLR sampled subgraph:

$$w_{cp} = \begin{cases} 1, & \text{if } s \in \mathcal{E}_{\text{TLR}} \text{ or } o \in \mathcal{E}_{\text{TLR}}, \\ 0, & \text{otherwise,} \end{cases}$$

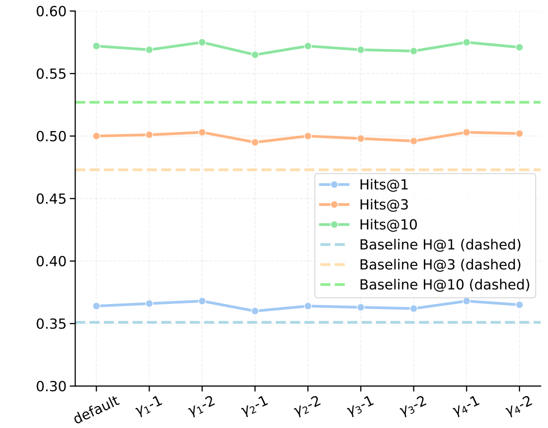


Figure 6: Performance of ICL-RBMH under different sampling hyperparameter configurations.

where $\mathcal{E}_{\text{TLR}}$ is the set of all 1-hop neighbors identified in the TLR stage. This guides the expansion toward semantically coherent subgraphs.

A.2.3 Hyperparameter Sensitivity Experiment

Figure 6 presents the performance in ICL-RBMH setting under varying sampling hyperparameters. We perturb each of the four γ_i parameters individually (two settings per parameter), while keeping others fixed, and compare them against the default configuration. Across all variants, model performance remains stable, indicating that *RBMH Sampling* is robust to hyperparameter choices. Moreover, ICL-RBMH consistently outperforms the baseline ICL-TLR across all settings.

The sampling hyperparameter configurations and their corresponding performance metrics are summarized in Table 6, including mean and standard deviation to reflect stability.

Table 6: Performance of ICL-RBMH under different sampling hyperparameter configurations on ICEWS14.

ID	γ_1	γ_2	γ_3	γ_4	Hits@1	Hits@3	Hits@10
default	0.6	0.6	0.01	0.1	0.364	0.500	0.572
γ_1-1	0.4	0.6	0.01	0.1	0.366	0.501	0.569
γ_1-2	0.8	0.6	0.01	0.1	0.368	0.504	0.575
γ_2-1	0.6	0.4	0.01	0.1	0.364	0.500	0.572
γ_2-2	0.6	0.8	0.01	0.1	0.364	0.500	0.572
γ_3-1	0.6	0.6	0.05	0.1	0.363	0.498	0.569
γ_3-2	0.6	0.6	0.002	0.1	0.368	0.506	0.573
γ_4-1	0.6	0.6	0.01	0.2	0.368	0.503	0.575
γ_4-2	0.6	0.6	0.01	0.05	0.365	0.502	0.571
	Mean				0.366	0.501	0.571
	Std				0.0020	0.0024	0.0021
Baseline (ICL-TLR)					0.351	0.473	0.527

A.3 RBMH Algorithm

Algorithm 2 Rule-based Multi-hop history sampling

Input: Temporal knowledge graph $\mathcal{G}$, query $(s_q, r_q, ?, T)$, learned rules $\mathcal{TR}$, maximum history length N, scoring function $\mathcal{F}$, a set of TLR retrieved facts $\mathcal{G}_{s_q}(s_q, r_q, T)$

Output: A set of retrieved facts $\mathcal{G}(s_q, r_q, T)$

```
$M \leftarrow N - len(\mathcal{G}_{s_q}(s_q, r_q, T))$
**if** $M = 0$ **then**
    $\mathcal{G}(s_q, r_q, T) \leftarrow \mathcal{G}_{s_q}(s_q, r_q, T)$
    **return** $\mathcal{G}(s_q, r_q, T)$
**end if**
$\mathcal{C} \leftarrow \{(s, r, o, t, \mathcal{F}(s, r, o, t)) \mid (s, r, o, t) \in \mathcal{G},\ t < T\}$
$\mathcal{C}_{\text{top}} \leftarrow \text{Top}_{10M}(\mathcal{C})$
$\mathcal{C}_{\text{sample}} \leftarrow \text{WeightedSample}(\mathcal{C}_{\text{top}},\ M)$
$\mathcal{G}_{\text{mh}}(s_q, r_q, T) \leftarrow \{(s, r, o, t) \mid (s, r, o, t, w) \in \mathcal{C}_{\text{sample}}\}$
$\mathcal{G}(s_q, r_q, T) \leftarrow \mathcal{G}_{s_q}(s_q, r_q, T) \cup \mathcal{G}_{\text{mh}}(s_q, r_q, T)$
**return** $\mathcal{G}(s_q, r_q, T)$
```

B Computational Efficiency Analysis

RECIPE-TKG is designed to be parameter-efficient and computationally lightweight while maintaining strong performance. This section quantifies various aspects of efficiency in our framework.

Parameter Efficiency Our framework fine-tunes a small fraction of the total parameters in the base LLM. For `LLaMA2-7B`, we update only LoRA adapters (with rank 8, applied to query and value projections across 32 transformer layers) and a self-attention pooling module for entity embedding aggregation. The trainable parameter count is approximately 54.3M, which constitutes just 0.81% of the base model's 6.74B parameters. This parameter-efficient design enables effective fine-tuning while keeping most of the pre-trained knowledge intact.

Rule Mining Efficiency The temporal logical rule mining process in our RBMH sampling strategy is highly efficient. Table 7 shows the time required for rule extraction across all datasets using 15 CPU processes (averaged over 5 runs). The process completes in under 20 seconds even for the largest dataset, representing negligible computational overhead. Furthermore, the extracted rules capture persistent temporal patterns and are not highly sensitive to minor dataset changes, allowing for infrequent updates when the knowledge graph evolves.

Table 7: Rule mining time across datasets (in seconds).

Dataset	ICEWS14	ICEWS18	GDELT	YAGO
Time (s)	6.89 ± 0.08	16.72 ± 0.07	10.78 ± 0.08	2.73 ± 0.02

Training Overhead Table 8 compares training time per epoch between standard supervised fine-tuning and our contrastive fine-tuning on 1024 samples. The contrastive objective introduces no additional training time, demonstrating its computational efficiency despite the improved semantic learning.

Table 8: Training time per epoch on 1024 samples.

Training Mode	Time (s)	Δ%
Fine-tuning (FT)	824.31	-
FT + Contrastive Loss	821.51	-0.34%

Inference Overhead Table 9 quantifies the runtime impact of our test-time filtering mechanism. On 1,000 test samples, filtering increases inference time by 16.6%, which is reasonable considering the consistent performance improvements in Hits@10 across all datasets. The filtering step provides a favorable trade-off between computational cost and accuracy gain.

Table 9: Inference time on 1,000 samples.

Setting	Time (s)	Δ%
No filtering	2316.48	-
With filtering	2700.67	+16.60%

C Training Details

C.1 Relation Classification

The prompt used for relation classification is provided in Figure 7.

In cases where a neighbor is connected to the anchor via both a positive and a negative relation, it is excluded in training to avoid ambiguity.

Figure 8 shows the distribution of relation types across four datasets. Positive and negative relations appear in roughly balanced proportions, while neutral relations are consistently less common. Notably, YAGO exhibits a distinct relation distribution where the majority of relations are classified as *neutral*. Upon inspection, we find that this reflects the actual semantic nature of the relations in the dataset, which are mostly descriptive or taxonomic rather than sentiment-oriented. Consequently, the contrastive learning component has limited impact on YAGO, as it relies on meaningful distinctions between positive and negative relations. The observed performance gain on YAGO is therefore primarily attributed to improvements in history sampling and *Test-time filtering*.

C.2 Prompt

To guide the language model in performing temporal knowledge completion, we adopt a structured, instruction-style prompt format shown in Figure 9. The prompt defines the task explicitly: given a chronological list of historical events represented as quadruples, the model must predict the missing object entity for a future temporal query.

Each historical fact is formatted as `{time}:[{subject}, {relation}, {object_label}.{object}]` where `{object_label}` is a unique identifier associated with the entity (e.g., `3380.Joseph_Robinette_Biden`). This labeling scheme facilitates consistent reference resolution and improves post-processing via regex-based extraction. The final input ends with the query, and the model is asked to generate the correct object in fully qualified form `{object_label}.{object}`.

This prompt format is applied consistently across both in-context learning and fine-tuning setups.

C.3 LoRA Formulation

We follow the standard LoRA setup (Hu et al., 2022). Given a frozen pretrained weight matrix $W_0 \in \mathbb{R}^{d\times k}$, LoRA introduces two trainable low-rank matrices $A \in \mathbb{R}^{d\times r}$ and $B \in \mathbb{R}^{r\times k}$ with $r \ll \min(d, k)$, such that the original forward transformation $h(x) = W_0 x$ is modified as:

$$\hat{h}(x) = W_0 x + ABx. \qquad (10)$$

This design allows efficient fine-tuning by only training A and B, while keeping the pretrained weights W_0 frozen. In our experiments, we adopt the default LoRA implementation from the PEFT library (Mangrulkar et al., 2022).

C.4 Implementation Details

We fine-tune `LLaMA-2-7B` and `LLaMA-3-8B` models using LoRA adapters. All trainings are conducted

Prompt for Relation Classification

You are analyzing relation labels from a political event knowledge graph, where each relation reflects an action or request within a geopolitical context.
Classify the sentiment of the given relation as one of the following:

- **positive** (e.g., promoting peace, aid, cooperation)
- **negative** (e.g., violence, repression, aggression)
- **neutral** (e.g., procedural or ambiguous actions)

Avoid selecting `"neutral"` unless the relation is genuinely ambiguous or purely procedural in nature.

Figure 7: Prompt used for relation classification.

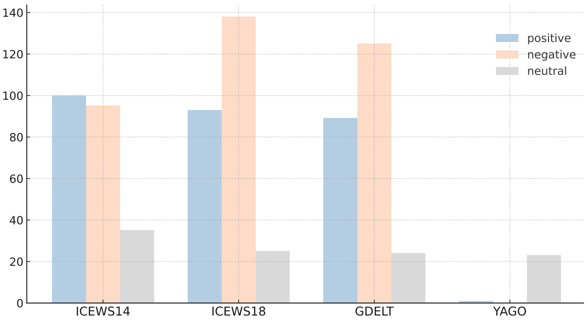


Figure 8: Distribution of relation types in four datasets after automatic classification.

on 2 H100 GPUs in `bfloat16` precision. We set maximum history length to 50 in history sampling according to the context length of `LLaMA-2-7B`. For fine-tuning, we train 1024-shots data for 50 epochs with the batch size of 512, the learning rate of 3e-4, the context length of 4096, the target length of 128, the LoRA rank of 8, the LoRA dropout rate of 0.05. For RECIPE-TKG, we train 6024-shots data (1024 aligned with GenTKG and 5000 randomly sampled by seed 42) for 10 epochs, and other settings keep unchanged. Contrastive tuning uses a margin of 1.0 and loss weight $\alpha = 0.2$ to balance cross-entropy and contrastive objectives.

Entities are tokenized using the native tokenizer of the LLM and embedded via the model's embedding layer. A lightweight attention aggregator produces final entity embeddings, jointly trained with the model.

C.5 Hyperparameter Sensitivity Experiment

As shown in Figure 10, varying α from 0.2 to 0.8 leads to marginal fluctuations across all evaluation metrics. These results suggest that the model is robust to the choice of α, and that *CFT* contributes consistently across a wide range of weighting schemes. Table 10 presents the sensitivity of model performance to the contrastive weight α. The consistently small standard deviations across metrics suggest that the model is robust to variations in α.

Table 10: Performance under different contrastive weight settings on ICEWS14.

Weight α	Hits@1	Hits@3	Hits@10
0.2	0.392	0.521	0.580
0.5	0.389	0.521	0.579
0.8	0.392	0.520	0.576
Mean	0.391	0.521	0.578
Std	0.0014	0.0006	0.0020

D Test-Time Filtering

Embedding Model. To compute semantic similarity between predictions and context, we use the `all-mpnet-base-v2` model (Song et al., 2020; Sentence-Transformers) from HuggingFace, a pretrained sentence transformer with 768-dimensional output. We treat both the generated prediction string and the full in-context prompt as input sequences and extract mean-pooled embeddings for similarity calculation.

Similarity Distribution Analysis. We analyze the cosine similarity $\phi(p, c)$ between prediction and context across 7,371 test samples from ICEWS14 using the contrastively tuned model. The average similarity score for correct predictions exceeds that of incorrect ones by $\Delta\mu = 0.057$. This supports

Prompt Example

You must be able to correctly predict the next `{object}` from a given text consisting of multiple quadruplets in the form of "`{time}:[{subject}, {relation}, {object_label}.{object}]`" and the query in the form of "`{time}:[{subject}, {relation},`" in the end. You must generate `{object_label}.{object}`.

```
2014-01-15: [Mehmet_Simsek, Make_statement, 5195.Other_Authorities_(Turkey)]
2014-01-20: [Nuri_al-Maliki, Consult, 3380.Joseph_Robinette_Biden]
2014-01-25: [Joseph_Robinette_Biden, Make_an_appeal, 3990.Massoud_Barzani]
2014-02-01: [Joseph_Robinette_Biden, Make_an_appeal_or_request,
```

Figure 9: Instruction-style prompt format for TKG forecasting.

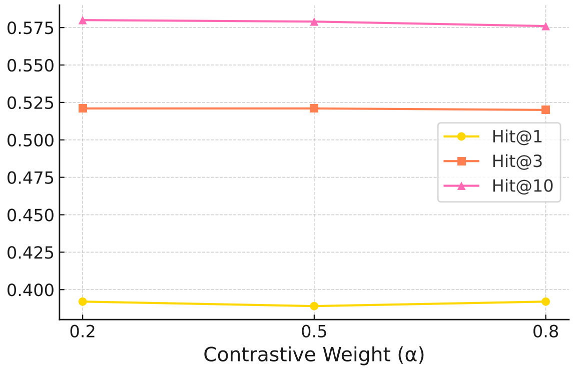


Figure 10: Effect of contrastive weight (α)

our assumption that similarity can serve as a proxy for semantic plausibility.

Novelty vs. Utility. We further observe that:

- 9.1% of predictions are non-historical despite the gold answer being present in $\mathcal{H}$.
- Among all non-historical predictions, only 1.5% are correct and improve Hits@10.

These findings indicate that many model generations deviate from the historical context unnecessarily and fail to yield substantial gains. They motivate fallback to more salient entities when regeneration fails.

Threshold Optimization. The optimal threshold τ^* is learned by maximizing separation between correct ($\mathcal{C}$) and incorrect ($\mathcal{I}$) prediction similarities:

$$\tau^* = \arg\max_{\tau} \left[F_{\mathcal{C}}(\tau) - F_{\mathcal{I}}(\tau)\right] \tag{11}$$

where F is the empirical CDF of cosine similarity values over samples from $\mathcal{C}$ and $\mathcal{I}$.

Fallback Scoring. If generation fails after k iterations (we use $k = 1$), the model selects a final answer from $\mathcal{H}$ using:

$$f(h) = \frac{\text{count}(h)}{|\mathcal{H}|}, \tag{12}$$

$$r(h) = 1 - \frac{\text{pos}(h)}{|\mathcal{H}|}, \tag{13}$$

$$\psi(h) = \beta \cdot f(h) + (1 - \beta) \cdot r(h), \tag{14}$$

where $\text{pos}(h)$ denotes the rank of h in its occurrence order. We set $\beta = 0.6$ in all experiments.

We compute cosine similarities between predicted entities and prompt context using the `all-mpnet-base-v2` sentence transformer from HuggingFace. The threshold τ^* is tuned on a development set by maximizing the separation between correct and incorrect predictions.

Figure 11 examines the effect of the semantic filtering threshold τ. As the threshold increases, Hits@10 improves, peaking near $\tau = 0.6$. Always falling back to historical entities ($\tau = 1.0$) slightly increases accuracy at the cost of exploration and computational efficiency. Threshold $\tau = 0.6$ balances correction with flexibility, enabling the model to revise low-quality outputs without overconstraining its generation space.

E Cross-Dataset Filtering Performance

To evaluate the robustness and generalization capability of our test-time filtering approach, we analyze its performance across all four benchmark datasets. While the filtering mechanism was introduced primarily to reduce hallucinations in open-ended generation, an important question is whether this component generalizes well across different temporal knowledge domains or if its effectiveness is dataset-dependent.

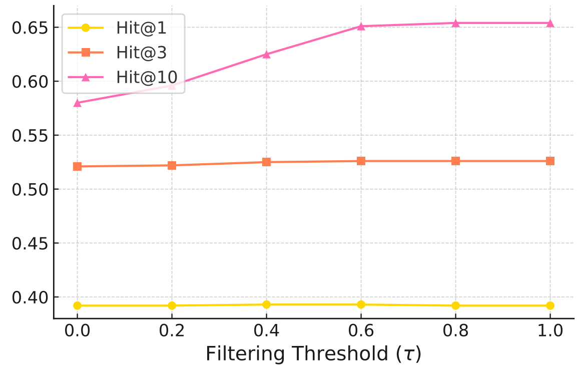


Figure 11: Effect of filtering threshold (τ)

Table 11: Effect of filtering across datasets.

Method	Hits@1	Hits@3	Hits@10
ICEWS14			
RECIPE-TKG	0.393	0.526	0.651
RECIPE-TKG w/o Filter	0.392	0.521	0.580
ICEWS18			
RECIPE-TKG	0.224	0.369	0.516
RECIPE-TKG w/o Filter	0.242	0.382	0.437
GDELT			
RECIPE-TKG	0.095	0.192	0.327
RECIPE-TKG w/o Filter	0.092	0.189	0.266
YAGO			
RECIPE-TKG	0.811	0.880	0.930
RECIPE-TKG w/o Filter	0.759	0.822	0.842

Table 11 shows the impact of our similarity-based filtering module across all datasets by comparing the full RECIPE-TKG framework against a variant without filtering. The filtering module consistently improves Hits@10 across all datasets, with gains ranging from 7.1 percentage points (ICEWS14) to 9.4 percentage points (GDELT). Most notably, on the YAGO dataset, the filtering mechanism substantially improves performance across all metrics (Hits@1/3/10), suggesting particular effectiveness on datasets with more descriptive entities and varied relation types.

These results demonstrate that the filtering mechanism's effectiveness is not dependent on dataset-specific properties, but rather reflects a general principle: by enforcing semantic consistency between predictions and input context, we can enhance model performance across diverse temporal knowledge domains. The observed consistency suggests that contextual alignment serves as a reliable signal for identifying and correcting implausible outputs, regardless of the specific entities and relations involved.

F Baseline Model Details

We compare RECIPE-TKG against several baseline methods that reflect the dominant modeling paradigms for TKG forecasting. Embedding-based methods include RE-GCN (Li et al., 2021), which applies relational graph convolutions to timestamped graph snapshots; xERTE (Han et al., 2020), which combines subgraph sampling and path-based reasoning using attention for explainability; TANGO (Han et al., 2021), which uses neural ODEs to learn continuous-time entity embeddings; and TimeTraveler (Sun et al., 2021), which employs reinforcement learning to explore multi-hop temporal paths. Rule-based method includes TLogic (Liu et al., 2022) relies on extracted symbolic rules for forecasting. The results of these models are derived from Liao et al. (2024)

We also replicate two recent LLM-based methods. ICL (Lee et al., 2023) applies in-context learning by prepending historical quadruples to a query and using greedy decoding with a regex-based answer extraction. GenTKG (Liao et al., 2024) performs parameter-efficient fine-tuning with LoRA adapters, and combines this with a rule-based history sampling module. We use their official codebases and replicate their evaluation pipelines for fair comparison.

G Baseline Model Details

We compare RECIPE-TKG against several baseline methods that reflect the dominant modeling paradigms for TKG forecasting.

Embedding-based methods include RE-GCN (Li et al., 2021), which applies relational graph convolutions to timestamped graph snapshots; RE-NET (Jin et al., 2020), which applies R-GCN (Schlichtkrull et al., 2018) for message passing for each snapshot and then uses temporal aggregation across multiple snapshots; xERTE (Han et al., 2020), which combines subgraph sampling and path-based reasoning using attention for explainability; TANGO (Han et al., 2021), which uses neural ODEs to learn continuous-time entity embeddings; and TimeTraveler (Sun et al., 2021), which employs reinforcement learning to explore multi-hop temporal paths.

Rule-based method TLogic (Liu et al., 2022) relies on extracted symbolic rules for forecasting.

LLM-based methods We implement several recent LLM-based approaches. ICL (Lee et al., 2023) applies in-context learning by prepending historical quadruples to a query and using greedy decoding with regex-based answer extraction. GenTKG (Liao et al., 2024) performs parameter-efficient fine-tuning with LoRA adapters, combined with rule-based history sampling. PPT (Xu et al., 2023) converts quadruples into natural language prompts and uses masked token prediction to leverage semantic information from pretrained language models. CoH (Xia et al., 2024b) explores high-order histories step-by-step to better utilize richer historical information for LLM reasoning. HFL (Xu et al., 2025) learns from historical facts across different time periods through a multi-perspective sampling strategy that focuses on mining relational associations. We use official codebases where available and replicate evaluation pipelines for fair comparison.

Note on embedding-based baselines Several specialized embedding models for TKG completion (e.g., RotateQVS (Chen et al., 2022), BoxTE (Messner et al., 2022), CGE (Ying et al., 2024)) have shown strong performance but are excluded from our main evaluation for three reasons. First, they use different dataset splits (e.g., ICEWS14 with 72,826/8,941/8,963 train/valid/test samples vs. our 74,845/8,514/7,371 split). Second, embedding methods require task-specific mathematical engineering, limiting cross-dataset generalizability, while LLM-based approaches benefit from pre-trained knowledge and adaptability. Third, there has been limited direct comparison between these paradigms in the literature. We include only embedding-based methods using consistent dataset splits for meaningful comparison.

H Dataset Statistics

We use four standard temporal knowledge graph benchmarks. ICEWS14 and ICEWS18 are subsets of the Integrated Crisis Early Warning System, containing geopolitical event records with daily granularity. GDELT provides global political event data, filtered to the most frequent events for tractability. YAGO consists of curated facts from a multi-year period. The statistics for these datasets are provided in Table 12.

I More Analysis

I.1 Analysis of Contrastive Fine-Tuning

To complement the ablation results in Section 6.1, we analyze how contrastive fine-tuning affects model behavior in low-history regimes—settings where standard exact-match metrics such as Hits@k may fail to capture the semantic relevance of model predictions.

Setup. We group ICEWS14 test samples by history length and compute the semantic distance between each model prediction and the gold entity. We compare three supervision settings: ICL, SFT, and contrastive FT, all evaluated under the same TLR history sampling.

We define semantic distance using cosine similarity between predicted and gold entities in a sentence embedding space:

$$\phi(p, o) = 1 - \text{cos-sim}(E(p), E(o)), \qquad (15)$$

where $E(\cdot)$ denotes the sentence transformer used in Section 4.3. Lower ϕ indicates higher semantic alignment, even if the prediction does not exactly match the gold entity.

Contrastive Tuning Improves Semantic Grounding. Figure 12 plots the semantic distance $\phi(p, o)$ against the retrieved history length. All models show the expected trend: greater history generally yields predictions closer to the gold entity in embedding space. However, the distinction between supervision strategies becomes clear in low-history regimes. In the encircled region (history length ≤ 3), contrastive fine-tuning produces fewer high-distance predictions than both ICL and SFT. This demonstrates that contrastive learning enhances the model's ability to infer plausible entities even when the input lacks strong historical evidence.

Multi-hop Sampling Further Stabilizes Model Behavior. To examine how our sampling strategy affects model reasoning on sparse-history inputs, we repeat the same experiment using our proposed *RBMH Sampling*. For comparability, we compute semantic distances on the same subset of samples originally identified as short-history under TLR.

As shown in Figure 13, contrastive-tuned models under *RBMH Sampling* exhibit more uniform semantic behavior across history lengths. Unlike the steep drop-off observed under TLR, the semantic distance remains relatively stable, indicating that many samples previously limited by shallow

Table 12: Dataset statistics used in our experiments. Time granularity varies by dataset and influences temporal resolution.

Dataset	#Train	#Valid	#Test	#Entities	#Relations	Time Gap
ICEWS14	74,845	8,514	7,371	7,128	230	1 day
ICEWS18	373,018	45,995	49,545	23,033	256	1 day
GDELT	79,319	9,957	9,715	5,850	238	15 mins
YAGO	220,393	28,948	22,765	10,778	24	1 year

context can now be grounded through richer structural and temporal cues. This supports our motivation in Section 2.1: one-hop sampling often fails to provide the necessary relational evidence, and multi-hop expansion is essential for enabling reliable reasoning, rather than the test instances being inherently harder.

Qualitative Support. Figure 14 presents qualitative examples where contrastive-tuned models produce predictions that are not exact matches but remain relationally and contextually appropriate. In contrast, ICL and SFT often produce surface-level or unrelated completions. These examples, paired with the distributional evidence above, underscore how contrastive fine-tuning improves semantic generalization and interpretability, particularly when Hits@k offers limited signal.

Case Study. To better understand the behavior of RECIPE-TKG, we provide a case study comparing the top-10 predictions of four methods on a specific query. The ground-truth object is `High_Ranking_Military_Personnel_(Nigeria)`, which is not explicitly present in the history. As shown in Figure 15, none of the models are able to perfectly predict the correct entity. However, the predictions made by RECIPE-TKG models are clearly more semantically aligned with the ground truth. For example, predictions such as `Military_(Nigeria)` and `Defense_Personnel_(Nigeria)` closely approximate the true answer in meaning, whereas other models (ICL and GenTKG) fail to capture such relevant semantics. This demonstrates the advantage of contrastive fine-tuning in shaping the embedding space, allowing the model to produce more relationally compatible predictions even when exact matches are not observed in history.

J Use of AI Tools

AI assistants were used to support writing (e.g., phrasing suggestions) and code generation (e.g., syntax templates). All such outputs were subject to thorough human verification, and the authors remain fully responsible for the content presented.

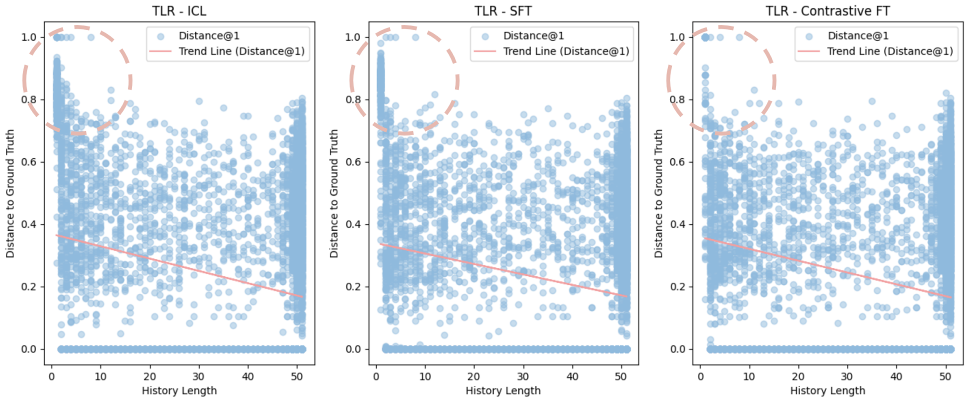


Figure 12: Semantic distance (ϕ) vs. history length on ICEWS14 under TLR sampling. The encircled region highlights CL's improved semantic grounding in sparse-history settings.

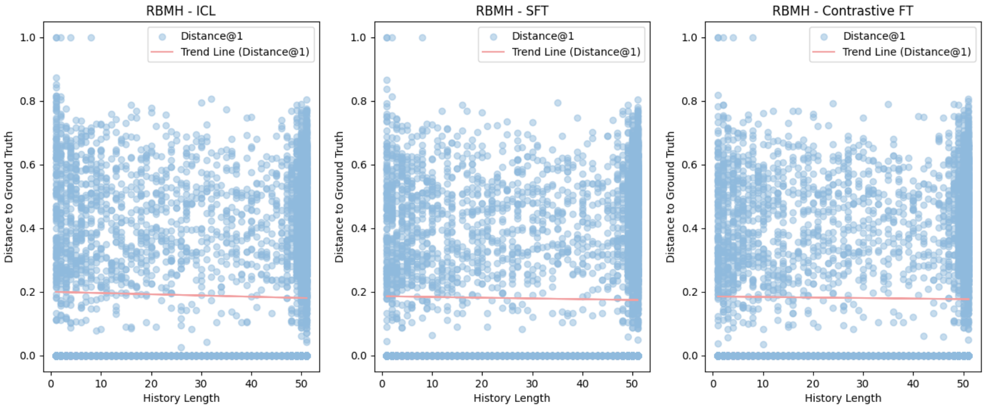


Figure 13: Semantic distance (ϕ) vs. history length for the same TLR-identified sparse samples, but evaluated under *RBMH Sampling*. The model exhibits more stable behavior across history lengths.

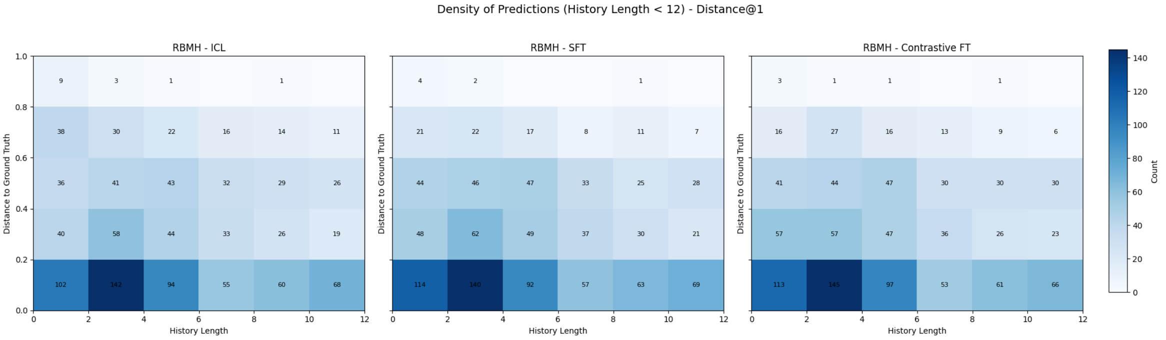


Figure 14: Semantic distance (ϕ) vs. history length for the same TLR-identified sparse samples, but evaluated under *RBMH Sampling*. *CFT* learns better with RBMH as it samples the deeper relationships between entities.

Model Outputs

ICL-LLaMA2-7b

1. Citizen_(Nigeria)
2. Boko_Haram
3. Suleiman_Abba
4. Other_Authorities_/_Officials_(Nigeria)
5. Aliyu_Mohammed_Gusau
6. Nigerian_Army
7. Nigerian_Army
8. Nigerian_Army
9. Nigerian_Army
10. Other_Authorities_/_Officials_(Nigeria)

GenTKG-LLaMA2-7b

1. Citizen_(Nigeria)
2. Boko_Haram
3. Suleiman_Abba
4. Other_Authorities_/_Officials_(Nigeria)
5. Nigeria
6. Aliyu_Mohammed_Gusau
7. Nigeria
8. Nigeria
9. Nigeria_Army
10. None

RECIPE-TKG-LLaMA2-7b

1. Citizen_(Nigeria)
2. Boko_Haram
3. Suleiman_Abba
4. Other_Authorities_/_Officials_(Nigeria)
5. Aliyu_Mohammed_Gusau
6. Government_(Nigeria)
7. Military_(Nigeria)
8. Abdul_Aziz_Yari
9. Chief_of_Staff_(Nigeria)
10. Abdul_Aziz_Yari

RECIPE-TKG-LLaMA3-8b

1. Citizen_(Nigeria)
2. Other_Authorities_/_Officials_(Nigeria)
3. Boko_Haram
4. Suleiman_Abba
5. Defense_/_Security_Ministry_(Nigeria)
6. Terrorist_(Boko_Haram)
7. Employee_(Nigeria)
8. Terrorist_(Nigeria)
9. Senior_Military_Official_(Nigeria)
10. Defense_Personnel_(Nigeria)

Ground-truth entity: High_Ranking_Military_Personnel_(Nigeria)

Figure 15: Top-10 predictions from four models. RECIPE-TKG produce semantically closer outputs to the ground truth.