LANTERN in the Event Stream: Training-Free Temporal Knowledge Graph Forecasting by Balancing Inertia and Shifts

Chengyuan Jin[1,2,3], Ao Chang[2,3], Daojian Zeng[4], Wenhao Teng[1], Xiangwen Liao[6], Kang Liu[1,2,3], Jun Zhao[2,3], Yubo Chen[5]*

[1]Department of Gastrointestinal Surgery, Fujian Provincial Cancer Hospital, China
[2]The Key Laboratory of Cognition and Decision Intelligence for Complex Systems, Institute of Automation, Chinese Academy of Sciences, Beijing, China
[3]School of Artificial Intelligence, University of Chinese Academy of Sciences, China
[4]Hunan Normal University, China
[5]School of Information Engineering, Minzu University of China
[6]College of Computer and Data Science, Fuzhou University, China

jinchengyuan2024@ia.ac.cn, changao2024@ia.ac.cn
zengdj@hunnu.edu.cn, fjtengwh@fjmu.edu.cn, liaoxw@fzu.edu.cn
kliu@nlpr.ia.ac.cn, jzhao@nlpr.ia.ac.cn, yubo.chen@muc.edu.cn

Abstract

Temporal knowledge graph forecasting (TKGF) asks a model to rank the most plausible future entity for a query such as $(s, r, ?, t)$ from historical events. Recent training-free methods use large language models (LLMs) for this task, but their accuracy depends heavily on which past events are shown in the prompt under a tight context budget. We present LANTERN, a training-free prompting framework that addresses this bottleneck by combining two complementary views of history: a long-window strength score for stable interaction patterns and a short-window novelty score for sudden changes. LANTERN first filters unhelpful events, then selects a compact evidence set with Pareto-greedy selection, and finally adds one structure-aware analogical demonstration. Across ICEWS14, ICEWS05-15, ICEWS18, and GDELT, LANTERN consistently outperforms the state-of-the-art training-free baseline AnRe under the same backbone and 2-hop candidate protocol, improving Hits@1 by up to 2.5 points and MRR by up to 1.2 points.

1 Introduction

Temporal knowledge graphs (TKGs) capture the evolution of relational facts as timestamped events (s, r, o, t) (Boschee et al., 2015; Leetaru and Schrodt, 2013). Temporal knowledge graph forecasting (TKGF) aims to predict missing entities in future queries such as $(s_q, r_q, ?, t_q)$ from historical evidence (García-Durán et al., 2018; Lee et al., 2023). While supervised models learn time-aware representations effectively (Jin et al., 2020; Zhu et al., 2021; Xu et al., 2023b), they usually require dataset-specific training and can be sensitive to distribution shifts. Recently, large language models (LLMs) have emerged as a promising *training-*

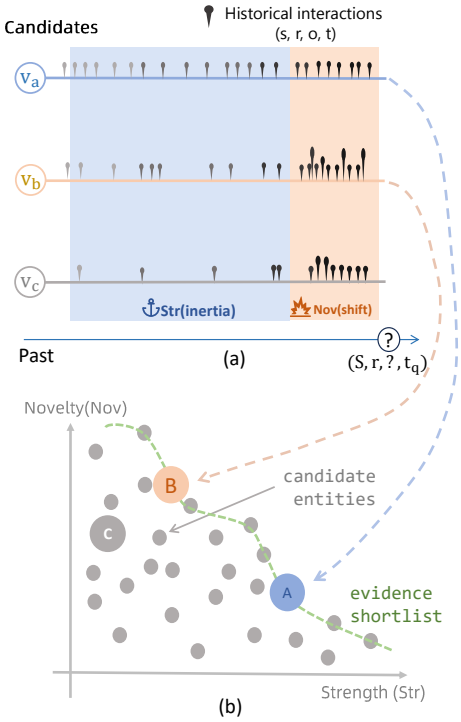

Figure 1: Interaction inertia vs. regime shift as complementary signals for evidence selection. (a) For a query $(s, r, ?, t_q)$, historical interactions (s, r, v, t) are summarized with fixed-count windows: a long window yields a smoothed strength prior (inertia), whereas a short window yields a novelty (surprise) signal that captures recent bursts. (b) Mapping candidate entities to the (Strength, Novelty) plane enables Pareto filtering to produce a compact evidence shortlist under a strict budget.

*Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 11519–11533

free alternative, performing forecasting through in-context learning without parameter updates (Lee et al., 2023). However, LLM performance is highly sensitive to context quality, making historical evidence selection and organization a key bottleneck under tight token budgets (Xia et al., 2024; Yu et al., 2024; Tang et al., 2025).

Problem formulation. At a high level, TKGF asks the model to answer a future question such as "On date t_q, subject s_q will have relation r_q with which entity?" by ranking candidate entities from historical event sequences. In our setting, the challenge is not to train a new predictor, but to decide *which few past events* to show the LLM for reliable ranking under a fixed context budget.

A fundamental challenge in evidence selection is the tension between *interaction inertia* and *regime shifts*. Real-world event streams often exhibit stable long-run preferences (inertia), but are punctuated by abrupt regime shifts—where short-lived bursts override historical regularities (Kleinberg, 2002; Koren, 2009). Existing prompting strategies, mainly based on semantic retrieval or simple recency, often fail to model this trade-off explicitly. As a result, prompts can be dominated by stale, repetitive patterns or distracted by noisy recent events, lacking the structural nuance required for accurate forecasting (Figure 1).

To address this limitation, we present LANTERN, a training-free prompting framework that balances inertia and shifts. LANTERN operationalizes this trade-off using two lightweight fixed-count-window scores: a smoothed *strength prior* from a long window to capture inertia, and a *novelty score* (based on a Beta-Binomial model) from a short window to quantify surprise and detect shifts. This dual-view approach helps distinguish stable trends from meaningful deviations.

Building on these signals, LANTERN constructs a compact prompt through a multi-stage process. We first filter irrelevant events using an LLM-based usefulness gate, then apply a Pareto-greedy selection strategy that prioritizes evidence maximizing both strength and novelty under a fixed budget. Additionally, we retrieve a structure-aware analogical demonstration—an example with a similar inertia-shift profile—to guide the LLM's reasoning. By default, we use a single demonstration to maintain efficiency, consistent with findings that additional demonstrations yield diminishing returns (Tang et al., 2025).

We evaluate LANTERN on four standard benchmarks: ICEWS14, ICEWS05-15, ICEWS18, and GDELT (Boschee et al., 2015; Leetaru and Schrodt, 2013). Experiments show that LANTERN consistently outperforms the state-of-the-art training-free baseline AnRe (Tang et al., 2025) using the same InternLM2-7B backbone and 2-hop candidate protocol. Notably, it improves Hits@1 by up to 2.5 points and MRR by up to 1.2 points, validating the benefit of explicitly modeling temporal dynamics in in-context learning.

Our contributions are summarized as follows:

(1) We propose LANTERN, a training-free framework for TKGF that leverages the balance between interaction inertia and regime shifts to improve evidence selection and reasoning.

(2) We introduce a scoring mechanism using fixed-count windows for strength and novelty, integrated with Pareto-greedy selection and structure-aware analogical demonstrations.

(3) Extensive experiments on multiple datasets show that LANTERN significantly outperforms strong baselines, offering a robust and efficient solution for temporal forecasting.

2 Related Work

Supervised temporal knowledge graph forecasting. Supervised TKGF learns time-aware representations from event sequences, including sequence encoders (García-Durán et al., 2018), autoregressive models (Jin et al., 2020), repetition-aware mechanisms (Zhu et al., 2021), evolutional representation learning (Li et al., 2021), recurrent GNN frameworks (Li et al., 2022), historical contrastive learning (Xu et al., 2023b), and temporal logical rules (Liu et al., 2022). These approaches require dataset-specific training or tuning and can be sensitive to non-stationarity. In contrast, our training-free framework avoids expensive retraining and handles distribution shifts through explicit inertia–shift modeling.

Training-free forecasting and demonstration-based prompting. Training-free TKGF uses LLMs as rankers through in-context learning without updating parameters (Lee et al., 2023). To handle strict context limits, recent work retrieves and organizes historical evidence, such as traversing higher-order histories (Xia et al., 2024) or adapting temporal rules (Yu et al., 2024; Wang et al.,

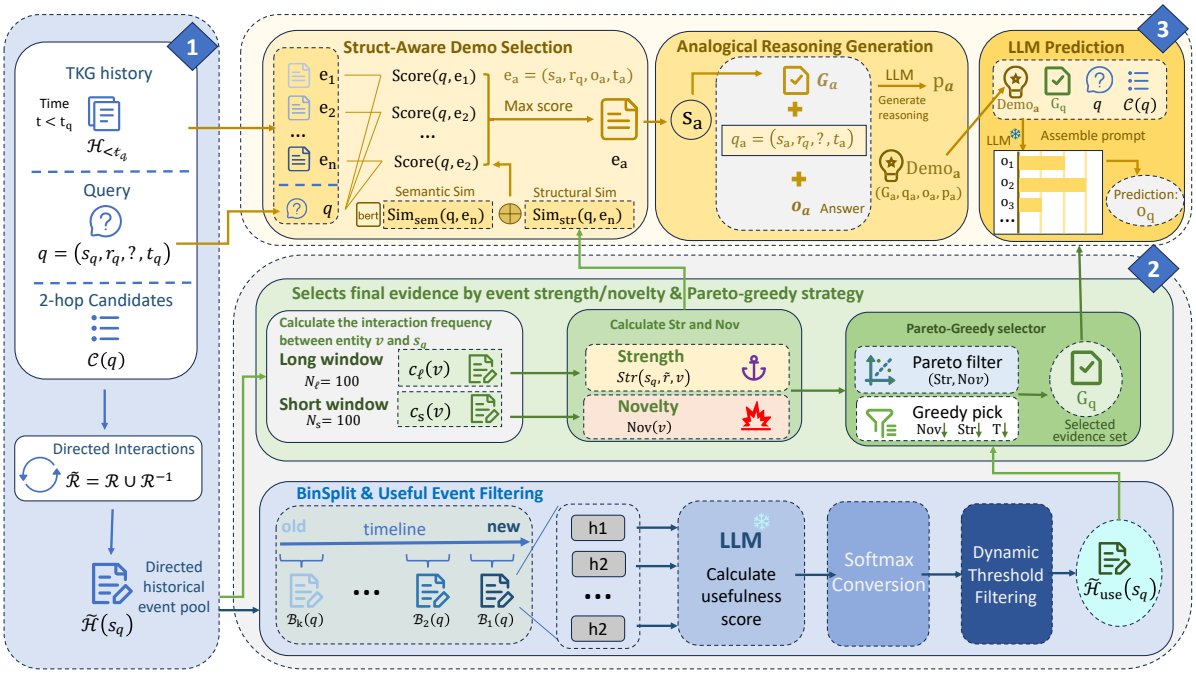

Figure 2: Overview of our training-free prompting framework. Given a query q, we first construct query-centered directed histories and a 2-hop candidate set $\mathcal{C}(q)$. We then select a compact evidence set G_q by combining strength/novelty scoring with an LLM-based usefulness gate and a Pareto-greedy selector under a fixed budget. In parallel, we retrieve a structure-aware analogical demonstration, build its evidence G_a, and prompt the LLM to generate a short process explanation. Finally, we assemble demonstrations, G_q, q, and $\mathcal{C}(q)$ into a single prompt for candidate ranking.

2024); related knowledge-selection work also studies retrieval and re-ranking to reduce semantic drift (Huang et al., 2025). Beyond history-only prompting, demonstration-based approaches provide analogical examples (Lee et al., 2023), with recent methods constructing demonstrations from multi-scale histories (Tang et al., 2025); more generally, retrieval-based in-context learning also shows the value of selecting demonstrations rather than using fixed examples (Luo et al., 2024). Other works explore prompt-based PLM methods (Xu et al., 2023a) or generative components (Liao et al., 2024; Bai et al., 2025; Ding et al., 2024). One broader example is EVCharging-GPT, which formulates sequential behavior prediction through structured prompting (Li et al., 2025). However, existing strategies often overlook the tension between interaction inertia and regime shifts in both evidence and demonstration selection. We address this by balancing smoothed strength and novelty signals and selecting structure-aware demonstrations that match the query's inertia–shift profile.

Non-stationarity, burstiness, and regime change. Outside TKGs, event-stream prediction under non-stationarity is often analyzed through burst detection, surprise, and changepoint/concept-drift frameworks (Kleinberg, 2002; Adams and MacKay, 2007; Gama et al., 2014). These perspectives motivate separating stable long-run preference signals from short-run deviations that may indicate a regime shift, and have been used broadly in temporal modeling (e.g., decomposing long-term preference vs. short-term drift in recommendation) (Koren, 2009). Our method brings this principle into training-free LLM prompting by defining explicit long-window *strength* and short-window *novelty* (surprise) signals. This allows us to construct prompts that capture both stable preferences and emerging shifts.

Benchmarks and temporal reasoning. Standard TKGF benchmarks are often derived from ICEWS and GDELT event streams (Boschee et al., 2015; Leetaru and Schrodt, 2013). Recent analyses further show that limited temporal reasoning and distribution shift can substantially affect LLM behavior, motivating methods that better control which historical signals are exposed to LLMs (Chu et al., 2024).

3 Method

The core bottleneck in training-free TKGF is choosing a few truly useful events from a long history. AnRe-style semantic retrieval works for stable patterns, but can miss sudden shifts; LANTERN addresses this gap with two complementary signals, *strength* for long-run interaction inertia and *novelty* for short-run regime shifts. We first score history with these views, then filter and select a compact evidence set under a fixed budget, and finally add one structure-aware analogical demonstration. Table 8 gives a concrete example.

We operationalize the inertia–shift trade-off with two lightweight fixed-count-window scores and build a compact prompt by selecting evidence (and optionally analogical demonstrations) under a strict context budget. Figure 2 overviews the pipeline: (i) setup and query-centered directed history construction with 2-hop candidate generation, (ii) fixed-count-window strength/novelty scoring, (iii) usefulness-gated Pareto-greedy evidence selection, (iv) structure-aware demonstration construction with analogical replay, and (v) prompting and inference for final LLM ranking.

3.1 Setup: Task, Candidates, and Directed Histories

A temporal knowledge graph (TKG) is a set of timestamped events (s, r, o, t), where $s, o \in \mathcal{E}$, $r \in \mathcal{R}$, and $t \in \mathcal{T}$. Given a query $q = (s_q, r_q, ?, t_q)$ (object prediction; subject prediction is analogous), the goal is to rank candidate entities $o \in \mathcal{C}(q)$ using historical events $\mathcal{H}_{<t_q}$. Following standard training-free LLM settings (Lee et al., 2023), we verbalize events into text prompts and score candidates by the LLM's label-token likelihoods, without parameter updates. We use a single analogical demonstration by default and focus on improving evidence selection and organization. For comparability, we construct $\mathcal{C}(q)$ with the standard 2-hop historical-neighbor protocol (Tang et al., 2025). This protocol is used by all compared training-free methods in our main experiments, so the comparison is direct; our framework is decoupled from candidate generation and can be paired with stronger retrieval modules in the future.

Query-centered directed interactions. We collect the query-centered history pool involving s_q:

$$\mathcal{H}(s_q) = \left\{(s, r, o, t) \in \mathcal{H}_{<t_q} \;\middle|\; s = s_q \;\vee\; o = s_q\right\} \tag{1}$$

To make interaction scoring directionally consistent, we augment the relation vocabulary with inverses: for each $r \in \mathcal{R}$, we denote its inverse by r^{-1}; let $\mathcal{R}^{-1} = \{r^{-1} \mid r \in \mathcal{R}\}$; and define $\tilde{\mathcal{R}} = \mathcal{R} \cup \mathcal{R}^{-1}$. We convert each event in $\mathcal{H}(s_q)$ into a directed interaction $(s_q, \tilde{r}, v, t)$ where s_q is always the source:

$$(s_q, \tilde{r}, v, t) = \begin{cases} (s_q, r, o, t), & \text{if } s = s_q, \\ (s_q, r^{-1}, s, t), & \text{if } o = s_q. \end{cases} \tag{2}$$

This yields a unified multiset $\tilde{\mathcal{H}}(s_q)$ of directed interactions.

3.2 Strength and Novelty from Fixed-count Windows

Fixed-count windows. Event streams have heterogeneous temporal density; we therefore define windows by *event counts*. Compared with time-span windows, fixed-count windows keep the sample size stable across dense and sparse periods, preventing high-density intervals from overwhelming the temporal signal. They also align naturally with a fixed prompt budget: a time-span window can contain wildly varying numbers of events, requiring ad hoc truncation and making comparisons across queries less consistent. This design is especially suitable for training-free prompting, where both stable sample size and stable token usage matter, though it can be less ideal for extreme cold-start cases or domains with highly irregular continuous-time dynamics. For each query-centered directed relation $(s_q, \tilde{r})$, let $\mathcal{L}_{q,\tilde{r}}$ be the list of all interactions $(s_q, \tilde{r}, *, t) \in \tilde{\mathcal{H}}(s_q)$ with $t < t_q$, sorted by decreasing t. We define:

- long-window list $\mathcal{L}^{\ell}_{q,\tilde{r}}$: the first N_ℓ interactions in $\mathcal{L}_{q,\tilde{r}}$;
- short-window list $\mathcal{L}^{s}_{q,\tilde{r}}$: the first N_s interactions in $\mathcal{L}_{q,\tilde{r}}$,

where $N_s \ll N_\ell$. If $|\mathcal{L}_{q,\tilde{r}}| < N_\ell$ (or N_s), we take as many most-recent interactions as available. In extremely sparse histories, the long-window "steady-state" estimate can be unreliable; we therefore use Dirichlet smoothing as an explicit backoff. For robustness under extreme sparsity, we optionally apply a relation-level hierarchical backoff; to mitigate stale evidence under distribution shift, we optionally apply drift-aware shrinkage of the long window. For clarity, these optional components

are described in Appendix A.5 and Appendix A.6, respectively. Let neighbor counts be

$$c_\ell(v) = \#\{(s_q, \tilde{r}, v, t) \in \mathcal{L}^\ell_{q,\tilde{r}}\},$$
$$c_s(v) = \#\{(s_q, \tilde{r}, v, t) \in \mathcal{L}^s_{q,\tilde{r}}\}, \quad (3)$$

with totals $C_\ell = |\mathcal{L}^\ell_{q,\tilde{r}}| = \sum_v c_\ell(v)$ and $C_s = |\mathcal{L}^s_{q,\tilde{r}}| = \sum_v c_s(v)$. Let the observed neighbor set in the long window be $V_{q,\tilde{r}} = \{v \mid c_\ell(v) > 0\}$.

Strength prior as steady-state preference. We model inertia as the steady-state preference distribution $p(v \mid s_q, \tilde{r})$. Using a symmetric Dirichlet prior with concentration $\alpha_0 > 0$, we define the strength score as the posterior mean:

$$\mathrm{Str}(s_q, \tilde{r}, v; t_q) = \frac{c_\ell(v) + \alpha_0}{C_\ell + \alpha_0 |V_{q,\tilde{r}}|} \quad (4)$$

This yields a smoothed long-run baseline capturing persistent interaction patterns under relation $\tilde{r}$. When the query-specific history is extremely sparse, we optionally apply a relation-level hierarchical backoff to avoid brittle estimates (Appendix A.5).

Novelty prior as Beta-Binomial surprise. To quantify regime shifts, we measure the surprise of the short-window concentration $c_s(v)$ under the long-run baseline. Simple metrics like z-scores fail to account for overdispersion in bursty political event streams (Kleinberg, 2002). We therefore use a Beta-Binomial model, which captures extra variability by treating the success probability as uncertain rather than fixed. We define novelty as the negative log-survival probability of observing at least $c_s(v)$ occurrences in the short window, with a Beta-Binomial prior parameterized by the strength baseline:

$$\mathrm{Nov}(s_q, \tilde{r}, v; t_q) = -\log \Pr(X \geq c_s(v)), \quad (5)$$

where X is a Beta-Binomial random variable with $n = C_s$ and (α, β) derived from $\mathrm{Str}(s_q, \tilde{r}, v; t_q)$. We compute this quantity in log-space and apply probability clipping for numerical stability; the full parameterization is in Appendix A.1. This score increases with $c_s(v)$ (rewarding bursts) and decreases with prior strength (penalizing expected behavior), matching the intuition of a regime shift. For brevity, when $(s_q, \tilde{r}, t_q)$ is clear, we write $\mathrm{Nov}(v)$.

3.3 Evidence Selection under a Fixed Budget

We formalize evidence construction as a constrained multi-objective selection problem over query-centered directed interactions.

LLM usefulness gate. Strength and novelty scores are lightweight and model-free, but they can still identify events that are statistically salient yet unhelpful for answering the query. We therefore adopt AnRe's probability-based usefulness scoring with dynamic threshold filtering (Tang et al., 2025) as a semantic gate before selection. We bin history by time quantiles, ask the LLM to judge usefulness on a short per-bin shortlist, and filter events with a stricter threshold for older bins (Appendix A.2).

Pareto-greedy selection. To avoid linear-weighting hyperparameters, we use Pareto filtering followed by budget-constrained greedy selection after usefulness filtering (Algorithm 1). Pareto sorting prioritizes *non-dominated* events under the two objectives (Novelty and Strength): events in the first Pareto frontier are preferred, followed by subsequent frontiers (iterative peeling). Concretely, each directed interaction event has the form $e = (s_q, \tilde{r}, v, t)$. We define event-level scores by lifting the neighbor-level definitions in Section 3.2: $\mathrm{Str}(e) \triangleq \mathrm{Str}(s_q, \tilde{r}, v; t_q)$ and $\mathrm{Nov}(e) \triangleq \mathrm{Nov}(s_q, \tilde{r}, v; t_q)$, computed from the short/long fixed-count windows of $(s_q, \tilde{r})$ ending at t_q. Pseudocode is provided in Appendix A.4 (Algorithm 1). We enforce deterministic time-coverage, per-relation diversity, and anti-redundancy constraints during greedy selection; details are in Appendix A.3.

3.4 Structure-aware Demonstrations and Analogical Replay

We include one analogical demonstration by default, consistent with AnRe's finding that one demonstration provides the best accuracy–cost trade-off under a fixed budget (Tang et al., 2025). Instead of using a fixed weight between semantic and structural similarity, we use an adaptive weight β based on dataset-level structural regularity (details in Appendix A.7).

Semantic and structural similarity. Given a target query $q = (s_q, r_q, ?, t_q)$ and a candidate demonstration event $e_a = (s_a, r_q, o_a, t_a)$ sharing the relation, we compute semantic similarity $\mathrm{Sim}_{\mathrm{sem}}(q, e_a)$ between the verbalized questions. We compute structural similarity $\mathrm{Sim}_{\mathrm{str}}(q, e_a)$ between low-dimensional inertia–shift profiles derived from $(\mathrm{Str}, \mathrm{Nov})$. We then combine them with

the adaptive weight β (details in Appendix A.7):

$$\begin{aligned}\text{Score}(q, e_a) = \beta \cdot \text{Sim}_{\text{sem}}(q, e_a) \\ + (1-\beta) \cdot \text{Sim}_{\text{str}}(q, e_a),\end{aligned} \tag{6}$$

and we pick the top-scoring candidate(s) subject to the minimum-history constraint $H_{\min}$.

Analogical replay with process explanations. Demonstrations are most effective when they teach the LLM *how* an answer follows from the temporal development of events, rather than only showing an input-output pair. Following AnRe (Tang et al., 2025), we construct a short LLM-generated *process explanation* for each selected demonstration.

Demo history and masked question. Let the selected demonstration event be $e_a = (s_a, r_q, o_a, t_a)$, retrieved from historical data such that it shares the relation r_q (and is semantically/structurally similar). We construct a demo query $q_a = (s_a, r_q, ?, t_a)$ by masking the object and treat the known object o_a as the answer. We then build the demonstration evidence G_a by running Algorithm 1 on the query-centered history around s_a (with the same B, K and usefulness gate).

Process explanation generation. Given (G_a, q_a, o_a), we prompt the LLM with θ_{arp} (Appendix D) to generate a short analysis p_a describing how the answer is implied by the event-chain evolution. The final analogical demonstration is thus a 4-tuple:

$$\text{Demo}_a = (G_a,\ q_a,\ o_a,\ p_a). \tag{7}$$

At inference time, we include one such demonstration before the target evidence G_q, allowing the LLM to learn an analogical reasoning pattern and transfer it to the target query.

3.5 Prompting and Inference

The final prompt contains: (i) one analogical replay demonstration (history $\rightarrow$ question $\rightarrow$ answer, followed by an LLM-generated process explanation), (ii) selected evidence interactions G_q in chronological (or reverse-chronological) order, (iii) the target query and candidate list $\mathcal{C}(q)$ mapped to numeric labels, and (iv) an instruction to output the most likely label. The LLM returns logits over candidate labels; we convert them to a distribution with softmax and rank candidates accordingly. Overall, the prompt exposes stable anchors (strength), recent bursts (novelty), and an LLM-judged usefulness gate, while analogical replay provides a transferable reasoning trajectory under the same budget.

4 Experiments

We evaluate our method on four standard temporal knowledge graph forecasting benchmarks. We detail the experimental setup, report main results against strong baselines, and present ablations and analyses on cross-LLM generalizability, regime-shift robustness, sparse-query backoff, and efficiency/cost.

4.1 Experimental Setup

LLM Backbone and Baselines. Following AnRe (Tang et al., 2025), we use InternLM2-7B(InternLM Team, 2024) as the default backbone and keep the same candidate protocol unless stated otherwise. We compare with training-free baselines: ICL(Lee et al., 2023), CoH (Xia et al., 2024), ONSEP (Yu et al., 2024), and AnRe (Tang et al., 2025). We also list representative supervised baselines (RE-NET (Jin et al., 2020), CyGNet (Zhu et al., 2021), TiRGN (Li et al., 2022)) for reference; unless stated otherwise, their numbers are taken from prior work.

Datasets. We evaluate on four standard TKGF benchmarks derived from political event streams: ICEWS14, ICEWS05-15, ICEWS18, and GDELT (Boschee et al., 2015; Leetaru and Schrodt, 2013). We use the same preprocessed splits as AnRe; note that some prior work uses a smaller ICEWS14 variant with fewer entities/relations, which is not used here. For ICEWS14, we follow prior work and tune hyperparameters on a held-out subset of the training data rather than an official validation split.

Evaluation metrics. We report Mean Reciprocal Rank (MRR) and Hits@k ($k \in \{1, 3, 10\}$) under the standard time-aware filtered protocol (Bordes et al., 2013; Han et al., 2020). For a query $(s_q, r_q, ?, t_q)$ with ground truth o^*, we rank o^* against candidate entities, filtering out other valid entities that appear in true facts (s_q, r_q, o', t_q) at the same timestamp.

Implementation details. We set the evidence budget $B = 100$ to match AnRe's "History Length=100" setting. We construct candidates $\mathcal{C}(q)$ with the standard 2-hop historical-neighbor protocol (Tang et al., 2025) and cap its size at $C_{\max} = 100$. Unless stated otherwise, we use $N_\ell = 100$, $N_s = 10$, and one demonstration ($m = 1$). For the core LANTERN-specific parameters, we use one shared default configuration across all four datasets, and the same selection defaults

Method	Train	ICEWS14				ICEWS05-15				ICEWS18				GDELT			
		MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
Supervised Methods (GNNs / Embeddings / Diffusion)																	
RE-NET (Jin et al., 2020)	✓	0.399	0.301	0.440	0.582	0.437	0.336	0.488	0.627	0.298	0.197	0.326	0.485	0.196	0.124	0.221	0.340
CyGNet (Zhu et al., 2021)	✓	0.381	0.274	0.426	0.579	0.413	0.294	0.461	0.616	0.278	0.172	0.310	0.469	0.190	0.117	0.219	0.334
xERTE (Han et al., 2020)	✓	0.408	0.327	0.457	0.573	0.466	0.378	0.523	0.639	0.293	0.210	0.335	0.465	0.195	0.119	0.220	0.342
TITer (Sun et al., 2021)	✓	0.418	0.328	0.465	0.584	0.476	0.383	0.528	0.649	0.317	0.221	0.335	0.448	0.195	0.127	0.220	0.331
TiRGN (Li et al., 2022)	✓	0.429	0.321	0.485	0.636	0.485	0.369	0.552	0.703	0.320	0.210	0.367	0.537	0.217	0.137	0.241	0.376
DiffuTKG (Cai et al., 2024)	✓	**0.485**	0.364	0.494	**0.727**	**0.527**	0.403	**0.602**	**0.759**	**0.367**	0.257	0.388	**0.578**	0.251	0.163	**0.275**	**0.423**
GenTKG (Liao et al., 2024)	✓	0.435	0.328	0.485	0.630	0.475	0.370	0.535	0.690	0.305	0.205	0.345	0.515	-	-	-	-
Training-free LLM Methods (InternLM2-7B)																	
ICL (Lee et al., 2023)	×	0.318	0.301	0.432	0.560	0.353	0.353	0.507	0.647	0.215	0.172	0.289	0.434	0.145	0.098	0.176	0.285
CoH (Xia et al., 2024)	×	0.439	0.331	0.496	0.649	0.497	0.380	0.564	0.713	0.330	0.218	0.378	0.549	-	-	-	-
ONSEP (Yu et al., 2024)	×	0.448	0.330	0.464	0.570	0.485	0.386	0.546	0.662	0.301	0.200	0.324	0.443	-	-	-	-
AnRe (1-hop) (Tang et al., 2025)	×	0.466	0.346	0.470	0.608	0.498	0.389	0.551	0.678	0.321	0.255	0.371	0.554	0.221	0.153	0.244	0.342
AnRe (2-hop) (Tang et al., 2025)	×	0.474	0.369	0.511	0.657	0.509	0.391	0.580	0.696	0.355	0.260	0.392	0.567	0.243	0.166	0.266	0.375
LANTERN (1-hop)	×	0.470	0.350	0.475	0.615	0.502	0.395	0.555	0.685	0.330	0.265	0.378	0.560	0.230	0.160	0.250	0.345
LANTERN (2-hop)	×	0.480	**0.375**	**0.515**	0.665	0.515	**0.405**	0.585	0.705	0.365	**0.285**	**0.400**	0.575	**0.255**	**0.175**	0.270	0.380

Table 1: Main results. Strict Pareto filtering boosts precision, improving H@1 while keeping H@10 competitive. Supervised/hybrid results are from (Li et al., 2022; Cai et al., 2024); LLM baselines are from (Tang et al., 2025) under the same InternLM2-7B backbone and candidate protocol. We reproduce GenTKG (Liao et al., 2024) with the same backbone. **Bold**: best overall; Underline: best among training-free methods. All H@1 gains over AnRe are significant ($p < 0.05$).

in Table 11. If the final ranking prompt exceeds the soft token budget (Appendix D.1), we drop the oldest selected evidence while preserving demonstrations and candidates. For LLM scoring, we map candidates to numeric labels and rank by label-token likelihoods. We report results on the standard test sets; for cost-intensive analyses, we follow Tang et al. (2025) and evaluate a stratified sample of 500 queries. Further details on the computing infrastructure are provided in Appendix D.1.

4.2 Main Results

Table 1 presents the performance comparison on four benchmarks. Under the matched backbone/candidate protocol, LANTERN consistently improves over strong training-free baselines. Paired t-tests on per-query Hits@1 indicators against AnRe confirm statistical significance across datasets ($p < 0.05$).

Key observations. On ICEWS18, which contains more queries whose correct answers are rare or unseen in the long window, the novelty component matters most: removing novelty lowers Hits@1 from 0.285 to 0.235 in Table 2, while Figure 3 shows that gains over AnRe grow from +0.5 points on frequent-answer queries (Q1) to +4.8 points on rare/unseen ones (Q5). In contrast, ICEWS14 is more inertia-dominated: the strength-only variant slightly exceeds the full model on some metrics (MRR 0.482 and H@1 0.378 vs. 0.480 and 0.375), indicating that long-run preference is often sufficient under stable conditions. For GDELT: Table 9 shows weaker structural regularity (R_{struct}=0.31), so the benefit there mainly comes from smoothing and filtering that suppress distractors under the same budget.

Variant	ICEWS18 (Unstable)				ICEWS14 (Stable)			
	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
LANTERN (Full)	**0.365**	**0.285**	**0.400**	**0.575**	0.480	0.375	**0.515**	0.665
Score Components								
w/o Usefulness Gate	0.345	0.265	0.385	0.565	0.465	0.355	0.495	0.640
w/o Novelty (Strength only)	0.320	0.235	0.360	0.550	**0.482**	**0.378**	0.512	**0.670**
w/o Strength (Novelty only)	0.330	0.250	0.365	0.530	0.430	0.310	0.450	0.580
Replace Nov w/ Freq Ratio	0.348	0.268	0.385	0.568	0.475	0.368	0.505	0.655
Replace Nov w/ Z-Score	0.350	0.270	0.388	0.570	0.476	0.370	0.508	0.658
Demonstration Strategy								
Fixed $\beta = 1.0$ (Sem. Only)	0.355	0.275	0.390	0.570	0.472	0.365	0.500	0.655
Fixed $\beta = 0.0$ (Str. Only)	0.340	0.260	0.375	0.555	0.468	0.360	0.495	0.650
Fixed $\beta = 0.5$ (Hybrid)	0.360	0.280	0.395	0.572	0.478	0.372	0.510	0.662
Constraints								
w/o Time Coverage	0.358	0.278	0.392	0.570	0.475	0.370	0.505	0.660

Table 2: Ablation studies on ICEWS18 (shift-heavy by the answer-rarity analysis in Section 4.5) and ICEWS14 (stable under the same criterion). Removing novelty substantially hurts ICEWS18 (-5.0 H@1 points), while the strength-only variant reaches the best MRR/H@1 on ICEWS14, making the stable-data conclusion explicit.

Base Model (ICEWS18)	MRR	H@1	H@3	H@10
InternLM2-7B	0.365	0.285	0.400	0.575
Mistral-7B	0.358	0.279	0.395	0.570
Qwen2.5-7B-Instruct	0.375	0.295	0.410	0.585
Llama-3.1-8B-Instruct	0.380	0.300	0.418	0.595
Gemma-2-9B-It	**0.385**	**0.305**	**0.422**	**0.600**

Table 3: Generalization of LANTERN across different LLM architectures on ICEWS18. Performance scales positively with the capabilities of the underlying model, demonstrating the robustness of our training-free framework.

4.3 Ablation Study

We ablate key components on ICEWS18 and ICEWS14; Table 2 summarizes the results. Here we use dataset labels in a quantitative sense: *shift-*

Method	Tokens/query	it/s
ICL	∼2,500	4.2
ONSEP	∼3,500	1.9
AnRe	∼5,100	1.5
LANTERN	∼3,200	0.9

Table 4: Efficiency comparison. Tokens/query reports the *peak* token count of the final ranking prompt. While our end-to-end throughput (it/s) is lower due to the multi-stage gate, we reduce the token consumption by ∼37%, offering significant API cost savings.

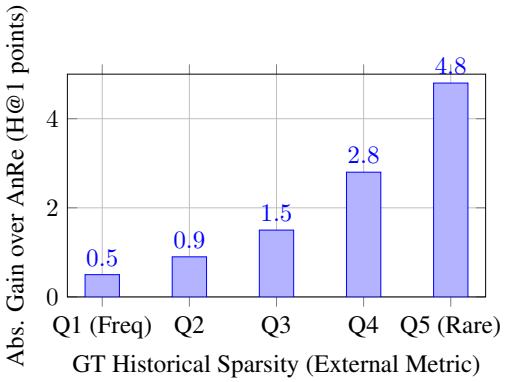

Figure 3: Absolute Hits@1 Gain (in Points) Over AnRe. Queries are binned by ground-truth sparsity; gains increase with the rarity (shift difficulty) of the true event.

heavy refers to datasets with more rare or unseen answers under the long-window rarity analysis in Section 4.5; *stable* refers to datasets where correct answers are more often supported by long-run repeated patterns; and *noisy* refers to weak structural regularity in Table 9, which increases distractors without necessarily implying genuine regime shifts. **Impact of Novelty vs. Strength.** On ICEWS18, removing novelty drops Hits@1 by 5.0 points (0.285 $\rightarrow$ 0.235), while removing strength reduces robustness and degrades broader ranking quality (e.g., Hits@10). On ICEWS14, by contrast, the strength-only variant reaches the best MRR/H@1 among the ablations, making the "stable dataset" interpretation explicit. Pareto selection balances both signals without introducing extra weighting hyperparameters. **Role of Analogical Demonstration.** With a fixed one-demo budget ($m = 1$), hybrid semantic+structural matching outperforms either alone, suggesting demonstrations should be both topical and inertia–shift aligned. **Time coverage.** The "w/o Time Coverage" row removes the per-bin cap that spreads selected evidence across K time-quantile bins; performance drops slightly on both datasets, showing that this simple constraint stabilizes evidence composition by preventing a narrow time span from dominating the prompt. **Parameter Sensitivity.** We find performance robust to $\alpha_0 \in [0.5, 2.0]$ and $\zeta \in [2.0, 3.5]$, indicating stability under moderate hyperparameter variation; we therefore present them as fixed defaults rather than claiming insensitivity beyond this range.

4.4 Generalizability across LLM Architectures

We investigate the generalizability of LANTERN by applying it to LLMs beyond the default InternLM2-7B. We conduct experiments on ICEWS18 (Table 3) using models with different architectures and instruction-following capabilities. Results show that our framework consistently benefits from stronger base models. Replacing InternLM2 with Llama-3.1-8B-Instruct (Meta, 2024) or Gemma-2-9B-It (Gemma Team, 2024) yields further gains in MRR and Hits@1, suggesting that our evidence selection is model-agnostic and scales with LLM capabilities. Mistral-7B (Mistral AI, 2023) shows slightly lower performance, consistent with baseline trends (Tang et al., 2025).

4.5 Analysis on Regime Shift Robustness

To avoid relying on model-internal scores, we partition ICEWS18 test queries using a ground-truth-based oracle metric: the long-window rarity of the correct answer. Let $\mathcal{H}_L$ be the query-specific long-window list $\mathcal{L}^{\ell}_{q,\tilde{r}_q}$ (Section 3.2). We define *GT long-window rarity* as the inverse frequency of the ground-truth object within this window: $1/(c_\ell(o^*)+1)$. Bin Q1 contains queries where o^* is highly frequent (predictable inertia); Bin Q5 contains queries where o^* is unseen or extremely rare (regime shift). Figure 3 reports absolute Hits@1 gains over AnRe: gains are small in Q1 (+0.5) but increase sharply in Q5 (+4.8), indicating that improvements concentrate on shift-heavy queries. This answer-rarity analysis is also the basis for the dataset-level labels used earlier: datasets with more mass in rare/unseen bins are described as *shift-heavy*, whereas datasets whose answers are concentrated in frequent bins are described as more *stable*.

4.6 Efficiency and Cost Analysis

We compare inference cost on the ICEWS14 test set (Table 4) and provide a detailed breakdown of LLM calls in Table 5. While our pipeline makes

Stage (ours)	# Calls/query	List size	Purpose
Usefulness gate (K bins)	K	$\leq M_{\text{use}}$ events/bin	filter semantically helpful evidence
Demo replay explanation	m	$\leq B$ events	generate process explanation p_a
Final prediction	1	$\leq B$ events, $\leq C_{\max}$ candidates	rank candidates by label-token likelihood
Total	$K + m + 1$	-	-

Table 5: LLM call budget decomposition. While we make more calls (K=4, m=1 $\Rightarrow$ 6 calls), the early calls are lightweight. This "filter-then-rank" strategy reduces the expensive final-step context.

Method (Sparse Partition)	MRR	H@1	H@3	H@10
w/o Backoff	0.310	0.215	0.345	0.490
w/ Backoff (LANTERN)	**0.335**	**0.240**	**0.375**	**0.520**

Table 6: Impact of relation-level backoff on sparse queries (bottom 25% history density) of ICEWS14. Backoff significantly improves robustness when specific history is lacking.

more calls (a usefulness gate per bin), it reduces the peak length of the final ranking prompt from 5,100 to 3,200 input tokens (compared to AnRe under the same setting). End-to-end throughput is lower due to multiple stages, but the gate prompts are short. We therefore present the comparison of peak token consumption at the final step in Table 4, while Table 5 makes the upstream call budget explicit; together they give a clearer picture of practical API cost than peak prompt length alone.

4.7 Effect of Backoff on Sparse Queries.

We stratify ICEWS14 test queries by history density (C_ℓ quantiles) and analyze the impact of our relation-level hierarchical backoff strategy on the sparse partition (0–25%). Table 6 shows that removing the backoff mechanism leads to a performance drop in sparse settings (MRR 0.335 $\rightarrow$ 0.310), confirming that borrowing relation-level priors is crucial when query-specific history is insufficient. In denser bins, backoff is rarely triggered by design, so we treat it as a targeted safeguard for the extreme sparse regime rather than a global improvement mechanism.

5 Conclusion

We propose a training-free prompting framework for temporal knowledge graph forecasting that explicitly balances interaction inertia and regime shifts. Using fixed-count windows, we combine a smoothed strength prior with a Beta-Binomial novelty score, and construct compact prompts through usefulness gating and Pareto-greedy evidence selection. Experiments on ICEWS and GDELT demonstrate consistent gains over strong training-free baselines under the same candidate protocol and prompt budget.

Limitations

Our fixed-count windows for approximating local steady-state preferences may be suboptimal in domains with highly irregular time intervals or continuous-time dynamics, where time-span windows may be more suitable. The usefulness gate, which relies on LLM-internal knowledge, may over-filter long-tail evidence, sacrificing recall for precision. We rely on candidate-set construction (e.g., 2-hop neighbors) for tractable scoring, so training-free ranking cannot recover true answers outside this set; this is a general limitation of the current task protocol rather than a special weakness of our method. Our validation is limited to political event streams (ICEWS, GDELT). Natural future directions include systematic fixed-count-window versus time-span-window comparisons, integrating open-world or zero-shot candidate retrieval for true cold-start entities, extending the framework beyond political event streams to other domains, and improving few-shot evidence selection under the same context budget, especially in low-resource settings.

Acknowledgments

This work is supported by the New Generation Artificial Intelligence-National Science and Technology Major Project (2025ZD0123301), the National Natural Science Foundation of China (No. 62576340, No. U24A20335, No. 62476060), the Beijing Nova Program (20250484750), and the Youth Innovation Promotion Association CAS.

References

Ryan Prescott Adams and David J. C. MacKay. 2007. Bayesian online changepoint detection. *Preprint*, arXiv:0710.3742.

Long Bai, Zixuan Li, Xiaolong Jin, Jiafeng Guo, Xueqi Cheng, and Tat-Seng Chua. 2025. G2S: A general-to-specific learning framework for temporal knowledge graph forecasting with large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 20927–20938, Vienna, Austria. Association for Computational Linguistics.

Antoine Bordes, Nicolas Usunier, Alberto Garcia-Duran, Jason Weston, and Oksana Yakhnenko. 2013. Translating embeddings for modeling multi-relational data. In *Advances in Neural Information Processing Systems*, volume 26.

Elizabeth Boschee, Jennifer Lautenschlager, Sean O'Brien, Steve Shellman, James Starz, and Michael Ward. 2015. Icews coded event data. Harvard Dataverse.

Yuxiang Cai, Qiao Liu, Yanglei Gan, Changlin Li, Xueyi Liu, Run Lin, Da Luo, and Jiaye Yang. 2024. Predicting the unpredictable: Uncertainty-aware reasoning over temporal knowledge graphs via diffusion process. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5766–5778, Bangkok, Thailand. Association for Computational Linguistics.

Zheng Chu, Jingchang Chen, Qianglong Chen, Weijiang Yu, Haotian Wang, Ming Liu, and Bing Qin. 2024. TimeBench: A comprehensive evaluation of temporal reasoning abilities in large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1204–1228, Bangkok, Thailand. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long and Short Papers)*, pages 4171–4186.

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.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. 2014. A survey on concept drift adaptation. *ACM Computing Surveys*, 46(4):44:1–44:37.

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.

Gemma Team. 2024. Gemma 2: Improving open language models at a practical size. *Preprint*, arXiv:2408.00118.

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 (ICLR)*.

Zhisheng Huang, Xudong Jia, Tao Chen, and Zhongwei Zhang. 2025. A general scalable approach for knowledge selection based on iterative hybrid encoding and re-ranking. *Data Intelligence*, 7(1).

InternLM Team. 2024. Internlm2 technical report. *Preprint*, arXiv:2403.17297.

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, Online. Association for Computational Linguistics.

Jon Kleinberg. 2002. Bursty and hierarchical structure in streams. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 91–101. ACM.

Yehuda Koren. 2009. Collaborative filtering with temporal dynamics. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 447–456. ACM.

Dong-Ho Lee, Kian Ahrabian, Woojeong Jin, Fred Morstatter, and Jay Pujara. 2023. Temporal knowledge graph forecasting without knowledge using in-context learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 544–557, Singapore. Association for Computational Linguistics.

Kalev Leetaru and Philip A. Schrodt. 2013. Gdelt: Global data on events, location and tone, 1979–2012. Paper presented at the International Studies Association Annual Convention.

Houzhi Li, Jinyu Wang, Zixin Jiang, Guorui Su, Chaowen Yan, Hao Su, and Zhichun Wang. 2025. Evcharging-gpt: Predicting electric vehicle user charging behavior using large language models. *Data Intelligence*, 7(2):336–357.

Yujia Li, Shiliang Sun, and Jing Zhao. 2022. Tirgn: Time-guided recurrent graph network with local-global historical patterns for temporal knowledge graph reasoning. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pages 2152–2158. International Joint Conferences on Artificial Intelligence Organization. Main Track.

Zixuan Li, Xiaolong Jin, Wei Li, Saiping Guan, Jiafeng Guo, Hua Hua, 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. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pages 4303–4317, Mexico City, Mexico. Association for Computational Linguistics.

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 AAAI Conference on Artificial Intelligence*, volume 36, pages 5806–5814.

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Zhao. 2024. Dr.icl: Demonstration-retrieved in-context learning. *Data Intelligence*, 6(4):909–922.

Meta. 2024. Llama-3.1-8b-instruct model card.

Mistral AI. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

SciPy 1.0 Contributors. 2020. Scipy 1.0: Fundamental algorithms for scientific computing in python. *Nature Methods*, 17(3):261–272.

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

Guo Tang, Zheng Chu, Wenxiang Zheng, Junjia Xiang, Yizhuo Li, Weihao Zhang, Ming Liu, and Bing Qin. 2025. AnRe: Analogical replay for temporal knowledge graph forecasting. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 4632–4650, Vienna, Austria. Association for Computational Linguistics.

Jiapu Wang, Kai Sun, Linhao Luo, Wei Wei, Yongli Hu, Alan Wee-Chung Liew, Shirui Pan, and Baocai Yin. 2024. Large language models-guided dynamic adaptation for temporal knowledge graph reasoning. *Preprint*, arXiv:2405.14170.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clément Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, and Jamie Brew. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45.

Yuwei Xia, Ding Wang, Qiang Liu, Liang Wang, Shu Wu, and Xiao-Yu Zhang. 2024. 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.

Wenjie Xu, Ben Liu, Miao Peng, Xu Jia, and Min Peng. 2023a. Pre-trained language model with prompts for temporal knowledge graph completion. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 7790–7803, Toronto, Canada. Association for Computational Linguistics.

Yi Xu, Junjie Ou, Hui Xu, and Luoyi Fu. 2023b. Temporal knowledge graph reasoning with historical contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 4765–4773.

Xuanqing Yu, Wangtao Sun, Jingwei Li, Kang Liu, Chengbao Liu, and Jie Tan. 2024. Onsep: A novel online neural-symbolic framework for event prediction based on large language model. *Preprint*, arXiv:2408.07840.

Cunchao Zhu, Muhao Chen, Changjun Fan, Guangquan Cheng, and Yan Zhan. 2021. Learning from history: Modeling temporal knowledge graphs with sequential copy-generation networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 4732–4740.

A Optional Components and Implementation Details

A.1 Beta-Binomial novelty computation

We model the short-window count X as a Beta-Binomial random variable with parameters $n = C_s$ and concentration ϕ (a hyperparameter tuned in experiments). To ensure $\alpha, \beta > 0$ even in degenerate cases, we clip the prior probability with a small ε and set

$$\tilde{p}(v) = \min\{1-\varepsilon, \max\{\varepsilon,\ \mathrm{Str}(s_q, \tilde{r}, v; t_q)\}\},$$
$$\alpha = \phi\,\tilde{p}(v), \qquad \beta = \phi\,(1-\tilde{p}(v)). \tag{A.1}$$

The novelty score is the negative log-survival function:

$$\begin{aligned}\mathrm{Nov}(s_q, \tilde{r}, v; t_q) &= -\log \Pr(X \geq c_s(v)) \\ &\approx -\log \sum_{k=c_s(v)}^{C_s} \exp\Big(\log \binom{C_s}{k} + \log \mathcal{B} \\ &\qquad (\alpha + k, \beta + C_s - k) - \log \mathcal{B}(\alpha, \beta)\Big),\end{aligned} \tag{A.2}$$

where $\mathcal{B}(\cdot,\cdot)$ is the Beta function. We compute Eq. A.2 in log-space for numerical stability and clip the survival probability by $\Pr(X \geq c_s(v)) \leftarrow \max(\Pr(X \geq c_s(v)), \delta)$ with $\delta = 10^{-6}$.

A.2 LLM usefulness scoring and dynamic threshold filtering

Strength and novelty scores are lightweight and model-free, but they can still identify events that are statistically salient yet semantically unhelpful for the specific query. We therefore adopt AnRe's probability-based usefulness scoring and dynamic threshold filtering (Tang et al., 2025) as a semantic gate.

Time-binned candidate pools. We partition the query-centered directed history $\tilde{\mathcal{H}}(s_q)$ into K time bins by quantiles of timestamps (most recent to oldest), denoted as $\{\mathcal{B}_j(q)\}_{j=1}^{K}$. To respect context budget, in each bin we preselect at most M_{use} events (e.g., by $(\text{Nov}\downarrow, \text{Str}\downarrow, t\downarrow)$) and only ask the LLM to judge usefulness on this short list.

Usefulness distribution from label-token likelihoods. For a bin $\mathcal{B}_j(q) = \{e_1, \ldots, e_F\}$ (after the preselection), we verbalize each event and assign it a numeric label. We use a structured prompt θ_{use} (Appendix D) that asks the LLM to *select the single most helpful historical event* for answering the query. Let s_ℓ be the logit of the label-token for event e_ℓ under the LLM. We convert these logits into a probability distribution:

$$p_{\text{use}}(e_\ell \mid q, j) = \frac{\exp(s_\ell)}{\sum_{k=1}^{F} \exp(s_k)} \tag{A.3}$$

Although the prompt requests one choice, the full softmax distribution provides a graded notion of usefulness, which we use for filtering.

Dynamic threshold filtering. Usefulness should be judged more strictly for events further away from the query time. Let t_q be the query time, and let $t^{\min}$ be the oldest timestamp in $\tilde{\mathcal{H}}(s_q)$, so that $T = t_q - t^{\min}$ is the maximum time span. For bin j with representative time t_j (e.g., the median timestamp in the bin), define $\Delta t_j = t_q - t_j$ and let $F = |\mathcal{B}_j(q)|$. We set a dynamic confidence threshold:

$$c_j = \frac{1}{F} + \left(1 - \frac{1}{F}\right)\left(\frac{\Delta t_j}{T}\right)^{\zeta}, \tag{A.4}$$

where $\zeta > 0$ controls how fast the threshold increases for older bins. We keep event $e \in \mathcal{B}_j(q)$ if $p_{\text{use}}(e \mid q, j) \geq c_j$. This yields a filtered pool $\tilde{\mathcal{H}}_{\text{use}}(s_q) \subseteq \tilde{\mathcal{H}}(s_q)$ which is then passed to Pareto-greedy selection (Algorithm 1).

Algorithm 1 Usefulness-Gated Pareto-greedy Evidence Selection

Require: Directed history $\tilde{\mathcal{H}}(s_q)$, query q, budget B
Require: Time bins K, per-bin cap b_{cap}, relation cap ratio ρ_{rel}, redundancy cap ρ_{cap}
Compute $\text{Str}(e), \text{Nov}(e)$ for all $e \in \tilde{\mathcal{H}}(s_q)$
Partition into $\{\mathcal{B}_j(q)\}_{j=1}^{K}$; compute $p_{\text{use}}(e \mid q, j)$; keep e if $p_{\text{use}}(e \mid q, j) \geq c_j$ (Eq. A.4)
$\tilde{\mathcal{H}}_{\text{use}}(s_q) \leftarrow$ kept events
Assign Pareto rank $r(e)$ to all $e \in \tilde{\mathcal{H}}_{\text{use}}(s_q)$ (Level 1 = non-dominated, Level 2 = non-dominated in remainder, etc.)
$P_q \leftarrow \text{Sort}(\tilde{\mathcal{H}}_{\text{use}}(s_q),\ \text{key} = (r(e)\uparrow,\ \text{Nov}\downarrow,\ \text{Str}\downarrow,\ t\downarrow))$
$G_q \leftarrow \emptyset$
Initialize counters $Count_{\text{bin}}[\cdot]$, $Count_{\text{rel}}[\cdot]$, and $Count_{\text{pair}}[\cdot,\cdot]$ to 0
ConstraintsSatisfied enforces time coverage, per-relation diversity, and anti-redundancy (Section A.3)
for $e \in P_q$ **do**
  if $|G_q| < B$ **and** ConstraintsSatisfied(e) **then**
    $G_q \leftarrow G_q \cup \{e\}$; update all counters
  end if
end for
return G_q

A.3 Deterministic constraints in ConstraintsSatisfied

In Algorithm 1, we implement ConstraintsSatisfied with three deterministic rules under a fixed event budget B. First, **time coverage**: each selected event e belongs to a time bin $j(e) \in \{1, \ldots, K\}$, and we enforce a per-bin cap b_{cap} such that $|\{e \in G_q : j(e) = j\}| \leq b_{\text{cap}}$ for all j (with the empty-bin rollover described in Appendix D.1). Second, **per-relation diversity**: letting $\tilde{r}(e)$ denote the directed relation of event e, we cap the number of selected events per relation by $|\{e \in G_q : \tilde{r}(e) = \tilde{r}\}| \leq \lceil \rho_{\text{rel}} \cdot B \rceil$ for all $\tilde{r} \in \tilde{\mathcal{R}}$. Third, **anti-redundancy**: for each ordered pair $(\tilde{r}, v)$, we cap repeated evidence about the same neighbor by $|\{e \in G_q : \tilde{r}(e) = \tilde{r},\ v(e) = v\}| \leq \rho_{\text{cap}}$. These constraints are applied greedily and prevent a single time region, relation, or neighbor from dominating the prompt.

A.4 Pseudocode: Usefulness-Gated Pareto-greedy Evidence Selection

Algorithm 1 provides pseudocode for evidence selection, including strength/novelty scoring, usefulness filtering, Pareto ranking, and greedy assembly under budget and diversity constraints.

A.5 Relation-level hierarchical backoff for strength under extreme sparsity

When the query-specific long window is extremely small, we optionally blend the query-specific estimate with a relation-level prior. Let $c_{\text{rel}}(v)$ be the count of neighbor v under relation $\tilde{r}$ across all interactions before t_q, with total $C_{\text{rel}} = \sum_v c_{\text{rel}}(v)$ and support $V_{\tilde{r}} = \{v \mid c_{\text{rel}}(v) > 0\}$. We define a smoothed relation-level distribution

$$\pi_{\text{rel}}(v \mid \tilde{r}) = \frac{c_{\text{rel}}(v) + \eta}{C_{\text{rel}} + \eta|V_{\tilde{r}}|}, \tag{A.5}$$

where $\eta > 0$ is a small smoothing constant. With sparsity threshold τ and backoff mass $\lambda \geq 0$, set $\lambda_{q,\tilde{r}} = \lambda \cdot \mathbb{I}[C_\ell < \tau]$. The backoff strength score is

$$\text{Str}_{\text{backoff}}(s_q, \tilde{r}, v; t_q) = \frac{c_\ell(v) + \alpha_0 + \lambda_{q,\tilde{r}}\, \pi_{\text{rel}}(v \mid \tilde{r})}{C_\ell + \alpha_0|V_{q,\tilde{r}}| + \lambda_{q,\tilde{r}}} \tag{A.6}$$

A.6 Drift detection and adaptive long-window shrinkage

Fixed-count windows assume a recent "steady state"; relation patterns may drift over time. We use a lightweight drift detector (KL divergence between the short-window empirical distribution and the relation-level historical distribution) to optionally shrink the long window when drift is detected. Specifically, if the KL divergence exceeds a threshold δ_{drift} (only checked when $C_\ell \geq \tau$), we shrink the effective long window size to $N'_\ell = \max\{N_s,\ \lfloor \gamma N_\ell \rfloor\}$ with $\gamma \in (0, 1)$. This prevents stale inertia from dominating the ranking during regime shifts.

A.7 Demonstration selection details

We set the minimum history threshold $H_{\min} = 300$ to ensure sufficient data for inertia/shift profiling of demonstrations. Let $H(\cdot)$ denote Shannon entropy (natural log) and $\sigma(x) = 1/(1 + \exp(-x))$ denote the logistic function. We define the *structural regularity* R_{struct} as $1 - H(\mathcal{R})/H_{\max}$, where $H(\mathcal{R})$ is the entropy of the relation distribution in the training set and $H_{\max} = \log|\mathcal{R}|$. We compute this on the original relation vocabulary $\mathcal{R}$ (excluding inverse relations) to capture dataset-level regularity rather than direction-specific artifacts. High entropy indicates weaker structural regularity (thus relying more on semantics, $\beta \to 1$); low entropy indicates stronger regularity (thus relying more on structure, $\beta \to 0$). We map this using a shifted sigmoid:

$$\beta = \sigma\big(\gamma \cdot (1 - R_{\text{struct}}) - \mu\big), \tag{A.7}$$

where we empirically set scale $\gamma = 5$ and shift $\mu = 2$. For semantic similarity, we use cosine similarity between frozen BERT `[CLS]` embeddings of the verbalized question sentence for q and the masked question sentence for e_a (encoder: `bert-base-uncased`; Devlin et al. 2019). For structural similarity, we compute low-dimensional inertia–shift profiles from the same (Str, Nov) signals (Section 3.2). Concretely, for each query-centered directed relation $(s, \tilde{r}_q)$ we compute a feature vector:

$$\mathbf{u}(s, \tilde{r}_q) = \begin{bmatrix} H(\text{Str}),\ \max_v \text{Str}(v),\ \max_v \dfrac{c_s(v)}{C_s}, \\ \max_v \text{Nov}(v),\ \text{mean}_v \text{Nov}(v) \end{bmatrix}, \tag{A.8}$$

where $H(\text{Str})$ is the entropy of the smoothed long-window strength distribution over neighbors v. We then set $\text{Sim}_{\text{str}}(q, e_a) = \cos(\mathbf{u}(s_q, \tilde{r}_q), \mathbf{u}(s_a, \tilde{r}_q))$. The adaptive weight β is computed using a shifted sigmoid based on dataset structural regularity R_{struct} (Table 9).

Variant (ICEWS14)	MRR	H@1	H@3	H@10
No drift adaptation	0.465	0.352	0.475	0.605
Hard shrink (LANTERN)	**0.480**	**0.375**	**0.515**	**0.665**
Soft mixing	0.475	0.365	0.490	0.620

Table 7: Comparison of drift adaptation strategies.

B Additional Experimental Results and Analyses

Unless stated otherwise, analyses in this appendix use a stratified sample of 500 test queries (following Tang et al. (2025)) to control LLM inference cost.

B.1 Analysis: Responding to Common Concerns

Drift handling variants. Table 7 shows that **Hard shrink** performs best, suggesting removing stale history can be more effective than soft mixing when drift is detected.

Window semantics. We discuss the limitation of fixed-count windows in the Limitations section and leave a systematic comparison to time-span

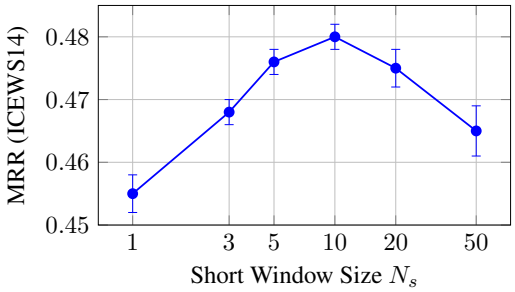

Figure 4: Sensitivity to short-window size N_s on ICEWS14 (mean $\pm$ std over 3 runs). Performance peaks at $N_s \approx 10$, balancing burst detection sensitivity and statistical robustness.

windows to future work; this keeps the current paper focused on the revised presentation of the same core method rather than introducing a new windowing framework.

Sensitivity to N_s (novelty stability). We vary the short-window size N_s in $\{1, 3, 5, 10, 20, 50\}$ and report MRR on ICEWS14. Figure 4 shows an inverted U-shaped curve. Very small windows ($N_s < 3$) are too noisy, while large windows ($N_s > 20$) dilute the novelty signal into a second strength signal. Regarding N_ℓ, performance is stable for $N_\ell \geq 100$, suggesting diminishing returns (and potentially stale noise) from older history.

Candidate-set analysis. Figure 5 analyzes the trade-off between recall and token cost. The 1-hop neighborhood caps oracle recall at ∼65%, while 2-hop (default) reaches ∼90% with manageable token cost (3,200 tokens); under the same hop setting, AnRe consumes ∼5,100 tokens. This analysis also clarifies that the candidate-set ceiling is shared by current training-free TKGF protocols: our contribution is to rank more effectively *within* this bounded set, not to change the candidate generator itself. Importantly, Figure 5 and our rare-answer analysis together suggest that most of our gains occur on rare or near-cold-start entities that are still recalled by this candidate set, rather than on frequent entities that are already easy to rank.

Case Study Table 8 presents a qualitative example illustrating how LANTERN counterbalances long-window alliance inertia with short-window novelty, aided by a structure-aware analogical demonstration.

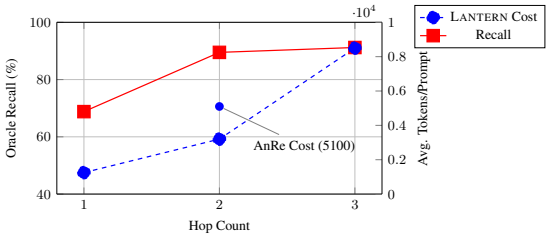

Figure 5: Trade-off between Oracle Recall (left axis) and Token Cost (right axis) on ICEWS14. The 2-hop setting captures ∼90% of answers. Crucially, at Hop=2, LANTERN (3.2k tokens) is significantly more efficient than AnRe (5.1k tokens, marked) due to Pareto filtering.

Field	Content
Query	(2018-06-15) France signs a clean energy cooperation agreement with which country?
Answer	Morocco (**regime shift**: emerging clean energy partnership overriding traditional alliance inertia).
Long-window inertia (2017–2018 Q1)	(2017-09-08) France–Germany launch gas pipeline project; (2017-11-12) sign cross-border electricity grid cooperation; (2018-02-20) announce fossil fuel efficiency research.
Short-window novelty (2018 Q2)	(2018-05-03) France trains Morocco on photovoltaic tech; (2018-05-28) high-level renewable energy talks; (2018-06-09) $200M loan for Morocco's solar plants.
Traditional selection	Retrieves 3 France–Germany energy cases, reinforcing inertial France-Germany energy association.
LANTERN selection	Selects 1 **structure-aware analogical** demo (Spain-Tunisia wind agreement: shifted from Portugal to North African renewables) + 2 key France-Morocco renewable interactions.
Traditional prediction	Ranks: 1) Germany (0.42), 2) Spain (0.18), ..., 4) Morocco (0.07) → wrong (over-reliance on long-term alliance inertia).
LANTERN prediction	Ranks: 1) Morocco (0.41), 2) Germany (0.29) → correct: analogical demo legitimizes "shift to emerging partners"; short-window confirms Morocco as focus.

Table 8: Case study on ICEWS18: evidence selection under traditional alliance inertia and cooperation shifts.

C Dataset Statistics, Component Origins, and Hyperparameters

We provide detailed statistics for all datasets in Table 9 and list the default hyperparameters used in our experiments in Table 11.

D Verbalization and Prompt Templates

We move the prompt templates to this dedicated section for clarity. We verbalize each event tuple (s, r, o, t) into a sentence and map entities/candidates to numeric labels to avoid multi-

Dataset	#Ent	#Rel	Train	Valid	Test	R_{struct}
ICEWS14	12,498	260	323,895	–	341,409	0.72
ICEWS05-15	10,094	251	368,868	46,302	46,159	0.69
ICEWS18	23,033	256	373,018	45,995	49,545	0.65
GDELT	7,691	240	1,734,399	238,765	305,241	0.31

Table 9: Dataset statistics and computed structural regularity R_{struct}, following the AnRe preprocessing. ICEWS14 does not use an official validation split; we tune on a held-out subset of its training data. In our analysis, lower R_{struct} indicates weaker structural regularity and therefore a more *noisy* environment in the sense used in Section 4.3.

ϕ (Concentration)	0.5	1.0	**2.0 (Def.)**	5.0
ICEWS18 MRR	0.355	0.362	**0.365**	0.360

Table 10: Sensitivity to Overdispersion ϕ. $\phi = 2.0$ offers the best balance, but the method is generally robust.

Hyperparameter	Default
N_ℓ (long window size)	100
N_s (short window size)	10
α_0 (Dirichlet smoothing)	1.0
B (evidence budget)	100
K (time bins)	4
b_{cap} (per-bin cap)	25
ρ_{rel} (per-relation cap ratio)	0.3
ρ_{cap} (redundancy cap)	3
M_{use} (LLM-use list cap per bin)	25
ζ (DTF exponent for usefulness)	2.75
β (demo score weight)	Adaptive
R_{struct} (dataset structural regularity)	Computed
τ (sparsity threshold)	5
λ (hierarchical backoff mass)	0.3
η (relation-prior smoothing)	0.1
δ_{drift} (drift threshold)	0.5
γ (drift shrink factor)	0.5
ϕ (Beta-Binomial concentration)	2.0
$C_{\max}$ (candidate cap)	100
δ (SF floor)	10^{-6}
$H_{\min}$ (min demo history)	300
Encoder (for Sim_{sem})	`bert-base-uncased`

Table 11: Default hyperparameters used across datasets. We use one shared configuration on all four benchmarks. Once the budget is fixed, most entries are engineering constants; the main behavior of LANTERN is governed by a small set of core parameters, notably N_ℓ, N_s, α_0, and ϕ. Parameters such as ζ and ϕ were selected on a held-out subset of the ICEWS14 training set.

token ambiguity and simplify inference.

D.1 Reproducibility Checklist

We provide the exact configuration details:

- **Novelty Calculation:** We compute Beta-function terms with `scipy.special.betaln` (SciPy 1.0 Contributors, 2020) and pre-cache the Beta-Binomial log-PMF for $n \in [1, 50]$. We set $\phi = 2.0$ by default and find it robust for $\phi \in [1, 5]$ (Table 10).
- **Binning Fallback:** In Algorithm 1, if a time quantile bin is empty (due to sparse history), its budget is rolled over to the most recent bin.
- **LLM Settings:** Temperature = 0 and top-p = 1.0. We use the official `Hugging Face` tokenizers (Wolf et al., 2020).
- **Token Budgeting:** "100 events" is an approximate cap. We enforce a soft cap of ∼3500 tokens for the final ranking prompt by truncating the oldest selected evidence when needed, and report the resulting peak prompt length in Table 4.
- **Prompt Design Rationale:** Numeric labels and single-number outputs reduce multi-token ambiguity in label scoring, keep decoding deterministic, and make the usefulness-gate / final-ranking prompts easier to parse consistently across models.
- **Computing Infrastructure:** All experiments were conducted on NVIDIA RTX 3090 GPUs.

Component	Template (Example)
Event sentence	On {date}, {subj} {relation-phrase} {obj}.
Query	On {date}, {subj} {relation-phrase} which entity?
Candidate list	Candidates: 0.EntityA, 1.EntityB, ...
Usefulness prompt (θ_{use})	There is a question and some historical events. Please select the single most helpful historical event for answering the question and output only its number. Question: {query} Historical events: {label:event} Your choice is:
Replay explanation prompt (θ_{arp})	Here are some time-ordered historical event chains and a question-answer pair. Please explain how the answer is implied by the historical development in the event chain. Historical Events: {G_a} Question: {q_a} Answer: {o_a} Your Analysis:

Table 12: Verbalization and prompt templates. Numeric labels let each candidate or event map to a single token when possible, reducing label ambiguity; single-label outputs also simplify inference and avoid noisy post-processing.