DOES FUTURE LINK PREDICTION ON TEMPORAL TEXT-ATTRIBUTED GRAPHS NEED HEAVYWEIGHT MODELING?

Anonymous authors
Paper under double-blind review

ABSTRACT

Future link prediction on Temporal Text-Attributed Graphs (TTAGs) increasingly relies on heavyweight models that process fine-grained interaction histories with temporal Transformers or large language models (LLMs). This modeling complexity incurs substantial computational cost, yet it remains unclear how much is necessary. We explore this question empirically and find that our lightweight Snapshot GNN built on reusable temporal snapshots and inexpensive global heuristics remains surprisingly competitive with substantially heavier models, while being much more efficient. To explain this result, we derive a theoretical bound on the optimal AUC advantage of the full temporal history over a compressed additive summary. Guided by the analysis, we perform controlled experiments to separate historical evidence from processing machinery and observe limited returns from both longer, finer-grained histories and heavier graph processing. LLMs still retain an advantage given comparable historical information, and graph models require additional structural cues to close the gap. This motivates us to examine where the remaining LLM advantage arises. We discover that GNN uncertainty alone can localize the predictions on which the LLM is particularly beneficial, which tend to correlate with weak structural support from the graph. When used for selective LLM invocation, these localized gains are substantially larger than their aggregate effect on global performance, while uncertainty remains highly competitive with learned localizers. Overall, our results suggest that future link prediction on TTAGs can often achieve strong accuracy at considerably lower computational cost across real-world benchmarks, with LLMs providing localized gain where graph models are least certain. Our code is available at `https://anonymous.4open.science/r/dyg-6686`

1 INTRODUCTION

Temporal graphs provide a natural abstraction for systems whose interactions evolve over time, such as knowledge graphs, communication networks, and recommendation systems (Holme & Saramäki, 2011). Beyond their dynamic connectivity, many temporal graphs are also associated with rich textual information, forming what are commonly known as Temporal Text-Attributed Graphs (TTAGs) (Zhang et al., 2024). Compared with structure-only temporal graphs, TTAGs provide an additional source of information: textual attributes, while the evolving graph captures how such information is organized and propagated over time. A fundamental task on TTAGs is future link prediction, which aims to predict future interactions from historical structural and semantic evidence and underlies applications such as interaction prediction and recommendation (Qin & Yeung, 2024).

Existing approaches to future link prediction on TTAGs broadly fall into two paradigms. The first builds upon temporal graph neural networks (TGNNs): textual attributes are encoded by pretrained language models such as BERT-style encoders, and the resulting embeddings are incorporated into temporal graph architectures (Zhang et al., 2024; Kumar et al., 2019; Trivedi et al., 2019; Rossi et al., 2020; Wang et al., 2021b;a; Cong et al., 2023; Yu et al., 2023). Representative methods such as DyGFormer (Yu et al., 2023) formulate temporal neighborhoods as interaction sequences and employ Transformer-based sequence modeling. More recent work places large language models (LLMs) more directly at the center of semantic modeling (Zhang et al., 2025; Wang et al., 2025;

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

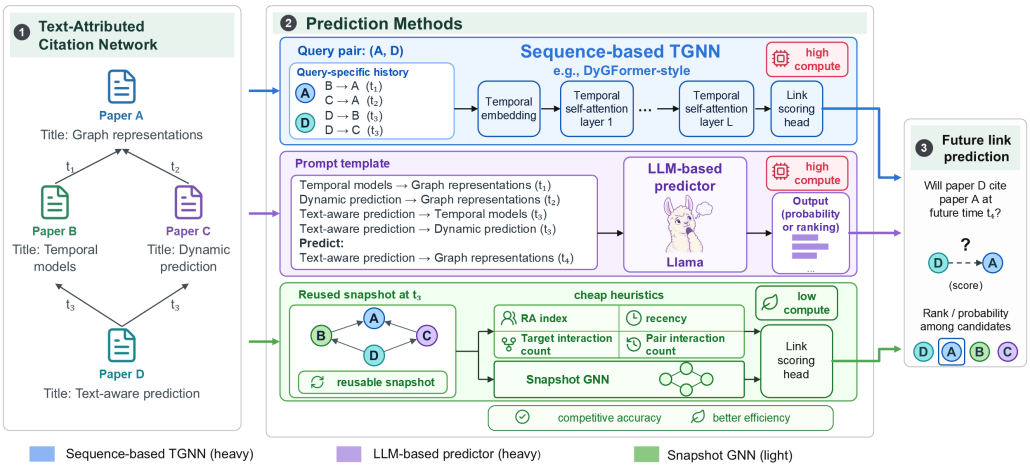


Figure 1: Three approaches to TTAG future link prediction: sequence-based TGNNs, direct LLM predictors, and our lightweight Snapshot GNN with inexpensive heuristics.

Roy et al., 2025; Lei et al., 2025; Huang et al., 2025). Some methods repeatedly summarize or update semantic representations over successive temporal windows using LLMs; others go further and construct LLM-based reasoning or agentic systems that directly predict future links from textualized graph histories. Together, these paradigms exemplify computation-intensive approaches to combining temporal structure with textual semantics for TTAG prediction.

Despite their modeling differences, both paradigms incur notable computational overhead, making **Efficiency** a central concern. Sequence-based TGNNs achieve expressive temporal modeling by processing complex interaction histories (Cong et al., 2023; Yu et al., 2023). LLM-based approaches introduce an even heavier semantic modeling pipeline. Repeatedly updating temporal semantic representations requires invoking an LLM throughout the graph timeline (Zhang et al., 2025; Wang et al., 2025), while LLM-as-predictor systems perform expensive reasoning for each query (Huang et al., 2025). Such overhead may be justified if the additional computation yields commensurate predictive gains. However, it remains unclear how much of this heavyweight machinery is necessary for accurate future link prediction. In particular, do we need to preserve increasingly fine-grained historical evidence and process it with increasingly heavy temporal or language-based models, or can much lighter-weight modeling recover most of the same predictive value? This motivates our central question: **Does future link prediction on TTAGs need heavyweight modeling?**

We begin by examining how far a lightweight design can go. We refer to this design as the **Snapshot GNN**. Rather than constructing and processing a separate fine-grained interaction sequence for every query, it represents temporal structure using reusable graph snapshots and applies a simple convolutional backbone, complemented by inexpensive global heuristic signals that summarize broader historical patterns. This design discards much of the query-specific sequence modeling employed by heavyweight TGNNs and avoids invoking LLMs as the default prediction mechanism. Figure 1 provides an overview of two heavyweight paradigms in prior work (Yu et al., 2023; Huang et al., 2025), alongside our Snapshot GNN. **Surprisingly, the predictor remains competitive with heavyweight models across real-world TTAG benchmarks at substantially lower end-to-end cost**. Across all eight settings, its average gap to the overall best is only 0.0019 AUC, with a maximum gap of 0.0057. Meanwhile, it achieves a 5–111$\times$ end-to-end runtime speedup over representative heavyweight baselines on GDELT and Enron (Zhang et al., 2024). This suggests that heavyweight machinery may provide less predictive value than its cost implies, motivating us to investigate why lightweight models perform so well.

To understand this behavior, we derive a theoretical bound on the optimal AUC advantage of the full temporal history over an additive summary. The bound separates errors from imperfect historical representation and history-dependent event effects. This motivates a controlled empirical study that isolates the effects of the **Historical Evidence** available and the **Processing Machinery** used to exploit it. We vary historical scope and fidelity, from heuristic summaries to snapshots and interaction sequences, and **find limited returns from longer histories and finer temporal detail**. We then

compare processing machinery given comparable historical evidence: mean pooling, exponential decay, graph convolution, and self-attention differ by at most 0.0038 AUC on GDELT and 0.0100 on Enron. These results suggest that coarse historical evidence and lightweight processing already capture much of the predictive value. Nevertheless, the entity-aware LLM remains substantially stronger. Much of this advantage disappears under anonymization, while graph-based processors recover most of the gap after adding inexpensive neighbor co-occurrence signals(Yu et al., 2023) and global structural heuristics. The resulting aggregate similarity may still mask differences across queries, motivating a closer look at where the LLM retains additional value.

We therefore characterize this residual advantage. A simple margin-based measure of GNN uncertainty (Loveland et al., 2025) identifies regions where the LLM is particularly beneficial, capturing 71.8–96.5% of LLM corrections at 20% coverage. These regions are also associated with weaker graph support such as fewer common neighbors. We next ask whether these localized gains translate into dataset-level improvements. For selected queries, we combine GNN and LLM predictions while retaining GNN predictions elsewhere. We find that the resulting dataset-level gain is relatively small, indicating that the benefit of selective LLM invocation is localized. At the same time, simple uncertainty remains competitive with learned localizers, whose stronger gains during training do not consistently transfer to test. **Together, these results suggest that GNN uncertainty is a practical signal for targeted LLM use, whose added value is concentrated in a subset of queries**.

Our main contributions are summarized as follows:

- **We find limited returns from heavyweight modeling across real-world TTAG benchmarks.** Our Snapshot GNN with inexpensive global heuristics remains competitive with substantially heavier TGNN and LLM-based models at much lower cost.
- **We explain why lightweight modeling works.** A theoretical bound motivates separating historical evidence from processing machinery. Controlled experiments show limited returns from finer-grained evidence and heavier processing; structural cues close much of the LLM gap.
- **We characterize where LLMs add value.** Their benefit concentrates on queries identified by GNN uncertainty, which tend to exhibit weaker graph support, while simple uncertainty signal remains competitive with learned localizers.

2 Preliminaries and Related Work

2.1 Preliminaries

Temporal Text-Attributed Graphs. A temporal text-attributed graph (TTAG) consists of a set of nodes $\mathcal{V}$ and a sequence of timestamped interactions

$$\mathcal{E} = \{e_i = (u_i, v_i, t_i)\}_{i=1}^{M}, \qquad t_1 \le t_2 \le \cdots \le t_M,$$

where $u_i, v_i \in \mathcal{V}$ are the interacting nodes and t_i is the interaction timestamp. Each node u has a textual attribute d_u. Let $\mathcal{G}_{<t}$ denote the TTAG observed before t.

Future Link Prediction. Given a candidate pair (u, v), query time t, and history $\mathcal{G}_{<t}$, future link prediction estimates their interaction likelihood at t. A predictor produces a score

$$s_\theta(u, v, t \mid \mathcal{G}_{<t}),$$

where larger scores indicate higher interaction likelihood. Under standard ranking-based evaluation, the model should rank the future interaction above sampled negative candidates (Zhang et al., 2024).

2.2 Related Work

Temporal Graph Neural Networks. TGNNs extend GNNs to evolving interaction graphs by modeling time-aware neighborhoods or dynamic node states. Representative methods include JODIE (Kumar et al., 2019), DyRep (Trivedi et al., 2019), TGAT (Xu et al., 2020), CAWN (Wang et al., 2021b), TCL (Wang et al., 2021a), GraphMixer (Cong et al., 2023) and DyGFormer (Yu et al., 2023). A notable recent approach, DyGFormer (Yu et al., 2023), models queried nodes' first-hop interaction sequences using neighbor co-occurrence encoding and a patched Transformer, with patching improving long-history modeling efficiency. However, prediction still requires constructing fine-grained, query-specific interaction histories and applying costly self-attention. Our work revisits the predictive benefit of such fine-grained temporal modeling when text is also available.

Modeling for Temporal Text-Attributed Graphs. TTAGs extend temporal graphs with rich textual attributes, requiring models to jointly capture evolving structures and semantics. Early adaptations typically encode raw texts into fixed features for existing TGNNs. DTGB (Zhang et al., 2024) provides a systematic TTAG benchmark, encoding textual attributes into dense embeddings as TGNN features. Recent methods model semantic dynamics explicitly. CROSS (Zhang et al., 2025) uses LLMs to extract evolving semantics and jointly encodes them with graph structures, while DyGRASP (Wang et al., 2025) further reasons over recent and global semantic dynamics with LLMs. More recently, LLMs have begun to be used directly for temporal link prediction. TGTalker (Huang et al., 2025) textualizes temporal neighborhoods and queries a pretrained LLM to predict future links. While these approaches demonstrate the potential of LLMs for temporal graph prediction, they also introduce substantial language-model computation into the prediction pipeline. Our work instead asks how much of this heavyweight semantic modeling is actually necessary.

Lightweight Graph Modeling. Recent work also explores lower-cost ways to incorporate semantics into graph prediction. MoMent (Xu et al., 2026) develops an efficient multimodal architecture to jointly model temporal, structural, and textual information. Beyond temporal graphs, GLANCE (Loveland et al., 2025) adaptively combines GNNs and LLMs for text-attributed graph prediction, and Refined-GAE (Ma et al., 2025) more broadly shows that carefully designed lightweight graph models can rival much more complex alternatives. Our work instead tests a more lightweight yet competitive predictor and focuses on explaining the factors behind its strong performance.

3 LIGHTWEIGHT TTAG MODELING

3.1 THE SNAPSHOT GNN

We begin by testing whether a deliberately lightweight design can remain competitive with heavyweight temporal and LLM-based models. We construct the Snapshot GNN, a simple predictor that avoids query-specific sequence modeling and routine LLM inference, while retaining the major sources of structural and semantic evidence. Our goal at this stage is not to introduce a new architecture, but to establish whether heavyweight modeling is necessary for competitive performance.

Snapshot-based Temporal Encoding. We batch prediction queries and deliberately use a shared historical cutoff for all queries in each batch. Specifically, for a batch starting at t_b, the snapshot encoder accesses only interactions observed before t_b and keeps its graph representation fixed throughout the batch. This strictly causal design allows all batch queries to share one temporal snapshot, at the cost of withholding some interactions already observable to later queries under query-specific history construction. Given the cutoff t_b, we build the snapshot from interactions in a preceding window of length W and collapse repeated interactions into a weighted graph $\mathcal{G}_{t_b}^{(W)}$. To retain coarse temporal information, the weight between nodes u and v is

$$w_{uv}(t_b) = \sum_{\tau \in \mathcal{T}_{uv}^{(W)}(t_b)} \exp[-\lambda(t_b - \tau)]\,. \tag{1}$$

where $\mathcal{T}_{uv}^{(W)}(t_b)$ contains their interactions within the historical window. By abstracting away fine-grained event ordering and fixing the representation within each batch, the snapshot needs to be constructed and encoded only once and can be reused across all batch queries, trading a modest loss of immediately available evidence for substantially reduced query-specific computation.

Lightweight Graph Encoder. Given the temporally weighted snapshot, we use a shallow Graph Isomorphism Network (GIN) (Xu et al., 2019) to aggregate local structural and semantic information, with pretrained E5 (Wang et al., 2022) embeddings $\mathbf{X}$ as initial node features. For a queried pair (u, v), we concatenate its endpoint representations to obtain a pair representation,

$$\begin{aligned} \mathbf{H} &= \mathrm{GIN}\Big(\mathcal{G}_{t_b}^{(W)}, \mathbf{X}\Big)\,, \\ \mathbf{z}_{uv}^{\mathrm{snap}} &= [\,\mathbf{h}_u \| \mathbf{h}_v\,]. \end{aligned} \tag{2}$$

where $\mathbf{h}_u$ and $\mathbf{h}_v$ are the GIN representations of the queried endpoints. The encoder contains no query-specific sequence modeling; it simply extracts a reusable structural-semantic representation from the shared temporal snapshot. We deliberately keep this component lightweight, leaving broader query-specific historical signals to the inexpensive heuristics introduced next.

Table 1: Transductive and inductive future link prediction AUC-ROC. Per-column best and second-best means are **bold** and underlined, respectively.

Method	Transductive				Inductive			
	GDELT	ICEWS1819	Enron	Googlemap_CT	GDELT	ICEWS1819	Enron	Googlemap_CT
Temporal Graph Models								
GraphMixer (Cong et al., 2023)	0.9637 ± 0.0013	0.9869 ± 0.0014	0.9625 ± 0.0030	0.8130 ± 0.0012	0.8815 ± 0.0015	0.9646 ± 0.0013	0.8719 ± 0.0096	0.7615 ± 0.0019
DyGFormer (Yu et al., 2023)	0.9714 ± 0.0001	0.9907 ± 0.0013	0.9766 ± 0.0004	0.9158 ± 0.0112	0.9068 ± 0.0016	0.9731 ± 0.0050	0.9283 ± 0.0031	0.8937 ± 0.0140
DyGFormer + Heuristics	$\underline{0.9735 \pm 0.0001}$	0.9906 ± 0.0009	0.9794 ± 0.0013	$\underline{0.9159 \pm 0.0038}$	0.9141 ± 0.0006	0.9714 ± 0.0032	0.9348 ± 0.0011	$\underline{0.8947 \pm 0.0057}$
TTAG Models								
MoMent (Xu et al., 2026)	0.9673 ± 0.0001	0.9912 ± 0.0001	0.9697 ± 0.0006	0.8302 ± 0.0003	0.8848 ± 0.0001	0.9703 ± 0.0003	0.8864 ± 0.0015	0.7776 ± 0.0006
CROSS (Zhang et al., 2025)	0.9493 ± 0.0119	0.9915 ± 0.0004	0.9716 ± 0.0056	$> 2\,\mathrm{d}$	0.8054 ± 0.0483	0.9764 ± 0.0011	0.9230 ± 0.0078	$> 2\,\mathrm{d}$
DyGRASP (Wang et al., 2025)	0.9660 ± 0.0003	0.9881 ± 0.0007	0.9688 ± 0.0017	$> 2\,\mathrm{d}$	$\mathbf{0.9190 \pm 0.0008}$	0.9717 ± 0.0028	0.9095 ± 0.0067	$> 2\,\mathrm{d}$
LLM-based Predictors								
TGTalker (Huang et al., 2025)	0.7866 ± 0.0012	0.9044 ± 0.0015	0.7512 ± 0.0005	$> 2\,\mathrm{d}$	0.6627 ± 0.0003	0.7467 ± 0.0005	0.8932 ± 0.0014	$> 2\,\mathrm{d}$
Llama-8B†	0.8959 ± 0	0.9602 ± 0	0.9406 ± 0	$> 2\,\mathrm{d}$	0.8152 ± 0	0.9263 ± 0	0.8565 ± 0	$> 2\,\mathrm{d}$
Llama-8B-FT†	0.9728 ± 0.0004	$\mathbf{0.9932 \pm 0.0002}$	$\mathbf{0.9822 \pm 0.0018}$	$> 2\,\mathrm{d}$	0.9136 ± 0.0011	$\mathbf{0.9803 \pm 0.0008}$	$\mathbf{0.9431 \pm 0.0017}$	$> 2\,\mathrm{d}$
Snapshot GNN	$\mathbf{0.9736 \pm 0.0004}$	$\underline{0.9920 \pm 0.0001}$	$\underline{0.9819 \pm 0.0002}$	$\mathbf{0.9321 \pm 0.0001}$	$\underline{0.9148 \pm 0.0015}$	$\underline{0.9765 \pm 0.0004}$	$\underline{0.9374 \pm 0.0009}$	$\mathbf{0.9151 \pm 0.0001}$

† Direct-LLM baselines implemented by us; Llama-8B-FT additionally uses our task-specific fine-tuning.

Global Heuristics. We further complement the snapshot representation with inexpensive historical statistics. Inspired by recent observations that simple heuristics can be surprisingly strong for temporal link prediction (Cornell et al., 2025), we consider signals capturing complementary aspects of the interaction history: *recency*, the *number of previous interactions* between the queried pair, the target node's *historical interaction count*, and the *Resource Allocation (RA) score*. All heuristics are computed causally at query time t using only past interactions, but may cover a broader history than the local snapshot. We concatenate the heuristic vector $\mathbf{r}_{uv}$ with $\mathbf{z}_{uv}^{\mathrm{snap}}$ as input to the MLP:

$$s(u, v) = \mathrm{MLP}([\,\mathbf{z}_{uv}^{\mathrm{snap}} \| \mathbf{r}_{uv}\,])\,. \tag{3}$$

This simple fusion adds current global cues to the reusable snapshot at negligible extra compute. Additional implementation details are provided in Appendix B.

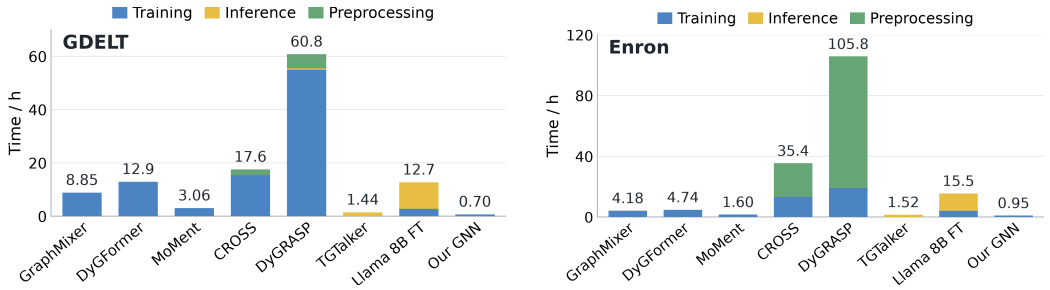
3.2 Lightweight Modeling Is Surprisingly Competitive

We test whether these substantial simplifications significantly hurt prediction. We compare our lightweight predictor with representative temporal graph models, including GraphMixer (Cong et al., 2023) and DyGFormer (Yu et al., 2023); recent TTAG models such as MoMent (Xu et al., 2026), CROSS (Zhang et al., 2025) and DyGRASP (Wang et al., 2025); and LLM-based predictor TGTalker (Huang et al., 2025). We also evaluate vanilla and task-finetuned Llama-3.1-8B-Instruct under the same protocol for direct LLM prediction. Baseline implementations are detailed in Appendix C.

Experimental Protocol. We evaluate all methods on four real-world TTAG benchmarks from DTGB (Zhang et al., 2024): GDELT, ICEWS1819, Enron, and Googlemap_CT. These datasets span knowledge graphs, communication networks, and bipartite recommendation. We follow the DTGB protocol for chronological splitting, negative sampling, and both transductive and inductive evaluation. Applicable methods share E5 embeddings; direct LLM predictors instead consume entity names and textualized causal histories. Our model uses the shared batch-level snapshot cutoff above, while heuristics are updated per query. We report five-seed mean ROC-AUC for all graph and LLM methods. Unless noted otherwise, subsequent experiments follow this protocol. Further details, including the batch-level aggregation, are in Appendix A.

Competitive prediction with lightweight modeling. Table 1 reports the main transductive and inductive results. Despite its deliberately lightweight design, **Snapshot GNN remains highly competitive across all eight settings**, with an average gap of only 0.0019 AUC to the overall best and a maximum gap of 0.0057. Among non-direct-LLM methods, it achieves or ties the best AUC in seven of eight settings and trails the best by only 0.0005 AUC on average. Notably, our task-adapted Llama-8B baseline becomes one of the strongest methods in the comparison, substantially outperforming vanilla Llama, TGTalker and often surpassing recent TTAG baselines. Thus, even against this strong LLM reference, lightweight graph modeling remains close to the accuracy frontier.

A large cost gap for a small performance gap. The similarity in predictive performance contrasts sharply with the difference in computational cost. Figure 2 reports end-to-end preprocessing, training, and inference time on GDELT and Enron. **Our lightweight predictor is the fastest method in both settings.** It is 18.4×/5.0× faster than DyGFormer, 18.1×/16.3× faster than Llama-8B-FT, and 86.3×/111.0× faster than DyGRASP on GDELT/Enron, respectively. Although Llama-8B-FT achieves slightly higher accuracy in several settings, it requires approximately 3,973× and 862× the



(a) GDELT (b) Enron

Figure 2: **Computational cost of TTAG prediction.** Wall-clock time on (a) GDELT and (b) Enron, decomposed into training, inference, and preprocessing. Measurement details are in Appendix D.

inference time of GIN on GDELT and Enron. Such costly inference limits deployment throughput and increases serving costs, whereas our predictor incurs negligible inference overhead.

4 Understanding Lightweight Performance

The competitive performance of our lightweight model raises the question of why such a simple model remains effective. One possible explanation is that simple aggregation already preserves most of the information in the full interaction history needed for future link prediction. To examine this possibility theoretically, we consider additive summaries formed by summing embeddings of timestamped events. Under stated assumptions, we bound the optimal AUC advantage of the full history over such a summary. This analysis then guides our empirical investigation.

Sample an evaluation group B by its averaging weight and then a balanced query label $Y \in \{0, 1\}$, so $\mathbb{P}(Y = 1 \mid B) = 1/2$; all expectations use this distribution. With cutoff 0 and $T > 0$, write the history as $H = \{(m_i, t_i)\}_{i=1}^{N}$, where $t_i \in [-T, 0)$, $a_i = -t_i$, and m_i is the visible event information. The shared context C includes B and textual node representations.

We represent the history's predictive effects through a latent state, using a tractable model that accumulates decaying event contributions with history-dependent corrections. Specifically, let

$$Z_0 = \sum_i e^{Aa_i}\big(v_C(m_i) + r_i\big), \qquad \|r_i\|_2 \le \varepsilon, \tag{4}$$

where $v_C(m_i) \in \mathbb{R}^d$ is event i's contribution given C, and $r_i \in \mathbb{R}^d$ may depend on preceding history. Here $\varepsilon \ge 0$, and $A \in \mathbb{R}^{d\times d}$ satisfies $\|e^{Au}\|_{\mathrm{op}} \le e^{-\beta u}$ for all $u \ge 0$, with decay rate $\beta > 0$. We assume the positive-label probability $\eta(H, C) := \mathbb{P}(Y = 1 \mid H, C)$ depends on the history only through Z_0, given C, so $\eta(H, C) = q_C(Z_0)$. The response $q_C : \mathbb{R}^d \to [0, 1]$ is L-Lipschitz but may be nonlinear; events need not be independent.

We next ask how well the positive-label probability can be approximated from compressed history and C. Encoding each timestamped event as $\phi(m, a) \in \mathbb{R}^p$ gives the additive summary

$$S_\phi(H) = \sum_i \phi(m_i, a_i). \tag{5}$$

To approximate the event-wise component of Z_0, we use linear reconstruction, allowing per-event error bounds to be combined after summation. Assume a linear map $D_C : \mathbb{R}^p \to \mathbb{R}^d$ satisfies

$$\big\|D_C\phi(m, a) - e^{Aa}v_C(m)\big\|_2 \le \rho e^{-\beta a}, \qquad a \in [0, T], \tag{6}$$

for all admissible m, uniformly over C, with $\rho \ge 0$.

We consider histories with uniformly bounded event counts and quantify temporal accumulation by

$$\mathcal{M}_{\beta,T} := \mathbb{E}\left[\left(\sum_i e^{-\beta a_i}\right)^2\right]. \tag{7}$$

For a measurable real-valued score f, let $\mathrm{AUC}_b(f)$ be the probability that a positive query scores above an independently drawn negative query conditional on $B = b$, with half credit for ties. Let $\mathrm{AUC}^*_{H,C}$ and $\mathrm{AUC}^*_{S,C}$ denote the maxima of $\mathbb{E}_B[\mathrm{AUC}_B(f)]$ over all measurable real-valued scoring functions f of (H, C) and $(S_\phi(H), C)$, respectively.

Table 2: AUC across evidence scopes and representations. **H-L/H-G**: local/global heuristic summaries; **Snap-S/L** and **Seq-S/L**: snapshots and sequences with short (47) or long (200) histories.

Dataset	Setting	Heuristic		Snapshot		Sequence	
		Local (H-L)	Global (H-G)	Short (Snap-S)	Long (Snap-L)	Short (Seq-S)	Long (Seq-L)
GDELT	Transductive	0.9331 ± 0.0000	0.9652 ± 0.0002	0.9704 ± 0.0001	0.9707 ± 0.0001	0.9734 ± 0.0001	0.9732 ± 0.0001
	Inductive	0.8424 ± 0.0000	0.8721 ± 0.0007	0.9028 ± 0.0001	0.9038 ± 0.0002	0.9138 ± 0.0008	0.9139 ± 0.0012
Enron	Transductive	0.9622 ± 0.0001	0.9631 ± 0.0000	0.9748 ± 0.0006	0.9749 ± 0.0003	0.9762 ± 0.0013	0.9756 ± 0.0012
	Inductive	0.8825 ± 0.0001	0.8853 ± 0.0001	0.9304 ± 0.0012	0.9313 ± 0.0003	0.9252 ± 0.0022	0.9204 ± 0.0012

Theorem 1 (Bounded advantage of full-history prediction)**.** *For any $\delta \in (0, 1/2)$ satisfying*

$$\sqrt{2}\, L(\varepsilon + \rho)\sqrt{\mathcal{M}_{\beta,T}} \leq \delta,$$

the optimal AUC values obey

$$\boxed{0 \leq \mathrm{AUC}^*_{H,C} - \mathrm{AUC}^*_{S,C} \leq \delta.} \tag{8}$$

The theorem gives a sufficient condition for a compact summary to approach the full history's optimal AUC, given shared context C. Here, ε bounds history-dependent corrections, while ρ controls reconstruction error for decayed event contributions from their embeddings. The proof in Appendix E uses these quantities to bound state approximation errors from omitted history-dependent corrections and imperfect reconstruction of decayed event-wise contributions, respectively. Small ε and ρ therefore limit the full-history advantage when temporal accumulation $\sqrt{\mathcal{M}_{\beta,T}}$ and probability sensitivity L remain controlled. This regime permits dependent events and nonlinear prediction. The appendix additionally establishes low-rank conditions for lossless predictive compression and shows that AUC gaps can vanish even with history-dependent interactions.

The two per-event error terms, ρ and ε, motivate complementary controlled experiments. The representation error ρ motivates examining whether richer historical evidence better preserves event-wise predictive information, while the correction bound ε motivates examining whether more complex processing better captures history-dependent contributions. We therefore first vary **how much historical information is retained**, then fix the history and vary **how its events are processed**.

How much evidence is needed? The first source of potential full-history advantage is information lost by a compact representation. We therefore examine the value of retaining progressively richer historical evidence. Although long interaction histories, broader neighborhoods, and rich semantic context can be intuitively relevant, their predictive value need not accumulate proportionally. If a limited view combining semantics, structure and time already resolves most positive and negative candidates, exposing substantially broader evidence may provide only diminishing returns.

We test this intuition by systematically varying representation fidelity and historical scope in lightweight predictors using three evidence representations. Inspired by prior work showing the strength of temporal heuristics (Cornell et al., 2025), **Heuristic** summarizes history with structural and temporal statistics and a scalar semantic similarity score. **Snapshot** preserves richer semantic and neighborhood representations while abstracting away fine-grained interaction order. **Sequence** further retains the queried nodes' ordered interaction histories. Snapshot and Sequence also include the structural and temporal heuristics. We use XGBoost for Heuristic and mean pooling followed by an MLP for Snapshot and Sequence. Implementation details are in Appendix F.1.

Table 2 shows rapidly diminishing returns from richer history. Extending history from 47 to 200 events yields only marginal changes for both snapshots and sequences. Sequences offer only small gains over snapshots on GDELT and no consistent gains on Enron. This helps explain our Snapshot GNN's competitiveness despite omitting event order: snapshots preserve most useful local evidence and are reusable across batch queries, while inexpensive global heuristics supply broader history, improving over local heuristics by about 3 AUC points on GDELT. The design thus trades fine-grained temporal detail for broader evidence and much cheaper processing.

How much processing is needed? We test whether heavier processing adds value under roughly fixed temporal evidence. Graph-based processors share each query's recent interaction histories and semantic node features, without co-occurrence encodings or global heuristics. We compare uniform pooling, exponential decay, graph convolution, and self-attention. LLMs receive the same histories and query in text form. *LLM (Anonymous)* uses anonymous entity identifiers, preserving interaction structure and temporal information; *LLM (Entity)* retains entity identities and may additionally exploit entity-specific associations learned during pretraining. See Appendix F.2 for more details.

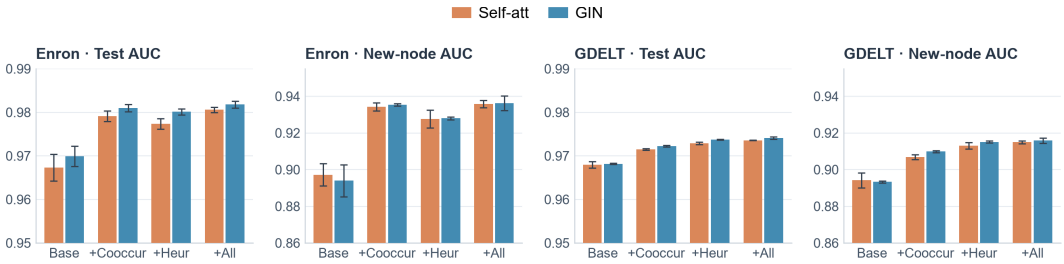


Figure 3: Transductive/inductive AUC of self-attention and GIN on Enron and GDELT. +Cooccur adds neighbor co-occurrence encodings, +Heur adds global heuristics and +All adds both.

Table 3: ROC-AUC under comparable histories. Graph methods share histories and node features; LLM (Anonymous) masks entity names, while LLM (Entity) retains them.

Method	GDELT		Enron	
	Transductive	Inductive	Transductive	Inductive
Mean Pool	0.9680 ± 0.0005	0.8934 ± 0.0015	0.9664 ± 0.0008	0.8970 ± 0.0068
Exp-Decay	0.9686 ± 0.0004	0.8971 ± 0.0012	0.9674 ± 0.0026	0.9040 ± 0.0035
GINConv	0.9682 ± 0.0001	0.8933 ± 0.0005	0.9699 ± 0.0023	0.8940 ± 0.0088
Self-Attn	0.9680 ± 0.0008	0.8942 ± 0.0040	0.9673 ± 0.0030	0.8972 ± 0.0061
LLM (Anonymous)	0.9670 ± 0.0004	0.8929 ± 0.0012	0.9746 ± 0.0011	0.9170 ± 0.0033
LLM (Entity)	0.9728 ± 0.0004	0.9136 ± 0.0011	0.9822 ± 0.0018	0.9431 ± 0.0017

Table 3 shows similar performance across graph-based processors: mean pooling, exponential decay, GINConv, and self-attention differ by at most 0.0038 AUC on GDELT and 0.0100 on Enron across evaluation settings. **This suggests that heavier cross-event processing alone accounts for little of the remaining gap.** The entity-aware LLM remains substantially stronger, while anonymization removes much of this advantage, suggesting the importance of additional semantic world knowledge acquired during pretraining. Nevertheless, the anonymized LLM retains a modest advantage on Enron, suggesting direct LLM processing of anonymized sequences can still help in some settings.

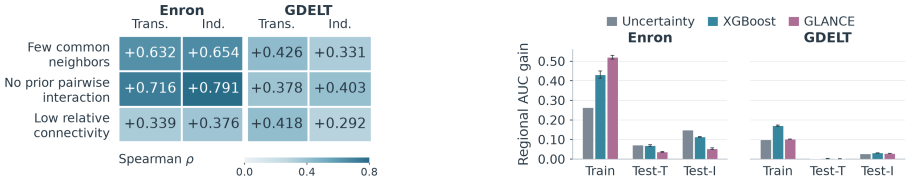
Given the LLM advantage under controlled evidence, we ask how much of this gain can be recovered with inexpensive graph-side signals rather than costly LLM inference. We augment self-attention and GIN with neighbor co-occurrence encodings and global historical heuristics, adding broader structural context at low cost. Figure 3 shows that these signals substantially improve graph-based prediction. Adding both raises GIN's mean AUC across four settings from 0.9314 to 0.9520, approximately 0.0010 below the entity-aware LLM. With both signals available, GIN also slightly outperforms self-attention across all four settings. **This suggests that much of the LLM advantage can be recovered through simple structural evidence not automatically exposed by self-attention over local histories.** With this evidence, the choice of processor becomes less consequential.

Together, these controlled results help explain the lightweight model's performance on these controlled settings. Finer-grained histories yield limited returns, while heavier processing of the same evidence adds little. Our Snapshot GNN thus nears saturation on both dimensions, retaining broad predictive evidence without costly sequence processing. Though structural cues largely close the aggregate LLM gap, similar overall performance may hide query-level differences. We next examine where this residual value arises and whether it can be localized efficiently.

5 Characterizing and Locating LLM Advantage

A key question left is whether the remaining LLM advantage is diffuse or concentrated in identifiable regions. We study this instance-level structure and ask if these regions can be localized efficiently.

Prior work on static node classification links GNN uncertainty to selective LLM invocation for structurally difficult regions such as low-homophily neighborhoods (Loveland et al., 2025). We ask whether uncertainty similarly identifies where LLMs help in temporal link prediction and why these queries challenge graph-based models. For a candidate edge e, we define GNN uncertainty as $U(e) = -|z_G(e) - \tau_G|$, where $z_G(e)$ is the GNN logit and τ_G is a decision threshold fixed on held-out validation data; larger values therefore indicate smaller margins and greater uncertainty. Figure 4 shows that selecting only the 20% most uncertain candidates captures 71.8–96.5% of all



(a) **(b)**

Figure 5: (a) Spearman correlations between GNN uncertainty and weak-structure indicators. (b) Regional AUC gains of LLM over GNN at 20% coverage under uncertainty and learned routing.

Table 4: AUC gains under 20% uncertainty routing. Selected 20% reports LLM–GNN gain within routed queries. Full test reports affine/MLP fusion gains before/after quantile alignment.

Setting	Selected 20%	Full test set			
	Standalone	Before alignment		After alignment	
	LLM − GNN	Affine	MLP	Affine	MLP
Enron T	+0.071890	+0.001198	+0.002286	+0.003121	+0.003032
Enron I	+0.148774	+0.004421	+0.005090	+0.005753	+0.005712
GDELT T	+0.002457	+0.000308	+0.000418	+0.000404	+0.000463
GDELT I	+0.027186	+0.000290	+0.000731	+0.000962	+0.001057

LLM corrections across the test settings. Thus, the corrections are strongly concentrated near the GNN decision boundary, making uncertainty a high-recall localizer. Figure 5(a) further shows that, among all observed positive test edges, uncertainty correlates positively with weak historical support, including few common neighbors, no prior interaction, and low relative connectivity. This suggests that positive edges with limited structural support are particularly challenging for the GNN to predict. **Thus, the LLM's residual value concentrates on graph-challenging queries, with GNN uncertainty providing a cheap way to locate them.**

We next ask whether richer learned routers can further refine the regions identified by uncertainty under the 20% budget. XGBoost augments uncertainty with graph-side signals, while a GLANCE-style router incorporates information from both the GNN and LLM. As shown in Figure 5(b), the learned routers often identify regions with substantially larger LLM advantage during training, but these gains do not transfer consistently across datasets and test settings. As a result, GNN uncertainty remains competitive across all settings without learning an additional utility mapping. **These results establish GNN uncertainty as a strong localizer alone.**

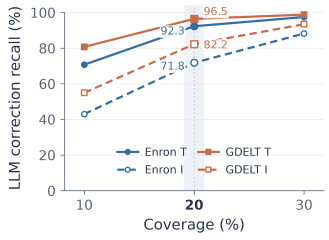


Figure 4: Correction recall under uncertainty routing. Recall is the fraction of LLM corrections captured at each uncertainty-routing coverage.

Finally, Table 4 examines how localized these gains remain at the dataset level. For routed queries, we replace the original GNN score with a fused GNN–LLM score, while unrouted queries retain their original predictions. The routed regions exhibit substantially larger LLM advantages than the aggregate improvement, indicating that **the identified benefit is much stronger locally than globally.** We also compare learned MLP fusion with a simple affine combination $as_{\text{GNN}} + bs_{\text{LLM}} + c$. The MLP initially performs better, but this comparison is confounded by differences in score scale. After separately quantile-aligning each fused output to the original GNN score distribution, the gap largely disappears. Thus, the apparent MLP advantage primarily reflects a more favorable score distribution rather than more effective nonlinear GNN-LLM interactions. **After score alignment, simple affine mixing achieves comparable performance.**

6 Conclusion

We revisit whether TTAG future link prediction requires heavyweight modeling. A lightweight predictor using reusable temporal snapshots and inexpensive heuristics remains competitive with heavier alternatives at much lower cost. Our theoretical analysis motivates separating historical evidence from processing. Controlled experiments then reveal limited returns from finer histories and heavier processing, while inexpensive structural cues recover much of the remaining LLM gap. The residual LLM advantage can be localized by GNN uncertainty and tends to occur where graph support is weaker. Overall, strong TTAG prediction often needs much lighter modeling across real-world benchmarks, with LLM gains concentrated where graph prediction is most uncertain.

AI USE STATEMENT

Generative AI tools were used to assist with copyediting, wording, and improving the readability of the manuscript. All AI-assisted text was reviewed and edited by the authors, who take full responsibility for the final content of the paper.

REPRODUCIBILITY STATEMENT

We provide detailed descriptions of the datasets, evaluation protocol, model configurations, and training settings in the main paper and Appendix. Additional implementation details for the evidence and processing analyses, LLM prompting, routing, and score alignment are also included in the Appendix. All experiments are conducted with fixed chronological splits and five random seeds unless otherwise specified. We additionally provide anonymized source code and scripts for data preprocessing, training, evaluation, and reproducing the reported experiments at `https://anonymous.4open.science/r/dyg-6686`.

REFERENCES

Weilin Cong, Si Zhang, Jian Kang, Baichuan Yuan, Hao Wu, Xin Zhou, Hanghang Tong, and Mehrdad Mahdavi. Do we really need complicated model architectures for temporal networks? In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=ayPPc0SyLv1.

Filip Cornell, Oleg Smirnov, Gabriela Zarzar Gandler, and Lele Cao. On the power of heuristics in temporal graphs. *CoRR*, abs/2502.04910, 2025. doi: 10.48550/ARXIV.2502.04910. URL https://doi.org/10.48550/arXiv.2502.04910.

Petter Holme and Jari Saramäki. Temporal networks. *CoRR*, abs/1108.1780, 2011. URL http://arxiv.org/abs/1108.1780.

Shenyang Huang, Ali Parviz, Emma Kondrup, Zachary Yang, Zifeng Ding, Michael M. Bronstein, Reihaneh Rabbany, and Guillaume Rabusseau. Are large language models good temporal graph learners? *CoRR*, abs/2506.05393, 2025. doi: 10.48550/ARXIV.2506.05393. URL https://doi.org/10.48550/arXiv.2506.05393.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In Ankur Teredesai, Vipin Kumar, Ying Li, Rómer Rosales, Evimaria Terzi, and George Karypis (eds.), *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pp. 1269–1278. ACM, 2019. doi: 10.1145/3292500.3330895. URL https://doi.org/10.1145/3292500.3330895.

Runlin Lei, Jiarui Ji, Haipeng Ding, Lu Yi, Zhewei Wei, Yongchao Liu, and Chuntao Hong. Exploring the potential of large language models as predictors in dynamic text-attributed graphs. *CoRR*, abs/2503.03258, 2025. doi: 10.48550/ARXIV.2503.03258. URL https://doi.org/10.48550/arXiv.2503.03258.

Donald Loveland, Yao-An Yang, and Danai Koutra. Glance for context: Learning when to leverage llms for node-aware GNN-LLM fusion. *CoRR*, abs/2510.10849, 2025. doi: 10.48550/ARXIV.2510.10849. URL https://doi.org/10.48550/arXiv.2510.10849.

Weishuo Ma, Yanbo Wang, Xiyuan Wang, and Muhan Zhang. Reconsidering the performance of GAE in link prediction. In Meeyoung Cha, Chanyoung Park, Noseong Park, Carl Yang, Senjuti Basu Roy, Jessie Li, Jaap Kamps, Kijung Shin, Bryan Hooi, and Lifang He (eds.), *Proceedings of the 34th ACM International Conference on Information and Knowledge Management, CIKM 2025, Seoul, Republic of Korea, November 10-14, 2025*, pp. 2052–2062. ACM, 2025. doi: 10.1145/3746252.3761108. URL https://doi.org/10.1145/3746252.3761108.

Meng Qin and Dit-Yan Yeung. Temporal link prediction: A unified framework, taxonomy, and review. *ACM Comput. Surv.*, 56(4):89:1–89:40, 2024. doi: 10.1145/3625820. URL https://doi.org/10.1145/3625820.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In Kentaro Inui, Jing Jiang, Vincent Ng, and Xiaojun Wan (eds.), *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing, EMNLP-IJCNLP 2019, Hong Kong, China, November 3-7, 2019*, pp. 3980–3990. Association for Computational Linguistics, 2019. doi: 10.18653/V1/D19-1410. URL https://doi.org/10.18653/v1/D19-1410.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael M. Bronstein. Temporal graph networks for deep learning on dynamic graphs. *CoRR*, abs/2006.10637, 2020. URL https://arxiv.org/abs/2006.10637.

Amit Roy, Ning Yan, and Masood S. Mortazavi. Llm-driven knowledge distillation for dynamic text-attributed graphs. *CoRR*, abs/2502.10914, 2025. doi: 10.48550/ARXIV.2502.10914. URL https://doi.org/10.48550/arXiv.2502.10914.

Rakshit S. Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=HyePrhR5KX.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. Text embeddings by weakly-supervised contrastive pre-training. *CoRR*, abs/2212.03533, 2022. doi: 10.48550/ARXIV.2212.03533. URL https://doi.org/10.48550/arXiv.2212.03533.

Lu Wang, Xiaofu Chang, Shuang Li, Yunfei Chu, Hui Li, Wei Zhang, Xiaofeng He, Le Song, Jingren Zhou, and Hongxia Yang. TCL: transformer-based dynamic graph modelling via contrastive learning. *CoRR*, abs/2105.07944, 2021a. URL https://arxiv.org/abs/2105.07944.

Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021b. URL https://openreview.net/forum?id=KYPz4YsCPj.

Yunan Wang, Jianxin Li, and Ziwei Zhang. Global-recent semantic reasoning on dynamic text-attributed graphs with large language models. *CoRR*, abs/2509.18742, 2025. doi: 10.48550/ARXIV.2509.18742. URL https://doi.org/10.48550/arXiv.2509.18742.

Da Xu, Chuanwei Ruan, Evren Körpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020. URL https://openreview.net/forum?id=rJeW1yHYwH.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ryGs6iA5Km.

Yuanyuan Xu, Wenjie Zhang, Ying Zhang, Xuemin Lin, and Xiwei Xu. Unlocking multi-modal potentials for link prediction on dynamic text-attributed graphs. In Sven Koenig, Chad Jenkins, and Matthew E. Taylor (eds.), *Fortieth AAAI Conference on Artificial Intelligence, Thirty-Eighth Conference on Innovative Applications of Artificial Intelligence, Sixteenth Symposium on Educational Advances in Artificial Intelligence, AAAI 2026, Singapore, January 20-27, 2026*, pp. 27386–27394. AAAI Press, 2026. doi: 10.1609/AAAI.V40I32.39956. URL https://doi.org/10.1609/aaai.v40i32.39956.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/d611019afba70d547bd595e8a4158f55-Abstract-Conference.html.

Table 5: Statistics of the DTGB datasets used in our experiments.

Dataset	# Nodes	# Edges	# Categories	# Timestamps	Domain	Bipartite
GDELT	6,786	1,339,245	237	2,591	Knowledge graph	No
ICEWS1819	31,796	1,100,071	266	730	Knowledge graph	No
Enron	42,711	797,907	10	1,006	E-mail	No
Googlemap_CT	111,168	1,380,623	5	55,521	Recommendation	Yes

Jiasheng Zhang, Jialin Chen, Menglin Yang, Aosong Feng, Shuang Liang, Jie Shao, and Rex Ying. DTGB: A comprehensive benchmark for dynamic text-attributed graphs. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*, 2024. URL http://papers.nips.cc/paper_files/paper/2024/hash/a65d054a407f94c34ecfb598fb540a0d-Abstract-Datasets_and_Benchmarks_Track.html.

Siwei Zhang, Yun Xiong, Yateng Tang, Jiarong Xu, Xi Chen, Zehao Gu, Xuehao Zheng, Zian Jia, and Jiawei Zhang. Unifying text semantics and graph structures for temporal text-attributed graphs with large language models. In Danielle Belgrave, Cheng Zhang, Laura N. Montoya, Hsuan-Tien Lin, Razvan Pascanu, Piotr Koniusz, Marzyeh Ghassemi, Nancy Chen, Iván Vladimir Meza Ruíz, and Arturo Loaiza-Bonilla (eds.), *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2025, NeurIPS 2025, San Diego, CA, USA, December 2-7, 2025 / Mexico City, Mexico, November 30 - December 5, 2025*, 2025. URL http://papers.nips.cc/paper_files/paper/2025/hash/4c5722bad9759216474df8fc46c97af2-Abstract-Conference.html.

A General Experimental Details

A.1 Datasets

We conduct experiments on four real-world text-attributed temporal graph benchmarks from DTGB (Zhang et al., 2024): **GDELT**, **ICEWS1819**, **Enron**, and **Googlemap_CT**. GDELT and ICEWS1819 are temporal knowledge graphs describing interactions among political entities, Enron contains timestamped e-mail communications, and Googlemap_CT is a bipartite user–business interaction graph constructed from timestamped reviews. These datasets span different graph structures, temporal granularities, and application domains, providing a diverse testbed for TTAG future-link prediction. Their basic statistics are summarized in Table 5.

A.2 Evaluation Protocol

We follow the DTGB (Zhang et al., 2024) protocol throughout, including chronological data splitting, negative sampling, and both transductive and inductive evaluation. In the transductive setting, future interactions are evaluated under the standard chronological split. The inductive setting follows the DTGB new-node protocol and evaluates interactions involving nodes that are unseen in the corresponding historical graph. We use identical evaluation candidates across applicable methods to ensure consistent comparison. Unless otherwise stated, ROC-AUC is computed separately within each evaluation batch and then averaged across batches. This aggregation is shared by graph-based and direct LLM methods, including the processing comparison in Table 3.

At test time, historical information is constructed causally: for a prediction at time t, a method may access only interactions available before its prescribed observation boundary.

A.3 Textual Features

DTGB (Zhang et al., 2024) initializes node and edge textual features using pooled representations from `bert-base-uncased`. While BERT provides strong contextual token representations, its vanilla pooled representations are not optimized as standalone semantic embeddings and have been

Table 6: AUC of DyGFormer using the original DTGB BERT features and E5 features. Only the text encoder is changed.

Setting	Text Features	GDELT	ICEWS1819	Enron	Googlemap_CT
Transductive	BERT	0.9699 ± 0.0004	0.9901 ± 0.0021	0.9768 ± 0.0038	0.8227 ± 0.0075
	E5	0.9714 ± 0.0001	0.9907 ± 0.0013	0.9766 ± 0.0004	0.9158 ± 0.0112
Inductive	BERT	0.9031 ± 0.0010	0.9725 ± 0.0029	0.9247 ± 0.0068	0.7754 ± 0.0076
	E5	0.9068 ± 0.0016	0.9731 ± 0.0050	0.9283 ± 0.0031	0.8937 ± 0.0140

shown to perform poorly on semantic similarity tasks (Reimers & Gurevych, 2019). We therefore use E5 (Wang et al., 2022), which is explicitly contrastively trained to produce general-purpose text embeddings.

To assess the effect of the textual backbone, we rerun DyGFormer while changing only the text encoder from the original BERT features to E5. As shown in Table 6, E5 improves performance in most settings, with particularly pronounced gains on Googlemap_CT: AUC rises from 0.8227 to 0.9158 under transductive evaluation and from 0.7754 to 0.8937 under inductive evaluation. We hypothesize that this gap partly reflects the limited discriminative value of the original BERT representations for business descriptions. The mean pairwise cosine similarity between business embeddings is approximately 0.99 for BERT versus 0.79 for E5, suggesting that BERT provides little angular separation between businesses and may obscure useful semantic distinctions. We therefore use the same fixed E5 embeddings for all applicable methods to ensure a common textual backbone.

A.4 Metrics and Repeated Runs

We use AUC as the primary evaluation metric. Unless otherwise specified, stochastic trainable models are evaluated over five random seeds, and we report mean $\pm$ standard deviation. The same chronological splits, textual features, and evaluation candidates are reused across random seeds and model variants.

A.5 General Training Environment

Neural models are implemented in PyTorch and optimized using AdamW unless the original implementation of a baseline specifies otherwise. Model-specific architectures, optimization hyperparameters, and stopping criteria are reported in their corresponding appendix sections. All local experiments are conducted on NVIDIA A100 PCIe 40 GB GPUs, using PyTorch 2.3.0 and CUDA 12.1.

B Lightweight Predictor Details

We initialize node representations with pretrained E5 embeddings, $\mathbf{h}_u^{(0)} = \mathbf{x}_u$. For each prediction batch, we use its starting time t_b as a shared causal cutoff: the snapshot contains only interactions occurring before t_b and is kept fixed for all queries in the batch. Given a historical window of length W, we construct

$$\mathcal{E}_{t_b}^{(W)} = \{(u, v, \tau) \in \mathcal{E} \mid t_b - W \leq \tau < t_b\}. \tag{9}$$

Repeated interactions between the same pair are collapsed into a single weighted edge. Its weight is computed using exponential temporal decay:

$$w_{uv}(t_b) = \sum_{\tau \in \mathcal{T}_{uv}^{(W)}(t_b)} \exp[-\lambda(t_b - \tau)], \tag{10}$$

where $\mathcal{T}_{uv}^{(W)}(t_b)$ denotes the timestamps of interactions between u and v within the snapshot window.

We encode the resulting weighted snapshot $\mathcal{G}_{t_b}^{(W)}$ using an L_{GIN}-layer GIN with residual connections. Starting from $\mathbf{h}_u^{(0)} = \mathbf{x}_u$, each layer updates node representations as

$$\begin{aligned} \mathbf{m}_u^{(\ell)} &= (1+\epsilon^{(\ell)})\mathbf{h}_u^{(\ell)} + \sum_{v \in \mathcal{N}_{t_b}^{(W)}(u)} w_{uv}(t_b)\mathbf{h}_v^{(\ell)}, \\ \mathbf{h}_u^{(\ell+1)} &= \mathbf{h}_u^{(\ell)} + \mathrm{LN}^{(\ell)}\Big(\mathrm{MLP}^{(\ell)}(\mathbf{m}_u^{(\ell)})\Big), \end{aligned} \tag{11}$$

for $\ell = 0, \ldots, L_{\mathrm{GIN}} - 1$, where $\epsilon^{(\ell)}$ is a learnable scalar and LN denotes layer normalization. The final snapshot representation is $\mathbf{h}_u = \mathbf{h}_u^{(L_{\mathrm{GIN}})}$. For a query (u, v, t), let $\mathcal{G}_{<t}$ denote the causal interaction history before t and $\mathcal{N}_u(t)$ denote the set of nodes that have interacted with u before t. We compute four inexpensive query-time heuristics.

Recency. Let t_{uv}^{last} denote the latest interaction time between u and v in the available history before t. For Snapshot and Sequence, we encode recency as

$$\begin{aligned} r_{uv}^{\mathrm{rec}}(t) &= -\log(1+\Delta_{uv}(t)), \\ \Delta_{uv}(t) &= \begin{cases} \max(0, t - t_{uv}^{\mathrm{last}}), & \text{if a historical interaction exists,} \\ 10^9, & \text{otherwise.} \end{cases} \end{aligned} \tag{12}$$

Larger values indicate more recent interactions.

Past Pair Interaction Count. The historical interaction count of the queried pair is

$$f_{uv}(t) = \sum_{(i,j,\tau)\in\mathcal{E}} \mathbf{1}[\tau < t]\, \mathbf{1}[(i = u \wedge j = v) \vee (i = v \wedge j = u)]. \tag{13}$$

For this heuristic, interactions in both directions are pooled, and repeated events are counted separately.

Target Activity. We use the target node's historical interaction count as an activity signal:

$$d_v(t) = \sum_{(i,j,\tau)\in\mathcal{E}} \mathbf{1}[\tau < t]\, (\mathbf{1}[i = v] + \mathbf{1}[j = v]). \tag{14}$$

This counts both incoming and outgoing interactions, including repeated interactions with the same neighbor.

Resource Allocation. We measure structural overlap using an interaction-count-weighted Resource Allocation (RA) score:

$$c_{uv}^{\mathrm{RA}}(t) = \sum_{z \in \mathcal{N}_u(t) \cap \mathcal{N}_v(t)} \frac{1}{d_z(t)}, \tag{15}$$

where $\mathcal{N}_u(t)$ is the set of distinct historical neighbors of u, and $d_z(t)$ is the historical interaction count defined above. Each common neighbor contributes once.

We collect these statistics into

$$\mathbf{r}_{uv}(t) = \left[r_{uv}^{\mathrm{rec}}(t) \,\|\, \log(1+f_{uv}(t)) \,\|\, \log(1+d_v(t)) \,\|\, \log(1+c_{uv}^{\mathrm{RA}}(t))\right]. \tag{16}$$

The final prediction concatenates the query-time heuristic vector with the two endpoint representations from the batch-shared snapshot:

$$s(u, v, t) = \mathrm{MLP}\left([\mathbf{h}_u \,\|\, \mathbf{h}_v \,\|\, \mathbf{r}_{uv}(t)]\right). \tag{17}$$

The snapshot representation is shared within each batch, whereas all four heuristics are updated causally at the individual query time. These query-time statistics are implemented with Numba-compiled routines and incur negligible overhead in practice, allowing us to retain the most recent historical information without sacrificing the efficiency of the batch-reused snapshot.

Table 7: Main hyperparameters of the lightweight GIN predictor for the four datasets.

Hyperparameter	GDELT	ICEWS1819	Enron	Googlemap_CT
GIN layers L_{GIN}	1	1	2	3
MLP hidden width H	128	256	64	256
Prediction MLP layers	4	2	2	2
Learning rate	5×10^{-5}	10^{-4}	10^{-4}	5×10^{-5}
Dropout	0	0	0.2	0

Hyperparameters. Table 7 summarizes the main settings of the lightweight GIN predictor. All datasets use frozen E5-large-v2 embeddings. Each GIN layer uses a two-linear-layer MLP ($1024 \rightarrow H \rightarrow 1024$), ReLU, LayerNorm, and an outer residual connection; its learnable ϵ is initialized to zero. We use AdamW with linear learning-rate decay, a 10% warmup fraction, a minimum learning-rate ratio of 0.05, zero weight decay, and gradient-norm clipping at 1.0. Each positive query is paired with one randomly sampled negative during training. Final tests restore the checkpoint with the highest validation AUC.

C Baseline Details

We compare against representative temporal graph models, recent TTAG-specific methods, and direct LLM-based predictors. To ensure a consistent and fair comparison, all methods are evaluated under the same strict-causal DTGB protocol. For a prediction query at time (t), graph histories, graph-derived features, and semantic context are constructed only from information available before (t). The same chronological data splits, negative candidates, and textual feature space are used across methods whenever applicable. For all methods requiring an LLM backend, we use Llama-3.1-8B-Instruct as the common backbone. Where official implementations are available, the released code and recommended configurations are used as references under this common protocol.

C.1 Temporal Graph Baselines

GraphMixer. GraphMixer (Cong et al., 2023) is a lightweight temporal graph architecture designed to model interaction histories without recurrent networks or self-attention. It consists of an MLP-based link encoder that summarizes temporal interaction information, a node encoder based on neighborhood mean-pooling, and an MLP link classifier that combines the resulting representations for prediction. Its link encoder adopts an MLP-Mixer-style architecture to mix information across sampled historical interactions, providing a simple alternative to more elaborate sequence models while still capturing temporal patterns. History length and standard training hyperparameters are selected on the validation split.

DyGFormer. DyGFormer (Yu et al., 2023) is a Transformer-based temporal graph model that operates directly on the historical first-hop interaction sequences of the queried nodes. A neighbor co-occurrence encoding records how frequently historical neighbors appear in the source and destination sequences, providing an explicit signal of their structural correlation. To capture longer temporal contexts efficiently, DyGFormer partitions interaction sequences into local patches and treats the resulting patches as Transformer tokens. This design preserves local temporal proximity while allowing the Transformer to model long-range dependencies over substantially longer interaction histories. The default history configuration is used unless otherwise specified.

C.2 TTAG Baselines

MoMent. MoMent (Xu et al., 2026) adopts a node-centric multi-modal view of dynamic text-attributed graphs, explicitly modeling textual, temporal, and structural information as complementary modalities. Its textual branch captures semantic context from node attributes, while the temporal branch characterizes node-specific behavioral patterns from interaction timestamps. In parallel, the structural branch represents local interaction context from neighboring events and their associated semantic and temporal information. Dedicated self-attention encoders transform the three modalities into modality-specific tokens, which are subsequently fused into a comprehensive node

representation. To promote coherence among the heterogeneous representations, MoMent further introduces a dual-domain alignment objective that aligns modality distributions globally and refines their agreement at the instance level.

CROSS. CROSS (Zhang et al., 2025) is an LLM-enhanced framework for temporal text-attributed graphs that explicitly models the evolution of textual semantics together with dynamic graph structure. Its Temporal Semantics Extractor uses an LLM to reason over the evolving textual neighborhoods of nodes and produces semantic representations at selected points along their interaction histories. These temporally evolving semantic features are then combined with graph representations through a Semantic-structural Co-encoder, which jointly propagates semantic and structural information and encourages mutual reinforcement between the two modalities. This design allows CROSS to serve as an extension of temporal graph backbones rather than replacing structural temporal modeling with direct LLM prediction.

DyGRASP. DyGRASP (Wang et al., 2025) targets the complementary roles of recent semantic dependencies and long-term semantic evolution in dynamic text-attributed graphs. For recent semantics, it employs node-centric implicit reasoning together with a sliding-window mechanism to capture dependencies among recent interaction texts without repeatedly invoking explicit long-horizon reasoning. For global semantics, DyGRASP uses explicit LLM reasoning with tailored prompts and an RNN-like chain structure, allowing the semantic state of a node to evolve incrementally over its interaction trajectory. Updating and merging layers subsequently integrate these recent and global semantic signals with temporal graph structural representations for downstream prediction.

C.3 LLM-Based Predictors

TGTalker. TGTalker (Huang et al., 2025) formulates temporal graph prediction as a language-model reasoning problem rather than learning a dedicated temporal graph encoder. Motivated by the strong recency bias observed in temporal networks, it extracts recent structural context around a query and serializes the relevant temporal interactions into natural language. The resulting description of the temporal neighborhood, together with the prediction query, is provided directly to a pretrained LLM, allowing temporal and structural patterns to be processed through the model's language interface. In addition to prediction, this formulation naturally supports textual explanations of the model's decisions.

Llama-8B. To provide a controlled direct-LLM baseline, we textualize the historical neighborhood of each query and ask Llama-3.1-8B-Instruct to predict whether the future interaction will occur. For each queried pair, the prompt contains the most recent 47 interaction events of the two endpoints, represented as compact neighbor lists together with their temporal information, followed by a binary prediction instruction of the form *"The answer is X"*, where $X \in \{0, 1\}$. Rather than using the generated hard label, we compute the link score from the model probabilities assigned to the two answer tokens:

$$s_{\mathrm{LLM}}(u, v, t) = \frac{p(X = 1)}{p(X = 0) + p(X = 1)}. \tag{18}$$

This provides a continuous prediction score while keeping the LLM input and output protocol identical across datasets. A concrete prompt example will be provided below for reference.

Llama-8B-FT. We further fine-tune the same Llama-3.1-8B-Instruct model for future-link prediction using the training portion of each dataset, while preserving the textualized history and binary output format of the zero-shot/in-context baseline above. For each dataset, we uniformly sample 10,000 positive training queries and pair each with one sampled negative, yielding 20,000 labeled examples. We apply LoRA to the attention projections (q, k, v, and o) and feed-forward projections (gate, up, and down) in all transformer layers, while keeping the pretrained weights frozen. Training minimizes the cross-entropy loss on the binary answer token, with prompt tokens excluded from the loss. We train for one epoch and evaluate the final adapter. This comparison measures the benefit of task-specific adaptation with the same base model and input/output format. The main fine-tuning hyperparameters are summarized in Table 8.

Table 8: Main hyperparameters for Llama-8B-FT. The effective batch size uses a micro-batch size of 1 and 8 gradient accumulation steps.

Hyperparameter	Value
Positive training queries	10,000 per dataset
Negative-to-positive ratio	$1:1$
Training epochs	1
Effective batch size	8
Learning rate	1×10^{-4}
Warmup ratio	0.03
Weight decay	0
LoRA rank / alpha	$r = 8$, $\alpha = 32$
LoRA dropout	0.05
Maximum sequence length	8,192 tokens

D Efficiency Evaluation Details

Training Cost. Training cost includes data and model loading, preparation within the training pipeline, optimization, validation, and checkpointing, excluding separately reported preprocessing. This includes training-time snapshot and heuristic construction for our GNN, and prompt preparation and tokenization for Llama fine-tuning. LLM-based semantic generation and representation preprocessing for CROSS and DyGRASP are reported separately under preprocessing, not training. Except for Llama-8B-FT, all trainable models are trained for up to 50 epochs. Llama-8B-FT retains one-epoch LoRA training on 20,000 balanced prompts. TGTalker requires no parameter training, but its setup costs remain part of inference.

Inference Cost. End-to-end inference covers the complete transductive and inductive evaluations, including necessary checkpoint loading, adapter merging, prompt construction, history retrieval, representation updates, prediction, and output processing. Our GNN includes batch-level snapshot construction and all required heuristic preparation or updates. Histories follow each evaluation protocol's causal visibility rules; future interactions cannot be used to prepare earlier predictions. The reported cost covers both evaluation protocols, not a single de-duplicated pass over their combined queries.

TGTalker's low inference cost in Figure 2 partly reflects its cached background context and minimal query-specific history: only up to two recent outgoing interactions of the source node are explicitly shown, with no separate history for the candidate destination. This design reduces query-time processing and enables fast inference, but is accompanied by substantially weaker predictive performance (Table 1), highlighting the need to consider accuracy alongside inference efficiency.

Preprocessing Accounting. For CROSS and DyGRASP, semantic generation and encoding are shown as a separate preprocessing segment, including both training-side and evaluation-side work. Those costs are then excluded from the corresponding training and inference segments, so their sum remains the full E2E cost. Other methods' preparation is included in their training or inference segments rather than treated as free.

E Full-History Advantage: Proof and Extensions

We first prove the main AUC bound, then give two complementary sufficient results for small compression gaps. The first studies shared representation capacity with exactly additive predictive states. The second studies history-dependent interactions while preserving the base event contributions exactly. Both results construct additive summaries with explicit AUC guarantees under their respective structural assumptions.

E.1 Proof of Theorem 1

We use the assumptions and evaluation distribution from the main text. Evaluation groups B are weighted according to the AUC averaging rule, $\mathbb{P}(Y = 1 \mid B) = 1/2$, and B is included in the

Table 9: Wall-clock cost (hours) on GDELT and Enron, decomposed into training, inference, and separately reported preprocessing. Total cost corresponds to the values visualized in Figure 2. An em dash indicates that preprocessing is included in training or inference rather than reported separately.

Dataset	Method	Training	Inference	Preprocessing	Total
GDELT	GraphMixer	8.8265	0.0266	—	8.8531
	DyGFormer	12.8745	0.0484	—	12.9229
	MoMent	3.0495	0.0063	—	3.0558
	CROSS	15.3520	0.0569	2.1528	17.5617
	DyGRASP	54.9211	0.6064	5.2528	60.7803
	TGTalker	0	1.4426	—	1.4426
	Llama-8B-FT	2.7750	9.9551	—	12.7302
	GIN (Ours)	0.7016	0.002506	—	0.7041
Enron	GraphMixer	4.1607	0.0194	—	4.1801
	DyGFormer	4.7148	0.0276	—	4.7424
	MoMent	1.5928	0.0050	—	1.5977
	CROSS	13.1909	0.0419	22.1720	35.4048
	DyGRASP	19.0061	0.1203	86.6818	105.8082
	TGTalker	0	1.5242	—	1.5242
	Llama-8B-FT	4.0324	11.5067	—	15.5391
	GIN (Ours)	0.9395	0.013352	—	0.9528

shared context C. All maps are measurable, including their dependence on C. Write $S = S_\phi(H)$ and define

$$\mathcal{F} = \sigma(H, C), \qquad \mathcal{G} = \sigma(S, C) \subseteq \mathcal{F}.$$

For either information set $\mathcal{A} \in \{\mathcal{F}, \mathcal{G}\}$, let

$$\mathrm{AUC}^*_{\mathcal{A}} := \sup_{f \ \mathcal{A}\text{-measurable}} \mathbb{E}_B[\mathrm{AUC}_B(f)],$$

where scores are real-valued. Within each group, AUC compares independent positive and negative draws, giving half credit for ties. Thus $\mathrm{AUC}^*_{\mathcal{F}} = \mathrm{AUC}^*_{H,C}$ and $\mathrm{AUC}^*_{\mathcal{G}} = \mathrm{AUC}^*_{S,C}$. As a proof device for controlling posterior approximation error, we use the auxiliary minimum Brier risks

$$R^*_{\mathcal{A}} := \inf_{f \ \mathcal{A}\text{-measurable},\ 0 \le f \le 1} \mathbb{E}[(Y - f)^2].$$

Proof. **Step 1: Express the auxiliary Brier gap as a conditional variance.** For any $\mathcal{A}$-measurable prediction $f \in [0, 1]$, let $\eta_{\mathcal{A}} = \mathbb{E}[Y \mid \mathcal{A}]$. Expanding the square gives

$$\mathbb{E}[(Y - f)^2] = \mathbb{E}[(Y - \eta_{\mathcal{A}})^2] + \mathbb{E}[(\eta_{\mathcal{A}} - f)^2], \tag{19}$$

because $\mathbb{E}[Y - \eta_{\mathcal{A}} \mid \mathcal{A}] = 0$ makes the cross term vanish. Hence $\eta_{\mathcal{A}} \in [0, 1]$ attains the minimum. In particular, write

$$\eta = \mathbb{E}[Y \mid \mathcal{F}] = q_C(Z_0), \qquad \eta_S = \mathbb{E}[Y \mid \mathcal{G}] = \mathbb{E}[\eta \mid \mathcal{G}],$$

where the last equality follows from the tower property. Applying equation 19 with $\mathcal{A} = \mathcal{F}$ and $f = \eta_S$ yields

$$\begin{aligned} \Delta_{\mathrm{Br}} &:= R^*_{\mathcal{G}} - R^*_{\mathcal{F}} \\ &= \mathbb{E}[(\eta - \eta_S)^2] = \mathbb{E}[\mathrm{Var}(\eta \mid \mathcal{G})] \ge 0. \end{aligned} \tag{20}$$

Step 2: Reconstruct the predictive state from the summary. Define the $\mathcal{G}$-measurable state estimate $\widehat{Z} = D_C S$ and the total decayed event weight

$$W_\beta(H) := \sum_i e^{-\beta a_i}, \qquad \mathbb{E}[W_\beta(H)^2] = \mathcal{M}_{\beta,T}.$$

The uniformly bounded event counts imply $N \le K$ almost surely for some finite constant K. Since $a_i \ge 0$ and $\beta > 0$,

$$0 \le W_\beta(H) \le N \le K, \qquad \mathcal{M}_{\beta,T} = \mathbb{E}[W_\beta(H)^2] \le K^2 < \infty.$$

By the linearity of D_C,

$$Z_0 - \widehat{Z} = \sum_i \big(e^{Aa_i} v_C(m_i) - D_C \phi(m_i, a_i)\big) + \sum_i e^{Aa_i} r_i.$$

Using the triangle inequality, the fidelity condition equation 6, and the bounds in equation 4, we obtain, for each admissible history,

$$\begin{aligned} \|Z_0 - \widehat{Z}\|_2 &\le \sum_i \|e^{Aa_i} v_C(m_i) - D_C \phi(m_i, a_i)\|_2 + \sum_i \|e^{Aa_i}\|_{\mathrm{op}} \|r_i\|_2 \\ &\le \rho \sum_i e^{-\beta a_i} + \varepsilon \sum_i e^{-\beta a_i} \\ &= (\varepsilon + \rho) W_\beta(H). \end{aligned} \tag{21}$$

Step 3: Bound the squared probability error. Consider the valid summary-based prediction $\widetilde{\eta} = q_C(\widehat{Z}) = q_C(D_C S) \in [0,1]$. The L-Lipschitz property of q_C and equation 21 imply

$$|\eta - \widetilde{\eta}| \le L \|Z_0 - \widehat{Z}\|_2 \le L(\varepsilon + \rho) W_\beta(H).$$

Since $\eta_S = \mathbb{E}[\eta \mid \mathcal{G}]$ and $\widetilde{\eta}$ is $\mathcal{G}$-measurable, orthogonal decomposition gives

$$\mathbb{E}[(\eta - \widetilde{\eta})^2] = \Delta_{\mathrm{Br}} + \mathbb{E}[(\eta_S - \widetilde{\eta})^2].$$

Squaring the preceding pointwise bound and taking expectations yields

$$\Delta_{\mathrm{Br}} \le \mathbb{E}[(\eta - \widetilde{\eta})^2] \le L^2 (\varepsilon + \rho)^2 \mathcal{M}_{\beta,T}. \tag{22}$$

Step 4: Convert the probability error into an AUC gap. Fix a group $B = b$. Let (H', C', Y') be an independent copy conditional on this group, and use primes for its probabilities and scores. Write $\mathbb{E}_b$ for expectation under these conditional draws. With $h(t) = \mathbf{1}\{t > 0\} + \frac{1}{2}\mathbf{1}\{t = 0\}$ and $\mathrm{sgn}(0) = 0$, balance and symmetrization give, for any $\mathcal{F}$-measurable score f,

$$\begin{aligned} \mathrm{AUC}_b(f) &= 4\mathbb{E}_b[\eta(1 - \eta') h(f - f')] \\ &= \frac{1}{2} + \mathbb{E}_b[(\eta - \eta')\,\mathrm{sgn}(f - f')]. \end{aligned} \tag{23}$$

For completeness, $h(t) = (1 + \mathrm{sgn}(t))/2$ and $\mathbb{E}_b[\eta(1 - \eta')] = 1/4$. Interchanging the two draws in the remaining term gives the second equality. The expectation in that equality is at most $\mathbb{E}_b|\eta - \eta'|$, attained by $f = \eta$. For a $\mathcal{G}$-measurable score, conditioning on the two summaries and contexts replaces η, η' by η_S, η'_S. Thus the corresponding maximum is attained by $f = \eta_S$, and

$$\mathrm{AUC}_b(\eta) = \frac{1}{2} + \mathbb{E}_b|\eta - \eta'|, \qquad \mathrm{AUC}_b(\eta_S) = \frac{1}{2} + \mathbb{E}_b|\eta_S - \eta'_S|.$$

Because B is available to both predictors, these posterior scores are admissible and simultaneously optimal in all groups. Averaging therefore gives the two optimal AUC values defined above.

Let $e = \eta - \eta_S$ and $\Delta_{\mathrm{Br}}(b) = \mathbb{E}[e^2 \mid B = b]$. Since $\sigma(B) \subseteq \mathcal{G}$, $\mathbb{E}[e \mid B] = 0$. Conditional independence of the two draws, the triangle inequality, and Cauchy–Schwarz imply

$$\begin{aligned} \mathrm{AUC}_b(\eta) - \mathrm{AUC}_b(\eta_S) &\le \mathbb{E}_b|e - e'| \\ &\le \sqrt{\mathbb{E}_b[(e - e')^2]} = \sqrt{2\Delta_{\mathrm{Br}}(b)}. \end{aligned}$$

The gap is nonnegative because the information sets are nested. Averaging over B and applying Jensen's inequality gives

$$\begin{aligned} 0 \le \mathrm{AUC}^*_{\mathcal{F}} - \mathrm{AUC}^*_{\mathcal{G}} &\le \mathbb{E}_B \sqrt{2\Delta_{\mathrm{Br}}(B)} \\ &\le \sqrt{2\mathbb{E}_B \Delta_{\mathrm{Br}}(B)} = \sqrt{2\Delta_{\mathrm{Br}}}. \end{aligned} \tag{24}$$

Every optimal AUC lies in $[1/2, 1]$, since a constant score attains $1/2$. Combining this universal bound with equation 22 and equation 24 yields

$$0 \le \mathrm{AUC}^*_{H,C} - \mathrm{AUC}^*_{S,C} \le \min\left\{\frac{1}{2},\ \sqrt{2}\, L(\varepsilon + \rho)\sqrt{\mathcal{M}_{\beta,T}}\right\}. \tag{25}$$

In particular, if $\delta \in (0, 1/2)$ satisfies $\sqrt{2}\, L(\varepsilon + \rho)\sqrt{\mathcal{M}_{\beta,T}} \le \delta$, then $0 \le \mathrm{AUC}^*_{H,C} - \mathrm{AUC}^*_{S,C} \le \delta$, as claimed. □

The state approximation holds separately for each history, so no independence among events or between H and C is required. Independent copies enter only in the definition of population AUC. One evaluation group gives pooled AUC; equal weights over groups give the mean of within-group AUC values. No group size enters the bound.

A consequence used by the extensions. The same argument applies to any measurable reconstruction $g(S, C)$ of a predictive state Z. Under the same balanced evaluation distribution, whenever $\eta = q_C(Z)$ with q_C globally L-Lipschitz,

$$0 \le \mathrm{AUC}^*_{H,C} - \mathrm{AUC}^*_{S,C} \le \min\left\{\frac{1}{2},\ \sqrt{2}\,L\left(\mathbb{E}\|Z - g(S,C)\|_2^2\right)^{1/2}\right\}. \tag{26}$$

Indeed, the projection identity bounds Δ_{Br} by $\mathbb{E}[(\eta - q_C(g(S,C)))^2]$, and equation 24 then bounds the AUC gap. Unlike $D_C S$ in the main proof, g may be nonlinear. This distinction is used in the interaction result below.

E.2 Shared Predictive Directions and Representation Error

This subsection isolates the representation issue by setting $\varepsilon = 0$. Fix the shared context $C = c_0$ and a single evaluation group, and suppress them in the notation. Consider M distinguishable event positions with fixed ages $a_i \in (0, T]$ and visible marks (i, x_i), where $x = (x_1, \ldots, x_M) \in [0,1]^M$. The coordinate i is a visible event key. For a finite family of queries $j \in \{1, \ldots, K\}$, assume that the binary label Y_j has conditional probability

$$\eta_j(x) := \mathbb{E}[Y_j \mid x] = (\Theta x)_j, \qquad \Theta_{ji} \ge 0, \qquad \sum_i \Theta_{ji} = 1, \tag{27}$$

where $\Theta \in \mathbb{R}^{K \times M}$ is fixed. Each row specifies the predictive weighting of events for a query. Allowing all points in the cube imposes no probabilistic independence among the event values.

For each query, this is a special case of the main predictive-state model. Take $A = -\beta$, $r_i = 0$, and

$$v_j(i,u) = e^{\beta a_i}\Theta_{ji}(u - \tfrac{1}{2}), \qquad q_j(z) = \mathrm{clip}_{[0,1]}(\tfrac{1}{2} + z),$$

where $\mathrm{clip}_{[0,1]}(s) = \min\{1, \max\{0, s\}\}$. The fixed age a_i is determined by the visible key. Then $q_j(\sum_i e^{-\beta a_i} v_j(i, x_i)) = (\Theta x)_j$, with $L = 1$. The index j labels the prediction query, not the semantic context C.

We require one additive summary $S_p(x) \in \mathbb{R}^p$ to serve every query. Its encoder may depend on the entire matrix Θ, but does not receive the selected query j; its readout may depend on j. This is an additional shared-encoding requirement across the query family. For a distribution $\mathbb{P}$ of x with $\mathbb{E}_{\mathbb{P}}\eta_j(x) = 1/2$, define

$$\Delta_j^{\mathrm{AUC}}(S_p; \mathbb{P}) := \mathrm{AUC}^*_{x,j}(\mathbb{P}) - \mathrm{AUC}^*_{S_p,j}(\mathbb{P}),$$

where the optimal AUC values concern Y_j with conditional law equation 27. This balance condition is imposed on each query–distribution pair to which the guarantee is applied.

Proposition 2 (Compression through shared predictive directions)**.** *Let $p \ge 1$ be an integer. If a matrix $\widetilde{\Theta}$ of rank at most p satisfies*

$$\tau := \max_{1 \le j \le K} \sum_{i=1}^{M} |\Theta_{ji} - \widetilde{\Theta}_{ji}|,$$

then there is a p-dimensional additive summary, chosen independently of $\mathbb{P}$, such that every query j and every distribution $\mathbb{P}$ with $\mathbb{E}_{\mathbb{P}}\eta_j(x) = 1/2$ satisfy

$$\boxed{0 \le \Delta_j^{\mathrm{AUC}}(S_p; \mathbb{P}) \le \min\left\{\frac{1}{2}, \frac{\tau}{\sqrt{2}}\right\}.} \tag{28}$$

In particular, a summary constructed from a truncated singular value decomposition satisfies

$$0 \le \Delta_j^{\mathrm{AUC}}(S_p; \mathbb{P}) \le \min\left\{\frac{1}{2}, \sqrt{\frac{M}{2}}\,\sigma_{p+1}(\Theta)\right\}, \tag{29}$$

where singular values are ordered decreasingly and extended by zeros. If $\mathrm{rank}(\Theta) \le p$, one shared additive summary attains zero gap for every such query and distribution.

Proof. Factor $\widetilde{\Theta} = UV$, with $U \in \mathbb{R}^{K\times p}$ and $V \in \mathbb{R}^{p\times M}$, padding with zeros when necessary. Let $z = x - \frac{1}{2}\mathbf{1}$ and define

$$S_p(x) = Vz = \sum_{i=1}^{M} V_{:i}(x_i - \tfrac{1}{2}). \tag{30}$$

This is an additive summary of the visible marked events. For query j, use the comparison prediction

$$\widehat{\eta}_j(S_p) = \mathrm{clip}_{[0,1]}(\tfrac{1}{2} + (US_p)_j).$$

The row-sum condition gives $\eta_j(x) = \frac{1}{2} + (\Theta z)_j$. Clipping cannot increase the error relative to a number in $[0,1]$, and $\|z\|_\infty \le 1/2$, so

$$|\eta_j(x) - \widehat{\eta}_j(S_p(x))| \le |((\Theta - \widetilde{\Theta})z)_j| \le \frac{1}{2}\sum_i |\Theta_{ji} - \widetilde{\Theta}_{ji}| \le \frac{\tau}{2}.$$

The projection identity implies

$$\mathbb{E}_{\mathbb{P}}\big[(\eta_j - \mathbb{E}_{\mathbb{P}}[\eta_j \mid S_p])^2\big] \le \mathbb{E}_{\mathbb{P}}[(\eta_j - \widehat{\eta}_j)^2] \le \frac{\tau^2}{4}.$$

For each balanced query–distribution pair, apply equation 24 with a single group. This gives $\Delta_j^{\mathrm{AUC}}(S_p;\mathbb{P}) \le \tau/\sqrt{2}$. Together with the universal upper bound $1/2$, this proves equation 28.

For equation 29, choose $\widetilde{\Theta}$ to retain the largest p singular values of Θ. Its residual $E = \Theta - \widetilde{\Theta}$ satisfies $\|E\|_{\mathrm{op}} = \sigma_{p+1}(\Theta)$. Each row therefore obeys

$$\sum_i |E_{ji}| \le \sqrt{M}\,\|E_{j:}\|_2 \le \sqrt{M}\,\sigma_{p+1}(\Theta).$$

Apply equation 28. If $\mathrm{rank}(\Theta) \le p$, choose $\widetilde{\Theta} = \Theta$ to obtain zero error. □

Connection to ρ. The factorization supplies a linear reconstruction $D_j s = U_{j:}s$. Extend the event encoder to all ages by

$$\phi((i,u),a) = e^{\beta(a_i - a)} V_{:i}(u - \tfrac{1}{2}).$$

At the observed age a_i this gives equation 30. For every $u \in [0,1]$ and $a \in [0,T]$,

$$|D_j\phi((i,u),a) - e^{-\beta a} v_j(i,u)| \le \rho_{\widetilde{\Theta}} e^{-\beta a}, \qquad \rho_{\widetilde{\Theta}} := \frac{1}{2}\max_{j,i} e^{\beta a_i}|\Theta_{ji} - \widetilde{\Theta}_{ji}|.$$

Thus a common approximate factorization controls the fidelity term in the main theorem. Proposition 2 gives a sharper accounting in this subclass by retaining the sum of individual event errors. Its spectrum-based guarantee is small when $\sqrt{M}\,\sigma_{p+1}(\Theta)$ is small; the factor $\sqrt{M}$ must be retained when comparing families of increasing size. The construction concerns one summary shared across queries at the fixed context. It does not assert that separately chosen factorizations at different contexts share the same encoder.

E.3 Recoverable History-Dependent Interactions

We now preserve the base event contributions exactly and examine a structured family of nonzero history-dependent corrections. Return to the shared context C, time window, and balanced evaluation distribution of the main theorem. Suppose event marks are types $m_i \in \{1,\dots,J\}$, with $z_i = (m_i, a_i)$ and $w_i = e^{-\beta a_i}$, where $\beta > 0$. Order events from oldest to newest, breaking time ties by a fixed rule. Consider the scalar predictive state

$$Z = \sum_i w_i v_C(m_i) + \lambda \sum_{j<i} w_i w_j K_C(z_i, z_j), \qquad \eta(H,C) = q_C(Z), \tag{31}$$

where $\lambda \in \mathbb{R}$, $v_C(m) \in \mathbb{R}$, and $q_C : \mathbb{R} \to [0,1]$ is globally L-Lipschitz. The kernel K_C is symmetric in its two complete timestamped events. Assume $|K_C| \le K_0$ and

$$W := \sum_i w_i \le W_0 < \infty$$

almost surely, uniformly over the allowed contexts. Then equation 31 has the form of equation 4 with $A = -\beta$ and

$$r_i = \lambda \sum_{j<i} w_j K_C(z_i, z_j), \qquad |r_i| \le |\lambda| K_0 W_0. \tag{32}$$

The bound on these corrections is not required to approach zero.

Regularity and accumulation. Write $k_{mn,C}(s,t) = K_C((m,s),(n,t))$. Assume each $k_{mn,C}$ is twice continuously differentiable in its age arguments on $[0,T]^2$, with derivatives continuous up to the boundary. Let $\omega_{mn}(C) = \omega_{nm}(C) \in [0,1]$ satisfy

$$\begin{aligned} |\partial_s k_{mn,C}|,\ |\partial_t k_{mn,C}| &\le \ell\, \omega_{mn}(C), \\ \|\nabla^2 k_{mn,C}\|_{\mathrm{op}} &\le K_2\, \omega_{mn}(C), \end{aligned} \tag{33}$$

uniformly over types, ages, and contexts, for finite nonnegative constants ℓ and K_2. Define

$$W_m := \sum_{i:m_i=m} w_i, \qquad D(H,C) := \max_m \sum_n \omega_{mn}(C) W_n, \qquad D(H,C) \le D_0 < \infty \tag{34}$$

almost surely. Here D measures the mass of interactions sensitive to time around each event type. Taking every $\omega_{mn} = 1$ gives $D = W$ and allows $D_0 = W_0$. Smaller D_0 is possible when each type interacts sensitively with a bounded amount of event mass. For example, at most d_0 such neighboring types, each with mass at most b_0, give $D_0 \le d_0 b_0$. A zero coefficient ω_{mn} permits nonzero interactions that are constant in the two age arguments.

An additive temporal summary. For an integer $r \ge 1$, partition $[0,T]$ into r intervals of width $h = T/r$, assigning boundary points by a fixed rule. A cell $b = (m,k)$ specifies an event type and a time interval. Let m_b denote its type and write $i \in b$ when event i belongs to that cell. Retain the $2Jr$ additive coordinates

$$U_b = \sum_{i \in b} w_i, \qquad V_b = \sum_{i \in b} w_i^2, \qquad S_r(H) = (U,V). \tag{35}$$

Each coordinate is a sum of an event-wise function of (m_i, a_i), and the encoder is independent of C. In particular, $\sum_i w_i v_C(m_i) = \sum_b v_C(m_b) U_b$ is preserved exactly.

Proposition 3 (Small compression gaps with nonzero corrections)**.** *Under equation 31–equation 35, let* $\Delta_r^{\mathrm{AUC}} = \mathrm{AUC}^*_{H,C} - \mathrm{AUC}^*_{S_r,C}$. *For every integer* $r \ge 1$,

$$\boxed{0 \le \Delta_r^{\mathrm{AUC}} \le \min\left\{\frac{1}{2}, \frac{\sqrt{2}\, L|\lambda| T^2 D_0 W_0}{r^2}\left(\frac{K_2}{2} + \frac{\beta\ell}{4}\right)\right\}.} \tag{36}$$

The guarantee is uniform over all joint distributions of (H,C) *satisfying these structural and evaluation conditions. With the displayed constants fixed, the AUC gap tends to zero as* r *increases, without decreasing* λ *or the correction bound in equation 32.*

Proof. We construct a nonlinear state reconstruction from the additive coordinates in equation 35.

Step 1: Recover a representative age for each cell. For a nonempty cell, define

$$\bar{a}_b = -\frac{1}{\beta} \log \frac{V_b}{U_b}. \tag{37}$$

Since V_b/U_b is a weighted average of $e^{-\beta a_i}$ within the cell, $\bar{a}_b$ lies in its interval. For an empty cell choose the midpoint of its interval. To control the first moments around these representative ages, fix a nonempty cell and let

$$F_b(s) = \log\left(\sum_{i \in b} \frac{w_i}{U_b} e^{-s a_i}\right).$$

Then $F_b(0) = 0$, and $F_b''(s)$ is the variance of ages under the weights proportional to $w_i e^{-s a_i}$. These ages lie in an interval of width h, so $0 \le F_b''(s) \le h^2/4$. The variance bound follows by comparing to the midpoint of that interval. Moreover,

$$\bar{a}_b = -F_b(\beta)/\beta, \qquad -F_b'(0) = \frac{\sum_{i \in b} w_i a_i}{U_b}, \qquad -F_b'(\beta) = \frac{\sum_{i \in b} w_i^2 a_i}{V_b}.$$

Integration of F_b'' gives

$$\begin{aligned} 0 \le -F_b'(0) + \frac{F_b(\beta)}{\beta} &= \frac{1}{\beta}\int_0^\beta (\beta - s) F_b''(s)\, ds \le \frac{\beta h^2}{8}, \\ 0 \le F_b'(\beta) - \frac{F_b(\beta)}{\beta} &= \frac{1}{\beta}\int_0^\beta s F_b''(s)\, ds \le \frac{\beta h^2}{8}. \end{aligned}$$

Writing $\delta_i = a_i - \bar{a}_b$ for $i \in b$, define $\mu_b = \sum_{i \in b} w_i \delta_i$ and $\nu_b = \sum_{i \in b} w_i^2 \delta_i$. The preceding inequalities imply

$$0 \le \mu_b \le \frac{\beta h^2}{8} U_b, \qquad -\frac{\beta h^2}{8} V_b \le \nu_b \le 0. \tag{38}$$

They also hold for empty cells, where both moments vanish.

Step 2: Reconstruct the interactions. Set

$$G_{bc} = k_{m_b m_c, C}(\bar{a}_b, \bar{a}_c), \qquad d_{bc} = \partial_s k_{m_b m_c, C}(\bar{a}_b, \bar{a}_c),$$

and define the measurable reconstruction

$$\widehat{Z} = \sum_b v_C(m_b) U_b + \frac{\lambda}{2} \left(U^\top G U - \sum_b G_{bb} V_b \right). \tag{39}$$

The subtracted diagonal removes each event's pairing with itself. Consequently, the second term equals the sum over distinct event pairs with their ages replaced by the corresponding representative ages.

For events $i \in b$ and $j \in c$, Taylor expansion and kernel symmetry give

$$K_C(z_i, z_j) - G_{bc} = d_{bc} \delta_i + d_{cb} \delta_j + R_{ij},$$

where

$$|R_{ij}| \le \frac{K_2 \omega_{m_b m_c}(C)}{2} (\delta_i^2 + \delta_j^2) \le K_2 \omega_{m_b m_c}(C) h^2.$$

Summing the linear terms over all distinct unordered pairs yields

$$\mathcal{T}_1 = \sum_{b,c} d_{bc} \mu_b U_c - \sum_b d_{bb} \nu_b.$$

The last sum again removes the self-pair terms. Using equation 38 and $|d_{bc}| \le \ell \omega_{m_b m_c}(C)$, and writing $Q_m = \sum_{i : m_i = m} w_i^2$, we obtain

$$|\mathcal{T}_1| \le \frac{\beta \ell h^2}{8} \left(\sum_{m,n} \omega_{mn}(C) W_m W_n + \sum_m \omega_{mm}(C) Q_m \right).$$

The first sum is at most DW. The second is also at most DW, because $Q_m \le W_m^2$ and $\omega_{mm}(C) W_m \le D$. Thus

$$|\mathcal{T}_1| \le \frac{\beta \ell h^2}{4} DW.$$

The total remainder satisfies

$$|\mathcal{T}_2| \le K_2 h^2 \sum_{j<i} \omega_{m_i m_j}(C) w_i w_j \le \frac{K_2 h^2}{2} DW.$$

Since the base contribution is reconstructed exactly, these two bounds give, separately for every admissible history and context,

$$|Z - \widehat{Z}| \le |\lambda| h^2 DW \left(\frac{K_2}{2} + \frac{\beta \ell}{4} \right). \tag{40}$$

Finally, use $D \le D_0$, $W \le W_0$, and $h = T/r$ to bound the root mean squared state error by the same uniform expression. Applying equation 26 gives equation 36. No independence among events is used. □

Interpretation and scope. The summary preserves the base event contributions exactly, isolating the effect of history-dependent corrections. These corrections can remain nonzero while their reconstruction error decreases. The rate comes from the temporal regularity of the interactions and from the two weighted statistics retained in each cell. The total state is reconstructed nonlinearly; the conclusion therefore uses equation 26 rather than treating the interaction as part of the main theorem's linear fidelity condition.

The representation has $2Jr$ coordinates, and the displayed readout evaluates $O(J^2r^2)$ cell pairs using exact statistics. Large mass, narrow temporal variation scales, or a growing number of types can increase the resources required for a small guaranteed gap. Thus the proposition provides a quantitative approximation result, not a claim that its sufficient dimension is always much smaller than the history. Together, the two propositions identify separate structural routes to small optimal-AUC gaps. They establish existence guarantees for the constructed summaries.

F Mechanism Analysis Details

F.1 Evidence Analysis

The goal of the evidence analysis is to vary the amount and representation of historical information while keeping the downstream predictor deliberately simple. We use two history scopes throughout the study. The short scope contains the most recent 47 interactions, following the default history length used by DyGFormer under the DTGB protocol, while the long scope extends the same history to 200 interactions. The latter provides a substantially broader temporal context without changing the underlying evaluation protocol.

Let $\mathbf{r}_{uv}(t)$ denote the four structural and temporal statistics: recency, target activity (historical interaction count), past pair interaction count, and resource allocation. For the **Heuristic** setting, an XGBoost classifier takes $[\mathbf{r}_{uv}(t) \,\|\, c_{uv}(t)]$ as input. To compute the semantic similarity $c_{uv}(t)$, we first propagate the frozen E5 embeddings over the available historical neighborhood:

$$\tilde{\mathbf{e}}_u(t) = \frac{\mathbf{e}_u}{d_u(t)+1} + \sum_w \frac{a_{uw}(t)}{\sqrt{(d_u(t)+1)(d_w(t)+1)}}\mathbf{e}_w, \qquad c_{uv}(t) = \frac{\tilde{\mathbf{e}}_u(t)^\top \tilde{\mathbf{e}}_v(t)}{\|\tilde{\mathbf{e}}_u(t)\|_2 \|\tilde{\mathbf{e}}_v(t)\|_2}. \quad (41)$$

Here, $\mathbf{e}_u$ is the frozen text embedding of node u, $a_{uw}(t)$ counts interactions with w within the evidence scope retained for u, and $d_u(t) = \sum_w a_{uw}(t)$ is the corresponding interaction count. In the local setting, this scope contains the most recent K interactions involving u strictly before t; in the global setting, it contains all such past interactions. The added 1 accounts for a self-loop. **Snapshot** and **Sequence** combine richer endpoint representations with $\mathbf{r}_{uv}(t)$, without separately including $c_{uv}(t)$. For both settings, we use the same uniform aggregation operator and prediction head. Let $\mathcal{S}_u^{(m)}(t) = \{\mathbf{x}_{u,1}^{(m)}, \ldots, \mathbf{x}_{u,n_u}^{(m)}\}$ denote the evidence associated with node u under representation $m \in \{\mathrm{snap}, \mathrm{seq}\}$, where each element contains the corresponding semantic and temporal attributes. We compute

$$\bar{\mathbf{h}}_u^{(m)} = \frac{1}{|\mathcal{S}_u^{(m)}(t)|} \sum_{\mathbf{x} \in \mathcal{S}_u^{(m)}(t)} \phi(\mathbf{x}), \qquad \bar{\mathbf{h}}_v^{(m)} = \frac{1}{|\mathcal{S}_v^{(m)}(t)|} \sum_{\mathbf{x} \in \mathcal{S}_v^{(m)}(t)} \phi(\mathbf{x}), \quad (42)$$

where $\phi(\cdot)$ is a shared lightweight input projection. The prediction is then obtained by

$$s^{(m)}(u,v,t) = \mathrm{MLP}\left(\left[\bar{\mathbf{h}}_u^{(m)} \,\|\, \bar{\mathbf{h}}_v^{(m)} \,\|\, \mathbf{r}_{uv}(t)\right]\right). \quad (43)$$

Thus, the four-dimensional heuristic vector is retained in the Snapshot and Sequence settings, while their learned endpoint representations replace the separately computed semantic-similarity feature used by the Heuristic setting.

Table 10 summarizes the shared pair scorer and training settings for the Snapshot and Sequence variants on Enron and GDELT. These variants share frozen E5 embeddings, heuristic fusion, the pair-scorer architecture, optimizer, learning rate, batch size, maximum training budget, and training seeds. Within each representation, the short and long variants use history caps of 47 and 200, respectively.

F.2 Processing Analysis

Historical Evidence and Input Representations. For a query (u, v, t), each graph-based processor retrieves the most recent causal one-hop interactions of each endpoint. Let

$$\mathcal{H}_u^{(K)}(t) = \left[(w_i, a_i, \tau_i)\right]_{i=1}^{m_u}, \qquad m_u \le K, \quad \tau_i < t, \quad (44)$$

Table 10: Shared scorer and training settings for Snapshot and Sequence on Enron and GDELT. S/L denote history caps $K = 47/200$ within each representation.

Setting	Shared value
Node text embeddings	Frozen E5-large-v2, 1024 dimensions
Pair-scorer input	Endpoint concatenation + 4 heuristics
Pair-scorer architecture	$2052 \to 1024 \to 1$
Pair-scorer activation	ReLU
Pair-scorer dropout / LayerNorm	0 / disabled
Heuristic fusion	Full-history recency, target activity, past pair count, interaction-count-weighted RA; late concat
Optimizer	Adam
Learning rate	10^{-4}, constant
Weight decay	0
Training / evaluation batch size	256 / 256
Negatives per positive	1 random negative
Maximum training budget	50 epochs

Table 11: Main hyperparameters for the processing study. Architecture-specific settings are described in the text.

Hyperparameter	Graph models	LLM
History cap per endpoint	47 events	47 events
Learning rate	10^{-4}	10^{-4}
Effective batch size	256	8
Training epochs	At most 50	1
Dropout	0.1	0.05 (LoRA)
Negatives per positive	1	1

denote the retained history, where w_i is the interaction partner and a_i identifies the historical edge. Repeated interactions remain separate events. We prepend a root token for the queried endpoint. Each event token concatenates learned projections of a frozen 1024-dimensional E5-large-v2 node embedding, a 768-dimensional edge feature, and a 100-dimensional relative-time encoding. Each channel is projected to 50 dimensions, giving a base token width of $D = 150$. Including the root, each endpoint has at most 48 tokens. The main hyperparameters are summarized in Table 11.

The base comparison disables neighbor co-occurrence encodings and all four historical heuristic features. No additional learned positional embeddings are used; relative-time encodings remain available. The graph-based models therefore access the same recent-history scope and feature channels. All graph processors and LLM variants follow the DTGB chronological 70/15/15 split. Training uses Adam with zero weight decay and no learning-rate scheduler.

Uniform and Exponentially Weighted Processing. **Mean Pool** uses fixed uniform cross-token aggregation in the sequence processor. Its two pre-LayerNorm residual blocks retain learned value and output projections and a $D \to 4D \to D$ feed-forward MLP, while bypassing content-dependent query-key weighting. The fixed mixing operates on the concatenated source and destination tokens, followed by separate endpoint readouts; it is not simply an average of raw endpoint histories followed by a classifier.

Exp-Decay uses fixed-weight mixing that emphasizes more recent events. If $\mathcal{I}$ indexes the valid tokens from both endpoints, its normalized mixing weights are

$$\alpha_j = \frac{\exp(-\gamma r_j)}{\sum_{k \in \mathcal{I}} \exp(-\gamma r_k)}, \qquad j \in \mathcal{I}. \tag{45}$$

Here r_j is the recency rank within the corresponding endpoint history: the root has rank zero, the newest event has rank one, and older events have larger ranks. Weights are normalized jointly across the endpoint sequences. This is decay by event rank rather than elapsed time; time differences remain available through the temporal input channel. Setting $\gamma = 0$ gives uniform weights under the same implementation. Exp-Decay also uses two sequence blocks, with $\gamma = 0.1$ for GDELT and Enron.

Graph Convolution and Self-Attention. The **GIN** results use an endpoint-star implementation. Each root receives messages from its own historical event tokens, and the aggregation module is shared by the two endpoints. With root representation $\mathbf{x}_{u,0}$ and event representations $\mathbf{x}_{u,i}$, the update is

$$\mathbf{z}_u = (1+\epsilon)\mathbf{x}_{u,0} + \sum_{i=1}^{m_u} \mathbf{x}_{u,i}, \tag{46}$$

$$\mathbf{h}_u = \mathbf{x}_{u,0} + \mathrm{MLP}_{\mathrm{GIN}}\big(\mathrm{LN}(\mathbf{z}_u)\big). \tag{47}$$

The scalar ϵ is trainable and initialized to zero. This is a history-to-root aggregation with normalization and a root residual, rather than an unmodified GINConv layer. The two endpoint histories do not exchange messages in the base model. Their root outputs are projected to 1024 dimensions and concatenated for a $2048 \to 1024 \to 1$ prediction MLP.

Self-Attn jointly processes the concatenated endpoint sequences using two pre-LayerNorm residual blocks and two attention heads (75 dimensions per head in Base), with a $D \to 4D \to D$ channel MLP. Learned query-key weights allow content-dependent interactions between events, including events from different endpoints. Token averaging produces the endpoint readouts before projection and prediction.

LLM Processing and Entity Anonymization. Both LLM variants use Llama-3.1-8B-Instruct with task-specific LoRA fine-tuning on 10,000 uniformly sampled positive training queries and an equal number of negatives per dataset. LoRA has rank 8 and scaling parameter 32, and is applied to the attention projections (q, k, v, o) and feed-forward projections (gate, up, down) in all transformer layers. Pretrained weights remain frozen; training minimizes cross-entropy on the binary answer token, excluding prompt tokens. We evaluate the final adapter after one epoch. The maximum input length is 8,192 text tokens.

The prompt supplies the query and recent histories of its endpoints. Separate mutual-history and common-neighbor sections, heuristic scores, expert predictions, and reasoning guidance are disabled. Relations inferable from repeated entities in the supplied histories remain observable. We request a binary answer without chain-of-thought generation and score a query as

$$s_{\mathrm{LLM}}(u,v,t) = \frac{p(X=1 \mid \mathcal{P}_{u,v,t})}{p(X=0 \mid \mathcal{P}_{u,v,t}) + p(X=1 \mid \mathcal{P}_{u,v,t})}, \tag{48}$$

where $\mathcal{P}_{u,v,t}$ denotes the input prompt.

LLM (Entity) retains the original entity names. **LLM (Anonymous)** uses prompt-local aliases: the endpoints are `Source` and `Target`, and other entities receive labels such as `N1`, `N2`, and so on, in order of first appearance. A repeated entity receives the same alias, preserving interaction structure and timestamps. The anonymous adapter is trained and evaluated with this representation; the experiment is not merely test-time renaming of the entity-trained adapter.

Making Structural Evidence Explicit. We independently enable two structural components for GIN and Self-Attn. **+Cooccur** adds the native neighbor co-occurrence channel. For each entity appearing in the endpoint sequences, it encodes its occurrence counts in both sequences, including root positions and excluding padding. This exposes repeated neighbors and overlap within the supplied histories without extending their temporal window. **+Heur** adds four query-level features from the full available causal history: recency, destination activity measured by historical interaction count, past pair interactions, and interaction-count-weighted resource allocation (RA) over common neighbors. These features reuse the existing heuristic extractor and its transformations and normalization. They are concatenated with the endpoint representations immediately before the prediction head. Unlike co-occurrence, this component can use interactions outside the 47-event window. Global history means all available events strictly before the query time, rather than future interactions. **Base** disables both components, and **+All** enables both. Within each backbone, the four settings share the training recipe and implementation, with only the feature switches changing. Enabling the native co-occurrence module adds a 50-dimensional channel and also changes parameter count, so this measures the benefit of the complete module.

Table 12: AUC (mean ± SD across seeds) for the structural augmentations in Figure 3. Each cell reports Test/New-node AUC.

Dataset	Setting	GIN	Self-Attn
GDELT	Base	$0.9682 \pm 0.0001/0.8933 \pm 0.0005$	$0.9680 \pm 0.0008/0.8942 \pm 0.0040$
	+Cooccur	$0.9722 \pm 0.0002/0.9099 \pm 0.0006$	$0.9715 \pm 0.0002/0.9068 \pm 0.0014$
	+Heur	$0.9737 \pm 0.0001/0.9151 \pm 0.0005$	$0.9729 \pm 0.0003/0.9131 \pm 0.0018$
	+All	$0.9741 \pm 0.0003/0.9158 \pm 0.0014$	$0.9736 \pm 0.0000/0.9150 \pm 0.0008$
Enron	Base	$0.9699 \pm 0.0023/0.8940 \pm 0.0088$	$0.9673 \pm 0.0030/0.8972 \pm 0.0061$
	+Cooccur	$0.9810 \pm 0.0008/0.9354 \pm 0.0007$	$0.9792 \pm 0.0012/0.9343 \pm 0.0023$
	+Heur	$0.9801 \pm 0.0007/0.9280 \pm 0.0009$	$0.9773 \pm 0.0012/0.9277 \pm 0.0048$
	+All	$0.9818 \pm 0.0008/0.9362 \pm 0.0039$	$0.9806 \pm 0.0006/0.9359 \pm 0.0020$

G LLM Complementarity Analysis Details

Evaluation and uncertainty routing. We use the GNN and entity-aware LLM from the main experiments and evaluate all candidate edges in the transductive and inductive test splits of Enron and GDELT, retaining the benchmark's negative samples. The entity-aware Llama-3.1-8B is fine-tuned on the strict DTGB training partition, excluding inductively reserved nodes from supervised training examples and the training-history graph. For each endpoint, we merge incoming and outgoing interactions strictly before the query time and retain the most recent 47 events in total. The prompt preserves event directions, timestamps, and multiplicities and lists the retained events chronologically, without entity-centric history compression. Each candidate (u, v, t) is scored and routed individually, without using its association with a positive–negative query pair. We use a common definition of GNN uncertainty throughout the structural, correction-recall, regional-routing, and score-fusion analyses. For each dataset, we fix the GNN checkpoint and evaluation procedure across these analyses and select a single decision threshold by maximizing balanced accuracy on its complete held-out validation split. The LLM threshold is selected separately on 5,000 held-out validation positive queries and one benchmark negative per query, giving 10,000 validation candidates per dataset. Both thresholds are frozen before test evaluation. We define GNN uncertainty as $U(e) = -|z_G(e) - \tau_G|$, where $z_G(e)$ is its logit and τ_G its validation threshold. For uncertainty-based selection at coverage c, we select the highest-scoring $\lfloor cN \rfloor$ candidate edges, where N is the number of candidates in the corresponding evaluation pool. Ties are broken using fixed random priorities independent of labels. The uncertainty baselines in the correction-recall, regional-routing, and score-fusion analyses share the same selected edges at matching coverage levels within each evaluation pool. A correction is an edge incorrectly classified by the GNN but correctly classified by the LLM under their respective thresholds. Correction recall is the number of selected corrections divided by the total number of corrections in the corresponding test split. This fixed-coverage evaluation uses an offline batch budget.

Table 13: Numerical values underlying the correction-recall figure. Recall (%) is the fraction of all single-edge LLM corrections captured by GNN-uncertainty selection at the indicated candidate-edge coverage. T/I denote the complete transductive/inductive test pools. Values are rounded to two decimal places.

Setting	10% coverage	20% coverage	30% coverage
Enron T	70.74	92.29	97.49
Enron I	43.02	71.79	88.21
GDELT T	80.73	96.45	98.82
GDELT I	54.96	82.23	93.45

Structural support. We compute structural indicators on the undirected simple graph containing interactions strictly before each query time. The three weakness scores are $-|N(u) \cap N(v)|$, $\mathbf{1}[v \notin N(u)]$, and $-[r(u) + r(v)]/2$, where

$$d_{\text{simple}}(x) = |N(x)|, \qquad r(x) = \frac{1}{d_{\text{simple}}(x)} \sum_{w \in N(x)} \sqrt{\frac{d_{\text{simple}}(x) + 1}{d_{\text{simple}}(w) + 1}}.$$

Here, $N(x)$ is the set of distinct historical neighbors and $d_{\text{simple}}(x)$ is the degree in this undirected simple graph; we set $r(x) = 0$ for isolated nodes. Larger scores indicate weaker structural support.

Figure 5(a) reports Spearman correlations between these scores and GNN uncertainty over all positive interactions in each test split, using average ranks for ties. This label-conditioned analysis is retrospective; labels do not enter the routing rule.

Router architecture. The regional-routing comparison uses the same uncertainty $U(e)$ and frozen held-out-validation threshold τ_G defined above. The chronological validation portion of the training context is used to select router checkpoints and temperatures, without refitting the GNN decision threshold. Both learned routers augment GNN uncertainty through a learned residual,

$$s(e) = b_e + f_\theta(x_e), \qquad b_e = \frac{U(e) - \mathrm{median}_{\mathrm{fit}}\, U}{\max(\mathrm{IQR}_{\mathrm{fit}}\, U, 10^{-6})}.$$

XGBoost receives the three structural indicators and b_e. The GLANCE-style router uses the richer input

$$x_e = [f_u \| f_v \| h_e \| d_{\mathrm{simple},t}(v) \| H_e \| M_e \| b_e] \in \mathbb{R}^{2052+d_h},$$

where f_u, f_v are 1024-dimensional normalized E5 features, h_e is the d_h-dimensional GNN pre-head representation, and $d_{\mathrm{simple},t}(v)$ is the target's number of distinct historical neighbors in the undirected simple graph. The updated checkpoints have $d_h = 64$ for Enron and $d_h = 128$ for GDELT, giving total router input dimensions of 2116 and 2180, respectively. Soft homophily is $H_e = p_Q(u,v)^\top \big(|N_t(v)|^{-1} \sum_{w \in N_t(v)} p_Q(u,w)\big)$, where Q is a feature-only binary classifier trained on the fitting data; isolated targets receive $H_e = 0$. The feature M_e measures predictive mutual information over ten MC-dropout passes through the complete GNN scoring path, including the graph encoder and heuristic fusion, wherever dropout is active. Enron uses the checkpoint's dropout probability 0.2; GDELT has no active dropout and therefore $M_e = 0$. The residual network has two hidden layers of width 2307, ReLU activations, dropout 0.2, and a scalar output. Its output layer is initialized to zero. LLM predictions provide training supervision but are not required as routing inputs.

Training objective. We keep both experts frozen and train the routers to maximize their regional AUC difference under a 20% coverage budget. During training, hard selection is relaxed to sigmoid weights $q_i \approx \sigma((s_i - \mu)/T)$, with μ chosen so that $n^{-1} \sum_i q_i = 0.2$. The objective is

$$\mathcal{J}(\theta) = \frac{\sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} q_i q_j \left[C(z_i^L, z_j^L) - C(z_i^G, z_j^G)\right]}{\left(\sum_{i \in \mathcal{P}} q_i\right)\left(\sum_{j \in \mathcal{N}} q_j\right)},$$

where $\mathcal{P}$ and $\mathcal{N}$ are positive and negative fitting examples, z^L, z^G are fixed expert scores, and $C(a,b) = \mathbf{1}[a > b] + \frac{1}{2}\mathbf{1}[a = b]$. Thus, the objective measures the LLM's weighted AUC advantage within the softly selected region. The MLP is optimized by backpropagation, while XGBoost fits trees to the objective's score gradients. Checkpoints and temperature $T \in \{0.25, 1.0\}$ are selected using validation regional AUC gain under hard 20% selection. At test time, each trained router selects the top 20% without further tuning; we report the mean and standard deviation of regional AUC gains across five seeds.

Table 14: Numerical values underlying the regional AUC-gain figure. Each entry is $\mathrm{AUC}_{\mathrm{LLM}} - \mathrm{AUC}_{\mathrm{GNN}}$ within that selector's selected 20% of candidate edges, on the 0–1 scale. T/I denote the complete transductive/inductive test pools. XGBoost and GLANCE are the uncertainty-augmented routers; GLANCE excludes its fusion refiner. Values are reported as mean ± standard deviation and rounded to six decimal places.

Setting	Uncertainty	XGBoost	GLANCE
Enron T	0.071890 ± 0.000000	0.068888 ± 0.003841	0.037366 ± 0.001449
Enron I	0.148774 ± 0.000000	0.113092 ± 0.001412	0.053212 ± 0.004010
GDELT T	0.002457 ± 0.000000	0.002610 ± 0.000294	0.002152 ± 0.000034
GDELT I	0.027186 ± 0.000000	0.031349 ± 0.000861	0.028821 ± 0.000033

Score fusion and alignment. For score fusion, we use the same uncertainty $U(e) = -|z_G(e) - \tau_G|$ and frozen held-out-validation threshold as in the preceding analyses. At 20% coverage, the selected region is identical to that of the uncertainty baseline in the correction-recall and regional-routing

analyses on the same evaluation pool. Within the uncertainty-selected region $\mathcal{S}$, we combine the frozen expert scores using either an affine model, $f_{\text{aff}}(e) = az_G(e) + bz_L(e) + c$, or a two-hidden-layer MLP with widths 64/64 and ReLU activations. Both models receive fitting-set-standardized expert scores, are trained with binary cross-entropy, and are selected by validation AUC. To examine score compatibility with the unrouted GNN predictions, we separately quantile-align each fusion model to a common GNN reference distribution:

$$\widetilde{f}_m(e) = F^{-1}_{G,\text{ref}}(F_{m,\text{ref}}(f_m(e))), \qquad m \in \{\text{aff}, \text{MLP}\}.$$

The empirical distributions are estimated on the uncertainty-selected portion of the fitting data, without using test data. The final prediction is

$$z_{\text{final}}(e) = \begin{cases} \widetilde{f}_m(e), & e \in \mathcal{S}, \\ z_G(e), & e \notin \mathcal{S}. \end{cases}$$

We also evaluate the unaligned variant using f_m directly. This monotone alignment preserves within-region ordering apart from ties, while adjusting comparisons between routed and unrouted edges. For regional evaluation, we compute AUC within each evaluation batch after restricting it to the selected examples in $\mathcal{S}$ and then average across batches. For full-test evaluation, we compute AUC on every complete evaluation batch and likewise average across batches. Reported gains are the differences between the corresponding AUCs and the GNN baseline.

H LLM Prompt Templates

For direct LLM prediction, each candidate edge is converted into a textual query containing the two endpoint identities, the prediction time, and their causal interaction histories. For each endpoint, we merge its incoming and outgoing interactions observed strictly before the query time and retain the most recent 47 events in total, or all available events if fewer exist. The retained events are listed from oldest to newest, with equal timestamps ordered by their original data-row order. We preserve the original interaction directions, timestamps, and repeated events. The source-side history is placed before the target-side history. Both LLM variants use this history-selection rule during training and evaluation. The model is instructed to output a binary answer in the fixed form *"The answer is: X"*, where $X \in \{0, 1\}$. Rather than using the generated hard label, we compute the prediction score from the probabilities assigned to the two answer tokens,

$$s_{\text{LLM}} = \frac{p(X = 1)}{p(X = 0) + p(X = 1)}. \tag{49}$$

We use the same prompt structure for the entity-aware and anonymized variants. **LLM (Entity)** retains the original entity surface forms, allowing the model to exploit their textual semantics and potentially entity-specific knowledge acquired during pretraining. **LLM (Anonymous)** replaces all entity surface forms with consistent anonymous identifiers while preserving interaction directions, timestamps, repeated-entity structure, and the prediction query. We first provide the generic prompt template and then show the entity-aware and anonymized versions of the same example query.

```
System:
You are a temporal link prediction analyst. Output only the final answer
in the format: "The answer is: X" (where X is 0 or 1).

User:
You are analyzing a temporal interaction graph. Your task is to predict
whether a future link between the source entity and the target entity
will occur at the prediction time using only historical interaction
patterns observed before that time.

ENTITY INFORMATION
Source: <SOURCE>
Target: <TARGET>
Prediction Time: <PREDICTION_TIME>
```

```
SOURCE HISTORY
The most recent actions received BY <SOURCE> or performed BY them
(up to 47 events):
<SOURCE_HISTORY>

TARGET HISTORY
The most recent actions received BY <TARGET> or performed BY them
(up to 47 events):
<TARGET_HISTORY>

PREDICTION TASK
Question: Will a link between the following entities occur at the
prediction time?
Link: <SOURCE> --> <TARGET>
Prediction Time: <PREDICTION_TIME>

Instruction:
1. The format MUST be exactly: "The answer is: X"
   (where X is 0 or 1).

0 = Link will NOT occur
1 = Link WILL occur

Response:
```

```
System:
You are a temporal link prediction analyst. Output only the final answer
in the format: "The answer is: X" (where X is 0 or 1).

User:
You are analyzing a temporal interaction graph. Your task is to predict
whether a future link between the source entity and the target entity
will occur at the prediction time using only historical interaction
patterns observed before that time.

ENTITY INFORMATION
Source: Presidential Press Service (United States)
Target: Klaus Kinkel (Germany)
Prediction Time: 624

SOURCE HISTORY
The most recent actions received BY Presidential Press Service
(United States) or performed BY them (up to 47 events):
- t=248: Presidential Press Service (United States) --> BBC
  (United Kingdom)
- t=621: Presidential Press Service (United States) --> Al Qaeda
  (None)

TARGET HISTORY
The most recent actions received BY Klaus Kinkel (Germany) or performed
BY them (up to 47 events):
- t=428: Christian Lindner (Germany) --> Klaus Kinkel (Germany)

PREDICTION TASK
Question: Will a link between the following entities occur at the
prediction time?
Link: Presidential Press Service (United States) -->
      Klaus Kinkel (Germany)
Prediction Time: 624

Instruction:
1. The format MUST be exactly: "The answer is: X"
```

```
   (where X is 0 or 1).

0 = Link will NOT occur
1 = Link WILL occur

Response:



System:
You are a temporal link prediction analyst. Output only the final answer
in the format: "The answer is: X" (where X is 0 or 1).

User:
You are analyzing a temporal interaction graph. Your task is to predict
whether a future link between the source entity and the target entity
will occur at the prediction time using only historical interaction
patterns observed before that time.

ENTITY INFORMATION
Source: Source
Target: Target
Prediction Time: 624

SOURCE HISTORY
The most recent actions received BY Source or performed BY them
(up to 47 events):
- t=248: Source --> N1
- t=621: Source --> N2

TARGET HISTORY
The most recent actions received BY Target or performed BY them
(up to 47 events):
- t=428: N3 --> Target

PREDICTION TASK
Question: Will a link between the following entities occur at the
prediction time?
Link: Source --> Target
Prediction Time: 624

Instruction:
1. The format MUST be exactly: "The answer is: X"
   (where X is 0 or 1).

0 = Link will NOT occur
1 = Link WILL occur

Response:
```

The two instantiated prompts correspond to exactly the same prediction query and the same selected historical events. Anonymization changes only the entity surface forms: the source and target are mapped to `Source` and `Target`, while other entities receive `N1`, `N2`, and so on in order of first appearance in the displayed histories. Repeated appearances of an entity use the same identifier across both history sections within a prompt. The mapping is rebuilt for every query, so identifiers do not carry persistent entity identities across prompts. The interaction directions, timestamps, multiplicities, and temporal ordering are unchanged.