TTGBENCH: BENCHMARKING TEMPORAL INTERACTION DYNAMICS IN TEXT-ATTRIBUTED TEMPORAL GRAPHS

Anonymous authors
Paper under double-blind review

ABSTRACT

Temporal graph learning models dynamic systems as sequences of timestamped interactions, where both structural and semantic dynamics evolve over time. However, existing benchmarks primarily emphasize structural dynamics through temporal link prediction (TLP), while support for semantic dynamics remains limited. Although temporal node classification (TNC) is sometimes included, it is often limited to simple binary settings. Moreover, commonly used datasets exhibit high link repetition, which can obscure whether models truly capture evolving interaction patterns. To address these limitations, we introduce **TTGBench**, a benchmark for temporal interaction dynamics in text-attributed temporal graphs. TTGBench comprises six real-world, text-rich datasets with high structural novelty and diverse semantic transition patterns, and jointly supports TLP as well as multi-class and multi-label TNC. We conduct a comprehensive evaluation of 17 state-of-the-art methods across Temporal Graph Neural Networks (TGNNs) and Large Language Model (LLM)-based paradigms. The results reveal a clear task-dependent performance asymmetry between the two paradigms: TGNN-based methods excel at structural prediction but fail at semantic dynamics, whereas LLM-based predictors show the opposite trend. Through in-depth analysis, we uncover their fundamental limitations and provide insights for developing more comprehensive temporal graph models. TTGBench datasets, leaderboards, and code are available at our project page https://ttgbench.netlify.app.

1 INTRODUCTION

Temporal graphs provide a powerful framework for modeling dynamic systems such as social networks (Zhang et al. (2023); Dileo et al. (2024); Mitra & Paul (2025)), e-commerce platforms (Ding et al. (2019); Zhao et al. (2024)), and financial transaction networks (Zhou et al. (2023); Trinh & Wang (2024)). At their core, these systems evolve along two fundamental dimensions: *structural dynamics* and *semantic dynamics*. Structural dynamics describes how nodes form and dissolve connections over time, typically modeled as *temporal link prediction* (TLP). In parallel, semantic dynamics captures how node labels or roles change over time—such as shifts in user interests—and can be formulated as *temporal node classification* (TNC). A comprehensive understanding of both dimensions is essential for uncovering the underlying dynamics of temporal graphs. Therefore, a benchmark that jointly evaluates TLP and TNC is crucial for assessing whether models can capture the full spectrum of temporal evolution.

Existing benchmarks have significantly advanced temporal graph learning by curating diverse datasets (e.g., DGB Poursafaei et al. (2022), TGB Huang et al. (2023)), introducing text-attributed graphs (DTGB Zhang et al. (2024a)), and establishing unified evaluation pipelines (DyGLib Yu et al. (2023)). Despite these contributions, two fundamental limitations remain. First, most benchmarks focus almost exclusively on structural dynamics through TLP, while largely overlooking semantic drift. Although DyGLib supports TNC, it is restricted to simple binary classification tasks (e.g., whether a user is banned), which fail to capture meaningful and continuous semantic drift. This limitation largely stems from the scarcity of datasets with rich, dynamically evolving semantic labels, as noted in prior work (Kazemi et al. (2020a); Longa et al. (2023). Second, existing datasets often contain a high proportion of repeated edges, leading to overestimated model performance. In fact, many

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.

methods achieve near-saturated results (e.g., over 98% on widely used benchmarks (Yu et al. (2023); Zhang et al. (2024a)), which obscures true model capability and fails to reflect realistic, high-novelty environments—a concern also highlighted by recent study (Yi et al. (2025)).

To address these limitations, we introduce the Text-attributed Temporal Graph Benchmark (TTGBench), a new benchmark that jointly evaluates structural and semantic dynamics. TTGBench comprises six real-world datasets spanning diverse domains and is characterized by a novel property we term *Dual Volatility* (Figure 1): (i) *Structural Volatility*, defined by high link novelty, limited repetition, and highly dynamic interaction patterns; and (ii) *Semantic Volatility*, where node labels evolve frequently with complex, non-trivial dynamics. These properties make TTGBench substantially more challenging than existing benchmarks. Under high structural volatility, memorization-based methods such as EdgeBank Poursafaei et al. (2022) fail due to the scarcity of repeated interactions, as confirmed in Section 6. Beyond global novelty, our datasets exhibit intricate structural evolution patterns (Section 4.1), requiring models to capture fine-grained temporal dependencies. On the semantic side, TTGBench initially supports both multi-class and multi-label TNC, enabling realistic and diverse semantic scenarios with complex label transitions (Section 4.2). Furthermore, all datasets include rich textual attributes, providing essential context for modeling both structural and semantic dynamics.

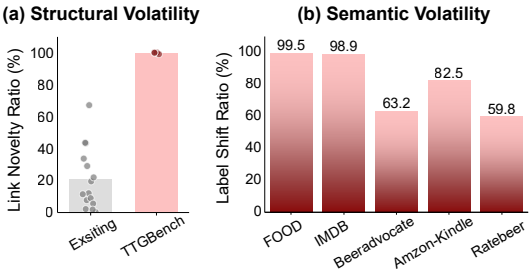

Figure 1: Illustration of dual volatility in TTGBench: structural and semantic volatility. (a) The proposed datasets exhibit high novelty, where previously unseen edges continuously emerge over time, leading to pronounced structural volatility. (b) A substantial proportion of nodes change their labels over time, highlighting the presence of semantic volatility in temporal graphs.

Building on TTGBench, we conduct a comprehensive evaluation of 17 state-of-the-art methods spanning different paradigms, including dominant TGNNs and recent LLM-based approaches. We categorize these methods into two groups based on the predictor: *TGNN-Predictors*, which include both pure TGNNs and LLM-as-Enhancer methods, and *LLM-Predictors*, which directly use LLMs for prediction. Our evaluation revealstask-dependent performance asymmetry: TGNN-Predictors excel at TLP but usually fail on TNC, whereas LLM-Predictors perform strongly on TNC but struggle with TLP. This dichotomy exposes a fundamental limitation—neither paradigm can simultaneously model structural and semantic dynamics. Our empirical results further reveal inherent limitations in how LLM-Predictors encode structural information. In addition, we conduct in-depth analyses to diagnose the failure of TGNNs on semantic tracking, identifying three intrinsic architectural bottlenecks that hinder their ability to utilize semantic signals. Finally, we perform a comprehensive efficiency analysis, providing practical insights into the deployment trade-offs of these methods. Collectively, our findings offer a deeper understanding of current approaches and provide guidance for designing future models that unify structural and semantic learning in temporal graphs.

We summarize our contributions as follows:

- **A unified benchmark:** We introduce **TTGBench**, the first benchmark that jointly evaluates structural and semantic dynamics. It comprises six real-world, text-attributed datasets exhibiting *Dual Volatility*, enabling a more rigorous and holistic assessment of existing models.

- **Comprehensive empirical evaluation:** We conduct a large-scale evaluation of 17 methods spanning both **TGNN-Predictors** and **LLM-Predictors**. Our results reveal task-dependent performance asymmetry: TGNN-Predictors excel at TLP but struggle with TNC, whereas LLM-Predictors show the opposite trend, exposing a fundamental limitation of current paradigms.

- **Diagnostic empirical insights:** We provide in-depth analyses to uncover the root causes of these limitations, including structural deficiencies in LLM-based predictors and architectural bottlenecks in TGNNs for semantic tracking, along with their performance and efficiency trade-offs. These findings offer actionable insights for developing more effective temporal graph models.

2 Related Work

Temporal Graph Learning. Temporal graph learning has attracted increasing attention due to its strong ability to model dynamic real-world systems Kazemi et al. (2020b); Skarding et al. (2021). Among existing approaches, TGNNs have emerged as the dominant paradigm owing to their expressive power Longa et al. (2023). In particular, continuous-time temporal graphs, which model interactions with arbitrary timestamps, provide greater flexibility and have enabled many state-of-the-art methods Kumar et al. (2019); Trivedi et al. (2019); Xu et al. (2020); Cong et al. (2023); Yu et al. (2023). Despite this progress, most studies focus primarily on temporal link prediction. To obtain a more comprehensive understanding of model capabilities, we additionally evaluate these methods on multi-class and multi-label temporal node classification.

Temporal Graph Benchmarks. Existing benchmarks have substantially advanced temporal graph research by providing diverse datasets and standardized evaluation protocols. DGB Poursafaei et al. (2022), DyGLib Yu et al. (2023), TGB Huang et al. (2023), and DGraph Huang et al. (2022) offer unified pipelines for training and evaluation, while DTGB Zhang et al. (2024a) introduces text-attributed temporal graph datasets. TGB-Seq Yi et al. (2025) further identifies excessive edge repetition in prior benchmarks and proposes datasets with higher novelty. However, these benchmarks mainly emphasize link-level tasks and, at the node level, are largely limited to binary classification, overlooking more realistic multi-class scenarios. In contrast, our benchmark introduces text-rich temporal graph datasets with high link novelty and supports both multi-class and multi-label node classification within a unified framework.

LLMs for Temporal Graph Learning. With the rapid advancement of large language models (LLMs), recent work has begun exploring their use in graph learning. LLM-as-Enhancer methods Roy et al. (2025); Zhang et al. (2025) integrate LLMs to improve temporal graph models, while TGTalker Huang et al. (2025) employs LLMs as predictors via in-context learning. However, most LLM-based graph learning studies focus on static graphs. To extend LLMs to temporal settings, we adapt recent LLM-as-Predictor methods originally designed for static graphs Chen et al. (2024); Tang et al. (2024) to temporal graphs, enabling empirical evaluation of their ability to directly model both structural and semantic dynamics.

3 Task Formulation

To comprehensively evaluate model capabilities in capturing both topological and semantic dynamics, TTGBench standardizes two fundamental tasks. We begin by formally defining the data structure.

Definition 1: Text-Attributed Temporal Graph. A text-attributed temporal graph is defined as $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathcal{T}, \mathcal{X}, \mathcal{Y})$, where $\mathcal{V}$ denotes the set of nodes and $\mathcal{E}$ is a chronologically ordered sequence of timestamped interactions. Each interaction is represented as an event $e = (u, v, t, x_e) \in \mathcal{E}$, indicating that nodes $u, v \in \mathcal{V}$ interact at time $t \in \mathcal{T}$, accompanied by an edge-level textual attribute x_e. In addition, each node u is associated with a time-dependent semantic label $y_u(t) \in \mathcal{Y}$, reflecting its instantaneous preference.

Based on this formulation, TTGBench defines two continuous-time tasks to evaluate a model's ability to capture structural evolution and semantic dynamics.

Temporal Link Prediction (TLP): Modeling Structural Evolution. TLP evaluates a model's ability to predict future interactions based on historical graph observations. Given the graph up to time t, denoted as $\mathcal{G}(\leq t)$, the objective is to predict whether an edge (u, v) will occur at a future time $t' > t$, i.e., whether $(u, v, t') \in \mathcal{E}$. Formally:

$$P(e = (u, v, t') \mid \mathcal{G}(\leq t)) = \sigma\left(f_\theta^{(\text{link})}(u, v, \mathcal{G}_{\leq t})\right), \tag{1}$$

where $\sigma(\cdot)$ is the sigmoid function mapping model outputs to link probabilities.

Temporal Node Classification (TNC): Modeling Semantic Evolution. In contrast to prior benchmarks that restrict node classification to simplistic, quasi-static binary tasks, TTGBench formulates TNC as a dynamic semantic prediction problem. As nodes interact over time, their semantic labels $y_u(t)$ evolve continuously. Given the historical graph observed up to time t, the goal is to predict node u's instantaneous semantic label $y_u(t')$ at a future time $t' > t$.

Table 1: Summary statistics of TTGBench datasets. C denotes node class count.

Dataset	# Nodes	# Edges	# Steps	Novelty	# C	Tasks
FOOD	33,773	401,057	6,252	1.0	9	TLP & multi-label TNC
IMDB	32,371	310,891	7,099	1.0	8	TLP & multi-label TNC
Librarything	51,201	785,690	2,914	1.0	-	TLP
Beeradvocate	99,361	1,586,573	1,577,920	0.99	7	TLP & multi-class TNC
Ratebeer	139,538	2,924,105	4,254	1.0	8	TLP & multi-class TNC
Amazon-Kindle	215,504	5,621,343	5,484,604	1.0	8	TLP & multi-class TNC

Importantly, TTGBench supports both multi-class and multi-label settings, enabling the modeling of complex and realistic semantic dynamics. The task is defined as:

$$P(y_u(t') = c \mid u, \mathcal{G}\leq t) = \text{Softmax}\left(f_\theta^{(\text{node})}(u, \mathcal{G}\leq t)\right)_c \quad \text{(multi-class)}, \tag{2}$$

$$P(\mathbf{Y}_{u,c}(t') = 1 \mid u, \mathcal{G}\leq t) = \sigma\left(f_{\theta,c}^{(\text{node})}(u, \mathcal{G}\leq t)\right) \quad \text{(multi-label)}, \tag{3}$$

where $\mathbf{Y}_u(t') \in \{0,1\}^{|\mathcal{C}|}$ is the ground-truth label vector, $\mathcal{C}$ denotes the label set, and $c \in \mathcal{C}$.

4 The Proposed Datasets: Unveiling Dual Volatility

To rigorously evaluate the capabilities of temporal graph learning models in highly dynamic environments, we construct six real-world, text-attributed temporal graph datasets spanning diverse domains, including culinary recipe feedback Majumder et al. (2019), movie reviews Misra (2019), book reading records Cai et al. (2017); Zhao et al. (2015), beer rating data McAuley et al. (2012); McAuley & Leskovec (2013), and online shopping interactions Hou et al. (2024). Key statistics are summarized in Table 1, with additional details provided in Appendix A.

Our datasets are fundamentally more challenging than existing benchmarks, as they exhibit a property we term *Dual Volatility*: (i) *Structural Volatility*, characterized by continuous and drastic reshaping of network topology, and (ii) *Semantic Volatility*, characterized by pervasive and complex semantic drift. Beyond the intuitive description introduced in Section 1, we provide a fine-grained, and micro-level analysis of these properties in this section.

4.1 Structural Volatility

We analyze structural dynamics from two complementary perspectives: continuous density variation and traffic concentration instability, as illustrated in Figure 2 (a) and (b).

Continuous Density Variation. Rather than remaining stable or growing uniformly, interaction volumes exhibit persistent and significant fluctuations over time. This continuous variation indicates that the temporal graphs undergo repeated phases of expansion and contraction. As a result, models must maintain strong temporal adaptability, avoiding overfitting to dense intervals while remaining robust during sparse periods.

Traffic Concentration Instability. Figure 2 (b) shows that the top 5% most active nodes (hubs) exhibit highly irregular and volatile activity patterns. The graphs continuously shift between centralized regimes—where a small number of nodes dominate interactions—and decentralized regimes with more evenly distributed activity. In other words, the "centers of gravity" of the network are constantly shifting. This instability causes structural patterns learned in one time window to quickly become obsolete, requiring models to adapt to highly non-stationary degree distributions.

4.2 Semantic Volatility

A key distinction between temporal and static graphs lies in the dynamic nature of node semantics: node labels can evolve over time. We next examine the semantic dynamics of our datasets in detail.

High-Frequency Semantic Drifts. Figure 2 (c) shows that frequent semantic drifts are a common phenomenon across datasets, although their intensity varies by domain. In datasets such as

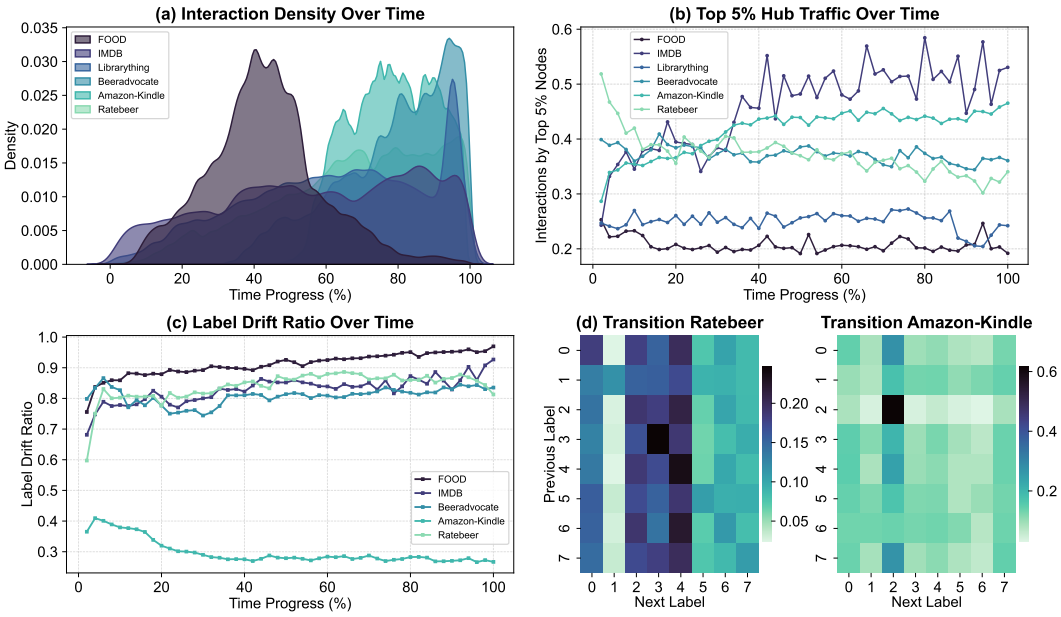

Figure 2: Micro-level visualization of dual volatility across the proposed datasets. (a) Interaction Density Over Time: Shows the temporal evolution of interaction density, where higher values indicate more frequent and concentrated interactions at a given time. (b) Top 5% Hub Traffic Over Time: Illustrates the temporal dynamics of interactions involving the top 5% most active nodes (hubs). (c) Label Drift Ratio Over Time: Depicts the proportion of nodes whose labels change over time. (d) Transition Matrices: Visualizes label transitions on two representative datasets, RateBeer and Amazon-Kindle, where the vertical axis denotes node labels at the previous time step, and the horizontal axis denotes labels at the subsequent time step.

FOOD and IMDB, the drift ratio remains consistently high (often above 0.8), indicating rapid and continuous changes in user preferences. In contrast, Amazon-Kindle exhibits a more moderate yet still substantial drift ratio (around 0.3), reflecting longer periods of semantic consistency. This variation aligns with domain-specific behaviors: while dining and movie consumption are fast-paced, reading typically involves sustained engagement, leading to slower semantic transitions. Importantly, this spectrum—from highly dynamic to moderately stable—demonstrates the diversity of semantic evolution patterns in our datasets, enabling a comprehensive evaluation of models under varying degrees of semantic volatility.

Non-Trivial Evolutionary Trajectories. Figure 2 (d) illustrates label transitions for the same node across consecutive time steps. The resulting heatmaps exhibit asymmetric, high-density clusters rather than uniform or trivially dominant patterns. This indicates that semantic evolution involves abrupt and heterogeneous transitions across distinct classes. Consequently, future states cannot be reliably inferred by simply copying the most recent label or applying a static global transition pattern, highlighting the intrinsic complexity of semantic drift in temporal graphs.

5 Evaluated Baselines

To systematically evaluate model performance on the proposed Dual Volatility datasets across both topological evolution (TLP) and semantic drift (TNC), we benchmark 17 representative algorithms spanning multiple modeling paradigms. Based on the type of predictor, we categorize these methods into two groups: *TGNN-Predictors* and *LLM-Predictors*.

5.1 TGNN-Predictors

This paradigm relies on Temporal Graph Neural Networks (TGNNs) as the predictive backbone, operating directly on graph structures via message passing or temporal aggregation. Most existing

methods specifically designed for temporal graph learning fall into this category, including both purely structural TGNNs and hybrid approaches that incorporate LLMs as auxiliary components.

Pure TGNNs. As the dominant paradigm in temporal graph learning, pure TGNNs serve as strong structural baselines. We evaluate nine state-of-the-art models covering the major mechanisms for modeling topological evolution. These include memory-based continuous-time models (JODIE Kumar et al. (2019), DyRep Trivedi et al. (2019), TGN Rossi et al. (2020)), attention-based temporal aggregation methods (TGAT Xu et al. (2020), DyGFormer Yu et al. (2023), FreeDyG Tian et al. (2024)), and walk-based or hybrid message-fusing approaches (CAWN Wang et al. (2021b), TCL Wang et al. (2021a), GraphMixer Cong et al. (2023)). These models primarily rely on structural patterns and temporal interaction frequencies to make predictions.

LLM-as-Enhancer. This hybrid subcategory leverages frozen LLMs to enrich semantic representations while retaining TGNNs as the final predictors. We evaluate two recent state-of-the-art methods: LKD4DyTAG Roy et al. (2025) and CROSS Zhang et al. (2025). LKD4DyTAG prompts an LLM to extract semantic embeddings from nodes' historical interactions and propagates them along the graph structure, using knowledge distillation from LLMs to guide TGNN optimization. In contrast, CROSS first generates textual summaries of interaction histories via an LLM, and then fuses the resulting embeddings with TGNN representations through multi-layer neural networks. Both approaches aim to enhance semantic awareness without altering the underlying structural modeling mechanism of TGNNs.

5.2 LLM-Predictors

An emerging line of work directly employs LLMs as predictors for graph learning, bypassing traditional message-passing frameworks. These methods typically convert graph structures into sequential text or embedding-based prompts, allowing LLMs to act as the primary reasoning engine. Although most existing methods in this paradigm are designed for static graphs, with the exception of the recent TGTalker Huang et al. (2025), their flexibility enables adaptation to temporal settings. We therefore extend representative approaches to temporal graphs to evaluate their ability to model both structural and semantic evolution.

In-Context Learning (ICL). These training-free methods leverage the pretrained knowledge and reasoning capabilities of LLMs, adapting to graph tasks through prompt-based demonstrations. TGTalker belongs to this category, constructing prompts from temporal interaction triplets (u, v, t) to guide the LLM in generating predictions. Building upon this, we evaluate two Disentangled Spatial-Temporal Thought variants from LLM4DyG Zhang et al. (2024b): DST-v1, which encodes structural information before temporal context, and DST-v2, which reverses this order. These methods serialize temporal graphs as linear sequences of triplets, requiring the LLM to infer structural relationships implicitly.

Supervised Fine-Tuning (SFT). We adapt two recent LLM-as-Predictor methods originally proposed for static graphs, LLaGA Chen et al. (2024) and GraphGPT Tang et al. (2024), to temporal settings. These methods construct prompts that include task instructions and sequences of structure-aware embeddings, and employ a learnable projector to align graph embeddings with the LLM embedding space. For LLaGA, we evaluate both variants: LLaGA-ND, which uses the initial embeddings of the center node and its neighbors, and LLaGA-HO, which uses embeddings obtained after multiple rounds of message passing over historical interaction structure. GraphGPT encodes graph structure through aligned graph and text embeddings, and additionally fine-tunes the LLM embedding layer. Unlike ICL methods, SFT approaches provide explicitly structured and hierarchical representations of graph topology.

Detailed descriptions of all 17 methods, along with prompt templates for LLM-Predictors, are provided in Appendix B.

6 Experiments and Analysis

Implementation Details. For TGNN-based methods, we follow the training protocols and pipelines established in DyGLib Yu et al. (2023), first training all models on the TLP task and then using the resulting checkpoints to initialize training for TNC. For all LLM-based methods, we adopt Qwen3-

Table 2: Temporal Link Prediction Results. Results are averaged over three independent runs (in %). The best and second-best results for each dataset are highlighted in **bold** and underlined, respectively.

Methods	FOOD		IMDB		Librarything		Beeradvocate		Amazon-Kindle		Ratebeer	
	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC	AP	AUROC
EdgeBank	50.00±0.00	49.70±0.00	50.00±0.00	46.29±0.00	50.00±0.00	49.77±0.00	50.51±0.00	50.57±0.00	50.00±0.00	49.98±0.00	50.07±0.00	49.74±0.00
JODIE	72.96±1.88	71.95±0.69	60.30±2.37	65.74±3.20	75.76±0.63	73.62±0.75	79.75±11.63	80.93±9.53	82.81±2.00	81.86±1.69	**92.62**±0.18	**91.56**±0.36
DyRep	72.83±0.88	69.01±1.53	**64.79**±1.30	**69.79**±2.03	73.81±1.55	72.34±1.13	**89.61**±0.20	**87.01**±0.40	78.35±2.63	76.63±2.69	85.87±0.33	83.84±1.18
TGAT	54.79±1.25	54.38±0.67	54.55±0.01	57.85±0.05	64.20±0.89	65.68±0.96	59.37±3.59	66.58±2.13	79.00±1.97	81.12±1.32	58.93±4.20	64.38±2.29
TGN	**77.74**±0.05	**75.74**±0.28	45.40±0.89	48.79±1.66	**82.60**±0.22	**82.02**±0.09	83.46±1.68	83.25±1.06	71.77±2.90	79.92±1.49	91.55±0.24	91.41±0.24
CAWN	51.71±0.38	53.96±0.23	54.00±1.17	55.60±2.75	62.01±0.21	63.21±0.07	66.15±0.85	71.29±0.05	79.34±1.50	80.79±1.22	59.94±2.40	65.78±0.67
TCL	49.12±0.79	50.98±1.23	55.78±0.49	58.83±0.44	54.51±4.02	56.63±2.91	55.40±0.18	64.62±0.29	79.68±1.51	80.96±0.95	57.61±3.77	62.63±2.39
GraphMixer	66.98±1.71	65.78±0.93	57.01±0.03	61.60±0.09	62.13±1.45	63.16±2.44	86.02±0.24	85.57±0.29	81.53±0.28	81.87±0.09	78.53±4.22	78.05±4.06
DyGFormer	59.48±0.24	57.26±0.52	55.32±0.11	58.42±0.10	54.53±2.16	54.35±3.14	50.53±0.08	56.80±0.05	72.78±0.32	74.71±0.09	53.28±0.91	57.65±0.73
FreeDyG	72.40±0.34	69.60±0.37	57.53±0.89	62.16±0.93	66.08±0.26	65.13±1.70	82.93±6.75	85.18±4.07	67.26±3.83	71.25±1.99	69.49±3.30	74.16±1.86
LKD4DyTAG	66.63±0.95	63.62±0.95	56.04±0.92	59.13±0.53	59.94±0.49	62.17±0.34	79.42±1.15	79.52±0.91	80.64±0.46	81.64±0.42	69.51±0.03	70.85±0.05
CROSS	58.15±0.13	56.48±0.24	55.98±0.05	59.14±0.12	59.04±0.83	60.25±0.94	51.36±1.03	57.52±1.01	**87.25**±0.13	**88.31**±0.12	52.57±0.09	57.19±0.21
TGTalker	51.04±0.02	52.02±0.03	50.22±0.04	50.43±0.07	50.63± 0.01	51.24±0.04	51.18±0.03	51.36±0.02	52.54±0.09	54.79±0.11	51.11±0.07	52.15±0.06
DST-v1	51.31±0.01	52.53±0.03	50.26±0.10	50.52±0.20	50.71±0.02	51.39±0.06	51.34±0.04	51.47±0.05	52.66±0.10	54.85±0.09	51.23±0.04	51.26±0.03
DST-v2	51.33±0.00	52.57±0.01	50.35±0.10	50.70±0.19	50.61±0.01	51.19±0.02	51.21±0.01	51.53±0.01	52.62±0.08	54.81±0.08	51.05±0.01	51.07±0.02
Llaga-ND	53.10±0.36	55.52±0.57	52.01±0.75	53.55±1.19	53.98±0.07	53.95±0.14	54.97±0.03	54.94±0.06	68.62±3.37	72.85±3.58	54.00±0.02	54.99±0.05
Llaga-HO	61.95±1.32	65.67±1.47	54.37±0.82	57.34±1.06	56.87±0.02	56.88±0.03	58.00±0.05	58.00±0.10	72.64±1.11	76.63±1.54	56.09±0.07	56.15±0.11
GraphGPT	55.09±0.43	57.95±1.02	51.80±0.18	53.19±0.29	54.96±0.04	54.89±0.11	56.80±6.80	58.35±8.35	70.72±0.72	71.17±1.13	56.03±6.03	59.29±9.29

8B Team (2025) as the backbone model. The hyperparameter configurations and training procedures for SFT-based approaches strictly follow their original implementations Chen et al. (2024); Tang et al. (2024). Sentence-BERT Reimers & Gurevych (2019) is used as the default text encoder to extract embeddings for all textual attributes. For both tasks, each dataset is chronologically split into training, validation, and test sets with a 40%/10%/50% ratio. All experiments are conducted on NVIDIA RTX A6000 GPUs (48GB). Additional implementation details are provided in Appendix C.1.

Evaluation Settings and Metrics. We follow standard evaluation protocols Poursafaei et al. (2022); Yu et al. (2023) by considering both transductive and inductive settings, along with three negative sampling strategies for TLP: random, historical, and inductive. For evaluation metrics, we adopt widely used measures including Average Precision (AP) and Area Under the ROC Curve (AUROC), as well as Mean Reciprocal Rank (MRR), which has gained increasing attention in recent work Huang et al. (2023); Yi et al. (2025). Due to space constraints and the consistent trends observed between MRR and the other metrics across different settings, we report AP and AUROC under the transductive setting with random negative sampling in the main text, while deferring MRR results and full experimental details under all settings to Appendix C.2. For the temporal node classification task, we use Macro-F1 and Balanced Accuracy (bACC) to account for class imbalance, following prior studies Brodersen et al. (2010); Grandini et al. (2020); He & Garcia (2009); Hinojosa Lee et al. (2024); Opitz (2022); Tarekegn et al. (2021).

6.1 Main Results and Analysis

From the results in Tables 2 and 3, we derive several key observations.

TGNNs and LLM-Predictors exhibit task-dependent performance asymmetry. TGNN-based predictors achieve strong performance on TLP but degrade substantially on TNC, whereas LLM-based predictors excel on TNC while struggling on TLP. Since TLP primarily requires modeling temporal structural evolution and TNC emphasizes dynamics, this divergence suggests that each paradigm captures only one facet of temporal information. These results reveal a fundamental limitation of existing approaches in achieving balanced structural and semantic learning.

Naive memorization fails under high novelty. The pure memorization baseline, EdgeBank, performs close to random (AUROC ≈ 50) across all datasets, confirming its ineffectiveness in high-novelty scenarios. This underscores the necessity of models that can learn evolving temporal patterns rather than relying on historical repetition.

Continuous state tracking is critical for TLP under structural volatility. Memory-based TGNNs, including TGN, JODIE, and DyRep, consistently rank among the strongest methods on TLP. By maintaining evolving node states, these models capture cumulative temporal effects, which appears to be a key factor underlying their superior performance when interaction repetition is limited.

Explicit structural inputs help LLM-Predictors on TLP, but the learning paradigm remains the bottleneck. Table 2 reveals three key findings. (i) Linearized temporal-structural sequences in ICL inputs, regardless of ordering, perform close to random, indicating that prompting off-the-shelf

Table 3: Temporal Node Classification Results. Results are averaged over three independent runs (in %). The best and second-best results for each dataset are highlighted in **bold** and underlined, respectively.

Methods	FOOD		IMDB		Beeradvocate		Amazon-Kindle		Ratebeer	
	Macro-F1	mACC	Macro-F1	mACC	Macro-F1	mACC	Macro-F1	mACC	Macro-F1	mACC
JODIE	25.95±4.66	33.48±10.52	12.65±0.90	15.85±0.57	7.00±0.06	14.52±0.15	17.60±0.01	13.64±0.04	7.86±0.30	13.88±0.32
DyRep	21.22±0.10	22.93±0.10	13.74±0.82	15.97±0.68	6.50±0.34	14.28±0.01	13.36±1.43	13.51±0.89	7.38±1.33	13.58±0.98
TGAT	22.73±2.73	27.78±5.56	8.53±0.02	12.48±0.00	6.18±0.04	14.30±0.01	14.91±0.37	14.38±0.25	8.67±0.05	14.53±0.06
TGN	21.98±1.20	23.61±0.96	8.51±0.00	12.50±0.00	8.35±0.01	15.31±0.02	13.22±0.25	13.28±0.19	7.34±1.15	13.54±0.78
CAWN	20.28±0.29	22.38±0.15	11.48±2.98	18.75±6.25	8.77±0.27	15.42±0.02	14.96±0.33	14.43±0.25	8.79±0.04	14.53±0.08
TCL	19.99±0.02	22.22±0.01	14.56±1.19	15.46±0.04	9.26±0.28	16.03±0.33	24.25±0.68	21.76±0.75	14.34±0.67	19.35±0.96
GraphMixer	16.94±0.01	20.77±0.02	9.20±0.69	12.91±0.41	6.14±0.01	14.29±0.01	13.35±0.40	13.38±0.22	7.13±1.78	13.59±1.09
DyGFormer	36.58±3.02	36.39±3.92	20.05±11.54	20.01±7.51	12.91±1.59	18.93±0.48	20.19±0.60	18.14±0.43	15.52±1.37	18.42±1.12
FreeDyG	20.46±0.04	22.47±0.02	10.20±1.55	13.46±0.89	7.75±1.61	14.91±0.62	12.80±0.86	13.10±0.47	8.54±0.46	14.23±0.35
LKD4DyTAG	23.37±3.38	27.78±5.56	8.51±0.01	12.50±0.02	6.14±0.01	14.29±0.02	10.39± 0.01	11.11±0.04	5.80±0.07	12.57±0.01
CROSS	19.99±0.02	22.22±0.01	8.62±0.02	12.41±0.01	6.17±0.01	14.30±0.01	9.59±0.07	10.97±0.03	5.35±0.00	12.50±0.00
TGTalker	31.84±0.05	30.13±0.52	28.96±0.59	30.63±0.46	**16.52**±0.31	18.00±0.30	25.60±0.24	24.02±0.26	13.42±0.47	15.29±0.78
DST-v1	32.87±0.22	31.62±0.26	**29.37**±0.33	30.94±0.69	16.29±0.42	17.47±0.38	**26.00**±0.33	24.42±0.14	12.83±0.92	14.89±0.65
DST-v2	**37.59**±0.25	**39.91**±0.22	29.14±0.24	30.72±0.36	16.12±0.25	17.17±0.34	24.64±0.37	23.94±0.25	13.34±0.94	14.69±0.50
Llaga-ND	28.48±0.30	32.04±0.66	19.13±1.13	25.95±1.16	13.90±1.88	18.30±1.60	18.57±2.53	17.25±1.13	14.12±0.08	17.51±1.19
Llaga-HO	30.37±0.41	33.26±1.75	20.66±1.63	26.45±1.67	15.09±1.70	19.24±1.35	21.01±2.68	18.66±2.55	16.94±0.17	**20.63**±1.42
GraphGPT	36.74±2.83	38.40±1.47	25.79±5.59	**31.01**±0.08	16.31±0.28	**20.22**±0.37	24.28±4.63	**24.74**±0.16	**18.93**±0.16	20.21±0.20

LLMs is insufficient for extracting implicit structural signals from simple (u, v, t) serializations. (ii) SFT consistently outperforms ICL, suggesting that incorporating explicit structural encoding in the input benefits TLP performance. (iii) Within SFT-based methods, LLaGA-HO and GraphGPT outperform LLaGA-ND, demonstrating that structure-aware aggregation of node embeddings partially compensates for LLMs' limited structural understanding. Nevertheless, even the best-performing LLM-Predictors on TLP remain significantly behind TGNNs, indicating that input-level structural enhancements alone are insufficient and that the limitation lies in the underlying learning paradigm.

LLM-as-Enhancer struggles to address TGNN semantic limitations. LLM-enhanced TGNN methods such as CROSS and LKD4DyTAG achieve competitive performance on TLP but inherit the weaknesses of TGNNs on TNC. This suggests that simply injecting LLM-derived semantic information into an aggregation framework is insufficient to overcome the structural bias of TGNNs, limiting their ability to model rapidly evolving semantics.

6.2 Why Do TGNNs Fail at Semantic Evolution?

To systematically analyze why TGNNs fail catastrophically on TNC, we conduct in-depth probing studies to isolate their underlying bottlenecks. We select JODIE as a representative TGNN due to its consistently strong performance, and choose DST-v2 and LLaGA-HO—two top-performing LLM-Predictors on TNC—as representatives of ICL- and SFT-based methods, respectively. The comparative analysis (Figure 3) reveals several fundamental limitations of TGNNs.

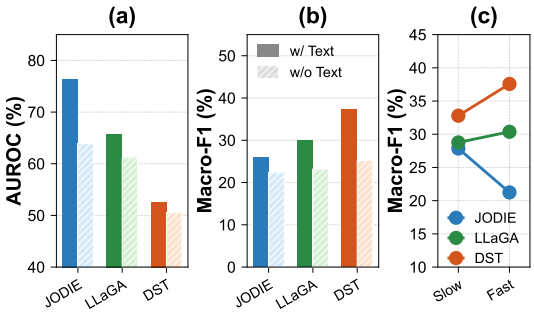

Figure 3: Comparative analysis of representative TGNN and LLM-Predictor methods. (a) and (b) show the performance of each method on TLP and TNC, respectively, with and without textual inputs. (c) illustrates how model performance varies as the frequency of semantic changes increases.

Architectural semantic blindness. We first examine whether models effectively utilize textual attributes for semantic evolution. As shown in Figure 3(a) and (b), removing textual attributes causes a sharp drop in TLP performance for TGNNs but a much smaller impact on TNC, whereas LLM-Predictors suffer substantial degradation on TNC with minor impact on TLP. This consistent trend holds across all methods in both paradigms (see Figure 7 for full results). The observed asymmetry suggests that, although TGNNs can exploit text as auxiliary signals for structural modeling, their message-aggregation architectures are inherently limited in leveraging textual information for semantic evolution.

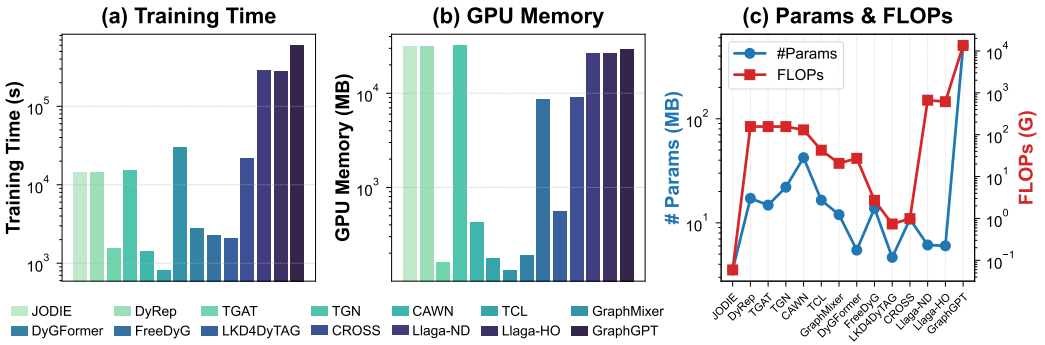

Figure 4: Efficiency Results. Scalability results of all methods are shown in Figure 8.

Representation lock-in. We next investigate whether this limitation can be mitigated through downstream supervision. Specifically, we compare two training strategies for TGNNs on TNC: linear probing (training only the classifier head on a TLP-pretrained backbone) and end-to-end fine-tuning (jointly optimizing the backbone and classifier). Surprisingly, end-to-end fine-tuning yields no meaningful improvement over linear probing and can even degrade performance (Table 22). This suggests a strong *representation lock-in*, where gradients from semantic supervision are insufficient to reshape representations that are already heavily biased toward structural patterns. As a result, the learned embeddings remain anchored in topological dynamics and are not adaptable to semantic dynamics.

Structural inertia under semantic volatility. Finally, we analyze model performance under varying degrees of semantic change based on the frequency of label drifts. As shown in Figure 3(c), TGNN performance deteriorates sharply as the environment transitions from slow (1–2 changes) to fast (¿2 changes) semantic dynamics. In contrast, LLM-Predictors maintain stable or even improved performance. This highlights an inherent *structural inertia* in TGNNs: their reliance on aggregating historical interactions becomes a liability when semantic states change rapidly. Conversely, LLM-Predictors exhibit strong semantic adaptability, dynamically leveraging immediate textual context to track evolving semantics without being constrained by historical structural dependencies.

6.3 Efficiency Analysis

We evaluate the practical applicability of all benchmarked methods by analyzing their computational efficiency. Figure 4 reports training time, GPU memory consumption, parameter counts, and FLOPs for all methods on the largest-scale Amazon-Kindle dataset.

Staggering overhead of LLM-Predictor paradigms. The most striking observation is the substantial computational cost incurred by LLM-Predictor paradigms due to the use of generative LLMs. In particular, GraphGPT—whose training involves fine-tuning the embedding layer of an LLM—requires nearly 600,000 seconds of training time and 13.5 teraFLOPs, exceeding even the most resource-intensive TGNNs by several orders of magnitude. Although LLM-Predictors achieve strong performance on TNC, their prohibitive computational overhead limits their practicality in real-world applications, highlighting the need for fundamental improvements to make this paradigm deployable in realistic settings.

Performance–efficiency trade-off within TGNNs. Clear trade-offs are observed among TGNNs. Memory-based architectures (e.g., JODIE, DyRep, and TGN) achieve dominant performance on TLP, but incur substantially higher computational and memory overhead due to the need to maintain and continuously update recurrent node states. This reflects the inherent cost of continuous-time structural tracking required to handle structural volatility effectively.

Scalability robustness. Beyond static efficiency, we further evaluate scalability by measuring runtime on progressively larger subsets of the Amazon-Kindle dataset, ranging from 1M to 5M interactions. Results show that all methods exhibit approximately linear scaling with respect to dataset size, indicating predictable and robust scalability. Detailed results are provided in Figure 8.

AI USE STATEMENT

In this work, we used generative AI tools for clean and reformat dataset, and assist with translation. We have not used generative AI tools for generate synthetic data sets, support qualitative and thematic data analysis, interpret results, and help develop theoretical models or conceptual frameworks, formulate mathematical claims, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments and implement methods are not applicable to this work. Additionally, we used generative AI tools for brainstorming, edit a research paper to improve readability. We have reviewed all AI-assisted work. For example, we checked LLM-polished sentences to ensure that they accurately conveyed our intended meaning.. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

All datasets used in this work are derived from publicly available sources that were released for research purposes. We strictly follow the respective data usage licenses and ensure that no personally identifiable information is included in the final released datasets. All user-generated contents are anonymized, and no attempts were made to re-identify any individual. No new data were collected from human participants for this work, and no Institutional Review Board approval was required under our institutional policies.

REPRODUCIBILITY STATEMENT

Section 6 and Appendix C disclose all the information needed to reproduce the main experimental results of the paper. `https://ttgbench.netlify.app` provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results. Section 6 and Appendix C specify all the training and test details necessary to understand the results.

AUTHOR CONTRIBUTIONS

If you'd like to, you may include a section for author contributions as is done in many journals. This is optional and at the discretion of the authors.

REFERENCES

Kay Henning Brodersen, Cheng Soon Ong, Klaas Enno Stephan, and Joachim M Buhmann. The balanced accuracy and its posterior distribution. In *2010 20th international conference on pattern recognition*, pp. 3121–3124. IEEE, 2010.

Chenwei Cai, Ruining He, and Julian McAuley. Spmc: Socially-aware personalized markov chains for sparse sequential recommendation. *arXiv preprint arXiv:1708.04497*, 2017.

Runjin Chen, Tong Zhao, Ajay Kumar Jaiswal, Neil Shah, and Zhangyang Wang. Llaga: Large language and graph assistant. In *International Conference on Machine Learning*, pp. 7809–7823. PMLR, 2024.

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? *arXiv preprint arXiv:2302.11636*, 2023.

Manuel Dileo, Matteo Zignani, and Sabrina Gaito. Temporal graph learning for dynamic link prediction with text in online social networks. *Machine learning*, 113(4):2207–2226, 2024.

Linlin Ding, Baishuo Han, Shu Wang, Xiaoguang Li, and Baoyan Song. User-centered recommendation using us-elm based on dynamic graph model in e-commerce. *International Journal of Machine Learning and Cybernetics*, 10(4):693–703, 2019.

Margherita Grandini, Enrico Bagli, and Giorgio Visani. Metrics for multi-class classification: an overview. *arXiv preprint arXiv:2008.05756*, 2020.

Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on knowledge and data engineering*, 21(9):1263–1284, 2009.

Maria Cristina Hinojosa Lee, Johan Braet, and Johan Springael. Performance metrics for multilabel emotion classification: comparing micro, macro, and weighted f1-scores. *Applied Sciences*, 14 (21):9863, 2024.

Yupeng Hou, Jiacheng Li, Zhankui He, An Yan, Xiusi Chen, and Julian McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

Shenyang Huang, Farimah Poursafaei, Jacob Danovitch, Matthias Fey, Weihua Hu, Emanuele Rossi, Jure Leskovec, Michael Bronstein, Guillaume Rabusseau, and Reihaneh Rabbany. Temporal graph benchmark for machine learning on temporal graphs. *Advances in Neural Information Processing Systems*, 36:2056–2073, 2023.

Shenyang Huang, Emma Kondrup, Ali Parviz, Zachary Yang, Zifeng Ding, Michael M. Bronstein, Reihaneh Rabbany, and Guillaume Rabusseau. Are large language models good temporal graph learners? In *New Perspectives in Graph Machine Learning*, 2025. URL https://openreview.net/forum?id=4ZithWQFQN.

Xuanwen Huang, Yang Yang, Yang Wang, Chunping Wang, Zhisheng Zhang, Jiarong Xu, Lei Chen, and Michalis Vazirgiannis. Dgraph: A large-scale financial dataset for graph anomaly detection. *Advances in Neural Information Processing Systems*, 35:22765–22777, 2022.

Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. Representation learning for dynamic graphs: A survey. *Journal of Machine Learning Research*, 21(70):1–73, 2020a. URL http://jmlr.org/papers/v21/19-447.html.

Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. Representation learning for dynamic graphs: A survey. *Journal of Machine Learning Research*, 21(70):1–73, 2020b.

Srijan Kumar, Xikun Zhang, and Jure Leskovec. Predicting dynamic embedding trajectory in temporal interaction networks. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1269–1278, 2019.

Antonio Longa, Veronica Lachi, Gabriele Santin, Monica Bianchini, Bruno Lepri, Pietro Lio, franco scarselli, and Andrea Passerini. Graph neural networks for temporal graphs: State of the art, open challenges, and opportunities. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL https://openreview.net/forum?id=pHCdMat0gI.

Bodhisattwa Prasad Majumder, Shuyang Li, Jianmo Ni, and Julian McAuley. Generating personalized recipes from historical user preferences. *arXiv preprint arXiv:1909.00105*, 2019.

Julian McAuley, Jure Leskovec, and Dan Jurafsky. Learning attitudes and attributes from multi-aspect reviews. In *2012 IEEE 12th International Conference on Data Mining*, pp. 1020–1025. IEEE, 2012.

Julian John McAuley and Jure Leskovec. From amateurs to connoisseurs: modeling the evolution of user expertise through online reviews. In *Proceedings of the 22nd international conference on World Wide Web*, pp. 897–908, 2013.

Rishabh Misra. Imdb spoiler dataset, 05 2019.

Anirban Mitra and Subrata Paul. Analyzing social networks with dynamic graphs: Unravelling the ever-evolving connections. In *Applied Graph Data Science*, pp. 195–214. Elsevier, 2025.

Juri Opitz. From bias and prevalence to macro f1, kappa, and mcc: A structured overview of metrics for multi-class evaluation. *Heidelberg University*, 2022.

Farimah Poursafaei, Shenyang Huang, Kellin Pelrine, and Reihaneh Rabbany. Towards better evaluation for dynamic link prediction. *Advances in Neural Information Processing Systems*, 35: 32928–32941, 2022.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. *arXiv preprint arXiv:2006.10637*, 2020.

Amit Roy, Ning Yan, and Masood Mortazavi. Llm-driven knowledge distillation for dynamic text-attributed graphs. *arXiv preprint arXiv:2502.10914*, 2025.

Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial. Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey. *iEEE Access*, 9:79143–79168, 2021.

Jiabin Tang, Yuhao Yang, Wei Wei, Lei Shi, Lixin Su, Suqi Cheng, Dawei Yin, and Chao Huang. Graphgpt: Graph instruction tuning for large language models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 491–500, 2024.

Adane Nega Tarekegn, Mario Giacobini, and Krzysztof Michalak. A review of methods for imbalanced multi-label classification. *Pattern Recognition*, 118:107965, 2021.

Qwen Team. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Yuxing Tian, Yiyan Qi, and Fan Guo. Freedyg: Frequency enhanced continuous-time dynamic graph model for link prediction. In *The twelfth international conference on learning representations*, 2024.

Toan Khang Trinh and Zhuxuanzi Wang. Dynamic graph neural networks for multi-level financial fraud detection: A temporal-structural approach. *Annals of Applied Sciences*, 5(1), 2024.

Rakshit Trivedi, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. Dyrep: Learning representations over dynamic graphs. In *International conference on learning representations*, 2019.

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. *arXiv preprint arXiv:2105.07944*, 2021a.

Yanbang Wang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. Inductive representation learning in temporal networks via causal anonymous walks. *arXiv preprint arXiv:2101.05974*, 2021b.

Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. Inductive representation learning on temporal graphs. *arXiv preprint arXiv:2002.07962*, 2020.

Lu Yi, Jie Peng, Yanping Zheng, Fengran Mo, Zhewei Wei, Yuhang Ye, Yue Zixuan, and Zengfeng Huang. TGB-seq benchmark: Challenging temporal GNNs with complex sequential dynamics. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=8e2LirwiJT.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. *Advances in Neural Information Processing Systems*, 36:67686–67700, 2023.

Jiasheng Zhang, Jialin Chen, Menglin Yang, Aosong Feng, Shuang Liang, Jie Shao, and Rex Ying. Dtgb: A comprehensive benchmark for dynamic text-attributed graphs. *arXiv preprint arXiv:2406.12072*, 2024a.

Siwei Zhang, Yun Xiong, Yateng Tang, Jiarong Xu, Xi Chen, Zehao Gu, Xuehao Zheng, Zi'an Jia, and Jiawei Zhang. Unifying text semantics and graph structures for temporal text-attributed graphs with large language models. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=9env0BdcDV.

Zeyang Zhang, Xin Wang, Ziwei Zhang, Haoyang Li, Yijian Qin, and Wenwu Zhu. Llm4dyg: can large language models solve spatial-temporal problems on dynamic graphs? In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4350–4361, 2024b.

Ziyi Zhang, Diya Li, Zhenlei Song, Nick Duffield, and Zhe Zhang. Location-aware social network recommendation via temporal graph networks. In *Proceedings of the 7th ACM SIGSPATIAL Workshop on Location-based Recommendations, Geosocial Networks and Geoadvertising*, pp. 58–61, 2023.

Tong Zhao, Julian McAuley, and Irwin King. Improving latent factor models via personalized feature projection for one class recommendation. In *Proceedings of the 24th ACM international on conference on information and knowledge management*, pp. 821–830, 2015.

Ziwei Zhao, Fake Lin, Xi Zhu, Zhi Zheng, Tong Xu, Shitian Shen, Xueying Li, Zikai Yin, and Enhong Chen. Dynllm: when large language models meet dynamic graph recommendation. *arXiv preprint arXiv:2405.07580*, 2024.

Yuyu Zhou, Me Sun, and Fan Zhang. Graph neural network-based anomaly detection in financial transaction networks. *Journal of Computing Innovations and Applications*, 1(2):87–101, 2023.

APPENDIX

A DATASETS DETAILS

A.1 DATASET DESCRIPTION

Table 4: Comparison between our datasets and existing temporal graph datasets.

	Dataset	Nodes	Total Edges	Unique Edges	Unique Steps	Text Attr	Novelty	Surprise	Node Labels
Previous	mooc	7,144	411,749	178,443	345,600	✗	0.433	0.785	N.A.
	lastfm	1,980	1,293,103	154,993	1,283,614	✗	0.12	0.369	N.A.
	enron	184	125,235	3,125	22,632	✗	0.076	0.402	N.A.
	SocialEvo	74	2,099,519	4,486	565,932	✗	0.002	0.027	N.A.
	uci	1,899	59,835	20,296	58,911	✗	0.339	0.796	N.A.
	Flights	13,169	1,927,145	395,072	122	✗	0.194	0.362	N.A.
	CanParl	734	74,478	51,331	14	✗	0.673	0.654	N.A.
	USLegis	225	60,396	26,423	12	✗	0.437	0.340	N.A.
	UNtrade	255	507,497	36,182	32	✗	0.09	0.051	N.A.
	UNvote	201	1,035,742	31,516	72	✗	0.056	0.017	N.A.
	Contacts	692	2,426,279	79,530	8,064	✗	0.023	0.291	N.A.
	tgbl-wiki	9,227	157,474	18,257	152,757	✗	0.116	0.108	N.A.
	tgbn-reddit	11,766	27,174,118	516,669	21,889,537	✗	0.02	0.013	N.A.
	GDELT	6,786	1,339,245	249,241	2,403	✓	0.22	0.562	N.A.
	ICEWS1819	31,796	1,100,071	314,011	730	✓	0.292	0.695	N.A.
Ours	FOOD	33,773	401,057	401,057	6,252	✓	1.0	1.0	9
	IMDB	32,371	310,891	310,891	7,099	✓	1.0	1.0	8
	Librarything	51,201	785,690	785,690	2,914	✓	1.0	1.0	N.A.
	Beeradvocate	99,361	1,586,573	1,571,767	1,577,920	✓	0.991	0.997	7
	Ratebeer	139,538	2,924,105	2,855,175	4,254	✓	1.0	1.0	8
	Amazon-Kindle	215,504	5,621,343	5,561,738	5,484,604	✓	1.0	1.0	8

The detailed descriptions of the six proposed datasets are provided below, with comparisons to existing datasets summarized in Table 4.

- **FOOD** Majumder et al. (2019) This dataset contains recipes and user reviews collected from Food.com (formerly GeniusKitchen), covering 18 years of temporal interactions. It provides rich signals for studying culinary trends, evolving user preferences, and long-term temporal dynamics. In the constructed graph, users and recipes are represented as nodes, and a timestamped edge is created when a user reviews a recipe at time t. Each recipe contains detailed cooking instructions, ingredient information, and category labels. We assign recipe categories as users' temporal interest labels, representing users' evolving interests at each interaction. Since each recipe may have multiple labels, FOOD supports *multi-label* temporal node classification.
- **IMDB** Misra (2019) This dataset is constructed from IMDB and consists of users, movies, and timestamped review interactions. Each movie includes plot summaries and genre tags, while each review contains textual content and temporal information. We use movie genre tags as users' temporal interest labels at the time of interaction, making IMDB another *multi-label* classification benchmark. The rich textual information in both movies and reviews provides abundant signals for modeling both structural evolution and semantic drift in temporal graphs.
- **LibraryThing** Cai et al. (2017); Zhao et al. (2015) Collected from the book review platform LibraryThing (`https://www.librarything.com/`), this dataset contains users, books, and timestamped textual review interactions. It captures diverse literary preferences and their evolution over time. In our temporal graph, users and books are nodes connected through review-based temporal edges. Since this dataset does not provide item category labels, it is used only for temporal link prediction.
- **BeerAdvocate** McAuley et al. (2012) This dataset consists of beer reviews involving users and beers as graph nodes. Each beer is associated with a unique style label, which we use as the user's temporal interest label during interaction. Because each beer has a single category label, BeerAdvocate supports *multi-class* temporal node classification.
- **Amazon-Kindle** Hou et al. (2024) This dataset is drawn from the Amazon Reviews 2023 collection (`https://amazon-reviews-2023.github.io/`) and focuses on user reviews of Kindle products. Users and products form the graph nodes, while products and reviews both provide rich textual information. The resulting product-review network spans from 1996 to September 2023.

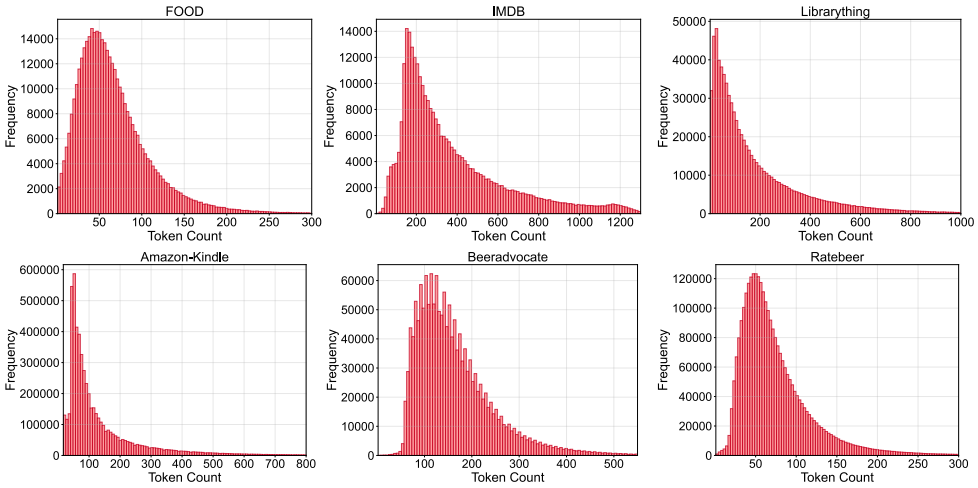

Figure 5: Distribution of edge text length (#tokens) on TTGBench datasets.

Each product has a specific category label, making this dataset suitable for *multi-class* temporal node classification.

- **Ratebeer** McAuley & Leskovec (2013) Ratebeer is another large-scale beer review dataset similar to BeerAdvocate, but with much denser temporal interactions, providing a complementary dynamic scenario for temporal graph modeling. Each beer also has a single category label, enabling *multi-class* temporal node classification.

A.2 Dataset Analysis

Distribution of Edge Text Lengths. Given that the temporal structure formed by sequential edges is crucial for learning in temporal graph models, we present the distribution of text lengths on edges (measured in tokens) across all datasets in Figure 5. As shown in the figure, our datasets exhibit a right-skewed distribution, which is a typical characteristic of user-generated content (UGC) in real-world scenarios. Most user interactions consist of brief feedback (e.g., a "5-star" rating or a short comment like "Good"), while only a small fraction of highly engaged users contribute long-tail texts. Additionally, the text length distributions vary across different datasets. These characteristics pose several challenges for temporal graph learning:

- The sparsity of information in short texts requires models to capture information over extended temporal windows to effectively associate with structural evolution.
- Long texts often contain multiple semantic cues (e.g., multi-faceted product descriptions), demanding that models identify the structure-relevant parts. Moreover, time-sensitive information embedded in long texts adds complexity for precise temporal alignment.
- The differing centers of text length distributions across datasets reflect domain-specific language habits, requiring models to adapt to domain variations. Notably, the existence of long-tail distributions also demands robustness from the models.

Node Label Distribution. As shown in Table 5, label distributions vary across datasets, with several exhibiting substantial class imbalance. As discussed in Section 6, we therefore adopt multiple imbalance-aware evaluation metrics for temporal node classification.

B Methods Details

B.1 TGNNs

The traditional temporal graph neural network methods are described as follows:

Table 5: Text labels of each dataset.

Datasets	Text Labels	Percentages
FOOD	Side Dishes & Vegetables	10.3 %
	Dietary & Specialty	32.4 %
	Soups & Stews	1.8 %
	Cuisine & Regional	14.4 %
	Desserts & Sweets	8.0 %
	Baking & Breads	3.6 %
	Beverages & Snacks	5.3 %
	Sauces & Condiments	2.4 %
	Main Dishes	21.8 %
IMDB	Horror	3.7 %
	Family & Animation	4.9 %
	Sci-Fi & Fantasy	13.0 %
	Drama & Realism	25.0 %
	Action & Adventure	21.5 %
	Thriller & Suspense	16.9 %
	Historical & War	2.7 %
	Comedy & Lighthearted	12.4 %
Beeradvocate	Dark Beers	21.9 %
	Strong Ales & Barley Wines	9.2 %
	Belgian & Specialty Ales	16.5 %
	Sour & Fruit Beers	5.4 %
	Wheat Beers	6.5 %
	Pale Ales & IPAs	26.3 %
	Lagers	14.3 %
Amazon-Kindle	Science, Technology & Professional	0.2 %
	Children & Young Adult	6.4 %
	Social Sciences & Humanities	4.7 %
	Lifestyle & Hobbies	0.5 %
	Entertainment & Arts	0.8 %
	Nonfiction & Self-Improvement	1.3 %
	Short Reads	0.04 %
	Fiction & Literature	86.2 %
Ratebeer	Strong Ales & Barley Wines	8.9 %
	Wheat Beers	6.7 %
	Dark Beers	19.7 %
	Lagers	16.1 %
	Belgian & Specialty Ales	15.8 %
	Pale Ales & IPAs	26.2 %
	Sour & Fruit Beers	5.4 %
	Ciders, Meads & Sakes	1.3 %

- **JODIE** Kumar et al. (2019) models the future evolution of entity embeddings by forecasting their trajectories over time to predict future interactions and entity states. It employs two interconnected recurrent neural networks to update the dynamic states of entities, along with a projection operation that estimates each entity's future embedding trajectory.
- **DyRep** Trivedi et al. (2019) introduces a recurrent framework to continuously update node states after each interaction. Additionally, it features a temporal-attentive aggregation component designed to capture the evolving structural patterns in temporal graphs.
- **TGAT** Xu et al. (2020) generates node representations by aggregating information from each node's temporally relevant neighbors, using a self-attention mechanism. It also integrates a time encoding function to model temporal dependencies within the graph.
- **TGN** Rossi et al. (2020) maintains a dynamic memory for every node, which is updated whenever the node participates in an interaction. This update process involves a message function, a message aggregator, and a memory updater. A dedicated embedding module is then used to produce time-aware node representations based on these evolving memories.
- **CAWN** Wang et al. (2021b) first extracts several causal anonymous walks for each node to uncover the causal dynamics of the network and produce relative node identities. These walks are then encoded using recurrent neural networks, and their outputs are aggregated to compute the final node representations.

- **EdgeBank** Poursafaei et al. (2022) is a memorization-based method without trainable parameters for transductive temporal link prediction. It records observed interactions in memory and predicts a positive link if the interaction exists in memory, and negative otherwise.
- **TCL** Wang et al. (2021a) constructs each node's interaction sequence by applying a breadth-first search on its temporally dependent sub-graph. It then employs a graph transformer that integrates both topological and temporal cues for learning node representations, further incorporating a cross-attention mechanism to capture interdependencies between interacting nodes.
- **GraphMixer** Cong et al. (2023) demonstrates that a fixed time encoding function outperforms a trainable one. This fixed function is embedded within a link encoder built on the MLP-Mixer architecture to model temporal interactions. Additionally, a node encoder using neighbor mean-pooling summarizes node features.
- **DyGFormer** Yu et al. (2023) generates node representations by leveraging each node's historical first-hop interactions. It introduces a neighbor co-occurrence encoding mechanism to capture relationships between nodes based on their interaction histories. Additionally, a patching strategy is applied to segment each interaction sequence into smaller patches, enabling the model to effectively utilize longer historical sequences.
- **FreeDyG** Tian et al. (2024) enhances task performance by adaptively weighting frequency components in the temporal structure's frequency domain. It extracts both high- and low-frequency components of dynamic patterns, then amplifies task-related frequencies and attenuates unrelated signals.

B.2 LLM-AS-ENHANCER

This line of work leverages frozen LLMs to enrich semantic representations for TGNNs, while TGNNs remain the final predictive models.

- **LKD4DyTAG** Roy et al. (2025) feeds textual descriptions of temporal graphs into a frozen teacher LLM to extract embeddings, which are then used to guide a student GNN through knowledge distillation, enhancing the GNN with semantic knowledge captured by the LLM.
- **CROSS** Zhang et al. (2025) first generates textual summaries of interaction histories using an LLM, and then employs a semantic-structure co-encoder to help GNNs incorporate the temporal semantic information summarized by the LLM.

B.3 LLM-AS-PREDICTOR

LLM-as-Predictor approaches directly employ LLMs to solve downstream tasks on temporal graphs. Depending on how LLMs are adapted, this paradigm mainly includes two settings: Supervised Fine-Tuning (SFT) and In-Context Learning (ICL). In both cases, temporal graphs must first be transformed into LLM-understandable sequences, after which temporal structural patterns are learned through either SFT or ICL.

Existing methods under this paradigm have primarily focused on static graphs and remain under-explored for temporal graphs. Given their strong potential, and to encourage further development of LLM-as-Predictor methods for temporal graph learning, we provide detailed descriptions of the six baselines evaluated in the main paper.

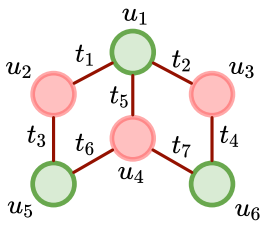

Figure 6: A temporal graph demo.

Structural Sequence. Since LLMs can only process token sequences, the structural information in temporal graphs must first be serialized. The ICL-based methods directly represent the temporal structure as a sequence of (u, v, t) triplets sorted in chronological order, while the three SFT-based methods describe the temporal structure of the target node using a multi-hop neighbor sequence obtained through breadth-

Table 6: Structural Sequences of Node u_4 at Time t_7 by ICL and SFT Methods (Based on Figure 6)

Methods	Sequences of temporal structure
ICL	$(u_1, u_2, t_1), (u_1, u_3, t_2), (u_2, u_5, t_3),$ $(u_3, u_6, t_4), (u_1, u_4, t_5), (u_4, u_5, t_6)$
SFT	(u_1, u_5, u_2, u_3)

Table 7: Differences between three spatial-temporal prompting strategies.

Method	Prompt
TGTalker	Analyze the historical interactions to identify patterns that might indicate a future interaction. Consider both the relationships between users and items and the timing of past interactions within the historical interactions to make your prediction.
DST-v1	Think about structure and then time: First analyze the structural information within the historical interactions to understand the relationships between users and items, then consider the temporal patterns to predict the future connection.
DST-v2	Think about time and then structure: First analyze the temporal sequence of interactions in the historical data to identify patterns over time, then examine the structural information within the historical interactions to predict the future connection.

first search (BFS). Taking the prediction of node u_4's property at time t_7 in Figure 6 as an example, the structural sequences generated by ICL and SFT approaches are summarized in Table 6.

Structural Information. As shown by the serialization strategies in Table 6, ICL-based methods linearize the temporal structure into a sequence of (u, v, t) triplets, encoding a linear structural representation. This representation is simple and direct, but it requires LLMs to infer temporal structural patterns from the sequential triplets to perform the two temporal graph tasks. In contrast, SFT-based methods represent the temporal structure of the target node through its neighbor sequences, obtained via BFS, which encode explicit hierarchical structural information. This hierarchical information facilitates better structural comprehension by the LLM.

Learning and Trainable Parameters. SFT-based and ICL-based methods employ distinct strategies for LLM training or tuning.

- *SFT-based methods fine-tune LLMs to learn structural knowledge for better downstream task performance.* To improve computational efficiency, SFT methods train a projector layer to align temporal structural knowledge with the token embedding space. Specifically, LlaGA-ND and GraphGPT convert the neighbor node sequences into initial text embeddings. LlaGA-HO converts the structural knowledge obtained by aggregating multi-hop neighbor information through message passing from the target node's embedding. In addition, GraphGPT further fine-tunes the embedding layer of the LLM itself.

- *ICL-based methods do not require training or fine-tuning the LLM.* Instead, they leverage the LLM's pre-trained knowledge and reasoning capabilities, guiding it to understand temporal structural patterns via proposed instructions. To thoroughly explore the LLM's ability to learn temporal graph structural knowledge through ICL, we adopt the Disentangled Spatial-Temporal Thoughts prompting technique from LLM4DyG Zhang et al. (2024b) to TGTalker Huang et al. (2025), which encourages the LLM to separately process spatial and temporal information. Specifically, the three variants are: a) TGTalker that jointly processes spatial and temporal information; b) DST-v1 that processes structural information first, followed by temporal information; and c) DST-v2, which processes temporal information first, then structural information. The differences among these three prompting strategies are summarized in Table 7.

Adaptation of SFT-based Methods. It is important to note that the three SFT methods were originally designed for static graphs. We have adapted them to the temporal graph setting, but due to the increased complexity of temporal graphs, these adaptations cannot fully capture all temporal graph properties. Our adaptations are as follows:

- For all methods, the neighbor sequences of the target node are time-sensitive. For example, in Figure 6, when predicting the property of node u_4 at time t_7, its first-order neighbors only include u_1 and u_5, but not u_6.

Table 8: Prompts of LLM-as-Predictor methods for predicting whether node u_4 will form a link with node u_6 at time t_7, based on the example in Figure 6

Method	Prompt
LlaGA-ND	Given two node-centered subgraphs: $\langle u_1, u_5, u_2, u_3 \rangle$ and $\langle u_3, u_1 \rangle$, we need to predict whether these two nodes connect with each other. Please tell me whether two center nodes in the subgraphs should connect to each other.
LlaGA-HO	Given two node-centered subgraphs: $\langle hop_0, hop_1, hop_2 \rangle$ and $\langle hop_0, hop_1, hop_2 \rangle$, we need to predict whether these two nodes connect with each other. Please tell me whether two center nodes in the subgraphs should connect to each other.
GraphGPT	Given a sequence of graph tokens: $\langle u_1, u_5, u_2, u_3 \rangle$ that constitute a user-item review subgraph, where the first token represents the central node (the user), and the remaining nodes represent the central node's first- and second-order neighbors. The first-order neighbors are the items that the central node has reviewed and the second-order neighbors are other users who have reviewed the same items. The other sequence of graph tokens: $\langle u_3, u_1 \rangle$, where the first token corresponds to the center node (the item), and the remaining tokens represent the item's first- and second-order neighbors. The first-order neighbors are the users who have reviewed the item and the second-order neighbors are the items that those users also have reviewed. If the connections between nodes represent the review relationships between users and items, are these two central nodes connected? Give me a direct answer of 'yes' or 'no'.

- Both LlaGA-HO and GraphGPT require a message-passing process during their Text-Graph Grounding phase, which depends on known graph structures. To accommodate this, we construct a static graph structure by aggregating interactions from the training set, ignoring their timestamps.
- Due to computational constraints, we omit the Self-Supervised Instruction Tuning phase in GraphGPT's training pipeline while retaining the other two phases.

Finally, in Table 8 and Table 9, we provide detailed prompts for all methods on the two tasks — temporal link prediction and temporal node classification — using the example of predicting whether there is a connection between u_4 and u_6 at time t_7, and predicting the label of u_4 at time t_7, respectively, as illustrated in Figure 6.

C Experiments Details

C.1 Setup Details

For the TGNN-Predictor methods, we primarily adopt the DyGLib codebase[1]. Following their workflow, we first perform a thorough grid search on the hyperparameters of nine traditional methods based on the temporal link prediction task to identify their optimal settings. After training the best-performing models on the temporal link prediction task, we use them as initial checkpoints to continue training for the temporal node classification task.

For the temporal link prediction task, we use supervised binary cross-entropy loss as the objective function. For the temporal node classification task, we treat each label in the FOOD and IMDB multi-label datasets as an independent binary classification problem, applying a weighted binary cross-entropy loss to account for class imbalance. For the other three multi-class datasets, we use a weighted cross-entropy loss as the objective function for the same reason.

For the LLM-Predictor methods, we follow the settings in their original papers as closely as possible, making appropriate adjustments to ensure they could be properly applied to temporal graphs. Specifically, for LlaGA-ND and LlaGA-HO, we consistently set the learning rate to 2e-5 and the batch size to 16 for all models, and trained them for one epoch on each dataset. For LlaGA-HO, since multi-hop

[1] https://github.com/yule-BUAA/DyGLib

Table 9: Prompts of LLM-as-Predictor methods for predicting the label of node u_4 at time t_7, based on the example in Figure 6.

Method	Prompt
LLaGA-ND	Given a node-centered graph: $\langle u_1, u_5, u_2, u_3 \rangle$, where the center node represents a user and the other nodes represent the user's historical interactions, including 1st-order neighbors (items that the user has reviewed) and 2nd-order neighbors (other users who have reviewed the same items), classify the center node into one or more of the following 9 interest categories: Side Dishes & Vegetables, Dietary & Specialty, Soups & Stews, Cuisine & Regional, Desserts & Sweets, Baking & Breads, Beverages & Snacks, Sauces & Condiments, Main Dishes.
LLaGA-HO	Given a node-centered graph: $\langle hop_0, hop_1, hop_2 \rangle$, where the center node represents a user and the other nodes represent the user's historical interactions, including 1st-order neighbors (items that the user has reviewed) and 2nd-order neighbors (other users who have reviewed the same items), classify the center node into one or more of the following 9 interest categories: Side Dishes & Vegetables, Dietary & Specialty, Soups & Stews, Cuisine & Regional, Desserts & Sweets, Baking & Breads, Beverages & Snacks, Sauces & Condiments, Main Dishes.
GraphGPT	Given a user-item review graph: $\langle u_1, u_5, u_2, u_3 \rangle$ where the 0th node is the target user, and the other nodes are its first- or second-order neighbors. The first-order neighbors are the items that the user has reviewed and the second-order neighbors are other users who have reviewed the same items. Classify the target user's interest into one or more of the following 9 categories: Side Dishes & Vegetables, Dietary & Specialty, Soups & Stews, Cuisine & Regional, Desserts & Sweets, Baking & Breads, Beverages & Snacks, Sauces & Condiments, Main Dishes.

embeddings of target nodes require the graph structure, we constructed a static graph by removing timestamp information from edges in the training set, which was then used as the graph structure during embedding computation.

For GraphGPT, due to the complexity of temporal graphs, we retained only two out of its original three pipeline stages: Structural Information Encoding with Text-Graph Grounding and Task-Specific Instruction Fine-tuning. Considering the higher computational cost of Task-Specific Instruction Tuning, we set the batch size to 4 for the Amazon-Kindle dataset and 8 for the other datasets, to fully utilize the 48GB memory of an NVIDIA A6000 GPU. The learning rate was kept at 2e-3, and we set the maximum output length of the LLM to 4096 tokens to accommodate as much structural information as possible. As with LlaGA, this model was also trained for one epoch on each dataset.

C.2 More Experiment Results and Analysis

Detailed Main Results. Regarding the TLP task, Tables 10 and 11 supplement Table 2 by providing results for an extended set of evaluation metrics.

TLP Inductive Results. In line with the evaluation protocol of DyGLib Yu et al. (2023), we also assess the performance of all methods under inductive scenarios in addition to transductive settings for TLP task. The transductive setting focuses on predicting future links among nodes encountered during training, whereas the inductive setting requires predicting future links for previously unseen nodes. The detailed inductive results for temporal link prediction task are presented in Tables 12 and 13.

Additional Negative Sampling Results. Following DyGLib Yu et al. (2023), we additionally evaluate all methods on the TLP task under both historical and inductive negative sampling strategies. The results are reported in Tables 14 to 21.

Table 10: Detailed main temporal link prediction results on small-scale datasets. Results are averaged over three independent runs (in %).

Methods	FOOD			IMDB			Librarything		
	AP	AUROC	MRR	AP	AUROC	MRR	AP	AUROC	MRR
JODIE	72.96±1.88	71.95±0.69	40.09±0.83	60.30±2.37	65.74±3.20	25.57±1.69	75.76±0.63	73.62±0.75	44.14±0.97
DyRep	72.83±0.88	69.01±1.53	40.55±0.44	64.79±1.30	69.79±2.03	27.05±0.60	73.81±1.55	72.34±1.13	44.13±0.73
TGAT	54.79±1.25	54.38±0.67	20.14±1.34	54.55±0.01	57.85±0.05	19.23±0.10	64.20±0.89	65.68±0.96	27.74±1.17
TGN	77.74±0.05	75.74±0.28	44.11±0.11	45.40±0.89	48.79±1.66	10.80±0.58	82.60±0.22	82.02±0.09	52.29±0.46
CAWN	51.71±0.38	53.96±0.23	17.43±0.38	54.00±1.17	55.60±2.75	19.52±0.49	62.01±0.21	63.21±0.07	26.70±0.08
TCL	49.12±0.79	50.98±1.23	15.43±0.55	55.78±0.49	58.83±0.44	20.18±0.21	54.51±4.02	56.63±2.91	18.88±4.14
GraphMixer	66.98±1.71	65.78±0.93	31.62±1.98	57.01±0.03	61.60±0.09	21.20±0.05	62.13±1.45	63.16±2.44	26.52±1.26
DyGFormer	59.48±0.24	57.26±0.52	24.51±0.13	55.32±0.11	58.42±0.10	20.24±0.02	54.53±2.16	54.35±3.14	19.75±2.34
FreeDyG	72.40±0.34	69.60±0.37	38.56±0.52	57.53±0.89	62.16±0.93	21.78±0.80	66.08±0.26	65.13±1.70	30.90±0.13
LKD4DyTAG	66.63±0.95	63.62±0.95	31.30±1.05	56.04±0.92	59.13±0.53	20.50±0.76	59.94±0.49	62.17±0.34	23.37±0.35
CROSS	58.15±0.13	56.48±0.24	23.17±0.23	55.98±0.05	59.14±0.12	20.62±0.06	59.04±0.83	60.25±0.94	24.16±0.53
TGTalker	51.04±0.02	52.02±0.03	16.24±0.15	50.22±0.04	50.43±0.07	15.12±0.10	50.63± 0.01	51.24±0.04	15.54±0.12
DST-v1	51.31±0.01	52.53±0.03	16.58±0.18	50.26±0.10	50.52±0.20	15.18±0.22	50.71±0.02	51.39±0.06	15.68±0.15
DST-v2	51.33±0.00	52.57±0.01	16.60±0.15	50.35±0.10	50.70±0.19	15.25±0.18	50.61±0.01	51.19±0.02	15.51±0.08
Llaga-ND	53.10±0.36	55.52±0.57	18.85±0.55	52.01±0.75	53.55±1.19	17.50±0.85	49.98±0.04	49.95±0.07	14.85±0.10
Llaga-HO	61.95±1.32	65.67±1.47	29.50±1.85	54.37±0.82	57.34±1.06	20.15±1.25	49.97±0.04	49.95±0.07	14.83±0.11
GraphGPT	55.09±0.43	57.95±1.02	25.05±0.85	51.80±0.18	53.19±0.29	17.20±0.45	49.96±0.04	49.89±0.11	14.81±0.18

Table 11: Detailed main temporal link prediction results on large-scale datasets. Results are averaged over three independent runs (in %).

Methods	Beeradvocate			Amazon-Kindle			Ratebeer		
	AP	AUROC	MRR	AP	AUROC	MRR	AP	AUROC	MRR
JODIE	79.75±11.63	80.93±9.53	57.49±16.89	82.81±2.00	81.86±1.69	60.05±3.18	92.62±0.18	91.56±0.36	74.20±0.10
DyRep	89.61±0.20	87.01±0.40	72.50±0.16	78.35±2.63	76.63±2.69	56.99±3.23	85.87±0.33	83.84±1.18	66.11±0.71
TGAT	59.37±3.59	66.58±2.13	27.90±4.64	79.00±1.97	81.12±1.32	44.37±2.71	58.93±4.20	64.38±2.29	28.63±3.66
TGN	83.46±1.68	83.25±1.06	59.35±2.48	71.77±2.90	79.92±1.49	39.67±3.72	91.55±0.24	91.41±0.24	72.00±0.31
CAWN	66.15±0.85	71.29±0.05	35.12±0.20	79.34±1.50	80.79±1.22	44.92±2.04	59.94±2.40	65.78±0.67	29.86±1.58
TCL	55.40±0.18	64.62±0.29	22.37±1.01	79.68±1.51	80.96±0.95	44.74±2.60	57.61±3.77	62.63±2.39	24.94±4.85
GraphMixer	86.02±0.24	85.57±0.29	60.51±0.84	81.53±0.28	81.87±0.09	48.81±0.44	78.53±4.22	78.05±4.06	48.52±7.06
DyGFormer	50.53±0.08	56.80±0.05	16.34±0.71	72.78±0.32	74.71±0.09	36.61±0.22	53.28±0.91	57.65±0.73	23.09±1.33
FreeDyG	82.93±6.75	85.18±4.07	58.86±8.56	67.26±3.83	71.25±1.99	32.80±4.85	69.49±3.30	74.16±1.86	40.10±3.65
LKD4DyTAG	79.42±1.15	79.52±0.91	47.51±1.41	80.64±0.46	81.64±0.42	46.44±0.59	69.51±0.03	70.85±0.05	37.59±0.07
CROSS	51.36±1.03	57.52±1.01	17.89±2.25	87.25±0.13	88.31±0.12	56.99±0.38	52.57±0.09	57.19±0.21	21.32±0.98
TGTalker	51.18±0.03	51.36±0.02	16.05±0.08	52.54±0.09	54.79±0.11	18.21±0.35	51.11±0.07	52.15±0.06	16.10±0.14
DST-v1	51.34±0.04	51.47±0.05	16.18±0.10	52.66±0.10	54.85±0.09	18.32±0.41	51.23±0.04	51.26±0.03	16.15±0.09
DST-v2	51.21±0.01	51.53±0.01	16.12±0.05	52.62±0.08	54.81±0.08	18.28±0.38	51.05±0.01	51.07±0.02	15.98±0.05
Llaga-ND	49.97±0.03	49.94±0.06	14.82±0.12	68.62±3.37	72.85±3.58	36.45±4.12	50.00±0.02	49.99±0.05	14.90±0.08
Llaga-HO	50.00±0.05	50.00±0.10	14.91±0.15	72.64±1.11	76.63±1.54	42.10±2.05	50.00±0.00	50.00±0.00	15.07±0.10
GraphGPT	56.80±6.80	58.35±8.35	22.15±7.50	50.72±0.72	51.17±1.17	15.75±1.05	56.03±6.03	59.29±9.29	21.80±8.15

Table 12: Temporal link prediction results on small-scale datasets under inductive settings. Results are averaged over three independent runs (in %).

Methods	FOOD			IMDB			Librarything		
	AP	AUROC	MRR	AP	AUROC	MRR	AP	AUROC	MRR
JODIE	76.02±0.07	74.28±0.06	41.71±0.32	88.39±0.21	87.47±0.35	61.51±0.31	84.46±0.02	84.47±0.08	57.68±0.63
DyRep	73.08±0.91	71.48±0.85	37.38±1.46	86.79±0.49	85.72±0.60	59.30±0.41	82.45±0.16	82.96±0.21	54.40±0.21
TGAT	54.75±0.65	54.30±0.23	20.12±0.67	74.53±0.21	73.75±0.30	39.85±0.06	71.31±0.95	71.94±0.93	33.97±1.28
TGN	73.77±0.44	72.14±0.46	38.43±0.30	79.66±1.92	79.23±2.39	47.26±1.20	83.38±0.03	84.35±0.20	54.87±0.16
CAWN	53.06±0.41	54.33±0.50	18.74±0.35	63.64±0.95	66.12±0.27	27.52±1.43	68.74±1.25	69.21±1.06	31.96±1.66
TCL	50.40±0.75	50.71±1.14	16.85±0.42	67.39±3.46	68.20±1.39	31.95±4.87	64.02±2.34	65.14±1.62	26.35±2.96
GraphMixer	68.41±1.54	67.19±1.10	33.08±1.68	86.67±0.03	84.91±0.02	60.07±0.03	77.25±1.73	76.95±2.01	43.78±2.03
DyGFormer	58.00±0.21	56.37±0.44	23.03±0.10	62.33±0.38	65.88±0.09	25.71±0.48	61.16±3.16	60.93±3.65	24.63±2.80
FreeDyG	72.65±0.18	70.38±0.13	38.47±0.35	86.90±0.08	85.15±0.10	60.26±0.07	79.81±0.71	78.54±1.32	47.81±0.98
LKD4DyTAG	62.13±0.67	59.74±0.65	26.74±0.61	61.81±0.76	65.71±0.22	25.20±0.80	67.44±0.19	68.06±0.24	29.98±0.03
CROSS	56.77±0.44	55.39±0.28	21.95±0.35	62.27±1.18	66.97±0.47	25.16±1.39	65.79±0.51	66.24±0.53	28.82±0.41
TGTalker	51.04±0.02	52.02±0.03	16.20±0.12	50.22±0.04	50.43±0.07	15.15±0.11	50.63± 0.01	51.24±0.04	15.50±0.08
DST-v1	51.31±0.01	52.53±0.03	16.55±0.15	50.26±0.10	50.52±0.20	15.20±0.20	50.71±0.02	51.39±0.06	15.65±0.12
DST-v2	51.33±0.00	52.57±0.01	16.58±0.10	50.35±0.10	50.70±0.19	15.28±0.15	50.61±0.01	51.19±0.02	15.48±0.06
Llaga-ND	53.10±0.36	55.52±0.57	18.80±0.50	52.01±0.75	53.55±1.19	17.45±0.80	49.98±0.04	49.95±0.07	14.85±0.12
Llaga-HO	61.95±1.32	65.67±1.47	29.45±1.80	54.37±0.82	57.34±1.06	20.10±1.20	49.97±0.04	49.95±0.07	14.83±0.10
GraphGPT	55.09±0.43	57.95±1.02	25.00±0.80	51.80±0.18	53.19±0.29	17.15±0.40	49.96±0.04	49.89±0.11	14.80±0.15

Text Ablation Study Results. Figure 7 presents detailed performance comparisons on both TLP and TNC before and after removing textual attributes. The results show that textual information

Table 13: Temporal link prediction results on large-scale datasets under inductive settings. Results are averaged over three independent runs (in %).

Methods	Beeradvocate			Amazon-Kindle			Ratebeer		
	AP	AUROC	MRR	AP	AUROC	MRR	AP	AUROC	MRR
JODIE	90.84±0.51	89.99±0.68	68.66±0.96	93.78±0.38	93.01±0.37	78.89±0.98	90.38±0.70	89.81±0.77	67.92±1.10
DyRep	85.81±2.03	84.59±1.78	62.14±3.80	91.25±0.43	90.17±0.46	73.40±0.85	83.59±0.11	81.82±0.08	57.22±0.03
TGAT	59.67±3.80	62.48±2.60	29.21±4.06	82.51±1.03	83.97±0.58	49.22±1.64	62.19±3.89	64.92±2.37	31.86±3.16
TGN	79.71±8.75	80.85±7.37	52.83±13.11	94.14±0.51	94.43±0.36	77.84±0.14	86.56±0.87	86.10±1.15	61.47±1.89
CAWN	66.89±0.95	68.29±0.09	36.61±0.28	80.89±0.11	81.90±0.18	46.82±0.33	63.92±2.23	66.57±0.79	33.90±1.23
TCL	54.64±0.16	59.35±0.57	24.08±1.07	81.60±0.22	82.53±0.13	47.51±0.67	61.99±4.64	63.42±3.14	30.17±5.63
GraphMixer	87.00±0.19	85.52±0.26	62.75±0.59	93.33±0.04	92.71±0.03	74.54±0.11	82.19±3.88	80.38±4.26	54.45±6.40
DyGFormer	49.72±0.31	52.27±0.24	17.36±0.81	74.79±0.40	76.32±0.17	38.63±0.31	57.53±0.92	59.29±0.80	26.90±1.26
FreeDyG	84.22±5.36	84.83±3.81	60.85±5.85	87.64±1.74	86.77±1.48	64.46±4.09	76.08±2.92	77.48±2.02	48.59±3.36
LKD4DyTAG	76.66±0.82	75.61±0.82	43.50±0.81	80.48±0.07	81.69±0.12	46.02±0.17	71.56±0.16	70.95±0.14	39.32±0.22
CROSS	50.99±1.61	53.34±1.42	19.13±2.48	87.47±0.22	88.73±0.21	57.42±0.36	56.94±0.11	59.06±0.06	25.55±0.74
TGTalker	51.18±0.03	51.36±0.02	16.00±0.09	52.54±0.09	54.79±0.11	18.15±0.30	52.41±0.13	54.09±3.92	17.80±2.50
DST-v1	51.34±0.04	51.47±0.05	16.15±0.11	52.66±0.10	54.85±0.09	18.25±0.38	52.96±0.94	55.60±1.22	18.65±1.05
DST-v2	51.21±0.01	51.53±0.01	16.10±0.04	52.62±0.08	54.81±0.08	18.20±0.35	51.72±1.08	54.15±2.89	17.15±1.85
Llaga-ND	49.97±0.03	49.94±0.06	14.80±0.15	68.62±3.37	72.85±3.58	36.40±4.10	50.00±0.02	49.99±0.05	14.90±0.05
Llaga-HO	50.00±0.05	50.00±0.10	14.90±0.12	72.64±1.11	76.63±1.54	42.05±2.00	50.00±0.00	50.00±0.00	14.90±0.00
GraphGPT	56.80±6.80	58.35±8.35	22.10±7.45	50.72±0.72	51.17±1.17	15.70±1.00	56.03±6.03	59.29±9.29	27.75±8.10

Table 14: Temporal link prediction results on small-scale datasets under transductive settings with historical negative sampling. Results are averaged over three independent runs (in %).

Methods	FOOD			IMDB			Librarything		
	AP	AUROC	MRR	AP	AUROC	MRR	AP	AUROC	MRR
JODIE	55.00±0.11	55.13±0.30	20.64±0.17	66.09±8.32	72.60±7.93	30.45±9.29	53.86±1.56	53.43±1.58	19.97±1.10
DyRep	55.13±0.49	53.93±0.49	21.10±0.40	57.57±3.78	57.83±4.89	23.09±2.91	56.62±1.13	56.83±0.87	21.91±0.93
TGAT	51.38±0.05	50.69±0.09	17.92±0.04	46.59±0.37	45.36±1.13	14.18±0.13	45.51±0.06	45.35±0.22	13.11±0.16
TGN	55.73±0.21	57.26±0.23	20.58±0.16	40.40±0.71	37.98±1.74	8.39±0.32	53.66±0.36	54.44±0.17	19.25±0.30
CAWN	52.26±0.01	52.06±0.04	18.50±0.05	45.85±0.25	43.61±0.47	14.19±0.07	48.06±0.16	48.74±0.10	14.98±0.10
TCL	50.49±0.04	49.83±0.12	17.29±0.14	46.07±1.06	44.17±0.93	14.32±0.86	46.04±0.32	46.53±0.54	13.39±0.22
GraphMixer	58.75±1.17	57.56±1.90	23.79±0.76	51.95±0.26	57.20±0.77	16.65±0.03	51.01±0.28	49.59±0.55	17.90±0.41
DyGFormer	50.85±0.66	49.79±1.16	17.57±0.36	46.75±0.28	44.53±0.50	14.80±0.25	49.37±1.25	50.20±0.52	16.00±1.09
FreeDyG	57.66±0.31	55.46±0.59	23.30±0.22	55.06±0.83	59.51±1.48	19.67±0.30	51.03±0.45	49.23±0.00	18.06±0.38
LKD4DyTAG	52.30±0.00	51.96±0.08	18.62±0.02	45.77±0.60	42.98±0.36	14.12±0.46	46.09±0.14	46.16±0.21	13.71±0.12
CROSS	50.86±0.28	49.75±0.11	17.66±0.21	47.71±0.04	45.20±0.02	15.63±0.07	46.78±0.05	47.45±0.08	14.03±0.06

substantially affects TGNN performance on TLP but has relatively limited impact on TNC, whereas for LLM-Predictors the opposite trend holds: text contributes significantly to TNC while providing smaller gains for TLP.

Comparison of LP and E2E for TGNNs on TNC. As shown in Table 22, end-to-end fine-tuning does not yield significant improvements over linear probing on the TNC task for TGNNs. This suggests that gradients from semantic supervision are insufficient to overcome the strong structural bias learned during TLP pretraining. Consequently, the representations remain anchored in topological dynamics and fail to adapt effectively to semantic objectives.

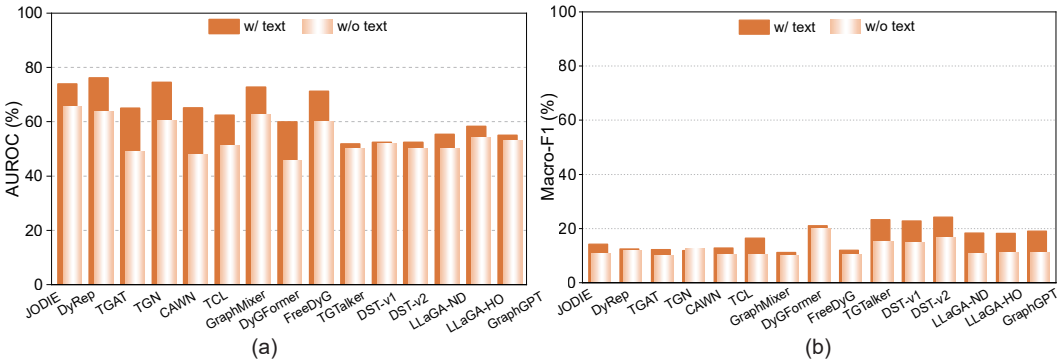

Figure 7: Effect of Text Attributes on TLP. Impact of textual attributes on model performance across different methods for the TLP task.

Table 15: Temporal link prediction results on small-scale datasets under inductive settings with historical negative sampling. Results are averaged over three independent runs (in %).

Methods	FOOD			IMDB			Librarything		
	AP	AUROC	MRR	AP	AUROC	MRR	AP	AUROC	MRR
JODIE	50.08±0.39	46.65±0.10	17.65±0.47	74.19±0.85	71.07±1.83	41.40±0.35	52.27±0.06	47.60±0.69	19.80±0.09
DyRep	48.32±0.29	45.69±0.06	16.11±0.29	69.73±0.52	65.67±1.27	36.57±0.31	50.86±0.17	46.70±0.74	18.40±0.04
TGAT	51.01±0.25	50.25±0.03	17.66±0.30	64.16±0.67	60.85±0.71	30.34±0.56	47.20±0.22	45.20±0.09	15.05±0.15
TGN	49.07±0.34	46.38±0.32	16.72±0.23	69.85±2.88	64.18±1.71	37.97±4.38	46.31±0.19	43.16±0.34	14.64±0.16
CAWN	54.97±0.22	53.61±0.08	20.31±0.22	55.21±1.39	54.88±1.44	21.38±1.18	50.89±1.13	50.06±1.36	17.55±0.81
TCL	53.10±0.17	49.76±0.48	19.31±0.30	57.06±4.53	54.56±2.23	23.76±4.87	47.35±0.47	44.99±0.48	15.14±0.41
GraphMixer	56.46±1.19	53.37±1.60	22.80±0.94	75.58±0.02	71.56±0.10	44.36±0.21	61.46±2.92	59.97±2.19	26.70±2.84
DyGFormer	48.37±1.76	46.33±3.88	15.92±0.80	53.50±0.41	53.75±0.04	19.53±0.49	53.57±0.32	56.43±0.14	18.51±0.29
FreeDyG	54.50±0.49	50.54±1.00	21.36±0.24	76.96±0.31	73.35±0.45	46.00±0.55	58.12±1.22	56.72±1.11	23.62±1.13
LKD4DyTAG	50.34±0.08	49.50±0.13	17.12±0.01	51.36±0.19	51.15±0.01	17.98±0.19	48.17±0.04	46.87±0.00	15.57±0.01
CROSS	49.78±0.42	48.83±0.34	16.78±0.33	55.17±1.15	56.38±0.67	20.51±1.17	50.15±0.24	50.27±0.43	16.61±0.14

Table 16: Temporal link prediction results on large-scale datasets under transductive settings with historical negative sampling. Results are averaged over three independent runs (in %).

Methods	Beeradvocate			Amazon-Kindle			Ratebeer		
	AP	AUROC	MRR	AP	AUROC	MRR	AP	AUROC	MRR
JODIE	57.46±3.08	62.36±0.84	21.12±3.28	68.84±1.66	68.96±1.42	32.45±1.80	64.03±0.51	67.15±0.24	26.89±0.48
DyRep	68.04±2.86	71.01±2.67	32.16±2.35	65.26±0.94	64.98±1.07	29.23±0.81	69.96±2.88	72.18±3.36	33.95±2.39
TGAT	52.55±0.13	52.82±0.14	18.40±0.09	53.77±0.52	54.89±0.54	19.24±0.37	50.26±0.08	49.82±0.23	17.01±0.00
TGN	53.54±2.03	57.49±0.89	18.18±1.89	44.79±0.59	46.94±0.86	11.13±0.64	57.87±3.88	62.78±2.82	21.51±3.49
CAWN	52.34±0.05	53.17±0.01	18.26±0.04	54.31±0.90	54.83±1.17	19.78±0.63	51.02±0.18	51.85±0.10	17.30±0.19
TCL	52.68±0.17	52.85±0.18	18.57±0.13	55.02±0.81	54.65±0.78	20.49±0.61	53.63±1.84	52.81±1.25	19.66±1.55
GraphMixer	54.82±1.62	57.02±1.22	19.70±1.33	64.62±0.03	65.97±0.01	28.16±0.06	54.79±0.71	55.97±1.02	19.96±0.48
DyGFormer	52.47±0.24	52.58±0.14	18.45±0.20	53.75±0.64	54.18±0.53	19.29±0.50	53.67±0.18	53.52±0.33	19.43±0.14
FreeDyG	50.56±1.71	54.11±0.91	15.86±1.66	65.16±0.09	65.98±0.29	28.79±0.18	54.35±0.29	57.11±0.04	19.33±0.29
LKD4DyTAG	51.97±0.10	52.35±0.13	18.01±0.05	54.95±0.50	55.12±0.63	20.29±0.38	49.85±0.16	49.30±0.28	16.74±0.11
CROSS	52.14±0.34	52.16±0.30	18.23±0.21	56.22±0.01	56.53±0.17	21.18±0.01	54.36±1.19	54.11±0.89	20.03±1.05

Table 17: Temporal link prediction results on large-scale datasets under inductive settings with historical negative sampling. Results are averaged over three independent runs (in %).

Methods	Beeradvocate			Amazon-Kindle			Ratebeer		
	AP	AUROC	MRR	AP	AUROC	MRR	AP	AUROC	MRR
JODIE	53.42±0.25	53.05±0.41	19.51±0.14	78.75±0.39	75.38±0.08	47.83±0.95	53.14±0.30	52.69±0.53	19.28±0.44
DyRep	50.82±0.86	49.19±0.43	17.78±0.78	73.53±1.42	68.86±1.20	41.20±2.08	54.15±0.03	53.32±0.88	20.03±0.20
TGAT	54.06±0.47	52.84±0.14	19.76±0.37	56.59±0.09	56.15±0.15	21.88±0.05	52.97±0.79	52.02±0.25	19.04±0.68
TGN	49.39±0.31	48.17±0.39	16.48±0.23	63.52±0.44	58.78±0.36	29.33±0.76	50.70±0.90	48.66±0.41	17.66±0.79
CAWN	52.81±0.17	53.49±0.22	18.60±0.12	56.49±0.47	55.43±0.34	21.89±0.41	54.89±0.16	56.34±0.14	19.94±0.12
TCL	54.23±0.33	52.83±0.14	20.06±0.18	55.04±0.35	54.30±0.27	20.66±0.28	56.75±0.76	55.37±0.69	22.38±0.55
GraphMixer	55.63±1.05	56.63±0.78	20.74±0.88	78.70±0.73	76.63±0.39	46.89±1.27	58.13±0.03	59.04±0.21	22.71±0.12
DyGFormer	55.95±0.21	54.07±0.19	21.27±0.20	56.36±0.52	55.74±0.50	21.55±0.37	57.23±0.11	57.84±0.20	21.90±0.20
FreeDyG	54.44±0.68	56.03±0.00	19.73±0.62	81.28±0.69	78.73±0.51	51.06±1.32	60.23±0.56	62.79±0.98	23.98±0.34
LKD4DyTAG	51.19±0.05	50.61±0.13	17.70±0.01	56.21±0.10	55.15±0.07	21.63±0.09	51.12±0.25	50.64±0.35	17.64±0.17
CROSS	54.58±1.13	52.96±0.70	20.32±0.81	59.80±0.01	59.52±0.16	24.25±0.07	58.00±1.19	58.44±0.76	22.67±1.14

Scalability Results. Figure 8 reports scalability evaluations of all methods. Specifically, on the largest dataset, Amazon-Kindle, we subsample 1M, 2M, 3M, 4M, and 5M edges and measure the runtime of each method. Results show that all methods exhibit approximately linear growth in computation time as data size increases, demonstrating good scalability. LLM-Predictor methods incur higher time cost due to LLM inference, but still maintain favorable scaling behavior.

D LIMITATIONS

While TTGBench enables a comprehensive evaluation of LLM-based approaches for temporal graph learning under both paradigms, there are limitations worth noting. Since no LLM-as-Predictor methods are specifically designed for temporal graphs, we adapt state-of-the-art models originally developed for static graphs. To accommodate the increased structural and temporal complexity, we introduce necessary simplifications to ensure computational feasibility, which may limit their performance. Addressing these limitations will be crucial for developing more capable, dedicated LLM-as-Predictor frameworks tailored to temporal graph learning.

Table 18: Temporal link prediction results on small-scale datasets under transductive settings with inductive negative sampling. Results are averaged over three independent runs (in %).

Methods	FOOD			IMDB			Librarything		
	AP	AUROC	MRR	AP	AUROC	MRR	AP	AUROC	MRR
JODIE	52.27±1.48	51.54±0.87	19.00±1.18	54.04±9.34	57.97±13.71	19.38±7.05	57.33±1.47	57.08±1.90	23.53±1.14
DyRep	55.10±0.51	52.88±0.78	21.37±0.36	57.29±3.41	50.87±3.44	26.33±3.64	63.21±1.50	63.42±1.17	28.00±1.70
TGAT	51.24±0.20	50.62±0.21	17.71±0.21	44.95±0.38	43.83±1.01	13.27±0.12	43.18±0.24	41.18±0.70	12.04±0.06
TGN	55.34±0.24	56.03±0.49	20.67±0.10	35.32±0.12	20.45±0.41	6.44±0.14	59.17±0.46	60.54±0.29	23.69±0.45
CAWN	53.06±0.07	53.83±0.10	18.77±0.04	46.32±0.41	44.02±0.38	14.62±0.29	47.85±0.06	48.47±0.06	14.92±0.11
TCL	50.53±0.01	50.22±0.21	17.13±0.12	46.19±1.44	44.56±1.49	14.55±1.24	43.74±0.11	42.97±0.46	11.76±0.09
GraphMixer	53.81±0.68	50.34±1.35	20.70±0.30	42.63±0.27	40.60±0.85	11.52±0.05	45.64±0.55	41.81±0.15	14.65±0.46
DyGFormer	49.32±1.83	47.18±3.92	16.67±0.79	46.97±0.17	44.98±0.25	15.29±0.14	49.46±0.01	51.30±1.58	15.74±0.38
FreeDyG	53.45±0.38	49.25±0.98	20.80±0.18	44.83±0.08	43.89±0.16	13.00±0.01	45.92±0.42	42.00±0.04	15.11±0.40
LKD4DyTAG	52.74±0.10	52.29±0.13	18.98±0.09	46.33±0.89	43.57±0.76	14.75±0.78	44.29±0.06	43.15±0.08	12.64±0.02
CROSS	50.92±0.10	50.07±0.03	17.54±0.10	48.93±0.08	47.19±0.07	16.50±0.05	46.54±0.04	47.15±0.11	13.82±0.07

Table 19: Temporal link prediction results on small-scale datasets under inductive settings with inductive negative sampling. Results are averaged over three independent runs (in %).

Methods	FOOD			IMDB			Librarything		
	AP	AUROC	MRR	AP	AUROC	MRR	AP	AUROC	MRR
JODIE	50.08±0.39	46.65±0.10	17.65±0.47	74.19±0.85	71.07±1.83	41.40±0.35	52.27±0.06	47.60±0.69	19.80±0.09
DyRep	48.32±0.29	45.69±0.06	16.11±0.29	69.73±0.52	65.67±1.27	36.57±0.31	50.86±0.17	46.70±0.74	18.40±0.04
TGAT	51.01±0.25	50.25±0.03	17.66±0.30	64.16±0.67	60.85±0.71	30.34±0.56	47.20±0.22	45.20±0.09	15.05±0.15
TGN	49.07±0.34	46.38±0.32	16.72±0.23	69.85±2.88	64.18±1.71	37.97±4.38	46.31±0.19	43.16±0.34	14.64±0.16
CAWN	54.97±0.22	53.61±0.08	20.31±0.22	55.21±1.39	54.88±1.44	21.38±1.18	50.89±1.13	50.06±1.36	17.55±0.81
TCL	53.10±0.17	49.76±0.48	19.31±0.30	57.06±4.53	54.56±2.23	23.76±4.87	47.35±0.47	44.99±0.48	15.14±0.41
GraphMixer	56.46±1.19	53.37±1.60	22.80±0.94	75.58±0.02	71.56±0.10	44.36±0.21	61.46±2.92	59.97±2.19	26.70±2.84
DyGFormer	48.37±1.76	46.33±3.88	15.92±0.80	53.50±0.41	53.75±0.04	19.53±0.49	53.57±0.32	56.43±0.14	18.51±0.29
FreeDyG	54.50±0.49	50.54±1.00	21.36±0.24	76.96±0.31	73.35±0.45	46.00±0.55	58.12±1.22	56.72±1.11	23.62±1.13
LKD4DyTAG	50.34±0.08	49.50±0.13	17.12±0.01	51.36±0.19	51.15±0.01	17.98±0.19	48.17±0.04	46.87±0.00	15.57±0.01
CROSS	49.78±0.42	48.83±0.34	16.78±0.33	55.17±1.15	56.38±0.67	20.51±1.17	50.15±0.24	50.27±0.43	16.61±0.14

Table 20: Temporal link prediction results on large-scale datasets under transductive settings with inductive negative sampling. Results are averaged over three independent runs (in %).

Methods	Beeradvocate			Amazon-Kindle			Ratebeer		
	AP	AUROC	MRR	AP	AUROC	MRR	AP	AUROC	MRR
JODIE	57.75±2.59	63.12±0.18	23.14±3.13	58.69±2.39	55.79±2.66	25.10±2.06	66.28±0.28	70.43±0.20	30.59±0.44
DyRep	71.73±2.56	74.04±2.53	38.10±2.50	62.65±1.46	61.27±1.96	27.58±1.08	72.86±1.78	75.20±2.50	38.07±1.33
TGAT	50.57±0.25	50.73±0.16	17.17±0.28	48.21±0.78	47.43±1.01	16.15±0.66	49.80±0.22	49.28±0.25	16.90±0.23
TGN	54.88±0.29	60.31±1.15	20.32±0.72	35.99±0.65	23.46±2.47	7.83±0.52	64.44±3.75	71.05±2.62	27.93±3.69
CAWN	51.43±0.06	52.82±0.10	17.77±0.04	49.99±1.19	49.20±1.82	17.48±0.79	52.54±0.12	55.03±0.13	18.23±0.18
TCL	50.23±0.39	50.38±0.23	16.77±0.33	49.56±0.54	47.92±0.81	17.44±0.40	50.93±0.85	50.36±0.63	17.65±0.89
GraphMixer	51.71±0.75	52.29±0.73	19.50±0.58	50.51±0.13	49.41±0.07	18.45±0.14	52.58±0.20	52.42±0.44	19.84±0.39
DyGFormer	50.30±0.33	50.62±0.15	16.80±0.27	50.89±0.94	50.79±0.94	17.57±0.64	52.44±0.20	53.81±0.34	18.13±0.13
FreeDyG	51.41±0.41	52.26±0.24	19.03±0.61	51.03±0.02	49.42±0.38	18.19±0.03	54.36±1.00	56.53±1.66	20.11±0.69
LKD4DyTAG	50.53±0.07	50.64±0.09	17.93±0.12	51.88±0.71	50.77±1.01	19.03±0.48	49.62±0.08	49.11±0.19	16.92±0.05
CROSS	50.13±0.39	50.02±0.57	16.81±0.20	55.64±0.05	56.70±0.47	21.64±0.10	53.31±0.73	54.90±0.66	18.83±0.64

Table 21: Temporal link prediction results on large-scale datasets under inductive settings with inductive negative sampling. Results are averaged over three independent runs (in %).

Methods	Beeradvocate			Amazon-Kindle			Ratebeer		
	AP	AUROC	MRR	AP	AUROC	MRR	AP	AUROC	MRR
JODIE	53.42±0.25	53.05±0.41	19.51±0.14	78.75±0.39	75.38±0.08	47.83±0.95	53.14±0.30	52.69±0.53	19.28±0.44
DyRep	50.82±0.86	49.19±0.43	17.78±0.78	73.53±1.42	68.86±1.20	41.20±2.08	54.15±0.03	53.32±0.88	20.03±0.20
TGAT	54.06±0.47	52.84±0.14	19.76±0.37	56.59±0.09	56.15±0.15	21.88±0.05	52.97±0.79	52.02±0.25	19.04±0.68
TGN	49.39±0.31	48.17±0.39	16.48±0.23	63.52±0.44	58.78±0.36	29.33±0.76	50.70±0.90	48.66±0.41	17.66±0.79
CAWN	52.81±0.17	53.49±0.22	18.60±0.12	56.49±0.47	55.43±0.34	21.89±0.41	54.89±0.16	56.34±0.14	19.94±0.12
TCL	54.23±0.33	52.83±0.14	20.06±0.18	55.04±0.35	54.30±0.27	20.66±0.28	56.75±0.76	55.37±0.69	22.38±0.55
GraphMixer	55.63±1.05	56.63±0.78	20.74±0.88	78.70±0.73	76.63±0.39	46.89±1.27	58.13±0.03	59.04±0.21	22.71±0.12
DyGFormer	55.95±0.21	54.07±0.19	21.27±0.20	56.36±0.52	55.74±0.50	21.55±0.37	57.23±0.11	57.84±0.20	21.90±0.20
FreeDyG	54.44±0.68	56.03±0.00	19.73±0.62	81.28±0.69	78.73±0.51	51.06±1.32	60.23±0.56	62.79±0.98	23.98±0.34
LKD4DyTAG	51.19±0.05	50.61±0.13	17.70±0.01	56.21±0.10	55.15±0.07	21.63±0.09	51.12±0.25	50.64±0.35	17.64±0.17
CROSS	54.58±1.13	52.96±0.70	20.32±0.81	59.80±0.01	59.52±0.16	24.25±0.07	58.00±1.19	58.44±0.76	22.67±1.14

Table 22: Performance comparison of TGNNs on the TNC task under two training strategies: linear probing (LP) and end-to-end fine-tuning (E2E). Linear probing trains only the classifier head on a TLP-pretrained backbone, while end-to-end fine-tuning jointly optimizes both the backbone and the classifier. Results are reported in terms of Macro-F1.

Methods	FOOD		IMDB		Beeradvocate		Amazon-Kindle		Ratebeer	
	LP	E2E	LP	E2E	LP	E2E	LP	E2E	LP	E2E
JODIE	25.95	25.94	12.65	12.65	7.00	6.95	17.60	13.18	7.86	8.03
DyRep	21.22	21.40	13.74	13.77	6.50	6.49	13.36	14.02	7.38	8.62
TGAT	22.73	22.73	8.53	8.53	6.18	6.14	14.91	15.34	8.67	8.66
TGN	21.98	21.80	8.51	8.51	8.35	7.95	13.22	OOM	7.34	8.22
CAWN	20.28	19.99	11.48	11.49	8.77	8.66	14.96	15.96	8.79	9.04
TCL	19.99	19.99	14.56	13.60	9.26	8.17	24.25	27.06	14.34	14.45
GraphMixer	19.99	19.99	9.20	9.29	6.14	6.14	13.35	14.62	7.13	7.27
DyGFormer	36.58	26.58	20.05	12.85	12.91	10.16	20.19	19.16	15.52	13.79
FreeDyG	20.46	20.88	10.20	10.02	7.75	7.24	12.80	13.77	8.54	8.28

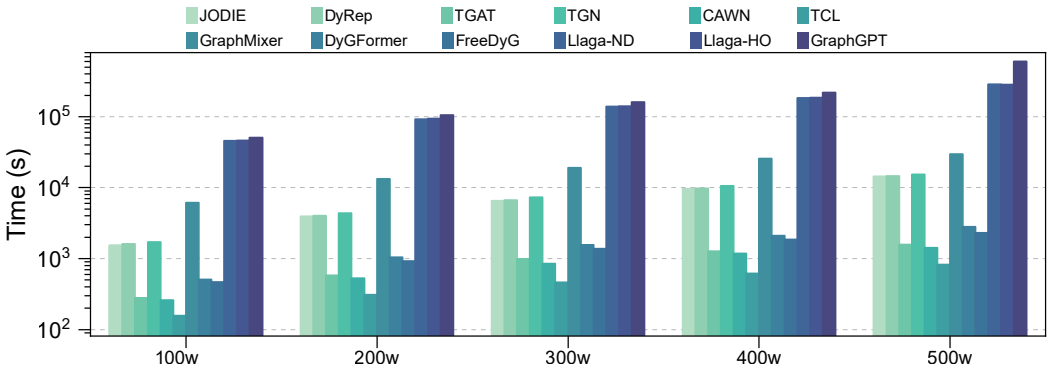

Figure 8: Scalability results across all methods.

E Broader Impact

This work introduces TTGBench, a comprehensive benchmark for evaluating temporal graph learning under both structural evolution and semantic drift. By curating six text-rich datasets exhibiting dual volatility, TTGBench addresses an important limitation of existing benchmarks, which largely emphasize structural memorization while underrepresenting dynamic semantic transitions. In doing so, it provides a more realistic and challenging testbed for developing models with broader temporal reasoning capabilities.

A key broader impact of TTGBench is methodological: by jointly evaluating two complementary tasks and systematically benchmarking representative methods spanning both TGNN-based models and emerging LLM-as-Predictor paradigms, the benchmark can help drive the development of more general, robust, and capable graph learning systems. We hope this contributes to progress toward foundation models for temporal graphs and encourages research beyond narrowly optimized single-task solutions.

Beyond academia, TTGBench may also have practical value for industrial applications involving dynamic relational systems, such as recommendation, financial risk analysis, and evolving user-interest modeling. In particular, our comprehensive evaluation of predictive performance together with efficiency trade-offs can inform practitioners in selecting models appropriate for deployment under different resource and application constraints.