TGB-SEQ BENCHMARK: CHALLENGING TEMPORAL GNNS WITH COMPLEX SEQUENTIAL DYNAMICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Future link prediction is a fundamental challenge in various real-world dynamic systems. To address this, numerous temporal graph neural networks (temporal GNNs) and benchmark datasets have been developed. However, these datasets often feature excessive repeated edges and lack complex sequential dynamics, a key characteristic inherent in many real-world applications such as recommender systems and "Who-To-Follow" on social networks. This oversight has led existing methods to inadvertently downplay the importance of learning sequential dynamics, focusing primarily on predicting repeated edges.

In this study, we demonstrate that existing methods, such as GraphMixer and DyGFormer, are inherently incapable of learning simple sequential dynamics, such as "a user who has followed OpenAI and Anthropic is more likely to follow AI at Meta next." Motivated by this issue, we introduce the Temporal Graph Benchmark with Sequential Dynamics (TGB-Seq), a new benchmark carefully curated to minimize repeated edges, challenging models to learn sequential dynamics and generalize to unseen edges. TGB-Seq comprises large real-world datasets spanning diverse domains, including e-commerce interactions, movie ratings, business reviews, social networks, citation networks and web link networks. Benchmarking experiments reveal that current methods usually suffer significant performance degradation and incur substantial training costs on TGB-Seq, posing new challenges and opportunities for future research. The datasets and benchmarking code are available at `https://anonymous.4open.science/r/TGB-Seq-3F23`.

1 INTRODUCTION

Future link prediction (Divakaran & Mohan, 2020) is a fundamental challenge in various real-world dynamic systems, such as social networks (Daud et al., 2020), e-commerce (Bai et al., 2020), financial systems (Rajput & Singh, 2022). For instance, an online shopping website must decide which items to recommend to users based on their click history, while a social networking platform needs to identify which users may be interested in connecting based on their existing relationships. Among the various approaches for future link prediction, temporal Graph Neural Networks (GNNs) are particularly notable for their flexibility in modeling diverse applications and their representation learning capabilities (Zheng et al., 2024; Skarding et al., 2021; Kazemi et al., 2020). Recently, several temporal GNN methods (Yu et al., 2023) have demonstrated impressive performance in future link prediction on existing benchmarks (Poursafaei et al., 2022). However, most existing datasets are not derived from real-world recommender systems, despite recommendations being a natural and essential application of future link prediction.

Observations. To assess the capability of current temporal GNNs in recommendation tasks, we evaluate their performance on future link prediction using two widely used recommendation datasets, including the user-product interaction network Taobao (Zhu et al., 2018) and the business review network Yelp [1]. Figure 1 presents the performance of three state-of-the-art temporal GNN approaches across these datasets, including EdgeBank (Poursafaei et al., 2022), GraphMixer (Cong et al., 2023) and DyGFormer (Yu et al., 2023). We split these datasets chronologically and randomly sample 100 negative destination nodes for each positive instance, utilizing the Mean Reciprocal Rank (MRR) as the evaluation metric. Besides, we also include SGNN-HN (Pan et al., 2020),

[1] https://www.yelp.com/dataset

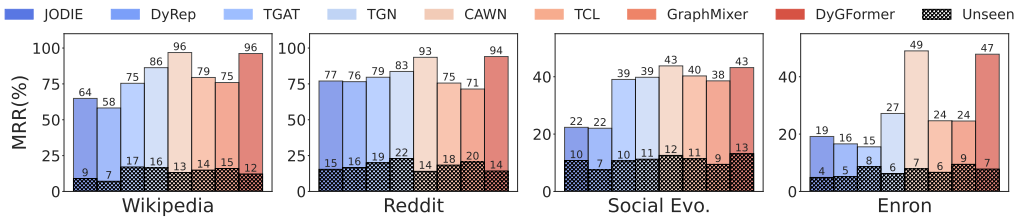

Figure 2: The MRR scores of eight popular temporal GNNs for predicting repeated historical edges on four previously established datasets. "Unseen" denotes the performance of unseen edges.

one of the state-of-the-art methods for sequential recommendation to compare with temporal GNNs. Intuitively, these recommendation datasets are comparable to existing datasets (e.g., Wikipedia and Reddit), as all represent typical dynamic systems, and thus, temporal GNNs are expected to perform in a similar trend on these recommendation datasets. However, Figure 1 shows that temporal GNN methods present significant performance degradation compared to their strong results on two previously established datasets and present a substantial performance gap compared with SGNN-HN, which contradicts our intuition. This observation raises a critical question: *Why do existing temporal GNN methods, which demonstrate superior performance on established temporal graph datasets, fail to perform well in a typical downstream application, i.e., recommendation?*

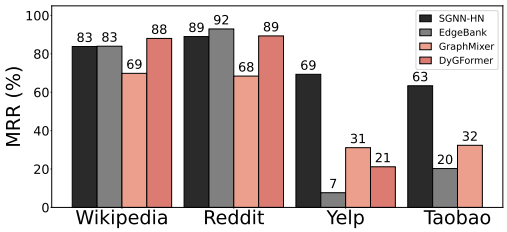

Figure 1: The MRR scores of three selected temporal GNNs and SGNN-HN on two existing datasets (Wikipedia, Reddit) and two recommendation datasets (Yelp and Taobao).

We conjecture that this is because existing datasets, e.g., Wikipedia and Reddit, contain excessive repetitions of historical edges compared to these evaluated recommendation datasets. Consequently, temporal GNNs tend to predict these repeated historical edges via memorizing or aggregating historical edges and perform well on existing datasets. To validate our assumption, we use existing temporal GNNs to predict both repeated and unseen edges and report their MRR scores separately across four widely used datasets: Wikipedia, Reddit, Social Evo. and Enron, following the experimental settings of Figure 1. The results in Figure 2 indicate a substantial prediction performance gap between historical and unseen edges, with differences reaching up to eightfold. This phenomenon implies that existing methods are effective on graphs dominated by repeated edges but fail to generalize to those that emphasize unseen edges. The underlying reason is probably that existing methods tend to rely on the information of historical neighbors, which limits their generalizability. Thus, they can only associate query nodes with their historical neighbors but fail on unseen edges.

Motivations. However, future links are typically not simple repetitions of historical ones in many real-world dynamic systems. Instead, the evolution of many dynamic systems often exhibits intricate *sequential dynamics*. For example, on an e-commerce platform, an *entity* (i.e., a customer) who has purchased a smartphone and a phone case is likely to buy a screen protector next. In this context, future interactions of entities typically involve new purchases rather than simply repeating past ones. Therefore, a model must capture the inherent sequential dynamics in these systems to accurately predict future links. *Capturing sequential dynamics involves modeling the evolution of the intentions of entities based on their historical interactions and forecasting unseen interactions.* However, we find that existing temporal GNNs struggle to effectively capture even simple sequential dynamics that exclude repeated edges, despite these dynamic patterns being frequently present in the training set. The observed cases are provided in later Section 3 and Figure 3. On the other hand, existing datasets often contain an excessive number of repeated edges, which undermines the critical aspect of complex sequential dynamics. Evaluating temporal GNN models solely based on these datasets cannot adequately assess their ability to capture complex sequential dynamics.

Contributions. To address this gap, we present the Temporal Graph Benchmark with Sequential Dynamics (TGB-Seq), a collection of new benchmark datasets designed to evaluate the systems' ability to capture complex sequential dynamics. TGB-Seq includes four widely-used recommendation datasets and four non-bipartite datasets derived from typical future link prediction scenarios that

inherently exhibit complex sequential dynamics, including a movie rating network (ML-20M), an e-commerce interaction network (Taobao), two business review networks (Yelp and GoogleLocal), two "Who-To-Follow" social networks (Flickr and YouTube), a citation network (Patent) and a web link network (WebLink). The TGB-Seq datasets are carefully curated to minimize repeated edges. Only Yelp and Taobao contain a small number of repeated edges with the natural behavior that users potentially review or click items multiple times. All TGB-Seq datasets are ensured with medium to large scale toward the practical situation, comprising millions to tens of millions of edges. Overall, we make the following contributions in this paper:

- We demonstrate that existing temporal GNNs fail to capture sequential dynamics in temporal graphs, limiting their generalizations to various real-world scenarios.
- We propose TGB-Seq, a collection of eight benchmark datasets for future link prediction, carefully curated from diverse application domains with intricate sequential dynamics. TGB-Seq focuses on evaluating temporal GNNs' capability to capture sequential dynamics and generalize to unseen edges, addressing the limitations of existing datasets that contain excessive repetitions of edges and overlook the intricate sequential dynamics present in real-world dynamic systems.
- Comprehensive evaluations on TGB-Seq reveal that existing temporal GNNs experience substantial performance declines compared to their impressive results on existing benchmarks. This observation underscores the limited ability of existing methods to capture complex sequential dynamics and demonstrate the distinguishing functionality of TGB-Seq in evaluating such ability.

2 Related Work

Temporal Graph Datasets and Benchmarks. Several studies (Poursafaei et al., 2022; Huang et al., 2024b) pointed out that existing benchmarks for dynamic graph learning lead to overly optimistic assessments of current approaches (Yu et al., 2023; Poursafaei et al., 2022; Huang et al., 2024b; Yu, 2023). Specifically, commonly used datasets, such as Reddit (Kumar et al., 2019), Wikipedia (Kumar et al., 2019), MOOC (Kumar et al., 2019), and LastFM (Kumar et al., 2019), suffer from inconsistent preprocessing and simplistic negative sampling, resulting in inflated performance metrics and unreliable comparisons. To address these issues, BenchTeMP (Huang et al., 2024a) provides a unified evaluation framework with consistent datasets and comprehensive performance metrics. Poursafaei et al. (Poursafaei et al., 2022) construct six dynamic graph datasets across diverse fields, such as politics, economics, and transportation, and introduce two negative sampling strategies to make the evaluations more challenging. TGB (Huang et al., 2024b) introduces several large-scale datasets for future link prediction, establishing a comprehensive benchmark with reproducible evaluation protocols to fairly assess several machine learning models on temporal graphs across multiple domains. Different from previous work that emphasizes expanding dataset diversity, scale, and evaluation protocols, we construct a collection of challenging benchmark datasets that originate from typical application scenarios of future link prediction, challenging the existing temporal GNNs with complex sequential dynamics inherently in these applications.

Temporal Graph GNNs for future link prediction. Future link prediction is a critical task in various dynamic systems, which aim to predict future interactions or relationships between entities based on historical data. To capture the evolution pattern, memory-based methods such as TGN (Rossi et al., 2020), Jodie (Kumar et al., 2019), DyRep (Trivedi et al., 2019), and APAN (Wang et al., 2021c), use dynamic memory modules to store and update node information during interactions, allowing for more effective modeling. On the other hand, approaches like TGAT (Xu et al., 2020), CAWN (Wang et al., 2021d), TCL (Wang et al., 2021a), GraphMixer (Cong et al., 2023), and DyGFormer (Yu et al., 2023), aggregate historical neighbor information directly during prediction without memory modules. These methods employ contrastive learning and Transformer-based techniques to capture evolving node interactions and temporal dependencies. The Hawkes process (Hawkes, 1971; Mei & Eisner, 2017) is another widely used technique for capturing the impacts of historical events on current events. TREND (Wen & Fang, 2022) utilizes the Hawkes process to model the exciting effects between sequential interactions and captures both individual and collective characteristics of events by integrating event and node dynamics. In contrast to prior methods that emphasize intricate module designs for modeling dynamic evolutions, SimpleDyG (Wu et al., 2024) draws inspiration from natural language processing (NLP) studies and models dynamic graphs as a sequence modeling problem, using a simple Transformer architecture without complex modifications. Poursafaei et al.

(2022) observe that edges reoccur over time in the existing datasets and propose a simple memory-based heuristic approach, EdgeBank, without any learnable components. This method predicts edges based solely on past observations, yet it demonstrated remarkable performance in current evaluations. This further highlights the need for more comprehensive benchmarks that assess models' ability to generalize to unseen edges, thereby ensuring robust performance in real-world scenarios.

Repeat and Exploration Behaviors in Recommender Systems. Repeat and exploration behaviors of users have been extensively studied in the context of recommender systems. Repeat behavior refers to users consistently engaging with items they have previously interacted with (i.e., the reoccurrence of seen edges in temporal graphs), while exploration behavior involves users discovering new items they have not interacted with before (i.e., the appearance of unseen edges for the first time in temporal graphs). Existing studies reveal an imbalance in accuracy and difficulty between repetition and exploration in sequential recommendation tasks (Li et al., 2023b). Several methods have been proposed to better address repeat and exploration behaviors, particularly in session-based or sequential recommendation (Ren et al., 2019; Chang et al., 2024), as well as next-basket recommendation (Li et al., 2024; 2023a). However, while repeat and exploration behaviors have been extensively studied in recommendation scenarios, their conclusions may not directly apply to the future link prediction task in temporal graphs due to differences in model design and task settings. For example, many recommendation methods are tailored for bipartite graphs without features or interaction timestamps, whereas temporal GNNs often focus on general graphs that may include single or multiple node types and fully leverage temporal graph information such as features and interaction timestamps. Therefore, it is essential to investigate repeat and exploration behaviors in the context of future link prediction tasks on temporal graphs and to comprehensively evaluate the performance of existing temporal GNNs in handling these challenges.

3 Task Formulation and Current Pitfalls

Temporal graphs represent entities in the dynamic systems as nodes and interactions among entities as edges. Each edge is labeled with a timestamp to indicate the time of interaction occurred. Existing studies mainly categorize temporal graphs into two types: continuous-time temporal graphs and discrete-time temporal graphs. In this paper, we focus on continuous-time temporal graphs since they better reflect how dynamic graphs form incrementally in real-world scenarios and discrete-time temporal graphs can be directly converted to continuous-time temporal graphs without information loss. Formally, a continuous-time temporal graph can be denoted as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the edge set $\mathcal{E}$ can be represented as a stream of timestamped edges, i.e., $\mathcal{E} = \{(s_0, d_0, t_0), (s_1, d_1, t_1), \cdots, (s_T, d_T, t_T)\}$ with $s_i, d_i \in \mathcal{V}$ representing the source and destination nodes, respectively. The t_i denotes the timestamp of the i-th edge with $t_0 \leq t_1 \leq \cdots \leq t_T$.

3.1 Future Link Prediction Formulation and Evaluation

Future Link Prediction. The task of future link prediction is formulated as predicting the existence of a link between two nodes at a given timestamp in literature (Kumar et al., 2019). Specifically, given a temporal graph $\mathcal{G}$, a query edge (s, d, t), and all edges appeared before time t, the model is required to predict the likelihood of the edge (s, d) appearing at time t. However, in real-world applications, the fundamental objective is to determine which entities the query entity is most likely to interact with. For instance, in the "Who-To-Follow" scenario within social networks, the task is to predict which users the query user is likely to follow next. The users with the highest predicted likelihood are then recommended to the query user. Given the high computational costs associated with calculating the likelihood of all potential entities in a large-scale graph, current literature in recommendation and knowledge graphs He et al. (2017); Kang & McAuley (2018); Teru et al. (2020) treats the future link prediction task as a ranking problem among multiple negative samples. Specifically, given a query edge (s, d, t), the model needs to rank the positive destination node d higher among the sampled k negative destinations based on the likelihood. The current temporal graph benchmark study, TGB (Huang et al., 2024b), adopts these settings and sets k to 20. In this work, we set k to 100 in our evaluation setting for a more robust evaluation.

Negative Sampling Strategies. Previous studies (Poursafaei et al., 2022; Huang et al., 2024b) leverage historical edges as negative samples to increase the difficulty for the models to predict the

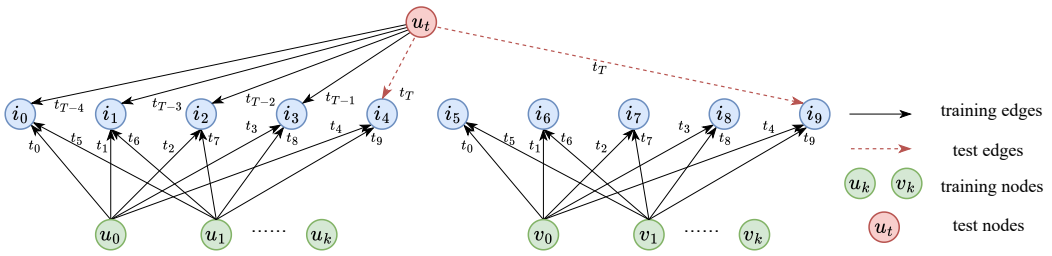

Figure 3: Toy example of sequential dynamics in a temporal graph. The bipartite graph consists of users and items. The first user in group u, u_0, interacts sequentially with items $\{i_k\}_{k=0}^{k=4}$ at time $\{t_k\}_{k=0}^{k=4}$, respectively. Similarly, the first user in group v, v_0, interacts sequentially with items $\{i_k\}_{k=5}^{k=9}$ at the same timestamps $\{t_k\}_{k=0}^{k=4}$ as u_0. The second users, u_1 and v_1, follow a similar interaction pattern but interact with items at different times compared to the first users. All other users interact with items in a comparable sequential manner. A test sample queries whether the test node will interact with i_4 or i_9 at time t_T, based on its four previous interactions from t_{T-4} to t_{T-1}.

potential link, based on the assumption that the positive edges are likely to be repetitions of historical edges. However, this is not always the case in the domains of our TGB-Seq datasets. Because historical edges are not likely to reoccur again in the future time. Thus, we randomly sample the negative destination nodes from all possible nodes, i.e., all nodes in non-bipartite datasets and all items in bipartite recommendation datasets.

Evaluation Metrics. Most existing studies leverage Area Under the Receiver Operating Characteristic curve (AUROC) and Average Precision (AP) for link prediction performance evaluation with a single negative sample, while Yang et al. (2015); Huang et al. (2024b) argue that they are not proper metrics for link prediction with multiple negative samples. Thus, we deploy the commonly used ranking metric, Mean Reciprocal Rank (MRR), to evaluate future link prediction, following (Cong et al., 2023; Huang et al., 2024b). The MRR score is defined as the average of the reciprocal ranks of the positive destination nodes among the negative destination nodes, and thus emphasizes the relative high likelihood of the positive edge among the candidate edges.

3.2 Current Pitfalls in Temporal GNNs

In this section, we aim to demonstrate that existing temporal GNNs are unable to capture even simple sequential dynamics. Figure 3 illustrates a toy example of sequential dynamics in a temporal graph. To empirically evaluate whether existing temporal GNNs can learn the simple sequential dynamics, we construct a dataset that mirrors the dynamics depicted in Figure 3. Specifically, the dataset consists of items $\{i_k\}_{k=0}^{k=9}$ and multiple nodes in both group u and group v, as in the toy example. To ensure that the sequential dynamics can be effectively modeled, the number of nodes in both group u and group v is set to 500. Each u_k interacts sequentially with items $\{i_k\}_{k=0}^{k=4}$, while each v_k interacts sequentially with items $\{i_k\}_{k=5}^{k=9}$. Note that each u_k and v_k always interact at the same timestamps as stated in the caption of Figure 3. Both nodes and edges lack features. The dataset is chronologically split into training set, validation set, and test set. The training set contains the complete interactions of 70% of the users in both group u and group v. Given the four historical interactions, i.e, $\{i_k\}_{k=0}^{k=3}$ or $\{i_k\}_{k=5}^{k=8}$, a temporal GNN model is required to predict the interaction likelihood of the query user with i_4 and i_9. Despite these straightforward sequential dynamics appearing commonly in the training set and thus considered as simple patterns, existing methods cannot correctly predict item i_4 instead of i_9 given that a test node has interacted with $\{i_k\}_{k=0}^{k=3}$ sequentially. We use the AP metric to evaluate nine temporal GNNs and SGNN-HN. All temporal GNNs achieve an AP score of approximately 50% as shown in Table 1, indicating that they cannot distinguish between i_4 and i_9.

Table 1: The AP metric on the toy example dataset. ℓ indicates the length of the temporal random walk of CAWN.

Method	AP (%)
JODIE	51.19 ± 0.32
DyRep	51.30 ± 0.27
TGAT	51.06 ± 0.23
TGN	51.25 ± 0.48
CAWN ($\ell = 2$)	50.00 ± 0.00
CAWN ($\ell = 3$)	52.80 ± 0.05
EdgeBank	50.00 ± 0.00
TCL	50.00 ± 0.00
GraphMixer	50.00 ± 0.00
DyGFormer	50.66 ± 0.50
SGNN-HN	100.00 ± 0.00

The shortcomings of current temporal GNNs in capturing sequential dynamics might relate to the functionality of their structures. Generally, the temporal GNN models can be partitioned into two components: i) a memory module to represent the interaction history of the nodes, and ii) an aggregation module to aggregate neighborhood information when predicting future interactions. Among the existing studies, the designed temporal GNN models might contain both or either of these two components. The limitations of each component in capturing sequential dynamics to distinguish items i_4 and i_9 are discussed as follows.

Notations. We denote the node feature of u as $\mathrm{F_n}(u)$, the edge feature of (u, v) as $\mathrm{F_e}(u, v)$, and the time difference between the interaction (u, v) and the query time as Δt. $\mathrm{Mem}(u)$ and $\mathrm{Emb}(u)$ denote the memory and embedding of node u, respectively. $\mathcal{N}_t^k(u)$ denotes the set of k-hop historical neighbors of node u before time t. In memory module, we use $\mathcal{N}_{\mathbf{b}}(u)$ to denote the set of u's neighbors within a batch.

Memory module. The memory module is designed to memorize the interaction history of nodes using a low-dimensional representation, called *memory*. Formally, a node s's memory is updated when processing a batch of incoming edges that involves s:

$$\mathrm{Mem}(s) = f_{\mathrm{mem}}\left(\mathrm{Mem}(s), \mathrm{F_n}(s), \{(\mathrm{Mem}(d), \mathrm{F_e}(s, d), \Delta t) \mid d \in \mathcal{N}_{\mathbf{b}}(s)\}\right), \tag{1}$$

where f_{mem} is typically an RNN model such as LSTM and GRU (Cho et al., 2014; Graves & Graves, 2012). The memory and feature of s, $\mathrm{Mem}(s)$ and $\mathrm{F_n}(s)$, are treated as the hidden state of the RNN. The information of incoming edges in the batch, $\{(\mathrm{Mem}(d), \mathrm{F_e}(s, d), \Delta t) \mid d \in \mathcal{N}_{\mathbf{b}}(s)\}$, serve as the input to RNN. Typically, only the last edge for each node in the batch is considered. In the toy example, items i_4 and i_9 always interact at the same timestamps and lack distinguishing features, resulting in identical memories for both items. Therefore, the memory module could not distinguish the difference between item i_4 and i_9 by only utilizing the information of historical edges.

Aggregation module. Given a prediction request for potential edges (s, d, t), the aggregation module aggregates the information from the historical neighbors of node s and d before time t to generate their current embeddings. The aggregation for node s is formulated as:

$$\mathrm{Emb}(s) = f_{\mathrm{agg}}\left(\mathrm{F_n}(s), \left\{(\mathrm{F_n}(d), \mathrm{F_e}(s, d), \Delta t) \mid d \in \mathcal{N}_t^k(s)\right\}\right), \tag{2}$$

where f_{agg} is commonly implemented as a Transformer (Vaswani, 2017) (e.g., in DyGFormer) or its variants, an MLP-mixer (Tolstikhin et al., 2021) (e.g., in GraphMixer), or time projection function (e.g., in JODIE). Note that if the memory is available, $\mathrm{F_n}(s)$ is replaced by a combination of the memory and node feature. In addition to interaction information, several studies compute the correlations between the neighborhoods of s and d to capture their structural and temporal dependencies:

$$\mathrm{Co}(s, d) = f_{\mathrm{co}}\left(\mathcal{N}_t^k(s), \mathcal{N}_t^k(d)\right). \tag{3}$$

In DyGFormer, f_{co} computes the number of common neighbors between s and d, i.e., the co-occurrence frequency between $\mathcal{N}_t^k(s)$ and $\mathcal{N}_t^k(d)$, while in CAWN, f_{co} leverages anonymous temporal random walk to establish the correlation between network motifs of s and d. For computational efficiency, aggregation modules typically consider only one-hop neighbors, i.e., $k = 1$ in both Equation (2) and Equation (3).

Such aggregation modules are insufficient in capturing the sequential dynamics in our toy dataset. The underlying issue is similar to that of memory modules: a node is represented solely by its features and interaction timestamps. However, i_4 and i_9, their one-hop neighbors $\{u_k\}$ and $\{v_k\}$, interact in a similar manner at identical timestamps, respectively. As a result, the aggregation modules generate the same embeddings for i_4 and i_9, as well as for $\{u_k\}$ and $\{v_k\}$, respectively. While computing correlations between the source and destination nodes may seem helpful, both DyGFormer and CAWN fail in the toy example. DyGFormer's f_{co} is ineffective since the source and destination nodes have no common neighbors. Though CAWN's f_{co} employ a sophisticated anonymous random walk technique, it fails to distinguish between i_4 and i_9 because their one-hop neighborhoods mirror each other. Therefore, the aggregation modules cannot capture even the simple sequential dynamics in the toy example.

Aggregation module with high-order historical neighbors. Leveraging high-order historical neighbor information can modestly enhance the capture of sequential dynamics. For example, extending the length of the temporal random walk from 2-hop to 3-hop in CAWN enables the incorporation of

Table 2: Statistics of TGB-Seq datasets.

Dataset	Nodes (users/items)	Edges	Timestamps	Repeat ratio(%)	Density(%)	Bipartite	Domain
ML-20M	100,785/9,646	14,365,034	9,864,096	0	1.48×10^{0}	✓	Movie rating
Taobao	760,617/863,016	16,447,721	124,412	16	2.51×10^{-3}	✓	E-commerce interaction
Yelp	1,338,688/405,081	18,727,939	13,627,978	26	3.45×10^{-3}	✓	Business review
GoogleLocal	206,244/267,336	1,870,421	1,727,614	0	3.39×10^{-3}	✓	Business review
Flickr	105,974	6,084,535	111	0	5.42×10^{-2}	×	Who-To-Follow
YouTube	388,066	3,288,028	203	0	2.18×10^{-3}	×	Who-To-Follow
Patent	1,810,841	10,818,819	1,468	0	3.30×10^{-4}	×	Citation
WikiLink	1,358,870	34,163,774	2,198	0	1.85×10^{-3}	×	Web link

higher-order temporal and structural entangled information, resulting in a slight performance improvement from 50.00% to 52.80%. The limited gain arises because an increased number of high-order neighbors introduces excessive noise. Consequently, CAWN is unable to effectively differentiate the subtle differences between the local structures of (u_k, i_4) and (u_k, i_9). Furthermore, utilizing high-order information results in substantial computational resource consumption (Besta et al., 2024). CAWN encounters memory issues on a GPU with 80GB of memory when the walk length is extended to four, even on this small graph. Therefore, effectively capturing intricate sequential dynamics through high-order neighbors appears to be feasible; nonetheless, it remains an open problem.

In summary, neither the memory module nor the aggregation module can distinguish items i_4 and i_9 in the toy example. Consequently, temporal GNNs that incorporate either or both of these modules are unable to effectively capture the simple sequential dynamics, resulting in suboptimal performance on the toy dataset. These findings suggest that current methods are insufficient for future link prediction tasks that involve complex sequential dynamics. This highlights the urgent need to develop robust temporal Graph Neural Networks (GNNs) and establish new benchmark datasets to effectively evaluate the ability of temporal GNNs to capture sequential dynamics.

4 Proposed Datasets

Our proposed TGB-Seq aims to challenge temporal GNNs with intricate sequential dynamics that are inherently exhibited in various real-world dynamic systems. TGB-Seq comprises eight temporal graph datasets, including four bipartite datasets derived from recommender systems and four non-bipartite datasets curated from diverse application domains. All TGB-Seq datasets focus on interactions between entities and exclude node and edge features. Table 2 presents the statistics of TGB-Seq datasets. Besides, we also provide a selected list of datasets used for continuous-time temporal graph learning in Table 5 for comparison.

The most distinguishable feature of TGB-Seq datasets is *the low repeat ratio*, where only the Yelp and Taobao datasets contain repeated edges due to the natural behavior of users who may review or click on items multiple times. The repeat ratio r is defined as the portion of the number of repeated edges to the total number of edges in the dataset, i.e., $r = \frac{|\mathcal{E}_{\text{seen}}|}{|\mathcal{E}|}$, where an edge $e_i = (s_i, d_i, t_i) \in \mathcal{E}_{\text{seen}}$ if there exists an edge $e_j = (s_j, d_j, t_j)$ and satisfies that $s_i = s_j, d_i = d_j, t_j < t_i$.

Remark. The phenomenon of existing datasets that contain excessive repeated edges and its impact on overly optimistic evaluations has been highlighted in previous studies. To address the issues, these studies challenge the existing temporal GNNs with new evaluation protocols and new datasets from diverse domains. Specifically, Poursafaei et al. (2022) proposes a historical negative sampling strategy to challenge existing methods with hard negative samples, and Huang et al. (2024b) further employs multiple negative sampling strategies. Both of them propose new datasets from diverse domains and of diverse scales. However, most of the proposed datasets still contain numerous repeated edges as shown in Table 5. In contrast, we address this issue by proposing new challenging datasets curated to minimize repeated edges. Our TGB-Seq datasets emphasize the intricate sequential dynamics, a key characteristic of many real-world applications. Consequently, TGB-Seq datasets provide a robust benchmark for evaluating the ability of temporal GNNs to capture sequential dynamics and generalize to unseen edges, a capability that is often lacking in existing benchmark datasets.

In addition to the low repeat ratio, another notable feature of the TGB-Seq datasets is their origin in *diverse domains that represent typical real-world applications of future link prediction*. Besides classical applications like recommendations, the proposed non-bipartite datasets also represent

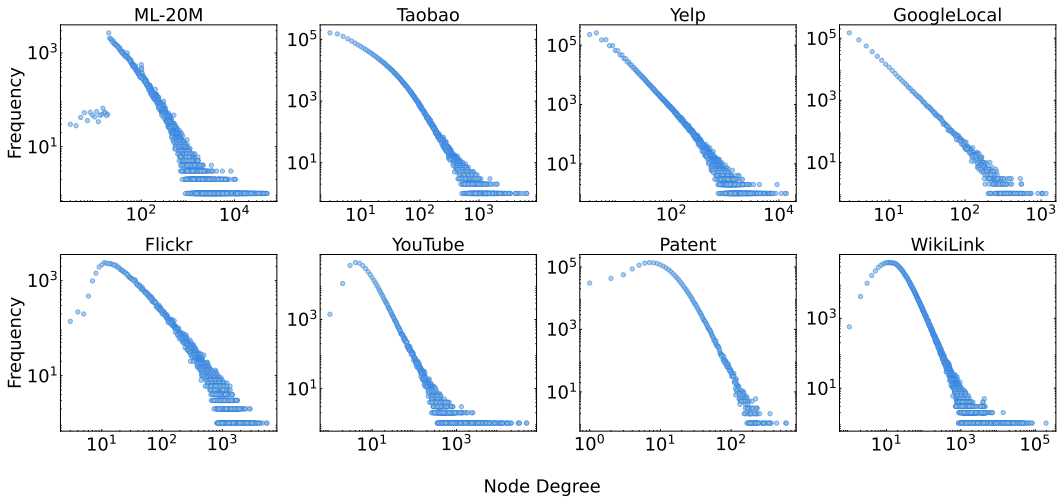

Figure 4: Distribution of node degree on our TGB-Seq datasets.

fundamental applications in real-world contexts. As a crucial task for online social networking platforms, "Who-To-Follow" aims to recommend a list of users that a given user may be interested in following (Gupta et al., 2013). Effective prediction of relevant connections between users can significantly enhance user experience by fostering engagement and interaction within the platform. Moreover, future link prediction in citation networks and web link networks can be applied to knowledge graph completion, thereby enriching knowledge representations and enabling more comprehensive information retrieval (Wang et al., 2021b).

Furthermore, the TGB-Seq datasets exhibit several key attributes of real-world networks. Figure 4 illustrates the node degree distribution of the TGB-Seq datasets, all of which *follow a power-law distribution*, a common characteristic of real-world networks (Barabási, 2013). This power-law behavior indicates that while a few hub nodes have a high degree of connections, the majority of nodes possess significantly fewer connections. Consequently, TGB-Seq is *highly sparse*, exhibiting low density, as shown in Table 2. Each TGB-Seq dataset is *of medium to large scale* and contains millions or tens of millions of edges, which aligns with typical real-world networks.

Dataset preprocessing. We split the datasets chronologically into training, validation, and test sets with a ratio of 70%/15%/15%. In the training sets, we retain only nodes with a degree of at least three. Besides, only nodes appearing in the training set are included in the validation and test sets. These settings are designed to mitigate the effects of cold-start nodes and the high sparsity of the datasets on the evaluation. Descriptions of TGB-Seq datasets are shown below.

ML-20M [2] is a widely used benchmark dataset in recommendation research, derived from the MovieLens website. It contains movie rating data, where each record includes the rating score of a user, ranging from 1 to 5, for a specific movie along with the timestamp of the rating. While the ratings represent explicit feedback, we transform this data into implicit feedback for our analysis, following He et al. (2017). Consequently, the ML-20M network is represented as a bipartite graph where users and movies serve as nodes, and an edge represents a user's rating of a movie at a given time. The task of ML-20M and the following recommendation datasets is to predict whether a given user will interact with a given item at a given time.

Taobao [3] (Zhu et al., 2018; 2019; Zhuo et al., 2020) is a user behavior dataset derived from the e-commerce platform Taobao. It contains user click data on products from November 25, 2017, to December 3, 2017. The dataset is a bipartite graph where users and products are nodes, and an edge represents a user's click on a product at a given time.

Yelp [4] is a business review dataset sourced from Yelp, a prominent platform for business recommendations, including restaurants, bars, and beauty salons. It contains user reviews of businesses from

[2] https://grouplens.org/datasets/movielens/20m/
[3] https://tianchi.aliyun.com/dataset/dataDetail?dataId=649
[4] https://www.yelp.com/dataset

2018 to 2022. The dataset is a bipartite graph where users and businesses are nodes, and an edge represents a user's review of a business at a given time.

GoogleLocal (Li et al., 2022; Yan et al., 2023) is a business review dataset derived from Google Maps, with a smaller scale compared to Yelp. It contains user reviews and ratings of local businesses. Following the settings for the ML-20M dataset, we treat these ratings as implicit feedback. Similar to the Yelp dataset, the GoogleLocal dataset is a bipartite graph where users and businesses are nodes, and an edge indicates a user's review of a business at a given time.

Flickr (Cha et al., 2009) is a "Who-To-Follow" social network dataset derived from Flickr, a photo-sharing platform with social networking features. The dataset was crawled daily from November 2 to December 3, 2006, and from February 3 to March 18, 2007 by Cha et al. (2009). It is estimated to represent 25% of the entire Flickr network. The Flickr dataset is a non-bipartite graph where users are nodes, and an edge represents the friendship established between users at a given time. The task for the "Who-To-Follow" datasets, including Flickr and YouTube, is to predict whether a given user will follow another specified user at a particular time.

YouTube (Mislove et al., 2007) is another "Who-To-Follow" social network dataset derived from YouTube, a video-sharing platform that includes a user subscription network. Similar to Flickr, the YouTube dataset is a non-bipartite graph where users are nodes, and an edge indicates the subscription of a user to another user at a given time.

Patent (Hall et al., 2001) is a citation network dataset of U.S. patents, capturing the citation relationships between patents from 1963 to 1999. The dataset is organized as a non-bipartite graph where patents are nodes, and an edge represents a citation made by one patent to another at the time of publication. The task for the Patent dataset is to predict whether a given patent will cite another given patent, given several of their established citations.

WikiLink (Boldi et al., 2004; 2011; Boldi & Vigna, 2004) is a web link network dataset derived from Wikipedia, containing the hyperlink relationships between Wikipedia pages. This dataset is a non-bipartite graph, where pages are nodes and edges indicate hyperlinks established from one page to another at a given time. The task for WikiLink is to predict whether a given page will link to another given page at a given time.

5 EXPERIMENTS

In this section, we evaluate the performance of existing temporal GNNs on the TGB-Seq datasets. The selected temporal GNN models includes JODIE (Kumar et al., 2019), DyRep (Trivedi et al., 2019), TGAT (Xu et al., 2020), TGN (Rossi et al., 2020), CAWN (Wang et al., 2021d), EdgeBank (Poursafaei et al., 2022), TCL (Wang et al., 2021a), GraphMixer (Cong et al., 2023), and DyGFormer (Yu et al., 2023). The descriptions of these methods are provided in Appendix C. We employ the DyGLib Yu et al. (2023) framework to conduct the experiments. We limit the running time of each method to 48 hours and omit the methods that require more than 24 hours to finish one training epoch, which are denoted as OOT (out of time). Each result is the average of three runs with different random seeds with reported standard deviation.

Implementation details. We follow Rossi et al. (2020) to set a relatively small batch size to ensure timely updates for the memory module. Specifically, we set the batch size to 200 for the GoogleLocal dataset across all methods. For larger datasets, however, a batch size of 200 is too small and would incur unacceptable training costs for most methods. Thus, we increase the batch size to 400 for all other datasets to accelerate the training process. Following DyGFormer, we use a learning rate of 1e-4 across all methods and datasets. A grid search is performed to tune the hyper-parameters of each method on the validation set. Detailed configurations are provided in Appendix B.1.

5.1 FUTURE LINK PREDICTION PERFORMANCE

Performance on recommendation datasets. Table 3 presents the results on four recommendation datasets, ML-20M, Taobao, Yelp, and GoogleLocal. We can find that existing temporal GNNs underperform on these datasets, with a large margin compared to SGNN-HN, one of the state-of-the-art methods for sequential recommendation. Such a phenomenon highlights the limitations of current

Table 3: MRR of nine popular temporal GNN methods and SGNN-HN on four recommendation datasets and two previously established datasets (e.g., Wikipedia and Reddit). "OOT" denotes that the method failed to complete one epoch of training within 24 hours.

Datasets	ML-20M	Taobao	Yelp	GoogleLocal	Wikipedia	Reddit
JODIE (Kumar et al., 2019)	OOT	OOT	OOT	36.84 ± 1.87	59.18 ± 1.90	72.39 ± 2.53
DyRep (Trivedi et al., 2019)	OOT	OOT	OOT	28.77 ± 3.93	53.82 ± 1.57	72.39 ± 0.96
TGAT (Xu et al., 2020)	13.27 ± 0.56	30.29 ± 0.20	20.57 ± 0.47	19.49 ± 0.22	70.37 ± 0.24	75.25 ± 0.13
TGN (Rossi et al., 2020)	OOT	OOT	OOT	51.59 ± 0.62	78.36 ± 0.99	79.00 ± 0.93
CAWN (Wang et al., 2021d)	17.04 ± 0.30	41.70 ± 0.30	25.87 ± 0.42	18.96 ± 0.06	88.16 ± 0.22	88.65 ± 0.11
EdgeBank (Poursafaei et al., 2022)	0.99 ± 0.00	20.24 ± 0.00	7.66 ± 0.00	0.99 ± 0.00	83.98 ± 0.00	92.97 ± 0.00
TCL (Wang et al., 2021a)	17.20 ± 0.04	38.65 ± 0.66	18.05 ± 1.61	18.90 ± 0.17	72.83 ± 0.68	70.82 ± 2.02
GraphMixer (Cong et al., 2023)	21.69 ± 0.37	32.36 ± 0.09	31.12 ± 0.24	20.32 ± 0.23	69.85 ± 0.46	68.42 ± 0.31
DyGFormer (Yu et al., 2023)	OOT	OOT	21.17 ± 0.21	18.89 ± 0.02	88.04 ± 0.33	89.34 ± 0.15
SGNN-HN (Pan et al., 2020)	34.80 ± 0.04	63.37 ± 0.06	69.34 ± 0.09	64.59 ± 0.23	83.83 ± 0.55	89.01 ± 0.17

Table 4: MRR score of nine popular temporal GNNs on four non-bipartite datasets.

Datasets	Flickr	Youtube	Patent	WikiLink
JODIE (Kumar et al., 2019)	43.38 ± 0.47	OOT	OOT	OOT
DyRep (Trivedi et al., 2019)	38.56 ± 0.31	OOT	OOT	OOT
TGAT (Xu et al., 2020)	15.44 ± 1.37	46.10 ± 2.30	9.40 ± 1.87	12.30 ± 2.41
TGN (Rossi et al., 2020)	44.64 ± 2.25	OOT	OOT	OOT
CAWN (Wang et al., 2021d)	15.87 ± 2.84	43.21 ± 1.02	11.11 ± 0.42	18.51 ± 6.90
EdgeBank (Poursafaei et al., 2022)	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00	0.99 ± 0.00
TCL (Wang et al., 2021a)	24.25 ± 2.38	46.68 ± 0.17	10.33 ± 0.25	40.00 ± 0.69
GraphMixer (Cong et al., 2023)	30.24 ± 0.49	54.16 ± 0.03	17.12 ± 0.39	47.63 ± 0.05
DyGFormer (Yu et al., 2023)	17.21 ± 3.20	41.59 ± 2.06	11.94 ± 0.75	39.59 ± 0.29

temporal GNNs in capturing intricate sequential dynamics present in various real-world applications, which may be attributed to their model architecture that heavily relies on historical edges.

Moreover, we observe that the performance of these methods varies significantly from that of existing datasets. For instance, DyGFormer achieves the best performance on Wikipedia, but underperforms all other methods except for EdgeBank on GoogleLocal. EdgeBank is a heuristic method that memorizes historical edges, achieving the best performance on Reddit. However, it struggles with datasets that exclude repeated edges, such as ML-20M and GoogleLocal. TGN shows a significant performance gap compared to the best method on Wikipedia and Reddit, yet it achieves the highest performance on GoogleLocal. These observations suggest that TGB-Seq effectively evaluates the capabilities of temporal GNNs across different dimensions, rather than focusing on a specific ability.

Furthermore, the performance of these methods varies significantly across different datasets. TCL outperforms other methods on Taobao, but suffers a large performance degradation on ML-20M and Yelp. GraphMixer achieves the best performance on ML-20M and Yelp, but exhibits a large performance gap with TCL on Taobao. These significant variations indicate that existing methods cannot effectively predict future links for various real-world applications and emphasize the necessity of TGB-Seq datasets that evaluate the ability of temporal GNNs to capture the sequential dynamics.

Performance on non-bipartite datasets. Table 4 presents the results on four non-bipartite datasets, Flickr, YouTube, Patent, and WikiLink. Compared to the recommendation datasets, the performance of temporal GNNs on these non-bipartite datasets is even worse. For example, on the Patent dataset, none of these methods achieves an MRR score higher than 20%. Moreover, the performance of any specific method varies significantly across these datasets. For instance, GraphMixer achieves an MRR score of approximately 54% on YouTube, but only 17.12% on the Patent dataset. These observations underscore the necessity of a diverse range of benchmark datasets from various domains to effectively evaluate temporal GNNs, particularly for applications involving complex sequential dynamics. This aligns with our objectives in proposing TGB-Seq.

5.2 Training cost

To comprehensively study the efficiency of existing temporal GNNs, we select three datasets with various sizes of edge sets and report the average training cost per epoch of the corresponding approach.

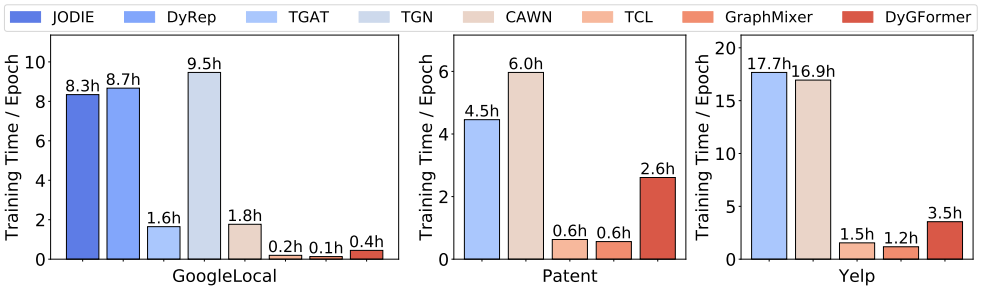

Figure 5: The average training cost per epoch of nine popular temporal GNN methods on GoogleLocal, Patent, and Yelp datasets, which consist of 1.87M, 10M, and 18.7M edges, respectively.

Figure 5 illustrates the results on the GoogleLocal, Patent, and Yelp datasets, where the methods that cannot finish one epoch in 24 hours are omitted.

We find that the training time of memory-based methods (JODIE, DyRep, and TGN) is significantly longer than that of aggregation-only methods (the remaining). The memory-based methods can only finish training on the GoogleLocal dataset, a relatively smaller dataset among TGB-Seq. This limitation might be attributed to the time-consuming memory updates. TGAT and CAWN are more efficient than JODIE, DyRep, and TGN, but one epoch on the Patent and Yelp datasets still requires a long time to finish. This is because their complex aggregation modules require computing multi-hop neighbor embeddings with self-attention mechanisms and temporal random walk, respectively. DyGFormer is more efficient than the above methods since it does not require the memory module and only aggregates the information of the first-hop neighbors. However, calculating the co-occurrence frequency of neighbors in DyGFormer is costly, taking 3.5 hours to finish a single epoch on the Yelp dataset. Among the investigated approaches, TCL and GraphMixer are the most efficient methods, as they only require simple aggregation operations.

These observations indicate that the memory and complex aggregation modules would significantly increase the training cost of existing temporal GNNs. As shown in Table 3 and Table 4, simple methods like TCL and GraphMixer can be more efficient in training, but they cannot achieve comparable performance with memory-based methods. This investigation suggests that achieving both efficiency and effectiveness in temporal GNNs simultaneously remains an open problem, underscoring the distinctive capability of TGB-Seq for comprehensive evaluations of these models.

6 Conclusion

In this paper, we demonstrate that current temporal GNNs struggle to capture intricate sequential dynamics that are inherently present in real-world dynamic systems, thereby limiting their abilities to generalize across various real-world applications of future link prediction. However, existing datasets often feature excessively repeated edges and thus are inadequate for evaluating such abilities of temporal GNNs. To address this gap, we propose TGB-Seq, a new challenging benchmark for temporal graph neural networks. TGB-Seq comprises eight datasets meticulously curated from diverse application domains characterized by complex sequential dynamics. Comprehensive evaluations on TGB-Seq reveal that existing temporal GNNs experience significant performance declines compared to their strong results on established benchmarks. This finding underscores the limitations of current methods' abilities in capturing intricate sequential dynamics and highlights the distinctive value of TGB-Seq in assessing these capabilities.

References

Ting Bai, Youjie Zhang, Bin Wu, and Jian-Yun Nie. Temporal graph neural networks for social recommendation. In *2020 IEEE International Conference on Big Data (Big Data)*, pp. 898–903. IEEE, 2020.

Albert-László Barabási. Network science. *Philosophical Transactions of the Royal Society A: Mathematical, Physical and Engineering Sciences*, 371(1987):20120375, 2013.

Maciej Besta, Afonso Claudino Catarino, Lukas Gianinazzi, Nils Blach, Piotr Nyczyk, Hubert Niewiadomski, and Torsten Hoefler. Hot: Higher-order dynamic graph representation learning with efficient transformers. In *Learning on Graphs Conference*, pp. 15–1. PMLR, 2024.

Paolo Boldi and Sebastiano Vigna. The webgraph framework i: compression techniques. In *Proceedings of the 13th international conference on World Wide Web*, pp. 595–602, 2004.

Paolo Boldi, Bruno Codenotti, Massimo Santini, and Sebastiano Vigna. Ubicrawler: A scalable fully distributed web crawler. *Software: Practice and Experience*, 34(8):711–726, 2004.

Paolo Boldi, Marco Rosa, Massimo Santini, and Sebastiano Vigna. Layered label propagation: A multiresolution coordinate-free ordering for compressing social networks. In *Proceedings of the 20th international conference on World Wide Web*, pp. 587–596, 2011.

Meeyoung Cha, Alan Mislove, and Krishna P Gummadi. A measurement-driven analysis of information propagation in the flickr social network. In *Proceedings of the 18th international conference on World wide web*, pp. 721–730, 2009.

Haw-Shiuan Chang, Nikhil Agarwal, and Andrew McCallum. To copy, or not to copy; that is a critical issue of the output softmax layer in neural sequential recommenders. In *Proceedings of the 17th ACM International Conference on Web Search and Data Mining*, WSDM '24, pp. 67–76, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400703713. doi: 10.1145/3616855.3635755. URL https://doi.org/10.1145/3616855.3635755.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pp. 1724–1734, 2014. doi: 10.3115/V1/D14-1179. URL https://doi.org/10.3115/v1/d14-1179.

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

Nur Nasuha Daud, Siti Hafizah Ab Hamid, Muntadher Saadoon, Firdaus Sahran, and Nor Badrul Anuar. Applications of link prediction in social networks: A review. *Journal of Network and Computer Applications*, 166:102716, 2020.

Aswathy Divakaran and Anuraj Mohan. Temporal link prediction: A survey. *New Generation Computing*, 38(1):213–258, 2020.

Alex Graves and Alex Graves. Long short-term memory. *Supervised sequence labelling with recurrent neural networks*, pp. 37–45, 2012.

Pankaj Gupta, Ashish Goel, Jimmy Lin, Aneesh Sharma, Dong Wang, and Reza Zadeh. Wtf: The who to follow service at twitter. In *Proceedings of the 22nd international conference on World Wide Web*, pp. 505–514, 2013.

Bronwyn H Hall, Adam B Jaffe, and Manuel Trajtenberg. The nber patent citation data file: Lessons, insights and methodological tools, 2001.

Alan G. Hawkes. Spectra of some self-exciting and mutually exciting point processes. *Biometrika*, 58:83–90, 1971.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pp. 173–182, 2017.

Qiang Huang, Xin Wang, Susie Xi Rao, Zhichao Han, Zitao Zhang, Yongjun He, Quanqing Xu, Yang Zhao, Zhigao Zheng, and Jiawei Jiang. Benchtemp: A general benchmark for evaluating temporal graph neural networks. In *2024 IEEE 40th International Conference on Data Engineering (ICDE)*, pp. 4044–4057. IEEE, 2024a.

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, 2024b.

Wang-Cheng Kang and Julian McAuley. Self-attentive sequential recommendation. In *2018 IEEE international conference on data mining (ICDM)*, pp. 197–206. IEEE, 2018.

Seyed Mehran Kazemi, Rishab Goel, Kshitij Jain, Ivan Kobyzev, Akshay Sethi, Peter Forsyth, and Pascal Poupart. Representation learning for dynamic graphs: A survey. *J. Mach. Learn. Res.*, 21: 70:1–70:73, 2020. URL https://jmlr.org/papers/v21/19-447.html.

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, KDD 2019, Anchorage, AK, USA, August 4-8, 2019*, pp. 1269–1278, 2019. doi: 10.1145/3292500.3330895. URL https://doi.org/10.1145/3292500.3330895.

Jiacheng Li, Jingbo Shang, and Julian J. McAuley. Uctopic: Unsupervised contrastive learning for phrase representations and topic mining. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pp. 6159–6169, 2022. doi: 10.18653/V1/2022.ACL-LONG.426. URL https://doi.org/10.18653/v1/2022.acl-long.426.

Jiayu Li, Aixin Sun, Weizhi Ma, Peijie Sun, and Min Zhang. Right tool, right job: Recommendation for repeat and exploration consumption in food delivery. In *Proceedings of the 18th ACM Conference on Recommender Systems*, RecSys '24, pp. 643–653, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400705052. doi: 10.1145/3640457.3688119. URL https://doi.org/10.1145/3640457.3688119.

Ming Li, Mozhdeh Ariannezhad, Andrew Yates, and Maarten de Rijke. Masked and swapped sequence modeling for next novel basket recommendation in grocery shopping. In *Proceedings of the 17th ACM Conference on Recommender Systems, RecSys 2023, Singapore, Singapore, September 18-22, 2023*, pp. 35–46, 2023a. doi: 10.1145/3604915.3608803. URL https://doi.org/10.1145/3604915.3608803.

Ming Li, Ali Vardasbi, Andrew Yates, and Maarten de Rijke. Repetition and exploration in sequential recommendation. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '23, pp. 2532–2541, New York, NY, USA, 2023b. Association for Computing Machinery. ISBN 9781450394086. doi: 10.1145/3539618.3591914. URL https://doi.org/10.1145/3539618.3591914.

Hongyuan Mei and Jason M Eisner. The neural hawkes process: A neurally self-modulating multivariate point process. *Advances in neural information processing systems*, 30, 2017.

Alan Mislove, Massimiliano Marcon, Krishna P Gummadi, Peter Druschel, and Bobby Bhattacharjee. Measurement and analysis of online social networks. In *Proceedings of the 7th ACM SIGCOMM conference on Internet measurement*, pp. 29–42, 2007.

Zhiqiang Pan, Fei Cai, Wanyu Chen, Honghui Chen, and M. de Rijke. Star graph neural networks for session-based recommendation. *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, 2020. URL https://api.semanticscholar.org/CorpusID:221339954.

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.

Nitendra Rajput and Karamjit Singh. Temporal graph learning for financial world: Algorithms, scalability, explainability & fairness. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4818–4819, 2022.

Pengjie Ren, Zhumin Chen, Jing Li, Zhaochun Ren, Jun Ma, and Maarten de Rijke. Repeatnet: A repeat aware neural recommendation machine for session-based recommendation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pp. 4806–4813, 2019. doi: 10.1609/AAAI.V33I01.33014806. URL https://doi.org/10.1609/aaai.v33i01.33014806.

Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. Temporal graph networks for deep learning on dynamic graphs. In *ICML 2020 Workshop on Graph Representation Learning*, 2020.

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. doi: 10.1109/ACCESS.2021.3082932. URL https://doi.org/10.1109/ACCESS.2021.3082932.

Komal Teru, Etienne Denis, and Will Hamilton. Inductive relation prediction by subgraph reasoning. In *International Conference on Machine Learning*, pp. 9448–9457. PMLR, 2020.

Ilya O Tolstikhin, Neil Houlsby, Alexander Kolesnikov, Lucas Beyer, Xiaohua Zhai, Thomas Unterthiner, Jessica Yung, Andreas Steiner, Daniel Keysers, Jakob Uszkoreit, et al. Mlp-mixer: An all-mlp architecture for vision. *Advances in neural information processing systems*, 34:24261–24272, 2021.

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

A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*, 2017.

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.

Meihong Wang, Linling Qiu, and Xiaoli Wang. A survey on knowledge graph embeddings for link prediction. *Symmetry*, 13(3):485, 2021b.

Xuhong Wang, Ding Lyu, Mengjian Li, Yang Xia, Qi Yang, Xinwen Wang, Xinguang Wang, Ping Cui, Yupu Yang, Bowen Sun, et al. Apan: Asynchronous propagation attention network for real-time temporal graph embedding. In *Proceedings of the 2021 international conference on management of data*, pp. 2628–2638, 2021c.

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

Zhihao Wen and Yuan Fang. Trend: Temporal event and node dynamics for graph representation learning. In *Proceedings of the ACM Web Conference 2022*, pp. 1159–1169, 2022.

Yuxia Wu, Yuan Fang, and Lizi Liao. On the feasibility of simple transformer for dynamic graph modeling. In *Proceedings of the ACM on Web Conference 2024*, pp. 870–880, 2024.

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

An Yan, Zhankui He, Jiacheng Li, Tianyang Zhang, and Julian McAuley. Personalized showcases: Generating multi-modal explanations for recommendations. In *Proceedings of the 46th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 2251–2255, 2023.

Yang Yang, Ryan N Lichtenwalter, and Nitesh V Chawla. Evaluating link prediction methods. *Knowledge and Information Systems*, 45:751–782, 2015.

Le Yu. An empirical evaluation of temporal graph benchmark. *arXiv preprint arXiv:2307.12510*, 2023.

Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. Towards better dynamic graph learning: New architecture and unified library. In *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023. URL http://papers.nips.cc/paper_files/paper/2023/hash/d611019afba70d547bd595e8a4158f55-Abstract-Conference.html.

Yanping Zheng, Lu Yi, and Zhewei Wei. A survey of dynamic graph neural networks. *arXiv preprint arXiv:2404.18211*, 2024.

Han Zhu, Xiang Li, Pengye Zhang, Guozheng Li, Jie He, Han Li, and Kun Gai. Learning tree-based deep model for recommender systems. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 1079–1088, 2018.

Han Zhu, Daqing Chang, Ziru Xu, Pengye Zhang, Xiang Li, Jie He, Han Li, Jian Xu, and Kun Gai. Joint optimization of tree-based index and deep model for recommender systems. *Advances in Neural Information Processing Systems*, 32, 2019.

Jingwei Zhuo, Ziru Xu, Wei Dai, Han Zhu, Han Li, Jian Xu, and Kun Gai. Learning optimal tree models under beam search. In *International Conference on Machine Learning*, pp. 11650–11659. PMLR, 2020.

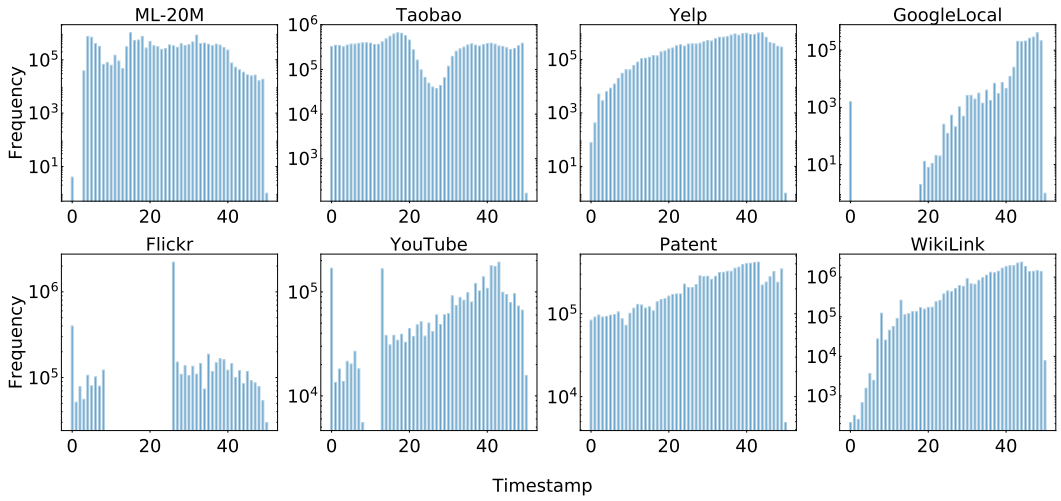

Figure 6: The variation of the number of edges in each discretized timestamp in the proposed TGB-Seq datasets.

A DATASETS

We provide a selected list of commonly used datasets for continuous-time temporal graph learning in Table 5 for reference. We also present the variation in the number of edges over discretized timestamps in the TGB-Seq datasets, as shown in Figure 6. Most datasets exhibit an increasing trend in the number of edges as time progresses, while the Taobao dataset demonstrates periodic fluctuations, with phases of increase and decrease. These fluctuations are likely attributed to shopping festivals and other popularity-driven factors. The Flickr and YouTube datasets contain periods without any edges appearing due to the crawling process of the original datasets.

Preprocessing of the Patent dataset. In the Patent dataset, all citations of one patent are labeled with the same timestamp, specifically the publication time of the patent. To address this, we carefully select test samples to ensure that each patent has prior citations, allowing temporal GNNs to leverage these historical edges for future link prediction. Specifically, we choose not to validate or test the first 50% of citations for the patents included in the validation and test sets; these citations serve solely as historical edges and are not used for model training. The remaining 50% of citations are then evenly divided into validation and test samples. Although the citations of a patent occur simultaneously at the publication time, temporal GNNs can utilize the relative publication times of these patents and their neighbors to capture inherent research trends, thereby enhancing future link prediction performance. The preprocessing code for the Patent dataset, along with other datasets, is provided in `https://anonymous.4open.science/r/TGB-Seq-3F23`.

A.1 DATASET LICENSES AND DOWNLOAD LINKS

In this section, we provide dataset licenses and download links as follows.

ML-20M: The data set may be used for any research purposes under the following conditions: (a) The user may not state or imply any endorsement from the University of Minnesota or the GroupLens Research Group. (b) The user must acknowledge the use of the data set in publications resulting from the use of the data set. (c) The user may not redistribute the data without separate permission. (d) The user may not use this information for any commercial or revenue-bearing purposes without first obtaining permission from a faculty member of the GroupLens Research Project at the University of Minnesota. (e) The executable software scripts are provided "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. The entire risk as to the quality and performance of them is with you. Should the program prove defective, you assume the cost of all necessary servicing, repair or correction. (f) In no event shall the University of Minnesota, its affiliates or employees be liable

Table 5: A selected list of datasets used for continuous-time temporal graph learning.

Dataset	Nodes (users/items)	Edges	Timestamps	Repeat ratio(%)	Density(%)	Bipartite	Domain
ML-20M	100,785/9,646	14,365,034	9,864,096	0	1.48×10^{0}	✓	Movie rating
Taobao	760,617/863,016	16,447,721	124,412	16	2.51×10^{-3}	✓	E-commerce interaction
Yelp	1,338,688/405,081	18,727,939	13,627,978	26	3.45×10^{-3}	✓	Business review
GoogleLocal	206,244/267,336	1,870,421	1,727,614	0	3.39×10^{-3}	✓	Business review
Flickr	105,974	6,084,535	111	0	5.42×10^{-2}	×	Who-To-Follow
YouTube	388,066	3,288,028	203	0	2.18×10^{-3}	×	Who-To-Follow
Patent	1,810,841	10,818,819	1,468	0	3.30×10^{-4}	×	Citation
WikiLink	1,358,870	34,163,774	2,198	0	1.85×10^{-3}	×	Web link
Wikipedia	8,227/1,000	157,474	152,757	88	1.91×10^{0}	✓	Social
Reddit	10,000/984	672,447	669,065	88	6.83×10^{0}	✓	Social
MOOC	7,047/97	411,749	345,600	57	6.02×10^{1}	✓	Interaction
LastFM	980/1,000	1,293,103	1,283,614	88	1.32×10^{2}	✓	Interaction
Enron	184	125,235	22,632	98	3.70×10^{2}	×	Social
Social Evo.	74	2,099,519	565,932	99	3.83×10^{4}	×	Proximity
UCI	1,899	59,835	58,911	66	1.66×10^{0}	×	Social
Flights	13,169	1,927,145	122	79	1.11×10^{0}	×	Transport
Contact	692	2,426,279	8,064	97	5.07×10^{2}	×	Proximity
tgbl-wiki	8,227/1,000	157,474	152,757	88	1.91×10^{0}	✓	Interaction
tgbl-review	352,636/298,590	4,873,540	6,865	3	4.63×10^{-3}	✓	Rating
tgbl-coin	638,486	22,809,486	1,295,720	83	5.60×10^{-3}	×	Transaction
tgbl-comment	994,790	44,314,507	30,998,030	20	4.48×10^{-3}	×	Social
tgbl-flight	18,143	67,169,570	1,385	97	2.04×10^{1}	×	Transport
Bitcoin-Alpha	3,783	24,186	24,186	0	1.69×10^{-1}	×	Finance
Bitcoin-OTC	5,881	35,592	35,592	0	1.03×10^{-1}	×	Finance

to you for any damages arising out of the use or inability to use these programs (including but not limited to loss of data or data being rendered inaccurate). The original dataset can be found here.

Taobao: CC BY-NC-SA 4.0 license (Creative Commons Attribution-NonCommercial-ShareAlike 4.0 International). The original dataset can be found here.

Yelp: MIT license. The original dataset can be found here.

GoogleLocal: The original dataset can be found here.

Flickr: CC BY-SA license (Creative Commons Attribution-ShareAlike). The original dataset can be found here.

YouTube: CC BY-SA license (Creative Commons Attribution-ShareAlike). The original dataset can be found here.

Patent: MIT license. The original dataset can be found here.

WikiLink: CC BY-SA license (Creative Commons Attribution-ShareAlike). The original dataset can be found here.

B Experiments Details

B.1 Experimental Configurations

We conduct a grid search to identify the optimal settings for key hyperparameters, with the search ranges and corresponding methods presented in Table 6. The final hyperparameter configurations of determined by the grid search for various methods are detailed in Table 7. For the configurations of the dropout rate and neighbor sampling strategies of different methods, most methods achieve the best performance with a dropout rate of 0.1 and the recent neighbor sampling strategy. However, TGAT performs well with a dropout rate of 0.3 on the ML-20M dataset. Meanwhile, GraphMixer achieves the best performance with a dropout rate of 0.3 on the ML-20M and the Yelp datasets, as well. Moreover, the best configurations of neighbor sampling strategies of CAWN and TCL on the ML-20M dataset are both uniform strategies.

For the ML-20M and the Flickr datasets, experiments are conducted on an Ubuntu machine equipped with Intel(R) Xeon(R) Gold 6240R CPU @ 2.40GHz. The GPU device is NVIDIA A100 with 80 GB memory. For the Taobao dataset, experiments are conducted on an Ubuntu machine equipped with Intel(R) Xeon(R) Gold 5218 CPU @ 2.30GHz. The GPU device is NVIDIA A100-SXM4 with

80 GB memory. For the Yelp dataset, experiments are conducted on an Ubuntu machine equipped with Intel(R) Xeon(R) Gold 6226R CPU @ 2.90GHz. The GPU device is NVIDIA RTX A6000 with 40 GB memory. For the GoogleLocal, the Patent, and the WikiLink datasets, experiments are conducted on an Ubuntu machine equipped with Hygon C86 7390 32-core Processor. The GPU device is NVIDIA A800 with 80 GB memory. For the YouTube dataset, experiments are conducted on an Ubuntu machine equipped with Intel(R) Xeon(R) Platinum 8369B CPU @ 2.90GHz. The GPU device is A100-SXM4 with 80 GB memory.

Table 6: Searched ranges of hyperparameters and the related methods.

Hyperparameters	**Searched Ranges**	**Related Methods**
Number of Sampled Neighbors	[20, 30, 40, 50, 60]	DyRep, TGAT, TGN, CAWN, TCL, GraphMixer
Dropout Rate	[0.0, 0.1, 0.2, 0.3, 0.4, 0.5]	JODIE, DyRep, TGAT, TGN, CAWN, TCL, GraphMixer, DyGFormer
Neighbor Sampling Strategies	[recent, uniform]	DyRep, TGAT, TGN, CAWN, TCL, GraphMixer
Length of Input Sequences & Patch Size	[32 & 1, 64 & 2]	DyGFormer

Table 7: Configurations of the number of sampled neighbors and the length of input sequences & the patch size of different methods.

Datasets	DyRep	TGAT	TGN	CAWN	TCL	GraphMixer	DyGFormer
ML-20M	40	50	40	60	60	60	32 & 1
Taobao	40	50	40	60	60	60	32 & 1
Yelp	40	60	40	60	60	60	32 & 1
GoogleLocal	20	60	20	60	60	20	64 & 2
Flickr	40	40	40	40	50	40	32 & 1
YouTube	40	40	40	50	40	50	32 & 1
Patent	40	40	40	40	40	40	64 & 2
WikiLink	40	40	40	40	60	50	64 & 2

C Temporal Graph Learning Methods

JODIE (Kumar et al., 2019) uses two coupled recurrent neural networks to dynamically update the states of users and items during interactions. It includes a novel projection operation that predicts future representation trajectories of both users and items, allowing the model to anticipate future behaviors. This architecture not only captures the evolution of user-item interactions but also facilitates the learning of representations that can be used for downstream tasks like recommendation and link prediction.

DyRep (Trivedi et al., 2019) introduces a dynamic representation learning framework that updates node states in real-time with each interaction. It leverages a recurrent neural network to capture node interactions and utilizes a temporal-attentive aggregation module to focus on evolving graph structures over time. DyRep is particularly effective in modeling dynamic relationships by considering both node communication and structural events, thus providing a comprehensive understanding of temporal graph changes.

TGAT (Xu et al., 2020) incorporates self-attention mechanisms to simultaneously model both the structural and temporal properties of dynamic graphs. Its design includes a time encoding function that uniquely represents temporal information, enabling the model to handle complex, evolving interactions among nodes. This combination allows TGAT to capture intricate temporal patterns and efficiently aggregate information from temporal-topological neighbors.

TGN (Rossi et al., 2020) introduces a memory-based approach for dynamic graph learning, where each node maintains an evolving memory that is updated through various interactions. Using a combination of message functions, aggregators, and memory updaters, TGN generates temporal node representations. The embedding module is crucial in capturing the temporal dynamics of nodes, which makes TGN adaptable for various dynamic graph tasks like link prediction and node classification.

CAWN (Wang et al., 2021d) performs random walks on continuous-time dynamic graphs and employs an attention mechanism to selectively focus on crucial segments of these walks. This allows it to capture both temporal relationships and causal dependencies in the network. By learning these patterns, CAWN is capable of generating relative node identities, making it effective for temporal graph tasks such as anomaly detection and node classification.

EdgeBank (Poursafaei et al., 2022) is a memory-centric approach tailored for transductive dynamic link prediction without relying on trainable parameters. It memorizes observed interactions and uses various strategies to update its memory. EdgeBank predicts future interactions based on whether the interaction is stored in its memory. Its simplicity lies in its rule-based decision-making, making it a lightweight yet competitive approach for link prediction in dynamic networks.

TCL (Wang et al., 2021a) employs contrastive learning on temporal graphs to learn robust node embeddings. Maximizing the agreement between node pairs that are temporally similar captures both temporal dependencies and topological structures. TCL uses a graph transformer to incorporate both graph topology and temporal information, along with cross-attention mechanisms to model interactions between nodes over time.

GraphMixer (Cong et al., 2023) focuses on enhancing node embeddings in dynamic graphs by mixing both temporal and structural features. It uses a fixed time encoding function rather than a trainable one, incorporating it into a link encoder based on MLP-Mixer to learn temporal links effectively. GraphMixer also includes a node encoder with neighbor mean-pooling to aggregate node features, offering a comprehensive method for dynamic graph analysis.

DyGFormer (Yu et al., 2023) adopts a Transformer-based approach to capture long-term temporal dependencies in dynamic graphs. It introduces neighbor co-occurrence encoding and patching techniques, which help in modeling both the local and global structure of evolving interactions. This allows DyGFormer to effectively capture complex patterns in dynamic environments, making it suitable for various temporal graph tasks.