T-GRAB: A Synthetic Diagnostic Benchmark for Learning on Temporal Graphs

Alireza Dizaji
alireza.dizaji@mila.quebec
Mila - Quebec AI Institute
University of Montreal

Benedict Aaron Tjandra
aaron.tjandra@mila.quebec
Mila - Quebec AI Institute
University of Oxford

Mehrab Hamidi
mehrab.hamidi@mila.quebec
Mila - Quebec AI Institute
University of Montreal

Shenyang Huang
shenyang.huang@mail.mcgill.ca
Mila - Quebec AI Institute
McGill University
University of Oxford

Guillaume Rabusseau
rabussgu@mila.quebec
Mila - Quebec AI Institute
DIRO - University of Montreal
CIFAR AI Chair

ABSTRACT

Dynamic graph learning methods have recently emerged as powerful tools for modelling relational data evolving through time. However, despite extensive benchmarking efforts, it remains unclear whether current Temporal Graph Neural Networks (TGNNs) effectively capture core temporal patterns such as periodicity, cause-and-effect, and long-range dependencies. In this work, we introduce the **Temporal Graph Reasoning Benchmark (T-GRAB)**, a comprehensive set of synthetic tasks designed to systematically probe the capabilities of TGNNs to reason across time. T-GRAB provides controlled, interpretable tasks that isolate key temporal skills: counting/memorizing periodic repetitions, inferring delayed causal effects, and capturing long-range dependencies over both spatial and temporal dimensions. We evaluate 11 temporal graph learning methods on these tasks, revealing fundamental shortcomings in their ability to generalize temporal patterns. Our findings offer actionable insights into the limitations of current models, highlight challenges hidden by traditional real-world benchmarks, and motivate the development of architectures with stronger temporal reasoning abilities. The code for T-GRAB can be found at: https://github.com/alirezadizaji/T-GRAB.

CCS CONCEPTS

• **Computing methodologies** → **Learning latent representations**.

KEYWORDS

Temporal Graph Learning, Graph Datasets, Benchmark Evaluation, Graph Time-Series, Temporal Reasoning

KDD 2025 MLoG-GenAI Workshop, August 06, 2025, Toronto, ON

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM
https://doi.org/XXXXXXX.XXXXXXX

ACM Reference Format:
Alireza Dizaji, Benedict Aaron Tjandra, Mehrab Hamidi, Shenyang Huang, and Guillaume Rabusseau. 2025. T-GRAB: A Synthetic Diagnostic Benchmark for Learning on Temporal Graphs. In *Proceedings of Machine Learning on Graphs in the Era of Generative Artificial Intelligence (KDD 2025 MLoG-GenAI Workshop).* ACM, New York, NY, USA, 8 pages. https://doi.org/XXXXXXX.XXXXXXX

1 INTRODUCTION

Many real-world networks, such as social media networks [8], human contact networks [23] and financial transaction [21] networks can be formulated as temporal graphs or graphs that evolve over time. Recently, Temporal Graph Neural Networks (TGNNs) have emerged as promising architectures to address the unique challenges associated with ML on temporal graphs, which necessitates the modeling of both spatial and temporal dependencies [4, 6, 20, 25, 30, 32]. Naturally, the development of TGNNs is quickly followed by an increased focus to design challenging benchmarks to understand their capabilities [5, 8, 19, 24, 31] across node, edge, and graph-level tasks. These benchmarks provide significant challenges for TGNNs in both scale and domain diversity with a focus on real-world tasks. However, current TGNNs have been shown to significantly struggle in these benchmarks and, in some cases, even underperform simple heuristics such as EdgeBank [19] and persistent forecast [8].

When compared to the increasing number of novel architectures proposed, exploring the weaknesses of TGNNs remains underexplored and often applies only to specific categories of methods [1, 26, 27]. Therefore, we argue that there is a strong need for a surgical and well-designed benchmark to highlight the weakness of existing models in performing crucial yet simple tasks on temporal graphs.

In the past, diagnostic benchmarks were developed with different task classes to provide crucial insights into model capabilities precisely when complex, real-world benchmarks proved insufficient for pinpointing specific failure modes. For instance, in computer vision, the CLEVR dataset [10] utilized synthetically generated scenes to test the compositional reasoning abilities of visual question-answering models, revealing limitations obscured by the biases and confounding factors present in natural images. Similarly, in early natural language processing, the bAbI dataset [29] provided a suite of 20 synthetic question-answering tasks generated algorithmically to probe basic reasoning skills (such as counting,

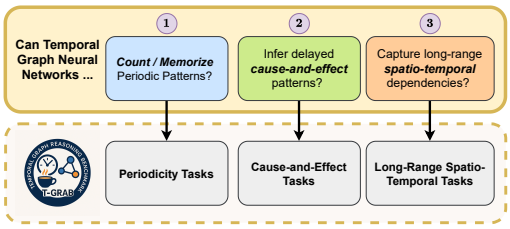


Figure 1: T-GRAB tests the capabilities of TGNNs to reason over time in three fundamental aspects and includes three carefully designed tasks.

induction, and deduction) essential for language understanding, offering metrics for progress on these core competencies. Reinforcement learning has also benefited from such focused evaluations; for instance, the Behaviour Suite for RL [16] includes controlled environments specifically designed to diagnose the memory and exploration capabilities of RL agents.

These examples demonstrate the power of purely synthetic datasets and environments designed for diagnostic evaluation: they allow for precise control over task complexity and the factors being tested, yielding clear insights into model strengths and weaknesses. Such a dedicated, synthetic diagnostic benchmark is currently missing for the domain of temporal graph learning (TGL). While existing benchmarks effectively test performance on complex, real-world dynamics, they inherently entangle various challenges, noisy interactions, complex graph structures evolving simultaneously with temporal patterns, and diverse event types. This makes it difficult to determine if a model's failure stems from an inability to handle graph complexity or from a fundamental deficit in capturing specific temporal patterns like periodicity, cause-and-effect relationships, or dependencies spanning long time horizons.

To address this gap and facilitate a deeper, more interpretable understanding of TGNN limitations, we introduce the Temporal Graph Reasoning Benchmark (T-GRAB). T-GRAB (Figure 1) comprises a suite of purely synthetic, dynamic graph datasets explicitly designed to probe the fundamental temporal reasoning capabilities essential for modeling real-world dynamic systems. By isolating core temporal patterns within controlled graph structures, T-GRAB allows for a clear assessment of how well current TGNNs capture and generalize these patterns.

Contributions. Our main contributions are as follows:

- We introduce T-GRAB , the first synthetic benchmark designed to systematically evaluate temporal reasoning capabilities of TGNNs in controlled settings. It features three carefully crafted dynamic link prediction tasks: 1) **periodicity** to assess *temporal pattern counting and memorization*, 2) **cause-and-effect** to evaluate *delayed dependency inference*, and 3) **long-range spatio-temporal** to test *long-range spatial-temporal dependency modeling*.
- T-GRAB provides a configurable environment where task difficulty can be precisely adjusted to identify the limitations of TGNNs. Our experiments reveal distinct behavioral patterns between CTDG and DTDG methods, and highlight that the number of temporal neighbors, an often overlooked hyperparameter, significantly impacts model performance across our benchmark tasks.
- For the periodicity tasks, GC-LSTM consistently performs best, even in the most challenging settings, indicating its recurrent structure is better suited for capturing periodic patterns and counting. In the cause-and-effect task, all models struggle with long-term memory, though DyGFormer, TGAT, and TGN degrade most gracefully. Finally, in the spatio-temporal task, DyGFormer's transformer-based architecture excels with short-range spatial dependencies, while TGAT and TGN outperform it as spatial dependencies grow longer.
- Notably, no single model consistently outperforms across all tasks in T-GRAB , contrasting with real-world benchmarks where leaderboards are typically dominated by a few methods. This finding underscores the value of T-GRAB as a diagnostic tool that can guide the development of more robust and versatile temporal graph learning methods capable of handling diverse temporal reasoning challenges.

Related Work Temporal Graph Neural Networks (TGNNs) are categorized into continuous and discrete time dynamic graph methods (CTDG and DTDG). CTDG methods process event streams with timestamps using neighbor sampling to model evolving relations. They include TGAT [30] with self-attention and temporal encoding, TGN [20] with memory modules, DyGFormer [32] using multi-head attention on temporal patches, and CTAN [6] employing ODEs. DTDG methods operate on regularly-spaced graph snapshots and typically use recurrent neural networks to track history; popular methods include GC-LSTM [2], T-GCN [33], and EvolveGCN [18]). Recently, [9] compared CTDG and DTDG methods and found that DTDG methods sacrifice accuracy for efficiency by computing on coarser-level snapshot information. Subsequently, they showed that DTDG methods can operate on CTDG datasets by using the Unified Temporal Graph (UTG) framework. In Section 3, we evaluate and compare the above methods from both continuous and discrete-time approaches on T-GRAB datasets to analyze their capabilities in capturing core temporal patterns. Dedicated benchmarks [13, 15, 17, 22] have significantly improved TGNN evaluation, though [19] noted overly optimistic results due to simple negative edges and introduced new sampling strategies and diverse datasets. Subsequently, TGB [8] and TGB 2.0 [5] presented larger, challenging datasets for link and node prediction, while TGB-Seq [31] proposed real-world with complex sequential dynamics and low edge repeatability, revealing limitations in current TGNN generalization. Our work complements these recent efforts by constructing synthetic tasks to more exactly pinpoint the current functional weaknesses of TGNNs.

2 TEMPORAL GRAPH PRELIMINARIES

Definition 2.1 (Discrete Time Dynamic Graphs). A *Discrete Time Dynamic Graph* $\mathcal{G}$ is a sequence of graph snapshots sampled at regularly-spaced time intervals [11]: $\mathcal{G} = G_1, G_2, G_3, \cdots, G_T$. Each $G_t = (V_t, E_t, \mathbf{X}_t)$ is the graph at snapshot $t = 1, \cdots, T$, where V_t, E_t are the set of nodes and edges in G_t, respectively, and $\mathbf{X}_t \in \mathbb{R}^{|V_t| \times d}$ is the matrix of node features at time t.

In this work, we focus on tasks designed for Discrete Time Dynamic Graphs (DTDGs). As Continuous Time Dynamic Graph

(CTDG) methods can be applied on DTDGs as well [9], we benchmark the performance of both types of methods for comprehensive evaluation. In our tasks, the set of vertices is the same at all time steps, i.e., $V_1 = V_2 = \cdots = V_t = \cdots$. Node features are also constant through time and consists of one-hot encoding of the N nodes (i.e., $d = N$ and $\mathbf{X}_t = \mathbf{I}$ is the identity matrix for all t). While dynamic graphs sometimes have edge features, we do not use any in T-GRAB . We follow the methodology outlined in [9] to evaluate CTDG methods on discrete time graphs by translating all edges in each graph snapshot G_t into a batch of edges $\{(u, v, t) \mid (u, v) \in E_t\}$.

3 TEMPORAL GRAPH REASONING BENCHMARK: T-GRAB

In this section, we introduce the Temporal Graph Reasoning Benchmark (T-GRAB), the first synthetic benchmark designed to systematically evaluate the temporal reasoning capabilities of TGNNs in a controlled environment. T-GRAB comprises three categories of dynamic link prediction tasks, each probing distinct aspects of temporal reasoning: 1) *periodicity* tasks, which assess counting and memorization capabilities; 2) *cause-and-effect* tasks, which evaluate the ability to identify causal relationships across time delays; and 3) *long-range spatio-temporal* tasks, which measure how effectively models capture dependencies spanning both spatial and temporal dimensions. These task families are examined in detail in Sections 3.1, 3.2, and 3.3, respectively. As summarized in Table 1, T-GRAB encompasses a diverse spectrum of graph characteristics and temporal patterns, providing a comprehensive framework to rigorously test the fundamental reasoning capabilities of TGNNs.

Methods in Comparison. We conduct a comprehensive evaluation of diverse Temporal Graph Learning (TGL) approaches on T-GRAB tasks . Our analysis encompasses four continuous-time (CTDG) architectures (DyGFormer [32], CTAN [6], TGN [20], and TGAT [30]), three discrete-time (DTDG) frameworks (EvolveGCN [18], T-GCN [33], and GC-LSTM [2]), two static graph methods (GCN [12] and GAT [28]), for which we use their DTDG implementations provided by UTG [9], and two established baselines (the *persistence* heuristic, which predicts edges from the previous timestep, and Edgebank$_\infty$ [19]). This selection represents the state-of-the-art across different temporal graph learning paradigms, enabling a rigorous assessment of their fundamental reasoning capabilities.

Evaluation Protocols. Prior research has demonstrated that evaluation results for dynamic link prediction can vary substantially depending on negative sampling strategies [19]. To ensure methodological rigor and reproducibility, we implement a comprehensive evaluation framework that calculates the average F_1 score across all possible node pairs at each test time step. This approach eliminates sampling bias and provides a more reliable performance assessment. For periodicity tasks, our evaluation incorporates all test edges in the F_1 calculation to capture the full spectrum of temporal patterns. In contrast, for cause-and-effect and long-range spatio-temporal tasks, we restrict the evaluation to edges involving the memory/target node, as they are the only predictable connections within the otherwise stochastically generated graph. This targeted evaluation ensures that model performance reflects genuine temporal reasoning capabilities rather than chance correlations in random edge formations.

3.1 Periodicity Tasks

We introduce a family of synthetic tasks designed to evaluate temporal graph learning (TGL) methods' ability to recognize periodic structures. These tasks assess two fundamental capabilities: *counting* and *memorization*, in both deterministic and stochastic environments.

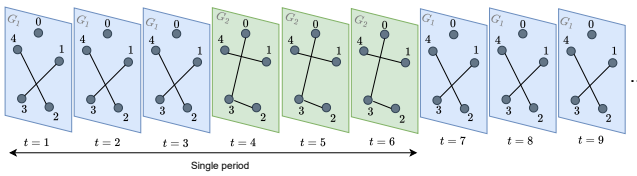


Figure 2: Periodic task in $\mathcal{P}^{\text{det}}(k = 2, n = 3)$ with 2 unique snapshots repeated 3 times within a period.

Definition 3.1 (Periodicity Tasks). Let $k, n \in \mathbb{N}$. The task families $\mathcal{P}^{\text{det}}(k, n)$ and $\mathcal{P}^{\text{sto}}(k, n)$ are defined based on a repeating pattern where integers $i = 1, \ldots, k$ each appears n consecutive times before cycling. For each t, let $i_t = (\lfloor t/n \rfloor \mod k) + 1$. In $\mathcal{P}^{\text{det}}(k, n)$, the dynamic graph $\mathcal{G} = G_1, G_2, \ldots$ is a periodic sequence alternating between k fixed static graphs $G_1, \ldots, G_k$, i.e., $G_t = G_{i_t}$. In $\mathcal{P}^{\text{sto}}(k, n)$, each G_t is sampled from one of k distributions $D_1, \ldots, D_k$ over static graphs (e.g., Erdős-Rényi (ER) [3], Stochastic Block Model (SBM) [7]), with $G_t \sim D_{i_t}$. The resulting sequence is stochastic, but the distribution pattern follows the periodic structure of $\mathcal{P}^{\text{det}}(k, n)$.

For example, a task in $\mathcal{P}^{\text{det}}(k = 3, n = 2)$ will correspond to a dynamic graph $G_1, G_1, G_2, G_2, G_3, G_3, G_1, G_1$,..., where G_1, G_2, G_3 are static graphs. A task in $\mathcal{P}^{\text{det}}(k, n)$ is illustrated in Figure 2.

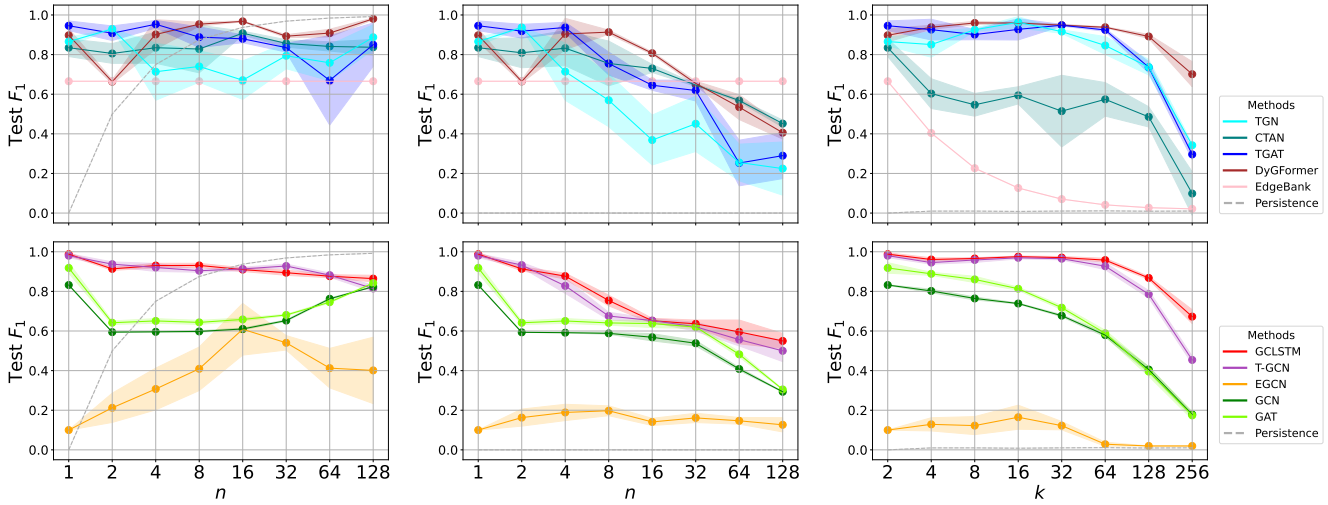
Task objectives. Tasks in $\mathcal{P}^{\text{det}}(k, n)$ test the counting and memory capacity of TGL methods. The parameter k controls the length of the pattern and thus the memory demand, while n governs how long each graph is repeated and tests the model's ability to count. For instance, solving a task in $\mathcal{P}^{\text{det}}(2, n)$ requires counting up to n before switching graphs, while solving a task in $\mathcal{P}^{\text{det}}(k, 1)$ requires memorizing k static graphs. Tasks in $\mathcal{P}^{\text{sto}}(k, n)$ introduce stochasticity, requiring models to reason over distributions rather than fixed structures, increasing the complexity while retaining the same periodic structure.

Can TGNNs count? To evaluate the counting ability of TGNNs, we construct tasks in $\mathcal{P}^{\text{det}}(k, n)$ with graphs sampled from the Erdős-Rényi (ER) model [3] (100 nodes, edge probability 0.01). We set $k = 2$ and vary n, where increasing n corresponds to greater task difficulty as models must count longer sequences.

Model performance on these $\mathcal{P}^{\text{det}}(2, n)$ tasks for n ranging from 1 to 128 is presented in Figure 3. Since models that simply repeat the previous timestep can perform well for large n, we evaluate both overall performance (left plot) and performance at change points (center plot) where the active graph switches (i.e., $t = n + 1, 2n + 1, \cdots$). High scores at change points indicate true counting and pattern understanding, whereas good overall performance with poor change point performance suggests the model is merely exploiting continuity rather than reasoning about periodicity.

Table 1: T-GRAB dataset statistics and characteristics.

Dataset	# Nodes	# Edges	# Timestamp	Counting	Memorizing	Spatial Understanding
Periodicity	100	10,560 - 9,144,440	96 - 12,288	✓	✓	✗
Cause-and-Effect	101	164,856 - 174,470	4,001 - 4,256	✗	✓	✗
Long-Range Spatio-Temporal	102	48,006 - 411,072	4,001 - 4,032	✗	✓	✓



(a) $\mathcal{P}^{\text{det}}(2, n)$ **- All time steps.** **(b)** $\mathcal{P}^{\text{det}}(2, n)$ **- Change points.** **(c)** $\mathcal{P}^{\text{det}}(k, 1)$ **- All time steps.**

Figure 3: Performance of CTDG methods (top row) and DTDG methods (bottom row) on the deterministic periodicity tasks.

The results reveal distinct behaviors across different TGNN families. Among CTDG models, TGAT excels for small n, while DyGFormer demonstrates more consistent performance at larger n. For DTDG methods, T-GCN and GC-LSTM are strongest overall. Notably, EvolveGCN performs worse than static graph learning methods (GCN and GAT), which lack temporal processing mechanisms, underscoring how challenging these seemingly simple periodic patterns can be for current TGNNs. EdgeBank remains constant across all n values, always predicting the union of both graphs. The persistence baseline improves over all timesteps as n increases by simply copying the previous snapshot, but scores zero at change points.

A key observation, evident when comparing performance over all timesteps versus at change points (Figure 3), is that as n grows, many models increasingly rely on repetition rather than explicit counting. While their overall scores might remain high or even improve for larger n (due to successfully predicting links during long static phases), their performance at change points often degrades. At $n = 32$, EdgeBank even starts to outperform all TGNNs. This divergence suggests that current TGNNs struggle to robustly count long sequences and instead learn a simpler heuristic reminiscent of persistence.

How much can TGNNs memorize? To evaluate the memorization capabilities of TGNNs, we fix $n = 1$ and vary k. The difficulty scales with k: as more unique graph structures are introduced, models must maintain a larger and more distinct set of representations to correctly predict links at each timestep.

Figure 3 (right) shows results on $\mathcal{P}^{\text{det}}(k, 1)$ for k ranging from 2 to 256. As expected, EdgeBank steadily degrades to zero performance at $k = 256$, eventually defaulting to predicting each snapshot as a clique. All TGNN models show a gradual decline as k increases. GC-LSTM and DyGFormer consistently perform best, demonstrating strong memorization of patterns. T-GCN, TGN, and TGAT remain robust up to $k = 128$, but drop sharply at $k = 256$, suggesting these models reach their maximum memorization capacity at this point.

In contrast, TGNNs such as CTAN and EvolveGCN struggle significantly as the number of unique graphs increases. CTAN's performance begins to degrade considerably after $k = 64$. Notably, both these methods are often outperformed by static GNNs like GCN and GAT, which exhibit a more gradual decline. This suggests that for tasks dominated by the need to memorize distinct states, an ineffective temporal mechanism can be more detrimental than no temporal mechanism at all.

Can TGNNs learn stochastic periodic structures? Finally, we investigate how models' memorization capabilities extend to probabilistic settings, where periodic structure emerges from stochastic

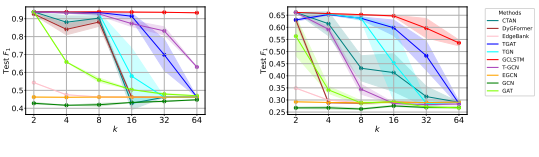


Figure 4: Performance of TGL methods on the stochastic periodicity tasks $\mathcal{P}^{\text{sto}}(k, 1)$ for different values of the intra-community edge probability: $p = 0.9$ (left) and $p = 0.5$ (right).

processes rather than deterministic patterns. We employ Stochastic Block Models (SBMs) with 100 nodes divided into 3 communities. While all SBM distributions share identical inter-community (0.01) edge probabilities, they differ in community structures . We examine two intra-community edge probability settings: $p = 0.9$ and $p = 0.5$ (the latter being more difficult).

Figure 4 presents model performance across these stochastic periodicity tasks. As expected, performance decreases as k increases, reflecting the growing challenge of memorizing multiple stochastic patterns. GC-LSTM consistently achieves the highest performance across both probability settings, with TGAT as the second-best performer. While T-GCN performs strongly with clearer community structure ($p = 0.9$), it struggles in the noisier setting ($p = 0.5$), falling behind TGAT and TGN. Overall, performance decreases under lower intra-community density, confirming that increased stochasticity challenges memorization capabilities. Notably, unlike in deterministic scenarios, static baselines (GCN and GAT) consistently underperform temporal models, highlighting the effectiveness of temporal modeling in capturing stochastic periodic structures within dynamic graphs.

3.2 Delayed Cause-and-Effect Tasks

To assess how effectively TGL methods capture delayed causal relationships, we introduce *delayed cause-and-effect tasks*. These tasks involve a sequence of randomly generated graphs (e.g., from an Erdős-Rényi distribution) with a designated memory node that connects to nodes participating in edges from ℓ time steps in the past. This design creates a clear temporal dependency that models must identify. We formalize this framework as follows:

Definition 3.2 (Delayed Cause-and-Effect Task). Let $\ell \in \mathbb{N}$. The *delayed cause-and-effect task* family $\mathcal{CE}(\ell)$ consists of dynamic graphs $\mathcal{G} = G_1, G_2, \ldots$ generated as follows: First, each graph $G_t = (V, E_t)$ is sampled independently from a distribution D over static graphs, where $V = \{v_1, \ldots, v_N\}$ is the set of nodes, shared across time steps. For $t > \ell$, the graph G_t is augmented by introducing a memory node $v_{\mathcal{M}}$ and connecting it to the nodes:

$$V \leftarrow V \cup \{v_{\mathcal{M}}\}$$

$$E_t \leftarrow E_t \cup \left\{(v_{\mathcal{M}}, u), (v_{\mathcal{M}}, v) \mid (u, v) \in E_{t-\ell}\right\} \text{ for } t > \ell.$$

Intuitively, the model needs to remember nodes that were connected to each other at $E_{t-\ell}$ to predict the edges of the memory node at time E_t. An example of this task is illustrated in Figure 5. As ℓ increases, the task becomes more challenging in terms of the memory capacity required from the model. While the number of nodes and edges in the graphs G_t also affect the task's difficulty, we focus on varying ℓ. Since only the edges involving the memory

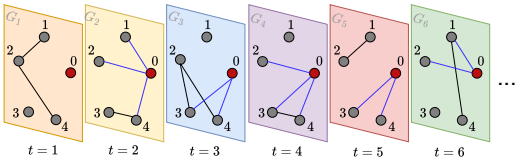


Figure 5: Illustration of a delayed cause-and-effect dataset with lag $\ell = 1$. Black edges show the cause subgraph (nodes 1 to N). Blue edges show the effect subgraph, where the memory node (0) connects to previously active cause nodes (degree ≥ 1).

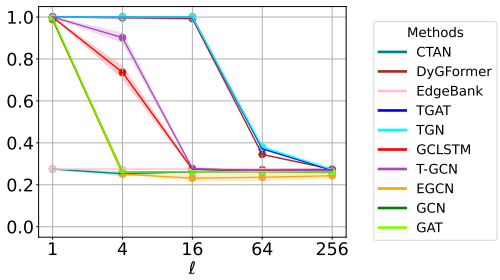


Figure 6: Methods' performance on cause-and-effect tasks $\mathcal{CE}(\ell)$ across five temporal lags ℓ.

node can be predicted (the other ones being completely random), only the F_1 over possible edges involving the memory node are reported in our experiments.

Task objectives and implementation. In the delayed cause-and-effect tasks $\mathcal{CE}(\ell)$, TGNNs must propagate information across time steps while identifying the causal relationship governing the memory node's connectivity. This requires models to recognize temporal patterns and maintain historical information. The task difficulty scales with ℓ: larger values require retaining information longer, making causal relationship identification increasingly challenging. For implementation, we generate the underlying graphs (excluding the memory node) using an Erdős-Rényi model with 100 nodes and edge probability 0.01, creating a controlled environment to isolate and evaluate temporal reasoning capabilities.

Results. Figure 6 illustrates performance on cause-and-effect tasks across varying temporal lags. For minimal lags ($\ell = 1$), most methods demonstrate strong performance, as the task primarily requires extracting information from the immediate past, a capability even static models possess. However, as lag increases, performance degrades systematically, with static and DTDG approaches struggling significantly with longer temporal dependencies. GC-LSTM and T-GCN exhibit particularly pronounced performance deterioration at $\ell = 4$ and $\ell = 16$, revealing limitations in recurrent architectures' capacity to maintain long-term temporal information. Conversely, DyGFormer, TGAT, and TGN consistently outperform other methods, underscoring the efficacy of attention mechanisms and memory modules in capturing dynamic temporal patterns. At extended lags ($\ell = 64$ and $\ell = 256$), all methods converge toward EdgeBank's performance level, indicating a common challenge in modeling distant causal relationships. Notably, CTAN shows inconsistent

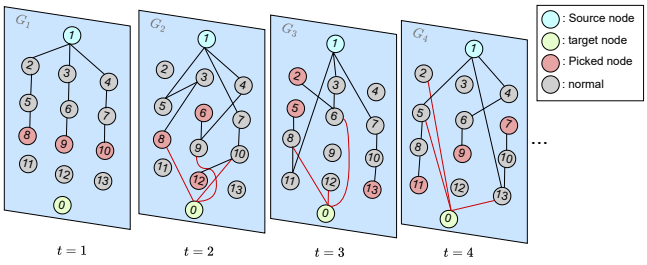


Figure 7: Illustration of a long-range spatio-temporal task $\mathcal{LR}(\ell, d)$ with $\ell = 1$, $d = 3$ and $P = 3$.

performance even at minimal lags, suggesting inherent limitations in this context. These findings emphasize the crucial importance of architectural design choices, particularly attention and memory components, for effective temporal reasoning in dynamic graphs.

3.3 Long-Range Spatio-Temporal Task

Finally, we introduce *long-range spatio-temporal tasks* to evaluate how effectively TGL methods reason across both temporal and spatial dimensions. These tasks extend the delayed cause-and-effect framework by incorporating multi-hop spatial paths alongside temporal dependencies. In this setting, each graph snapshot contains multiple paths originating from a source node $v_{\mathcal{S}}$, while a target node $v_{\mathcal{T}}$ connects to the endpoints of paths appearing ℓ time steps earlier. We formalize this task as follows:

Definition 3.3 (Long-Range Spatio-Temporal Task). Let $\ell, d, P \in \mathbb{N}$. The *long-range spatio-temporal task* family $\mathcal{LR}_P(\ell, d)$ is defined over dynamic graphs $\mathcal{G} = G_1, G_2, \ldots$, where each snapshot $G_t = (V, E_t)$ has a fixed node set $V = \{v_{\mathcal{S}}, v_{\mathcal{T}}, v_1, \ldots, v_N\}$. The edge set E_t consists of P disjoint paths of length d from the source node $v_{\mathcal{S}}$ through randomly chosen intermediate nodes. For $t > \ell$, E_t additionally includes P edges from the target node $v_{\mathcal{T}}$ to the endpoints of the P paths in $G_{t-\ell}$. Formally, for $t > \ell$: $E_t = \bigcup_{p=1}^{P} \{(v_{\mathcal{S}}, u_1^{(t,p)}), (u_1^{(t,p)}, u_2^{(t,p)}), \ldots, (u_{d-1}^{(t,p)}, u_d^{(t,p)})\} \cup \{(v_{\mathcal{T}}, u_d^{(t-\ell,p)})\}$, where the nodes $u_i^{(t,p)}$ for $1 \le i \le d$, $1 \le p \le P$ are drawn at random in $\{v_1, \ldots, v_N\}$ (without replacement).

Intuitively, we can treat node $v_{\mathcal{S}}$ as the *progenitor* of some signal that 1) reaches a set of nodes that are spatially separated from $v_{\mathcal{S}}$ by a distance d and 2) whose effect (connecting the nodes that it reached to node $v_{\mathcal{T}}$) is additionally delayed by ℓ timesteps. An illustration of a dynamic graph for this task is given in Figure 7. We will focus on the effect of the lag ℓ and distance d in our experiments and set the number of path P to 3 in all tasks, which we will simply denote by $\mathcal{LR}(\ell, d)$.

Task objectives and implementation. Dynamic graphs inherently contain two fundamental distance metrics: temporal (between time steps) and spatial (between nodes). Effective TGL methods must reason across both dimensions simultaneously to capture complex patterns. The $\mathcal{LR}(\ell, d)$ task family specifically evaluates this capability by requiring models to track information across both temporal lags and multi-hop spatial paths. The parameter ℓ determines the temporal memory requirement, while d defines the spatial propagation distance. This dual-parameter design creates a comprehensive benchmark for assessing spatio-temporal reasoning depth in TGL architectures. To systematically evaluate model capabilities, we generate tasks with increasing complexity using distance values $d \in \{1, 2, 4, 8, 16\}$ and temporal lags $\ell \in \{1, 4, 16, 32\}$.

Table 2: Average method rank (↓) for T-GRAB tasks (c.p. stands for change points). Top results are shown in first, second, *third*.

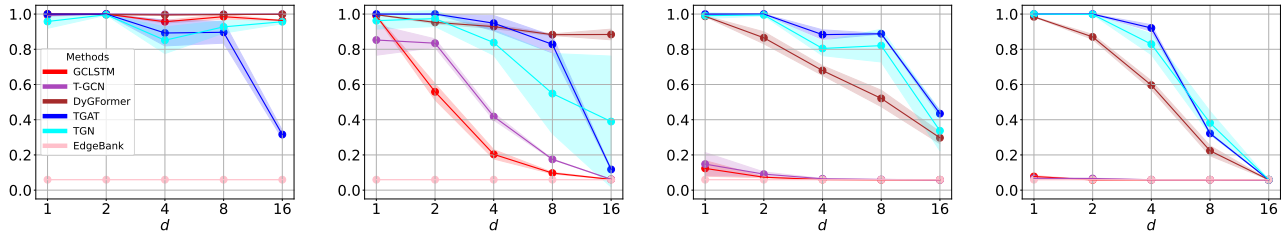
	Method	Periodicity $\mathcal{P}^{\text{det}}(2,n)$ all t	Periodicity $\mathcal{P}^{\text{det}}(2,n)$ c.p.	Periodicity $\mathcal{P}^{\text{det}}(k,1)$	Periodicity $\mathcal{P}^{\text{sto}}(2,n)$ $p=0.5$	Periodicity $\mathcal{P}^{\text{sto}}(2,n)$ $p=0.9$	Cause & Effect $\mathcal{CE}(\ell)$	Spatio-temporal long-range $\mathcal{LR}(\ell,d)$ $\ell=1$	$\ell=4$	$\ell=16$	$\ell=32$
CTDG	CTAN	5.625	*3.75*	7.625	3.167	5.167	6.4	10.0	8.6	8.4	6.8
CTDG	DyGFormer	3.25	3.5	*2.625*	7.167	7.5	*3.2*	2.2	1.8	*3.0*	3.4
CTDG	TGAT	4.75	5.0	4.375	*3.667*	3.167	2.6	3.8	**1.6**	**1.0**	**1.8**
CTDG	TGN	5.75	7.125	4.625	*3.667*	4.33	**2.4**	4.8	*2.8*	2.0	**1.8**
DTDG	GC-LSTM	**2.875**	**3.125**	**1.25**	**1.0**	**1.167**	4.8	*3.0*	4.6	5.6	6.2
DTDG	EGCN	10.875	10.0	9.75	6.5	8.167	5.0	8.2	7.8	7.0	8.0
DTDG	T-GCN	3.25	*3.75*	2.5	6.5	*3.333*	9.6	**1.8**	4.2	4.8	*6.0*
DTDG	GCN	8.875	8.0	7.0	9.833	9.833	7.8	6.8	8.8	9.2	9.0
DTDG	GAT	7.625	6.5	6.0	7.5	5.333	8.0	5.6	8.6	8.2	6.4
	EdgeBank	8.5	4.25	9.25	6.0	7.0	5.2	8.8	6.0	5.4	9.8
	Persistence	*4.625*	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0	11.0

Results. Figure 8 presents F_1 scores for top-performing methods and the EdgeBank baseline . At small temporal lags, both DTDG approaches and DyGFormer demonstrate strong performance. However, as ℓ increases, DTDG methods exhibit sharp performance declines, revealing limitations in recurrent architectures' capacity to maintain long-range temporal memory. DyGFormer's performance deteriorates more gradually but still struggles with extended temporal dependencies. In contrast, TGAT and TGN maintain robust performance even at substantial lags ($\ell = 16, 32$), highlighting the efficacy of attention-based message passing for complex spatio-temporal reasoning. Nevertheless, all models, even the strongest temporal reasoners, show significant performance degradation when spatial distance exceeds $d = 8$, emphasizing the persistent challenge of capturing long-range spatial dependencies in dynamic graphs.

4 DISCUSSION AND ANALYSIS

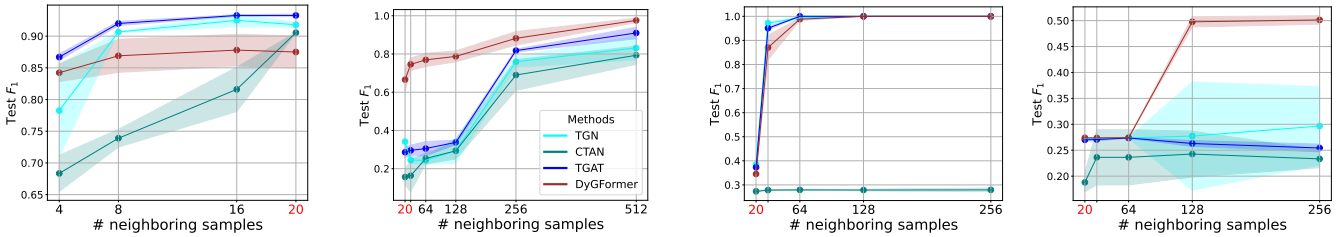
In this section, we synthesize key findings and insights across all three task categories in T-GRAB .

Bird's eye view of the results. Table 2 presents a comprehensive ranking of methods across our benchmark tasks. Our analysis reveals a striking dichotomy in model effectiveness across different temporal reasoning challenges. For periodicity tasks, GC-LSTM consistently outperforms all competitors across all five settings, challenging the prevailing assumption that CTDG methods outperform DTDG ones [9]. This suggests that for tasks requiring precise counting and pattern memorization, the simpler recurrent architecture of DTDG models may offer computational advantages over their more complex continuous-time counterparts. Conversely, for cause-and-effect and spatio-temporal long-range tasks, CTDG methods, particularly TGN and TGAT emerge as the dominant methods. Notably, CTAN, despite being explicitly designed for long-range temporal propagation, performs surprisingly poorly on spatio-temporal long-range tasks. This unexpected result highlights potential limitations in its ability to *jointly reason* over spatial and temporal dimensions.



(a) $(\ell = 1, d)$ **(b)** $(\ell = 4, d)$ **(c)** $(\ell = 16, d)$ **(d)** $(\ell = 32, d)$

Figure 8: Performance varying spatial and temporal distances in the long-range spatio-temporal tasks.



(a) $\mathcal{P}^{\text{sto}}(8, 1)$ **(b)** $\mathcal{P}^{\text{det}}(256, 1)$ **(c)** $C\mathcal{E}(64)$ **(d)** $C\mathcal{E}(256)$

Figure 9: Effect of the number of neighbors on periodic (left), and cause-and-effect (right) tasks .

These findings collectively demonstrate that no single architectural paradigm excels across all T-GRAB tasks. The observed performance variations suggest that different architectural inductive biases are better suited to specific temporal reasoning challenges. This insight points to a promising research direction: developing hybrid architectures that can effectively combine the strengths of different approaches to achieve superior performance across diverse temporal reasoning scenarios.

Effect of Number of Neighbors. Temporal neighbor sampling is a fundamental mechanism in CTDG methods [14, 20, 30, 32], that allows models to access historical interactions and model temporal dependencies, especially crucial in scenarios with sparse edge distributions. Despite its importance, the number of sampled neighbors remains an under-examined hyperparameter, commonly fixed at default values (e.g., 20 neighbors in established literature [32]). We maintained this conventional setting in our previous experiments but now systematically investigate how this parameter influences performance across T-GRAB tasks.

Figures 9a and 9b illustrate performance on periodic tasks as a function of neighbor count. Our analysis reveals that CTAN, TGN, and TGAT exhibit pronounced performance improvements with increased neighbor sampling across both stochastic and deterministic settings, demonstrating their substantial sensitivity to this parameter. For DyGFormer, this parameter directly determines the temporal context length for sequence construction, and it maintains relatively consistent performance, suggesting a more robust architecture with respect to neighbor sampling.

Extending our investigation to cause-and-effect tasks (Figures 9c and 9d), we observe that for $C\mathcal{E}(64)$, TGAT, TGN, and DyGFormer continue to benefit from sampling beyond 32 neighbors, while CTAN plateaus, indicating a fundamental limitation in its capacity and design. For the more challenging $C\mathcal{E}(256)$ task, DyGFormer demonstrates continued improvement with larger neighbor counts (particularly beyond 128), while other methods reach performance saturation. These findings suggest potential for further performance gains through even larger sampling windows (e.g., 512 neighbors), albeit with corresponding computational overhead.

5 CONCLUSION

We introduced T-GRAB , a synthetic benchmark designed to systematically probe the temporal reasoning abilities of TGNNs. Our results reveal that no single method excels across all tasks, with performance varying notably across tasks. Attention and memory-based architectures like DyGFormer, TGAT, and TGN show strong performance on long-range and causal tasks, while simpler recurrent models like GC-LSTM excel in periodic tasks. Our findings highlight limitations of current models and the need for architectures tailored to diverse temporal reasoning challenges.

ACKNOWLEDGMENTS

This research was supported by the Canadian Institute for Advanced Research (CIFAR AI chair program), the EPSRC Turing AI World-Leading Research Fellowship No. EP/X040062/1 and EPSRC AI Hub No. EP/Y028872/1. Shenyang Huang was supported by the Natural Sciences and Engineering Research Council of Canada (NSERC) Postgraduate Scholarship Doctoral (PGS D) Award and Fonds de recherche du Québec - Nature et Technologies (FRQNT) Doctoral Award. This research was also enabled in part by compute resources provided by Mila (mila.quebec).

REFERENCES

[1] Silvia Beddar-Wiesing, Giuseppe Alessio D'Inverno, Caterina Graziani, Veronica Lachi, Alice Moallemy-Oureh, Franco Scarselli, and Josephine Maria Thomas. 2024. Weisfeiler-Lehman goes Dynamic: An Analysis of the Expressive Power of Graph Neural Networks for Attributed and Dynamic Graphs. arXiv:2210.03990 [cs.LG] https://arxiv.org/abs/2210.03990

[2] Jinyin Chen, Xueke Wang, and Xuanheng Xu. 2021. GC-LSTM: Graph Convolution Embedded LSTM for Dynamic Link Prediction. arXiv:1812.04206 [cs.SI] https://arxiv.org/abs/1812.04206

[3] Paul Erdos and Alfréd Rényi. 1960. On the evolution of random graphs. *Publ. math. inst. hung. acad. sci* 5, 1 (1960), 17–60.

[4] ZhengZhao Feng, Rui Wang, TianXing Wang, Mingli Song, Sai Wu, and Shuibing He. 2024. A Comprehensive Survey of Dynamic Graph Neural Networks: Models, Frameworks, Benchmarks, Experiments and Challenges. arXiv:2405.00476 [cs.LG] https://arxiv.org/abs/2405.00476

[5] Julia Gastinger, Shenyang Huang, Mikhail Galkin, Erfan Loghmani, Ali Parviz, Farimah Poursafaei, Jacob Danovitch, Emanuele Rossi, Ioannis Koutis, Heiner Stuckenschmidt, Reihaneh Rabbany, and Guillaume Rabusseau. 2024. TGB 2.0: A Benchmark for Learning on Temporal Knowledge Graphs and Heterogeneous Graphs. arXiv:2406.09639 [cs.LG] https://arxiv.org/abs/2406.09639

[6] Alessio Gravina, Giulio Lovisotto, Claudio Gallicchio, Davide Bacciu, and Claas Grohnfeldt. 2024. Long Range Propagation on Continuous-Time Dynamic Graphs. arXiv:2406.02740 [cs.LG] https://arxiv.org/abs/2406.02740

[7] Paul W Holland, Kathryn Blackmond Laskey, and Samuel Leinhardt. 1983. Stochastic blockmodels: First steps. *Social networks* 5, 2 (1983), 109–137.

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

[9] Shenyang Huang, Farimah Poursafaei, Reihaneh Rabbany, Guillaume Rabusseau, and Emanuele Rossi. 2024. UTG: Towards a Unified View of Snapshot and Event Based Models for Temporal Graphs. arXiv:2407.12269 [cs.LG] https://arxiv.org/abs/2407.12269

[10] Justin Johnson, Bharath Hariharan, Laurens van der Maaten, Li Fei-Fei, C. Lawrence Zitnick, and Ross Girshick. 2016. CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning. arXiv:1612.06890 [cs.CV] https://arxiv.org/abs/1612.06890

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

[12] Thomas N. Kipf and Max Welling. 2017. Semi-Supervised Classification with Graph Convolutional Networks. arXiv:1609.02907 [cs.LG] https://arxiv.org/abs/1609.02907

[13] Srijan Kumar, Xikun Zhang, and Jure Leskovec. 2019. Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining (KDD '19)*. ACM, 1269–1278. https://doi.org/10.1145/3292500.3330895

[14] Yuhong Luo and Pan Li. 2022. Neighborhood-aware scalable temporal network representation learning. In *Learning on Graphs Conference*. PMLR, 1–1.

[15] Anmol Madan, Manuel Cebrian, Sai Moturu, Katayoun Farrahi, and Alex "Sandy" Pentland. 2012. Sensing the "Health State" of a Community. *IEEE Pervasive Computing* 11, 4 (2012), 36–45. https://doi.org/10.1109/MPRV.2011.79

[16] Ian Osband, Yotam Doron, Matteo Hessel, John Aslanides, Eren Sezener, Andre Saraiva, Katrina McKinney, Tor Lattimore, Csaba Szepesvari, Satinder Singh, Benjamin Van Roy, Richard Sutton, David Silver, and Hado Van Hasselt. 2020. Behaviour Suite for Reinforcement Learning. arXiv:1908.03568 [cs.LG] https://arxiv.org/abs/1908.03568

[17] Pietro Panzarasa, Tore Opsahl, and Kathleen M. Carley. 2009. Patterns and dynamics of users' behavior and interaction: Network analysis of an online community. *J. Am. Soc. Inf. Sci. Technol.* 60, 5 (May 2009), 911–932.

[18] Aldo Pareja, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao B. Schardl, and Charles E. Leiserson. 2019. EvolveGCN: Evolving Graph Convolutional Networks for Dynamic Graphs. arXiv:1902.10191 [cs.LG] https://arxiv.org/abs/1902.10191

[19] Farimah Poursafaei, Shenyang Huang, Kellin Pelrine, and Reihaneh Rabbany. 2022. Towards Better Evaluation for Dynamic Link Prediction. arXiv:2207.10128 [cs.LG] https://arxiv.org/abs/2207.10128

[20] Emanuele Rossi, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. 2020. Temporal Graph Networks for Deep Learning on Dynamic Graphs. arXiv:2006.10637 [cs.LG] https://arxiv.org/abs/2006.10637

[21] Kiarash Shamsi, Farimah Poursafaei, Shenyang Huang, Bao Tran Gia Ngo, Baris Coskunuzer, and Cuneyt Gurcan Akcora. 2024. GraphPulse: Topological representations for temporal graph property prediction. In *The Twelfth International Conference on Learning Representations*.

[22] Jitesh Shetty and Jafar Adibi. 2004. The Enron Email Dataset Database Schema and Brief Statistical Report. https://api.semanticscholar.org/CorpusID:59919272

[23] Razieh Shirzadkhani, Shenyang Huang, Abby Leung, and Reihaneh Rabbany. 2024. Static graph approximations of dynamic contact networks for epidemic forecasting. *Scientific Reports* 14, 1 (2024), 11696.

[24] Razieh Shirzadkhani, Tran Gia Bao Ngo, Kiarash Shamsi, Shenyang Huang, Farimah Poursafaei, Poupak Azad, Reihaneh Rabbany, Baris Coskunuzer, Guillaume Rabusseau, and Cuneyt Gurcan Akcora. 2024. Towards Neural Scaling Laws for Foundation Models on Temporal Graphs. *arXiv preprint arXiv:2406.10426* (2024).

[25] Joakim Skarding, Bogdan Gabrys, and Katarzyna Musial. 2021. Foundations and Modeling of Dynamic Networks Using Dynamic Graph Neural Networks: A Survey. *IEEE Access* 9 (2021), 79143–79168. https://doi.org/10.1109/access.2021.3082932

[26] Amauri H. Souza, Diego Mesquita, Samuel Kaski, and Vikas Garg. 2022. Provably expressive temporal graph networks. arXiv:2209.15059 [cs.LG] https://arxiv.org/abs/2209.15059

[27] Benedict Aaron Tjandra, Federico Barbero, and Michael Bronstein. 2024. Enhancing the Expressivity of Temporal Graph Networks through Source-Target Identification. arXiv:2411.03596 [cs.LG] https://arxiv.org/abs/2411.03596

[28] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. 2018. Graph Attention Networks. arXiv:1710.10903 [stat.ML] https://arxiv.org/abs/1710.10903

[29] Jason Weston, Antoine Bordes, Sumit Chopra, Alexander M. Rush, Bart van Merriënboer, Armand Joulin, and Tomas Mikolov. 2015. Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks. arXiv:1502.05698 [cs.AI] https://arxiv.org/abs/1502.05698

[30] Da Xu, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. 2020. Inductive Representation Learning on Temporal Graphs. arXiv:2002.07962 [cs.LG] https://arxiv.org/abs/2002.07962

[31] Lu Yi, Jie Peng, Yanping Zheng, Fengran Mo, Zhewei Wei, Yuhang Ye, Yue Zixuan, and Zengfeng Huang. 2025. TGB-Seq Benchmark: Challenging Temporal GNNs with Complex Sequential Dynamics. arXiv:2502.02975 [cs.LG] https://arxiv.org/abs/2502.02975

[32] Le Yu, Leilei Sun, Bowen Du, and Weifeng Lv. 2023. Towards Better Dynamic Graph Learning: New Architecture and Unified Library. arXiv:2303.13047 [cs.LG] https://arxiv.org/abs/2303.13047

[33] Ling Zhao, Yujiao Song, Chao Zhang, Yu Liu, Pu Wang, Tao Lin, Min Deng, and Haifeng Li. 2020. T-GCN: A Temporal Graph Convolutional Network for Traffic Prediction. *IEEE Transactions on Intelligent Transportation Systems* 21, 9 (Sept. 2020), 3848–3858. https://doi.org/10.1109/tits.2019.2935152