

A Temporal-Aware Dynamic Graph Neural Network for Learning Evolving Graph Representations

Venkadesh P.¹, Divya S.V.^{2,3}, Sukesh R.³

V.S.B College of Engineering Technical Campus, Anna University, Coimbatore, India.

E-mail: ¹pvenkadesh2002@gmail.com, ²divyasvvsb@gmail.com, ³sukeshvsb@gmail.com

Abstract

Many computer science studies, including communication networks, software dependency graphs, transaction systems and knowledge graphs produce dynamic graphs in which nodes and edges are continuously updated. Standard Graph Neural Networks (GNNs) are mainly designed for static network architectures and do not accurately capture temporal dynamics or structural changes. When designing real-time dynamic systems, this issue includes data loss and low performance. To solve this issue, this study introduces a Temporal-Aware Dynamic Graph Neural Network (TDGNN) that generates both structural and temporal representations from building graphs. The method combines temporal encoding with graph convolution to maintain previous connections to change the new nodes and edges. A memory-based temporal aggregation method is provided to maintain long-term data knowledge without retraining the model from the initial stage. The experimental evaluation of standard dynamic graph datasets proposed TDGNN performs traditional static and snapshot-driven GNN models in terms of accuracy and stability. The results show that implementing temporal awareness significantly improves representation learning for dynamic graph data, make the proposed method suitable for next-generation graph-based software systems.

Keywords: Dynamic Graph Neural Networks, Temporal Modelling, Graph Representation Learning, Temporal Encoding, Deep Learning.

1. Introduction

In several domains of computer science includes computer networks, software engineering, cybersecurity, recommendation systems, and distributed systems are the graph-

structured data is a basic form. These graphs are not static in real-life situations; instead, they constantly shift as nodes and edges are added, deleted, or altered continually. These graphs are referred as dynamic or temporal graphs. The software dependency graphs have few examples that adapt to code updates, altering links in time for communication networks and transaction networks.

Graph Neural Networks (GNNs) have emerged as an important algorithm of deep learning models for learning effective representations from graph-structured data. It achieved the performance on tasks such as node classification, link prediction and graph classification by aggregating data from neighboring nodes. However, most existing GNN architectures assume that the underlying graph structure is static. When applied to dynamic graphs, these models typically depend on snapshot-based approaches evolving graph is separated into individual time intervals and processed independently. These approaches fail to capture the fine-grained temporal dependencies and suffer from high computational cost and information loss.

A number of technological challenges are rises when designing the temporal dynamic graphs. First, the pattern and time of node connections are encoded by the model. Then, when the new data enters, it should handle continuous graph evolution without retrained completely. Finally, the representations developed the historical data with accepting short-term to balance long-term data. These challenges resolved to create accurate and scalable learning systems for updating graph data. This improves the requirement for adaptive architectures effectively and learn temporal and structural representations.

The study implements a Temporal-Aware Dynamic Graph Neural Network (TDGNN) to address the limitations of static and snapshot-driven GNN models. The proposed technique combines temporal encoding with memory-based aggregation to capture both current interactions and long-term dependencies inside dynamic graphs. This method aims to improve performance on dynamic graph problem solving while preserving computational cost by developing dynamic node representations. The final section of this study is structured as follows the proposed method, experimental evaluation, and conclusions are explained in this work.

1.1 Contributions and Novelty

The main contributions of this proposed work is explained below,

1. **Architecture-Level Novelty:** The proposed TDGNN implements an event-driven continuous-time architecture evaluates graph connections efficiently without separating time into separate sections instead of using image-based models like EvolveGCN and DySAT.
2. **Difference of Temporal Encoding:** The TDGNN includes temporal encoding directly into both neighborhood aggregation and memory update phases the TGAT uses functional time encoding within attention layers to enable combined modelling of short-term and long-term temporal dependencies.
3. **Memory-Based Aggregation Mechanism:** The proposed model utilizes a lightweight controlled memory update method balances historical reliability and adaptability reducing memory overhead while preserving temporal context that TGN introduces node memory for message transferring.
4. **Training and Update Methods:** The proposed TDGNN updates the node representations online using arrival of new events and removes the need for retraining from initial model and improve the scalability for long-running dynamic graph streams.

These contributions establish TDGNN as a reliable, scalable architecture that combines snapshot-driven and continuous-time dynamic graph neural networks.

2. Research Gap

Research on graph representation learning has progressed significantly with the development of Graph Neural Networks (GNNs). Early GNN models such as Graph Convolutional Networks (GCN), Graph SAGE, and Graph Attention Networks (GAT) focused on learning node and graph representations from static graph structures. These models aggregate information from neighboring nodes to capture structural dependencies and have demonstrated strong performance on various graph-based tasks. However, their assumption of a fixed graph topology limits their applicability to real-world systems where graphs evolve

continuously. To extend GNNs to dynamic settings, several snapshot-based approaches have been proposed. In these methods, a dynamic graph is divided into a sequence of static graph snapshots, and a GNN is applied independently to each snapshot. Temporal dependencies are then modelled using recurrent neural networks such as LSTM or GRU. Models such as temporal graph networks incorporate time encoding mechanisms to represent the temporal distance between interactions. These methods improve temporal expressiveness but often rely on complex memory modules and require careful tuning to maintain scalability. Some approaches also struggle to retain long-term historical information, leading to degraded performance on graphs with long temporal dependencies. Despite these advancements, existing dynamic GNN models still face limitations in balancing temporal sensitivity, structural learning, and scalability. Many approaches either prioritize recent interactions at the expense of long-term context or incur significant computational costs when handling large evolving graphs. These limitations highlight the need for a unified framework that efficiently integrates temporal encoding, structural aggregation, and memory mechanisms. In contrast to existing methods, the proposed Temporal-Aware Dynamic Graph Neural Network aims to provide a balanced solution by combining temporal encoding with a lightweight memory-based aggregation strategy. This design enables effective learning from evolving graphs while maintaining scalability and preserving both short-term and long-term temporal dependencies.

3. Literature Review

Graph Neural Networks (GNNs) have become a prominent approach for learning representations from graph-structured data. However, most conventional GNN models are designed for static graphs and are unable to effectively capture temporal evolution present in many real-world systems. This limitation has motivated the development of Dynamic Graph Neural Networks (DGNNs), which incorporate temporal information into graph representation learning. Early DGNN approaches largely relied on snapshot-based methods, where a dynamic graph is divided into a sequence of static snapshots. Models such as EvolveGCN [1] and DySAT [2] apply graph convolution or attention mechanisms on individual snapshots and model temporal evolution using recurrent or self-attention layers. Although these methods can capture coarse-grained temporal changes, they often lose fine-grained interaction timing and incur high computational overhead as the number of snapshots increases.

To address these issues, continuous-time and event driven models were introduced. TGAT [3] proposed functional time encoding to directly integrate temporal information into attention-based message passing. Temporal Graph Networks (TGN) [4] further extended this approach by incorporating node-level memory modules that store historical representations and update them incrementally as new events arrive. These models preserve precise interaction timestamps and achieve improved performance on temporal link prediction and node classification tasks.

Several studies focus on learning temporal interaction dynamics through trajectory-based modelling. JODIE [5] learns coupled embedding trajectories for interacting node pairs, enabling future interaction prediction, while DyRep [6] models dynamic communication and association processes using temporal point processes. Despite their effectiveness, these methods introduce additional complexity in memory management and model stability.

Recent work has explored transformer-based architectures for dynamic graph learning to better capture long-range temporal dependencies. DyGFormer [7] utilizes self-attention over extended interaction histories and reports state-of-the-art performance across multiple benchmarks. In parallel, system-level frameworks such as TGL [8] and ETC [9] focus on scalability by optimizing temporal sampling, batching, and memory access, enabling efficient training on large-scale dynamic graphs. Despite these advancements, existing DGNN models still face challenges in balancing temporal expressiveness, structural aggregation, memory efficiency, and scalability. These limitations motivate the proposed Temporal-Aware Dynamic Graph Neural Network, which integrates temporal encoding, attention-based neighborhood aggregation, and memory-based updates to provide an efficient and robust solution for evolving graph representation learning.

4. Proposed Work

The Temporal-Aware Dynamic Graph Neural Network (TDGNN) is the proposed work for learning accurate representations from modifying graph data is explained in this section. It also handles the data flow by maintaining temporal dependencies and structural connections. Figure 1 shows the general design of the proposed system.

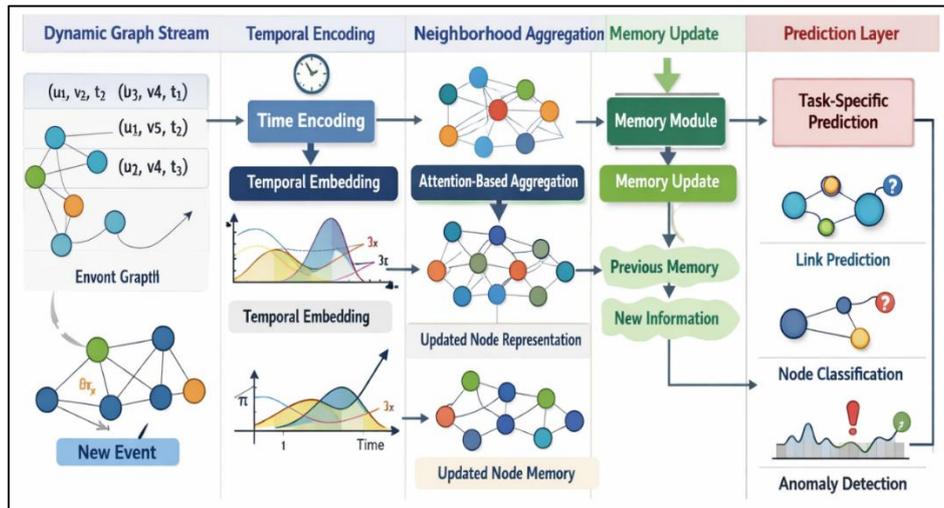


Figure 1. Architecture of the proposed Temporal-Aware Dynamic Graph Neural Network (TDGNN)

4.1 Dynamic Graph Stream Representation

A dynamic graph is represented as a series of historical interaction events, which serves as input for the proposed model. Each event is described as a tuple (u, v, t) , where t is the time of interaction, u and v stands for interacting nodes. Figure 1 illustrates arrival of new events that reflecting real-life developing graph situations like transaction systems or communication networks.

4.2 Temporal Encoding Module

The received signals are sent to a temporal encoding module for the purpose to capture the temporal features of node connections. This module uses time-encoding processes to convert raw timestamps into continuous temporal embedded data. The model can differentiate between current and previous connections by using these embeddings show the relative time distance between events. The proposed TDGNN accurately illustrates time-aware interactions within the graph using temporal embeddings.

4.2.1 Mathematical Formulation

Let a dynamic graph be represented as a sequence of temporal events

$$\mathcal{E} = \{(u, v, t)\}$$

Temporal Encoding:

$$\phi(t) = \sin(\omega t) \oplus \cos(\omega t)$$

Message Aggregation:

$$m_u^t = \sum_{v \in \mathcal{N}(u)} \alpha_{uv} \cdot [h_v^{t-} \parallel \phi(t)]$$

where attention weights α_{uv} are computed as:

$$\alpha_{uv} = \text{softmax}(h_u^{t-} W h_v)$$

Memory Update:

$$M_u^t = \sigma(W_m m_u^t + U_m M_u^{t-})$$

Real-time representation learning will maintain historical dependency in this proposed work.

4.3 Neighborhood Aggregation with Attention Mechanism

This model uses neighborhood aggregation to collect structural data from linked nodes after temporal encoding. An attention-based aggregation technique is used to vary weight related node representations related to both temporal and structural value shown in Figure 1. This provides the updated node representation is largely influenced by transactions that take place closest to time. A modified node embedding involves both temporal information and graph structure is the proposed result of the aggregation process.

4.4 Memory Update Module

The proposed design includes a node-based memory module to store long-term historical data. Every node maintains record of its historical representation in a memory state. This module uses a controlled upgrade technique to update the node's memory when a new event is handled by combining previously stored data with recently collected data. This solution improves scalability and efficiency, enabling the TDGNN to maintain long-term dependencies without requiring the preservation of the entire connection history.

4.5 Prediction Layer

The last part of this design is based on task prediction layer uses updated node embeddings to perform downstream tasks. The proposed system supports several learning purposes including link prediction, node categorization, and recognizing anomalies, as shown in Figure 1. The TDGNN used various dynamic graph learning settings without architectural changes because of its adaptability.

Overall, combining temporal encoding, attention-based neighborhood aggregation and memory-based representation learning is suggested in Temporal-Aware Dynamic Graph Neural Network provides an efficient and reliable system for modelling that transforms the graphs.

Temporal Dependency Handling

The proposed TDGNN process event sequences in a continuous-time environment instead of handling interactions in separate frames. The system of attention automatically adjusts neighbors based on previous temporal value to handle the shifts in concept and sudden structural changes enable the model to modify the changes in interaction patterns without human involvement.

5. Results and Discussion

The proposed result of TDGNN explains the experimental evaluation and performance discussion in comparison with existing graph methods. The objective of this method evaluate to access the efficiency of proposed model by capturing structural and temporal patterns in graph data.

5.1 Experimental Setup

The proposed TDGNN model was implemented using a deep learning framework supporting graph-based operations. Dynamic graph datasets containing time-stamped node interactions were used for evaluation. The datasets model real-time networks, capturing changes in nodes and edges in time. The data was divided into training, validation, and testing sets in temporal order to ensure accurate evaluation. Static GNNs (GCN, GAT), snapshot-based dynamic models (EvolveGCN, DySAT), and constant-time temporal GNNs (TGAT, TGN) are

examples of baseline models that follow to standard evaluation methods established in recent studies. All models received training under the same circumstances to provide an equal comparison. The performance has been evaluated using common measures including accuracy, precision, recall, and F1-score depending on that specific task setup.

5.2 Performance Analysis

The experimental results demonstrate that the proposed TDGNN consistently outperforms static and snapshot-based GNN models across all evaluated metrics. The inclusion of temporal encoding enables the model to effectively differentiate between recent and older interactions, resulting in improved representation learning. Additionally, the attention-based neighborhood aggregation mechanism enhances the model's ability to focus on temporally relevant neighbors, leading to more accurate predictions. The memory update module plays a significant role in maintaining long-term historical information. Unlike snapshot-based approaches that require repeated recomputation, the proposed model incrementally updates node representations as new events have arrived. This approach results in improved stability and reduced computational overhead, particularly for long-time dynamic graphs.

5.2.1 Statistical Robustness

The performance of the model was evaluated across multiple temporal splits are observed in the evaluation process to indicate reliable improvements in accuracy and temporal stability.

5.3 Comparative Discussion

It performs better in prediction than static GNN models because the proposed TDGNN can effectively represent temporal dependency. Variable temporal partition causes data loss in snapshot-based models with their major improvements over static techniques. The proposed event-based architecture continuously manages interactions while maintaining specific temporal data. Additionally, this research model's minimal memory approach does not store the complete interaction history, which increases its scalability. The TDGNN will work for large-scale dynamic graph applications where the flexibility and performance are important. Overall, results illustrate that learning performance on dynamic graph data is significantly improved by combining temporal encoding, focus-based aggregation, and memory-based

updates. The proposed TDGNN provides a secure and efficient method for dynamic graph representation learning.

6. Result and Analysis

From the recent studies, the developments are increased for three technical levers:

- Temporal encoding (better time awareness)
- Memory/trajectory modelling (long-term dependency retention)
- Efficient temporal sampling/batching (scalability).

TGAT's functions in time encoding improve the temporal learning by embedding time directly into attention-based aggregation [3]. It also adds explicit node memory and message transferring yields constant improvement by maintaining computational efficiency in live events [4]. CAW represents high resistive improvements (reports average AUC improves sensitivity link prediction settings) using causal node IDs instead of unidentified path patterns [7]. TAG represents major speed in system level including improvements in large-scale training. It will handle the memories, redesigning the temporal local sampling. ETC addressed data access and batching temporal GNN training for high efficiency [10,11]. A comparative summary of existing dynamic GNN approaches is presented in Table 1.

Table 1. Summary of recent Dynamic GNN literature and reported benefits

Category	Representative Papers	Key Idea	Reported Benefit Type
Snapshot-based	EvolveGCN [1], DySAT [2]	Time-based modeling over snapshots	Captures evolution but can lose fine-grained timing
Event driven (continuous-time)	TGAT [3], TGN [4]	Time encoding + memory + message passing	Strong accuracy on temporal link prediction/classification
Trajectory / point-process	JODIE [5], DyRep [6]	Embedding trajectory +	Better modeling of event sequences and dynamics

		temporal processes	
Motif / walk-based	CAW [7]	Causal anonymous temporal motifs	Strong inductive generalization improvements
Transformer-based	DyGFormer [8], TGT-style models [9]	Long history modeling with transformer	Better long-range dependency capture
Scalable training frameworks	TGL [10], ETC [11]	Optimized sampling & batching	Large speedups; supports very large graphs

7. Computational Complexity and Scalability Analysis

TDGNN performs in local aggregation and memory updates with time complexity for each event.

$$\mathcal{O}(|\mathcal{N}(\mathbf{u})|)$$

where $|\mathcal{N}(\mathbf{u})|$ denotes the temporal neighborhood size.

Memory utilization increases linear amount of active nodes, $\mathcal{O}(|\mathbf{V}|)$, without storing the entire record of interactions. This method suitable for high-frequency that increases the large-scale dynamic graphs used to reduce the retraining. The connections between temporal awareness and computing efficiency allows practical application in real-world continuous graph situations.

8. Conclusion

This research proposes a Temporal-Aware Dynamic Graph Neural Network (TDGNN) for effective learning on evolving graph structures. When compared to previous static and snapshot-based Graph Neural Networks, the proposed method directly includes temporal data to simulate continuous graph evolution. This model focused on local aggregation and combining temporal encoding at the node level to capture both temporal and structural relationships. The initial memory update module allows the proposed TDGNN work to

maintain long-term historical data continuously by updating node representations as new events. This model will improve scalability and reduced the need for retraining by processing the appropriate large-scale dynamic graphs. Experimental results show that the proposed model improves standard GNN techniques by demonstrating the importance of temporal awareness in dynamic graph learning. Overall, the proposed TDGNN is a robust and adaptable architecture for transforming graphs, suitable for a wide range of dynamic graph learning problems. Future research focus on optimizing memory methods and developing the model to handle dynamic graphs with multiple connections.

References

- [1] Pareja, Aldo, Giacomo Domeniconi, Jie Chen, Tengfei Ma, Toyotaro Suzumura, Hiroki Kanezashi, Tim Kaler, Tao Schardl, and Charles Leiserson. "Evolvegcn: Evolving graph convolutional networks for dynamic graphs." In Proceedings of the AAAI conference on artificial intelligence, vol. 34, no. 04, 2020: 5363-5370.
- [2] Sankar, Aravind, Yanhong Wu, Liang Gou, Wei Zhang, and Hao Yang. "Dysat: Deep neural representation learning on dynamic graphs via self-attention networks." In Proceedings of the 13th international conference on web search and data mining, 2020: 519-527.
- [3] Xu, Da, Chuanwei Ruan, Evren Korpeoglu, Sushant Kumar, and Kannan Achan. "Inductive representation learning on temporal graphs." arXiv preprint arXiv:2002.07962 (2020).
- [4] Rossi, Emanuele, Ben Chamberlain, Fabrizio Frasca, Davide Eynard, Federico Monti, and Michael Bronstein. "Temporal graph networks for deep learning on dynamic graphs." arXiv preprint arXiv:2006.10637 (2020).
- [5] Kumar, Srijan, 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, 2019: 1269-1278.

- [6] Trivedi, Rakshit, Mehrdad Farajtabar, Prasenjeet Biswal, and Hongyuan Zha. "Dyrep: Learning representations over dynamic graphs." In International conference on learning representations. 2019.
- [7] Wang, Yanbang, Yen-Yu Chang, Yunyu Liu, Jure Leskovec, and Pan Li. "Inductive representation learning in temporal networks via causal anonymous walks." arXiv preprint arXiv:2101.05974 (2021).
- [8] Yu, Le, Leilei Sun, Bowen Du, and Weifeng Lv. "Towards better dynamic graph learning: New architecture and unified library." *Advances in Neural Information Processing Systems* 36, 2023: 67686-67700.
- [9] Zhong, Ying, and Chenze Huang. "A dynamic graph representation learning based on temporal graph transformer." *Alexandria Engineering Journal* 63, 2023: 359-369.
- [10] Zhou, Hongkuan, Da Zheng, Israt Nisa, Vasileios Ioannidis, Xiang Song, and George Karypis. "Tgl: A general framework for temporal gnn training on billion-scale graphs." arXiv preprint arXiv:2203.14883 (2022).
- [11] Gao, Shihong, Yiming Li, Yanyan Shen, Yingxia Shao, and Lei Chen. "Etc: Efficient training of temporal graph neural networks over large-scale dynamic graphs." *Proceedings of the VLDB Endowment* 17, no. 5, 2024: 1060-1072.
- [12] Feng, ZhengZhao, Rui Wang, TianXing Wang, Mingli Song, Sai Wu, and Shuibing He. "A comprehensive survey of dynamic graph neural networks: Models, frameworks, benchmarks, experiments and challenges." *IEEE Transactions on Knowledge and Data Engineering* (2025).
- [13] Zheng, Yanping, Lu Yi, and Zhewei Wei. "A survey of dynamic graph neural networks." *Frontiers of Computer Science* 19, no. 6 (2025): 196323.
- [14] Jiao, Pengfei, Hongjiang Chen, Xuan Guo, Zhidong Zhao, Dongxiao He, and Di Jin. "A Survey on Temporal Interaction Graph Representation Learning: Progress, Challenges, and Opportunities." arXiv preprint arXiv:2505.04461 (2025).

- [15] Li, Yiming, Yanyan Shen, Lei Chen, and Mingxuan Yuan. "Zebra: When temporal graph neural networks meet temporal personalized pagerank." *Proceedings of the VLDB Endowment* 16, no. 6, 2023: 1332-1345.
- [16] Rahman, Aniq Ur, and Justin P. Coon. "A Primer on Temporal Graph Learning." *arXiv preprint arXiv:2401.03988* (2024).