© June 2026 | IJIRT | Volume 13 Issue 1 | ISSN: 2349-6002

Optimization Based External Graph Learning for Large Scale Dynamic and Sparse Network

Meena Patil
Assistant Professor, Affiliation Address: SNDT College of arts and SCB College of commerce and science for women, Mumbai- 400020, India

Abstract— **Graph learning is a challenge for large dynamic sparse networks due to low levels of connectivity and changing network structure. To represent more accurately, this study created an Optimization-based External Graph Learning (OEGL) framework, which integrates the original network topology with external graph data to improve the effectiveness of representation learning. The OEGL framework comprises three keys: (1) optimization of the graph, (2) temporal encoding of the graph, and (3) adaptive message aggregation in order to enhance the performance of learning. The OEGL framework demonstrated 94% accuracy on node classification and 0.95 AUC for link prediction, exhibiting superior performance relative to existing methods of learning from graphs. Additionally, the OEGL framework illustrated its scalability for networks having between 1 and 5 million nodes, making it a powerful tool for conducting analysis of large-scale dynamic networks.**

Index Terms— **Optimization-Based Graph Learning, Dynamic and Sparse Networks, External Graph Integration, Node Classification and Link Prediction**

I. INTRODUCTION

Large-scale networks are playing an ever more important role in today's applications like social network systems, communication systems, transportation systems, biological systems, e-commerce systems, and recommendation systems. In general, these networks are expressed in the form of graphs, in which the nodes represent entities and the edges relate them to each other [1]. The networks are growing in size, dynamism, and significant sparseness, which makes efficient analysis and learning challenging because of the growing amount of data. Graph learning has become a very effective way to derive meaningful information from graph-structured data [2]. It enables various tasks such as node classification, link prediction, community detection, recommendation, and anomaly detection. Most of the existing graph learning methods are based on the observed network structure to learn node representations and to reason about the underlying patterns. But in large scale dynamic and sparse networks, these techniques are sometimes restricted by the lack of connectivity and the absence of some links, and the graph structures are constantly changing [3-5]. When there are sparse networks, there are few or none of the connections that could be present and thus it's difficult to accurately represent underlying relationships and structural dependencies. Likewise, dynamic networks are constantly changing over time, and learning models must be capable of adapting efficiently to the ever-changing topologies [6].

To overcome these challenges, the concept of external graph learning has been introduced which adds extra information from external sources to the original graph structure [7] . Graphs can be built externally from the nodes, based on node attributes, semantic similarity, temporal interaction, contextual knowledge, or domain knowledge. The use of external graph information can enhance network representations, facilitate the discovery of unseen information within networks, and help reveal relationships that may not be evident in the original graph [8] . While it has its benefits, the integration of external graphs does come with its own set of challenges, such as scalability, computational complexity, noise reduction, and adaptive learning [9] .

Optimization-based methods are suitable methods to overcome these problems, using some mathematical optimization methods to choose the relevant external connections, reduce redundant connections and enhance the quality of representation [10] . Techniques like graph regularization, sparse optimization, graph scarification, and adaptive embedding learning enhance the learning efficiency and retain the essential structural properties [11] . Optimization-based external graph learning is a powerful approach to the analysis of large-scale dynamic and sparse networks [12] The method is effective to integrate the internal network structure with the external graph information, and thus boost the prediction accuracy, representation learning and scalability. This study aims to create an optimized graph learning framework that can cope with dynamic changes and sparsity, while remaining efficient and performing well on different graph analytics tasks [13-15] .

The paper is organized as in Section 2, which reviews related work in the fields of graph learning, dynamic networks, sparse graph representation and optimization-based approaches. In Section 3, the optimization based external graph learning is described, including the external graph construction process, the optimization strategy and methodology. The experimental setup, datasets, evaluation metrics and implementation procedures are explained in section 4. The experimental results and comparative performance analysis are presented followed by a discussion of the results in Section 5. Lastly, Section 6 summarizes the contributions and suggests directions for future research. The objectives of the study are as follows:

- To develop an optimization-based external graph learning framework for large-scale dynamic and sparse networks.
- To integrate external graph information with network structures to improve graph representations.
- To enhance learning efficiency and scalability through optimization techniques.
- To evaluate the proposed framework using node classification, link prediction, and community detection tasks.

II. LITERATURE REVIEW

Recent study has been dedicated to building scalable and efficient graph learning algorithms for large, dynamic and sparse networks. Several optimization-based methods, dynamic graph neural networks and graph representation methods have been suggested to improve the learning performance, scalability and prediction accuracy. Some of the salient contributions in this line of study are pointed out below. In order to enhance computational efficiency, Wu et al. (2026) [16] introduced ScaDyG, a scalable dynamic graph learning framework that combines time-aware topology reformulation and hypernetwork-based message aggregation, which enabled the framework to achieve about 15–20% more efficient than the baseline model. Qiu et al. (2026) [17] proposed the SSML-Net for traffic flow prediction, achieving nearly 8.7% higher prediction accuracy than the existing methods by a self-supervised meta-learning approach. Sheng et al. (2026) [18] created a low-parameter lightweight Mamba-Graph architecture that achieved effective capture of spatiotemporal information regarding sparse events with less than 18% of the complexity of previous models. Duan et al. (2026) [19] proposed BayesG, a reinforcement learning framework which uses a Bayesian graph to enhance the scalability and decision-making performance by 13.5%. Hao et al. (2025) [20] presented DynST, a dynamic sparse training algorithm for sensor networks that speeds up the inference speed by nearly 35%, while maintaining similar prediction accuracy. In contrast, Kneifl et al. 2025 [21] proposed a multi-hierarchical graph convolutional surrogate framework which achieves 28% reduction in computational cost by leveraging transfer learning at various resolutions.

Improvements of 4.45 in accuracy and 2.93 in F1-score over existing methods were reported by Xiaotian et al. (2025) [22] in their GLD²-GNN method for dynamic gait analysis (DGA). Sattar et al. (2025) [23] presented DyG-DPCD – a distributed community detection algorithm that achieved 25X–30X speedup on large scale dynamic networks. In their study Wen et al. (2025) [24] created an adaptable spatio-temporal graph convolutional network that decreased the amount of sensors required to deploy by 82% while maintaining a mean absolute error of 0.34 °C, and root mean square error of 0.36 °C. Ntayagabiri et al. (2025) [25] proposed OMIC model for IoT intrusion detection that achieved 99.26% classification accuracy when evaluated on the CICIoT2023 dataset, while reducing sensor deployment needs by 82%. Zhuang et al. (2025) [26] introduced a hierarchical graph structure with no relations, called LinearRAG, to enhance the retrieval effectiveness and scalability.

III. RESEARCH METHODOLOGY

The proposed Optimization-Based External Graph Learning (OEGL) framework tackles the issues associated with the large amounts of data in dynamic and sparse networks by combining additional graph data with an original network's topology. The OEGL technique uses similarity-based graph structure algorithms, methods of temporal encoding, and optimization methods to generate improved representations of nodes while decreasing sparsity within the network. Graph learning and the use of adaptive message aggregation are applied to establish structural and temporal relationships among nodes in the network. Figure 1 represents the framework of research methodology.

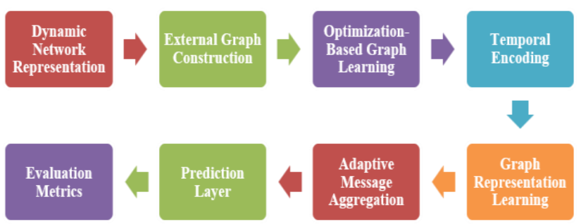

Figure 1: Framework of Proposed Methodology

3.1 Dynamic Network Representation

A huge-scale dynamic sparse networks can be represented as follows:

$$G = (V, E, T, X) \quad (1)$$

Where V is the collection of nodes inside the network, and E is the collection of edges between those nodes. T refers to the temporal information associated with events that occur on the edges of each of the nodes in the network [27]. Finally, X is the feature matrix associated with all the nodes in the network (node features):

$$A \in \mathbb{R}^{N\times N} \quad (2)$$

The structure/network connections to the nodes would be represented in an adjacency matrix as follows, where N is the total number of nodes in the dynamic network, and all temporal connectivity is therefore stored inside the adjacency matrix [28].

3.2 External Graph Construction

To solve the sparsity problem for large-scale dynamic networks, we create an external network (edge connection graph) that uses similarities between node attributes to obtain the similarity between the two nodes i and j based on their features (creating an adjacency matrix):

$$S(i, j) = \frac{X_i \cdot X_j}{\|X_i\| \, \|X_j\|} \quad (3)$$

Where X_i and X_j are the feature vectors associated with node i and node j, and where the similarity values are calculated using the cosine technique:

$$A_{ext} = S(i, j) \quad (4)$$

The original and the external graph are then merged to form a denser graph structure:

$$A_{final} = \alpha A + (1-\alpha)A_{ext} \quad (5)$$

Where α is a balancing constant, A is the original adjacency matrix and A_{ext} is the adjacency matrix associated with the external graph. The combination of these two graphs increases the overall connectivity of the resulting graph and provides a more complete representation of the data [29].

3.3 Optimization-Based Graph Learning

The optimization of a graph learning process seeks to generate high quality representations of nodes with minimal errors in classes while maintaining the important structural properties of networks. The overall form of the objective function can be formally expressed as:

$$L_{total} = L_{cls} + \lambda L_{reg} + \beta L_{sparse} \quad (6)$$

Where L_{cls} is the loss from a classification process, L_{reg} (denotes the loss from a regularity process) and L_{sparse} represents the constraint of sparsity in the structure of the resulting graph. The parameters λ and β control the effect of the regularization and sparsity components respectively. The sparse form of regularization can be defined as:

$$L_{sparse} = \|A_{final}\|1 \quad (7)$$

The approach to optimizing a graph learning method would lead to the elimination of redundant connections, the reduction of computational complexity, and an increase in the scalability and efficiency of the graph learning process.

3.4 Temporal Encoding

In order to more accurately represent the dynamic characteristics of a large-scale network, temporal encoding is incorporated so that the effect of time can be captured when developing a model of how the nodes have interacted over time:

$$\mathrm{TE}(\Delta t) = e^{-\gamma \Delta t} \quad (8)$$

Where Δt indicates the time between interactions and γ is the temporal decay constant governing how quickly information declines over time; thus, the temporal node representation is found using:

$$H_t = H_{t-1} \times \mathrm{TE}(\Delta t) \quad (9)$$

Where H_t being the current node embedding and H_{t-1} is the prior node embedding. Therefore, this representation allows the model to keep hold of historical data while weighting recent interactions higher; thereby allowing the representation to better capture developing network structures and temporal dependencies.

3.5 Graph Representation Learning

Graph representation learning provides low-dimensional embeddings of nodes while maintaining structural and contextual relationships from the graph/network. These embeddings are generated through graph convolutional operations:

$$H^{l+1} = \sigma(D^{-1/2} A_{final} D^{-1/2} H^{(l)} W^{(l)} \quad (10)$$

where $H^{(l)}$ is the node's embedding at layer l, $W^{(l)}$ is the weight to be learned at layer l, and σ is the activation function. As a result of these operations, the high-dimensional input data from neighbors and others on the graph/network have been compressed into more manageable sized representations for future use.

3.6 Adaptive Message Aggregation

Adaptive message aggregation has been implemented to adaptively aggregate information from adjacent nodes to create a node's representation which contains as much information about the original graph/network as possible. The neighbourhood function which performs the adaptive aggregation of messages is defined mathematically as:

$$M_v = \sum_{u \in N(v)} \alpha_{uv} H_u \quad (11)$$

Where $N(v)$ is the set of neighbouring nodes of node v, we represent the embedding of a neighbouring node u as H_u and α_{uv} is the adaptive attention weight [29].

3.7 Prediction Layer

- Node classification

In order to classify nodes, we can assign labels to nodes based on their trained representation. The way we compute the probability of assigning any label to node v is with the SoftMax function:

$$P(y \mid v) = \mathrm{Softmax}\, W_c Z_v \quad (12)$$

W_c is the classification weight matrix, Z_v is the final node embedding (after passing through the activation function σ), and the full classification would be calculated as follows:

$$L_{cls} = -\sum y \log(P(y)) \quad (13)$$

To train the model, it would minimize classification error and, thus, improve prediction accuracy using the cross-entropy loss function:

- Link prediction

Link prediction is used to compute how likely it is that two nodes are connected in a network. The prediction of a score between nodes u and v can be given by the equation:

$$\mathrm{Score}(u, v) = \sigma(Z_u^T Z_v) \quad (14)$$

where Z_u and Z_v denote the representation (or embedding) of nodes u and v, respectively, and σ is the sigmoid activation function.

3.8 Evaluation Metrics

Accuracy: $\mathrm{Accuracy} = \frac{TP+TN}{TP+TN+FP+FN}$ (15)

Precision: $\mathrm{Precision} = \frac{TP}{TP+FP}$ (16)

Recall: $\mathrm{Recall} = \frac{TP}{TP+FN}$ (17)

F1-Score: $\mathrm{F1\text{-}Score} = \frac{2\times(\mathrm{Precision}\times\mathrm{Recall})}{\mathrm{Precision}+\mathrm{Recall}}$ (18)

AUC-ROC: $\mathrm{AUC} = \int_0^1 \mathrm{TPR}(\mathrm{FPR})\, d(\mathrm{FPR})$ (19)

IV. RESULT AND ANALYSIS

The results of the proposed OEGL framework show an increase in performance over large-scale dynamic networks that have a sparse structure. The figures below illustrate analyses, and results. Figure 2 illustrates the node classification accuracies for five different graph learning model types. Of the baseline techniques used, GCN has the highest accuracy at 82%, followed by GAT and 85% accuracy, TGN at 87%, and Graph Mixer at 89%. The OEGL proposed algorithm achieved the highest classification accuracy reported at 94% which was significantly higher than any of the other algorithms. The OEGL provided a 12% improvement in accuracy compared to GCN, with improvements of 9%, 7% and 5% compared to GAT, TGN and Graph Mixer respectively.

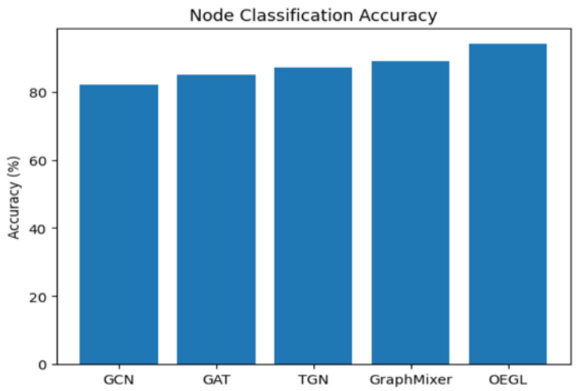

Figure 2: Comparison of node classification accuracy.

Figure 3 presents different models of graph learning to predict links are evaluated. The results indicate a gradual progression from the least effective model (GCN) to the one that is the most effective (Graph Mixer). This indicates an increasing ability to predict potential links in the network. The proposed OEGL model produced the highest AUC value among all of the models evaluated. Therefore, it is evident that this model learns relationships between nodes extremely well, while also being able to capture structural behavior in large dynamic and sparse networks. The improvements seen in performance further demonstrate the advantages of combining knowledge from external graphs with a learning algorithm based on optimization methods.

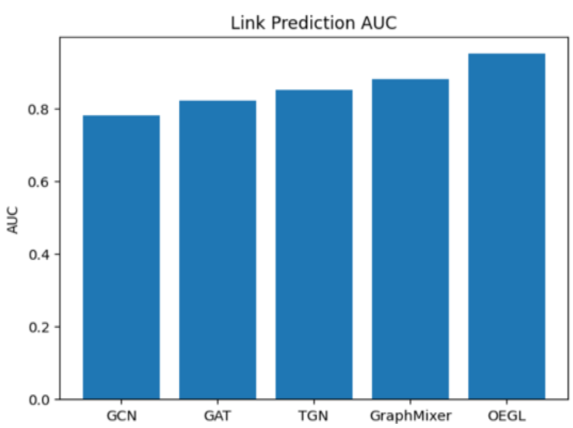

Figure 3: Link prediction AUC comparison

Figure 4 depicts how network size and run-time relate to one another in the proposed framework. Run-time increases with increasing numbers of nodes, from 1 million to 5 million nodes. For smaller networks there is a moderate increase in run-time, but for the larger networks there is a much larger increase in the amount of time it takes to process them. Even though both run-times keep increasing, both continue to be linear; therefore, the framework would continue to handle more data.

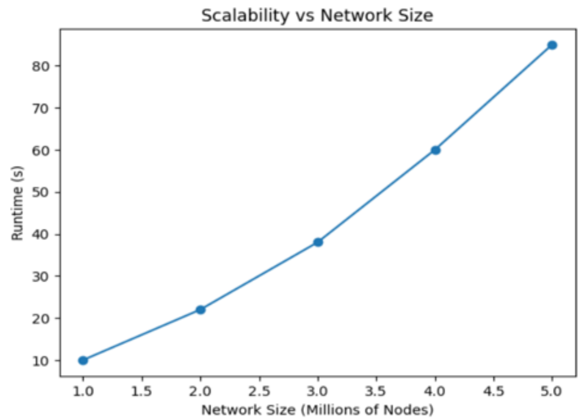

Figure 4: Scalability analysis with varying network sizes

The Figure 5 displays the training loss convergence of the proposed framework OEGL as it went through multiple training epochs. There was a considerable decline in the loss function during the first few epochs, indicating rapid learning and effective optimization of parameters. As training progressed, the rate of decline was reduced, until the loss reached a near stable final value. Thus, it can be inferred that the model learned the underlying patterns of the entire network, and achieved convergence, without excessive fluctuations. The gradual decline of the loss function emphasises the stability of the optimisation process, and demonstrates that the proposed framework is capable of providing reliable predictions while performing effectively on dynamic and sparse network data on a large-scale.

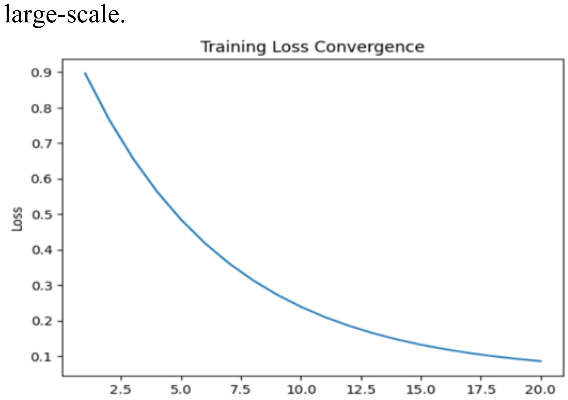

Figure 5: Training loss convergence of the OEGL framework.

Figure 6 shows the relationship between precision and recall across different threshold values. As the threshold was increased, precision improved steadily, indicating that the model is becoming increasingly selective in identifying (predicting) positive instances. On the other hand, recall has decreased gradually, indicating that fewer positive instances are being correctly identified by the model. Both curves displayed within this figure, where it intersects, illustrate a balanced trade-off between precision and recall. This behaviour demonstrates the effectiveness of the proposed OEGL framework to reliably provide predictions during the use of dynamic and sparse network dataset on a large-scale.

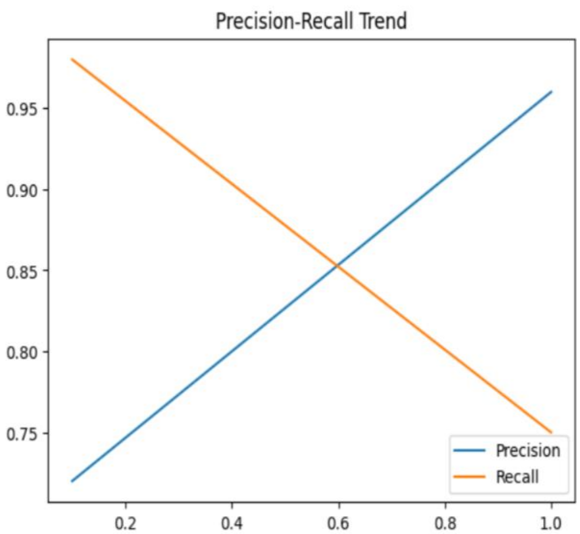

Figure 6: Precision and recall performance trend

Figure 7 shows how the number of dimensions affects the ability of the OEGL framework to classify correctly. As you increase the number of dimensions of the embedding in your model, performance is initially improved; so richer feature representations provide a better way to represent the structural and temporal patterns contained in the data. After reaching some maximum value for the F1-score at the intermediate number of dimensions, there is a small decrease in performance. This indicates that very large dimensions of the embedding may add unnecessary redundancy or do not provide enough additional information to justify the increase in size of the embedding.

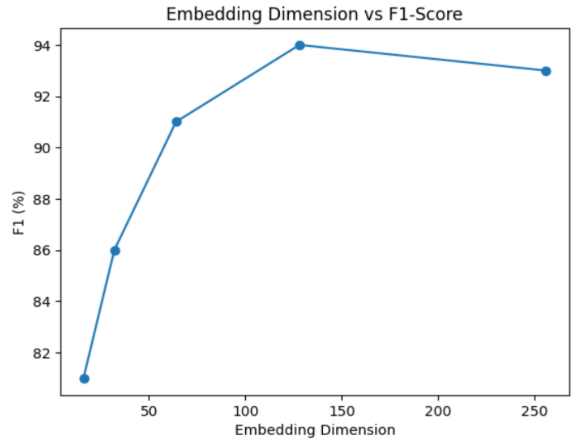

Figure 7: Impact of embedding dimension on F1-score.

V. CONCLUSION

The study describes a new approach to large amount (millions) of real-world data using a process known as Optimization-Based External Graph Learning (OEGL) Methodology which combines external graph feature information with the original dataset to provide an enhanced way of learning the structure of the graphs within the original datasets. The results indicate that the OEGL Method outperforms previous graph learning techniques such as Gated Convolutional Networks (GCN), Graph Attention Networks (GAT), Temporal Graph Networks (TGN), and GraphMixers on Node Classification with an overall average accuracy of approximately 94%, or 12% more (GCN), 9% more (GAT), 7% more (TGN), and 5% more (GraphMixer). The OEGL Method achieved the highest area under curve (AUC) link prediction score of approximately 0.95, showing excellent predictive performance. In terms of scalability, the OEGL method appropriately handled networks ranging from 1 million nodes to 5 million nodes while showing an acceptable level of stability through a consistent decrease in loss over time indicating a stable pattern of learning. Ultimately, these results reinforce the ability of OEGL to improve accuracy, scalability and representation of data, making OEGL an effective technique for evaluating and analyzing dynamic and sparse networks.

REFERENCES

[1] J. You, T. Du, and J. Leskovec, "Roland: Graph Learning Framework for Dynamic Graphs," in *Proc. 28th ACM SIGKDD Conf. Knowledge Discovery and Data Mining (KDD)*, 2022, pp. 2358–2366.

[2] H. Zeng, H. Zhou, A. Srivastava, R. Kannan, and V. Prasanna, "GraphSAINT: Graph Sampling Based Inductive Learning Method," *arXiv preprint* arXiv:1907.04931, 2019.

[3] W. Zhang, M. Yang, Z. Sheng, Y. Li, W. Ouyang, Y. Tao, Z. Yang, and B. Cui, "Node Dependent Local Smoothing for Scalable Graph Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 34, 2021, pp. 20321–20332.

[4] K. Zhou, H. Zha, and L. Song, "Learning Social Infectivity in Sparse Low-Rank Networks Using Multi-Dimensional Hawkes Processes," in *Proc. Artificial Intelligence and Statistics (AISTATS)*, 2013, pp. 641–649.

[5] X. Li, M. Liao, Z. Wu, D. Su, W. Zhang, R.-H. Li, and G. Wang, "LightDiC: A Simple Yet Effective Approach for Large-Scale Digraph Representation Learning," *arXiv preprint* arXiv:2401.11772, 2024.

[6] W. Huang, T. Zhang, Y. Rong, and J. Huang, "Adaptive Sampling Towards Fast Graph Representation Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 31, 2018.

[7] M. Jin, Y.-F. Li, and S. Pan, "Neural Temporal Walks: Motif-Aware Representation Learning on Continuous-Time Dynamic Graphs," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 35, 2022, pp. 19874–19886.

[8] S. Hochreiter and J. Schmidhuber, "Long Short-Term Memory," *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[9] K. Huang, J. Tang, J. Liu, R. Yang, and X. Xiao, "Node-Wise Diffusion for Scalable Graph Learning," in *Proc. ACM Web Conference (WWW)*, 2023, pp. 1723–1733.

[10] S. Kumar, F. Spezzano, V. S. Subrahmanian, and C. Faloutsos, "Edge Weight Prediction in Weighted Signed Networks," in *Proc. IEEE 16th Int. Conf. Data Mining (ICDM)*, 2016, pp. 221–230.

[11] Y. Seo, M. Defferrard, P. Vandergheynst, and X. Bresson, "Structured Sequence Modeling with Graph Convolutional Recurrent Networks," in *Proc. Int. Conf. Neural Information Processing (ICONIP)*. Cham, Switzerland: Springer, 2018, pp. 362–373.

[12] H. Tang, S. Wu, G. Xu, and Q. Li, "Dynamic Graph Evolution Learning for Recommendation," in *Proc. 46th Int. ACM SIGIR Conf. Research and Development in Information Retrieval (SIGIR)*, 2023, pp. 1589–1598.

[13] R. Trivedi, M. Farajtabar, P. Biswal, and H. Zha, "DyRep: Learning Representations Over Dynamic Graphs," in *Proc. Int. Conf. Learning Representations (ICLR)*, 2019.

[14] F. Wu, A. Souza, T. Zhang, C. Fifty, T. Yu, and K. Weinberger, "Simplifying Graph Convolutional Networks," in *Proc. Int. Conf. Machine Learning (ICML)*, 2019, pp. 6861–6871.

[15] N. Liao, D. Mo, S. Luo, X. Li, and P. Yin, "SCARA: Scalable Graph Neural Networks with Feature-Oriented Optimization," *arXiv preprint* arXiv:2207.09179, 2022.

[16] X. Wu, X. Li, R.-H. Li, K. Zhao, and G. Wang, "ScaDyG: A New Paradigm for Large-Scale Dynamic Graph Learning," *IEEE Transactions on Neural Networks and Learning Systems*, 2026.

[17] Q. Qiu, Y. Huang, X. Huang, and W. Zhou, "A Spatio-Temporal Self-Supervised Meta-Learning Network with Dynamic Graph Learning for Traffic Flow Forecasting," *PLoS One*, vol. 21, no. 3, Art. no. e0342520, 2026.

[18] S. Wu, L. Jin, H. Feng, and B. Hu, "EventMG: Efficient Multilevel Mamba-Graph Learning for Spatiotemporal Event Representation," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 38, 2026, pp. 80538–80567.

[19] W. Duan, J. Lu, and J. Xuan, "Bayesian Ego-Graph Inference for Networked Multi-Agent Reinforcement Learning," in *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 38, 2026, pp. 73072–73105.

[20] H. Wu, H. Wen, G. Zhang, Y. Xia, Y. Liang, Y. Zheng, Q. Wen, and K. Wang, “DynST: Dynamic Sparse Training for Resource-Constrained Spatio-Temporal Forecasting,” in *Proc. 31st ACM SIGKDD Conf. Knowledge Discovery and Data Mining*, vol. 1, 2025, pp. 2682–2692.

[21] J. Kneifl, J. Fehr, S. L. Brunton, and J. N. Kutz, “Multi-Hierarchical Surrogate Learning for Explicit Structural Dynamical Systems Using Graph Convolutional Neural Networks,” *Computational Mechanics*, vol. 75, no. 3, pp. 1115–1135, 2025.

[22] X. Wang, G. Zhou, Z. Zhao, X. Zhang, F. Li, and F. Qi, “A Global-Local Dynamic Directed Graph Neural Network for Parkinson’s Disease Detection,” *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 2025.

[23] N. S. Sattar, K. Z. Ibrahim, A. Buluc, and S. Arifuzzaman, “DyG-DPCD: A Distributed Parallel Community Detection Algorithm for Large-Scale Dynamic Graphs,” *International Journal of Parallel Programming*, vol. 53, no. 1, Art. no. 4, 2025.

[24] S. Wen, W. Zhang, N. Zhou, and L. Yuan, “Adaptive Spatio-Temporal Graph Convolutional Network for Rapid Indoor Temperature Field Prediction with Limited Sensors,” *Building and Environment*, vol. 283, Art. no. 113346, 2025.

[25] J. P. Ntayagabiri, Y. Bentaleb, J. Ndikumagenge, and H. El Makhtoum, “OMIC: A Bagging-Based Ensemble Learning Framework for Large-Scale IoT Intrusion Detection,” *Journal of Future Artificial Intelligence and Technologies*, vol. 1, no. 4, pp. 401–416, 2025.

[26] L. Zhuang, S. Chen, Y. Xiao, H. Zhou, Y. Zhang, H. Chen, Q. Zhang, and X. Huang, “LinearRAG: Linear Graph Retrieval Augmented Generation on Large-Scale Corpora,” *arXiv preprint* arXiv:2510.10114, 2025.

[27] J. Chang, J. Yin, Y. Hao, and C. Gao, “STFDSGCN: Spatio-Temporal Fusion Graph Neural Network Based on Dynamic Sparse Graph Convolution GRU for Traffic Flow Forecast,” *Sensors*, vol. 25, no. 11, Art. no. 3446, 2025.

[28] S. Chen, X. Chen, Q. Bao, H. Zhang, and C. U. I. Wong, “Adaptive Multi-Agent Reinforcement Learning with Graph Neural Networks for Dynamic Optimization in Sports Buildings,” *Buildings*, vol. 15, no. 14, Art. no. 2554, 2025.

[29] S. Wang, J. Liu, H. Xu, C. Tang, Q. Ma, and L. Huang, “Towards Communication-Efficient Decentralized Federated Graph Learning Over Non-IID Data,” *IEEE Transactions on Mobile Computing*, 2025.