2026 Volume: 02 Issue: 01 www.ijrpa.com ISSN 2456-9995 Review Article

International Journal Research Publication Analysis

Page: 01-10

# CHALLENGES AND ADVANCES IN GRAPH NEURAL NETWORK LEARNING ON LARGE GRAPHS

**[1*]Ms. A. Kamatchi, [2]Dr.V. Maniraj**

[1]Research Scholar, Department of Computer Science, A.V.V.M.Sri Pushpam College (Autonomous),Poondi,Thanjavur(Dt),Affiliated to Bharathidasan University,Thiruchirappalli,Tamilnadu).

[2]Associate Professor, Research Supervisor, Head of the Department, Department of Computer Science,A.V.V.M.Sri Pushpam College(Autonomous),Poondi,Thanjavur(Dt),Affiliated to Bharathidasan University,Thiruchirappalli,Tamilnadu.

**Article Received: 14 December 2025**

**Article Revised: 02 January 2026**

**Published on: 22 January 2026**

***Corresponding Author : Ms. A. Kamatchi**

Research Scholar, Department of Computer Science, A.V.V.M.Sri Pushpam College (Autonomous),Poondi,Thanjavur(Dt),Affiliated to Bharathidasan University,Thiruchirappalli,Tamilnadu).

DOI: https://doi-doi.org/101555/ijrpa.2897

## ABSTRACT

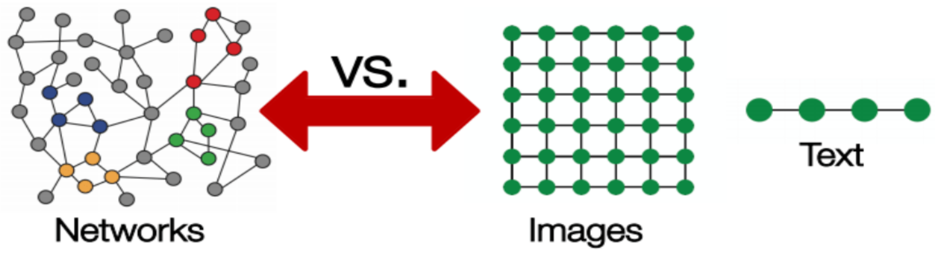
Graph Neural Networks (GNNs) have been widely adopted for learning from non-Euclidean data structures, particularly graphs. Numerous studies have sought to enhance training efficiency and reduce computational complexity when dealing with large-scale graph datasets. This paper surveys convolution-based GNN approaches developed for graph classification and prediction tasks on large graphs. Each method is critically examined with respect to its claimed efficiency gains and complexity reductions, and their limitations are systematically discussed. In addition, the reviewed techniques are evaluated from the perspective of graph dataset characteristics, providing insights into their practical applicability and constraints.

**KEYWORDS:** Graph Neural Networks, Graph Convolutional Networks, Graph Representation Learning, Large Graph Dataset.

## 1. INTRODUCTION

Graph-structured data differs fundamentally from grid-based or sequential data because relationships between entities are irregular, heterogeneous, and often dynamic. In such settings, the influence of a node is governed not by spatial proximity or fixed ordering, but by the topology of the graph itself and the nature of its connections. Capturing these relational patterns requires models that can aggregate and propagate information across connected nodes while preserving structural dependencies. Graph Neural Networks are specifically designed for this purpose, enabling the learning of node, edge, and graph-level representations through message passing mechanisms. By directly operating on graph topology, GNNs overcome the limitations of conventional neural architectures and provide a scalable framework for learning complex, non-Euclidean relationships inherent in real-world graph data.

Graph Neural Networks are broadly divided into Recurrent Graph Neural Networks (RecGNNs) and Convolutional Graph Neural Networks (ConvGNNs). RecGNNs rely on iterative message-passing mechanisms in which nodes repeatedly exchange information with their neighbors until a stable state is reached. In contrast, ConvGNNs learn node representations by aggregating features from local neighborhoods through convolution-like operations. ConvGNNs can be further categorized into spectral-based methods, which draw on spectral graph theory, and spatial-based methods, which perform direct neighborhood aggregation in the graph domain. Since spectral approaches require eigenvalue decomposition of adjacency or Laplacian matrices—leading to high computational overhead—spatial-based methods have become more widely adopted. These approaches also form the foundation of many spatio-temporal GNN models designed to capture the evolution of dynamic graphs over time. Despite significant progress in GNN research for unstructured data, efficiently handling large-scale and highly connected graphs remains a major challenge. Real-world graph datasets often contain millions of nodes and edges, making computation and memory management difficult. To address these scalability issues, several techniques such as neighborhood sampling, graph partitioning, and clustering have been proposed. This work aims to systematically examine and critically analyze these existing methods to better understand their effectiveness and limitations in processing complex, large-scale graphs.



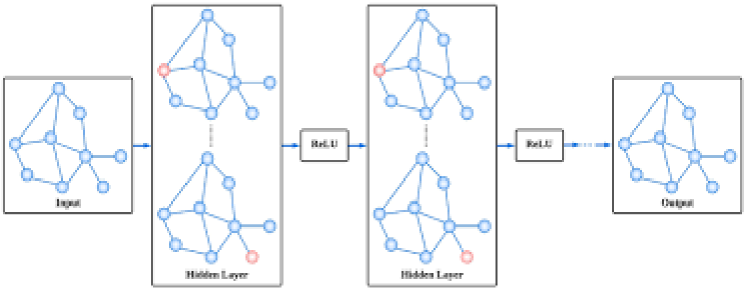
## 2. RELATED WORKS

The first notable survey of Graph Neural Networks was presented , which provided an overview of early graph convolutional models. Subsequent studies narrowed their focus to specific aspects of graph learning, with  examining network embedding techniques and concentrating exclusively on attention-based architectures. A more holistic treatment of GNNs was later attempted , though the analysis was limited to relational reasoning and combinatorial generalization. Broader surveys were introduced, offering a systematic classification of GNNs based on their core learning paradigms. An even more extensive review was conducted, which expanded the scope to include reinforcement learning–based and adversarial GNN frameworks.

While these surveys provided valuable comparisons and organizational frameworks for existing GNN models, they largely overlooked the challenges associated with scaling GNNs to large and dense datasets. Their primary objective was to categorize models and present a unified taxonomy rather than to critically evaluate performance and practicality in real-world, large-scale scenarios. In contrast, this work focuses specifically on GNN architectures designed for large graph analysis and offers a critical assessment of their methodologies, scalability claims, and empirical effectiveness.

### 2.1. CRITICAL REVIEW

This work not merely discusses working and results of existing models for learning large datasets, but does a thorough critical analysis of the same to learn their effectiveness and shortcomings in learning and reducing its complexity.



### 2.2. DATASET-WISE ANALYSIS

This study also examines the datasets employed in existing literature and evaluates their relevance and adequacy for representing complex, large-scale graphs. Detailed dataset statistics are presented, and the reported performance of various approaches is critically discussed in the context of their applicability to real-world graph data. By contrasting experimental results with the structural characteristics of practical datasets, this work highlights gaps between claimed effectiveness and real-world suitability.

The remainder of the paper is organized as follows. Section  introduces fundamental graph and dataset concepts, traces the evolution of Graph Neural Networks, and outlines commonly used notations. Section  reviews models that primarily aim to reduce computational and learning complexity, detailing their underlying algorithms, proposed solutions to earlier limitations, as well as their advantages, computational costs, and drawbacks. Section explores approaches that enhance learning effectiveness without explicitly targeting complexity reduction, noting how these techniques may still be applicable to large-scale graphs. Section  identifies common limitations across existing methods and analyzes their performance using dataset statistics to assess real-world effectiveness. Finally, Section 6 concludes the review.

## 3.DEFINITIONS

**Graph:** A graph $G = (V, E)$is defined by a set of vertices $V = \{v_1, v_2, \dots\}$, where each element represents a node, and a set of edges $E = \{e_1, e_2, \dots\}$, where each edge denotes a connection between one or more vertices in the graph.

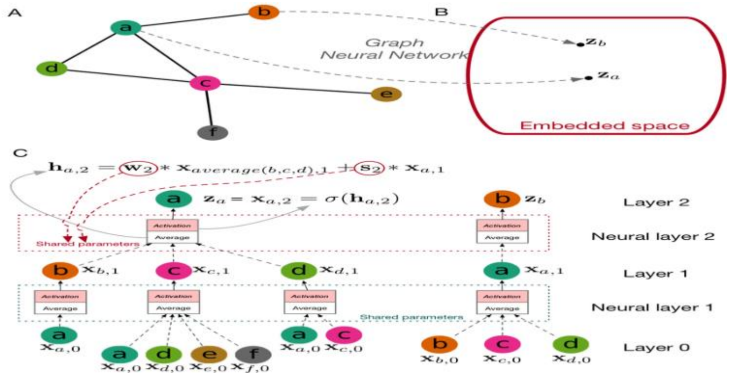
**Receptive Field:** In the context of Graph Neural Networks, the receptive field of a node refers to the collection of neighboring and multi-hop nodes whose features influence the computation of that node’s final representation.

**Seed Node:** A seed node is the initial or reference node from which the neighborhood sampling process is initiated, and for which the learned embedding or representation is the primary objective.

## 4.EVOLUTION OF GRAPH NEURAL NETWORKS

The earliest attempt to model graph-structured data can be traced, which focused on learning representations for directed acyclic graphs. The formal notion of Graph Neural Networks was later introduced  and subsequently extended , both of which employed recurrent architectures for iterative information propagation. The first convolution-based GNN model appeared  and relied on spectral graph theory. This approach was further refined, which demonstrated that both graph dimensionality and the computational cost of Fourier transformations could be reduced through simple mean or max pooling operations. Pooling strategies were later enhanced by optimizing max–min aggregation techniques.

Subsequent developments included, which proposed a semi-supervised framework for node classification, which introduced advanced spectral filters based on Cayley polynomials to improve representational capacity. A major conceptual breakthrough was achieved, which established the message-passing paradigm as a core mechanism for convolutional GNNs. Building on this foundation, numerous models have since been proposed that integrate convolution with diffusion processes, attention mechanisms, and other enhancements to further improve learning performance on graph-structured data.



## 5.NOTATIONS

| Notation | Description |
|---|---|
| *G* | A graph |
| *V* | The vertex set of *G* |
| *v*, *u* | Nodes belonging to *V* |
| *n* | The number of nodes, $n = \|V\|$ |
| *E* | The edge set of *G* |
| *e* | An edge e ∈ *E* |
| *m* | The number of edges, $m = \|E\|$ |
| *N*(*v*) | The neighbourhood set of *v* |
| *d* | The dimension of a node feature vector. |
| *b* | The dimension of a hidden node feature vector. |
| *K* | Number of layers in GNN |
| *c* | The dimension of an edge feature vector. |
| *k*, *l* | The layer index |
| *t* | The time step/iteration index |
| *s* | The batch size |
| *r* | The number of neighbors sampled for each node |
| $\sigma(\cdot)$ | The sigmoid activation function |
| $\sigma h(\cdot)$ | The tangent hyperbolic activation function |
| *A* | The graph adjacency matrix. |

| | |
|---|---|
| $A^T$ | The transpose of the matrix $A$. |
| $A^n$, $n \in Z$ | The $n^{th}$ power of $A$ |
| $D$ | The degree matrix of $A$ |
| $X \in R^{n\times d}$ | The feature matrix of a graph. |
| $x \in R^n$ | The feature vector of a graph in the case of $d = 1$. |
| $x_v \in R^d$ | The feature vector of the node $v$. |
| $X^e \in R^{n\times c}$ | The edge feature matrix of a graph. |
| $x^e_{(v,u)} \in R^c$ | The edge feature vector of the edge $(v, u)$. |
| $X^{(t)} \in R^{n\times d}$ | The feature matrix of a graph at time step $t$ |
| $H \in R^{n\times b}$ | The node hidden feature matrix |
| $h_v \in R^b$ | The hidden feature vector of node $v$ |
| $W$, $\Theta$, $w$, $\theta$ | Learnable model parameters. |

## 6. INDUCTIVE REPRESENTATION LEARNING ON LARGE GRAPHS (GRAPHSAGE)

The Graph Convolutional Network (GCN) employs full-batch gradient descent for graph convolution, requiring all nodes in the graph to be loaded into memory simultaneously. This design makes GCNs impractical for large-scale graphs due to high memory consumption. To address this limitation, GraphSAGE  was proposed, introducing a mini-batch training strategy that significantly reduces memory overhead. By enabling multiple parameter updates within a single epoch, GraphSAGE also achieves faster and more efficient convergence compared to full-batch approaches.

GraphSAGE is an inductive, spatial-based convolutional GNN that learns node representations by sampling a fixed-size subset of a seed node’s neighborhood. At each training iteration, it uniformly selects neighbors from the local vicinity of the target node and aggregates their features to generate embeddings. This neighborhood sampling mechanism allows GraphSAGE to scale to large graphs while maintaining the ability to generalize to unseen nodes.

### 6.1. FAST LEARNING WITH GRAPH CONVOLUTIONAL NETWORK VIA IMPORTANCE SAMPLING (FASTGCN)

FastGCN is graph convolution approach that learns features representations on graph's vertices and interpret convolutions as integral transformation of vertex embedding functions. FastGCN samples using importance-based sampling, unlike in GraphSAGE, a fixed number of vertices and not neighbours for each graph convolutional layer; it samples layer- wise. As the sampling is importance-based, the nodes which influence the seed node majorly, are selected. The model can be represented as in Eq.:

$$H^{(l+1)} = \sigma(\hat{A}H^{(l)}W^{(l)})$$

## 6.2. GATED ATTENTION NETWORKS FOR LEARNING ON LARGE AND SPATIOTEMPORAL GRAPHS (GAAN)

GaAN is a gated attention based convolutional GNN, modelled, performs sampling like GraphSAGE, however, with two major differences. At each sampling step it samples minimum of number of neighbouring nodes or, certain maximum number of nodes determined by a hyperparameter. It also merges any repeatedly sample node for a different seed node but of the same mini-batch. The gated attention, which is the main modification to previous models, can modulate the amount of attended content via the introduced gates. The model also involves transforming graph aggregators into Gated Graph Recurrent Unit (GGRU) which can be used for spatial-temporal learning.

## 6.3.ADAPTIVE SAMPLING TOWARDS FAST GRAPH REPRESENTATION LEARNING (ADAPT)

A variant of FastGCN, this general framework is an inductive top-down layer-wise sampling-based convolution framework that approximates optimal sampling by conditionally selecting lower layer nodes based on upper layer nodes. The approach is based on the premises of common neighbourhood for nodes across a layer, i.e., all parent nodes have same sampled neighbours. The effect is of having similar sampling flow for neighbourhood of nodes of a layer which intends to reduce learning complexity.

## 6.4.STOCHASTIC TRAINING OF GRAPH CONVOLUTIONAL NETWORKS WITH VARIANCE REDUCTION (STOGCN)

StoGCN is a stochastic approximation based convolutional GNN which improvises by employing the historical representation of nodes' activations to reduce variance in the sampled nodes in order to reduce the receptive-field size. Instead of recursively calculating a node's activation representation every time using its neighbours' activations at previous layers, it maintains an approximated representation for each node which is updated at every layer with newly learned representations. The model estimator has a zero variance and is referred to as control variate

## 6.5. GRAPH CONVOLUTIONAL NEURAL NETWORKS

PinSage is GCN based recommender algorithm that performs low-latency random walks on graphs for importance- based neighbourhood sampling of nodes. Here, the importance- based sampling is performed by selecting nodes with highest normalized count visits of multiple random walks. The model applies multiple convolutions in a localized set-up of small neighbourhood nodes to learn embeddings of each node for multiple features. The information gain in each convolution with respect to feature-type is stacked to get more comprehensive embeddings. The algorithm uses max-margin based loss function with an intent to maximize the inner product of embedding of the query item and the corresponding related item and minimize the inner product of the query item and an unrelated item.

## 4.6. LARGE-SCALE LEARNABLE GRAPH CONVOLUTIONAL NETWORK (LGCN)

LGCN [28] is a spatial based GCN that transforms generic graphs into grid-structure to apply standard one-dimensional CNN convolution for feature learning of graph nodes. It learns representations of neighbourhood nodes for the seed node and arranges them in matrix with features forming the columns and rows populated with feature-values for each neighbourhood node. The rows are then sorted based on the feature-values and top few rows and correspondingly nodes are selected for defining representation of the seed node.

## 5. GEOMETRIC GRAPH NEURAL NETWORKS

Geometric Graph Convolutional Networks (Geom-GCN) were proposed to address two key limitations of message-passing neural networks (MPNNs): the loss of structural information caused by indiscriminate neighborhood aggregation and the difficulty of capturing long-range dependencies in disassortative graphs. In such graphs, influential nodes may be several hops away, and conventional aggregation mechanisms often fail to account for their impact, leading to suboptimal representations. To overcome this, Geom-GCN introduces a geometric aggregation framework that projects nodes into a continuous latent embedding space and constructs structural neighborhoods based on geometric relationships within that space. A bi-level aggregation mechanism is then applied over these structurally informed neighborhoods to effectively update node feature representations, enabling improved learning of both local and long-range dependencies.

## 6. PERFORMANCE EVALUATION

Based on the preceding analysis, it is clear that most existing methods rely heavily on neighborhood sampling to perform convolution over large-scale graphs in order to reduce computational complexity. However, a key limitation of conventional sampling strategies is their tendency to overlook influential or informative nodes during the sampling process. The omission of such critical nodes can lead to incomplete neighborhood representations, ultimately degrading the quality of learned embeddings and adversely impacting model performance.

## 7. DATASETS REVIEW

It is observed that different data-statistics have been reported by the existing works for the same datasets. While where FastGCN, LCGN and StoGCN reported same *n* of 2,708 and 19,717 for Cora and PubMed datasets respectively; StoGCN reported higher *m* for both the datasets.

## 8. CONCLUSION

In this review, we have detailed the approaches to model large graphs. We have critically analysed each of these approaches and their claims of learning and reducing complexity in large graphs. It is observed that Adapt gives the best micro-F1 accuracy for comparatively smaller datasets Cora, CiteSeer and PubMed while GaAN has the best score in case of larger Reddit dataset. It is to be noted that Adapt’s Cora and PubMed datasets has lesser edges then the same datasets used by StoGCN; but StoGCN has reported only accuracy scores and not micro-F1 scores. Cluster-GCN has reported the best processing time with best memory and time complexity. However, they have processed only highly sparse datasets and not performed learning in dense graphs.

In the final analysis, it cannot be definitely stated that any of the existing approaches does quality learning along with substantially reduction of complexity in true real-world graphs.

## 9. REFERENCES

1. M. M. Bronstein, J. Bruna, Y. Lecun, A. Szlam and P. Vandergheynst, “Geometric Deep Learning: Going beyond Euclidean Data”, *IEEE Signal Processing Magazine*, Vol. 34, No. 4, pp. 18-42, 2017.
2. W. L. Hamilton, R. Ying and J. Leskovec, “Representation Learning on Graphs: Methods

and Applications", *IEEE Data Engineering Bulletin*, Vol. 24, No. 2, pp. 1-24, 2017.

3. J.B. Lee, R.A. Rossi, S. Kim, N.K. Ahmed and E. Koh, "Attention Models in Graphs: A Survey", *Proceedings of International Conference on Artificial Intelligence*, pp. 1-13, 2018.
4. P. W. Battaglia, "Relational Inductive Biases, Deep Learning, and Graph Networks", *Proceedings of International Conference on Artificial Intelligence and Machine Learning*, pp. 1-14, 2018.
5. J. Zhou and S. Hu, "Graph Neural Networks: A Review of Methods and Applications", *Proceedings of International Conference on Machine Learning*, pp. 1-22, 2018.
6. Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks", *Proceedings of International Conference on Network Embedding and Graph Neural Networks*, pp. 1-22, 2019.