Science & Technology Development Journal 2025, 28(4):3863-3869

Open Access Full Text Article

Research Article

A Graph-Based Framework for Complex Traffic Sign Arrangements in Vietnam

Chi Huy Kieu[1,2], Hong Quan Nguyen[3], Thanh Nguyen Tran[1,2], Chi Trung Nguyen[4], Kha Tu Huynh[1,2,*]

Use your smartphone to scan this QR code and download this article

ABSTRACT

As the demand for self-driving cars grows, the reliability of traffic sign recognition is essential for commuter safety. Researchers have explored several machine-learning and deep-learning approaches to traffic sign identification, but Vietnam's unique traffic environments, ranging from complex urban intersections to highways with vertically stacked signs, present unique challenges. While conventional object detection techniques can handle typical urban traffic signs, they struggle with the groups of stacked signs that are commonly found on Vietnamese highways. This study addressed this problem by treating each detected sign as a node in a graph and modeling its spatial and semantic relationships with edges using Graph Neural Networks, which can learn to identify patterns and groupings. This approach not only allows for the accurate detection of each sign but also captures the collective intent of grouped signs in both urban and highway contexts, thereby providing commuters with more reliable and contextually aware guidance when navigating Vietnam's complex traffic sign system.

Key words: Traffic-sign recognition, Graph Neural Network, Spatial relationships, Lane constraints, Sign grouping

[1]*International University, Ho Chi Minh, Vietnam*

[2]*Vietnam National University, Ho Chi Minh City, 700000, Vietnam*

[3]*Japan Advanced Institute of Science and Technology, Ishikawa, Japan*

[4]*University of Tennessee, Knoxville, USA*

Correspondence

Kha Tu Huynh, International University, Ho Chi Minh, Vietnam

Vietnam National University, Ho Chi Minh City, 700000, Vietnam

Email: hktu@vnuhcm.edu.vn

History

- Received: 24-08-2025
- Revised: 12-10-2025
- Accepted: 27-10-2025
- Published Online: 03-11-2025

DOI :

https://doi.org/10.32508/stdj.v28i4.4577

Check for updates

INTRODUCTION

The development of the autonomous car is expected to reshape society as profoundly as the invention of the motor vehicle. Consequently, the development of Advanced Driver Assistance Systems (ADAS) aims to improve driving comfort, safety, and the production of self-driving vehicles[1]. A key practical challenge for ADAS is the reliable detection of traffic signs, which are essential for safe and successful navigation; improving the identification of traffic signs is thus a crucial field in autonomous vehicle development.

Researchers have explored various methods, including machine learning and deep learning, to address this challenge[2–8]. However, detecting stacked signs that convey guiding instructions for multiple lanes remains problematic despite notable advances in recent years. This issue is especially pronounced on Vietnamese highways, where multiple signs—which can include information such as speed limits, lane-usage restrictions, and directional cues—are often stacked together.Correctly identifying and grouping these stacked signs provides ADAS with complete context about permissible actions in a lane for specific vehicle types.For example, combining a speed-limit sign with a motorbike restriction sign provides a complete picture of what actions are allowed in that lane, preventing potential misinterpretation if the signs are read in isolation.

Grouping signs requires an understanding of the spatial relationships of each sign as well as how they relate to each other. If this process is not handled carefully, this can increase complexity, reduce the speed of processing, and put additional strain on detection systems. Graph Neural Networks (GNNs) provide a potential solution for this issue since they excel at modeling interactions among components[9]. Specifically, GNNs can help the system identify the spatial and semantic relationships between neighboring signs, allowing for a more precise grouping and interpretation of their combined meaning. In a GNN framework, each sign is a node, while edges represent functional links or spatial distances.Furthermore, GNNs are capable of scaling efficiently, allowing them to handle scenarios with many stacked signs with minimal computational cost.

This study makes the following three contributions to the literature: first, the application of GNNs, specifically Attention Graph Neural Networks (AGNNs), for modeling spatial and semantic relationships between traffic signs, thus improving the grouping and interpretation of stacked signs. Second, the proposal of a method that addresses the challenges of accurately identifying Vietnamese traffic signals, particularly on expressways. This method utilizes a three-step pipeline involving YOLO for object detection, graph construction for representing relation-

Cite this article : Kieu C H, Nguyen H Q, Tran T N, Nguyen C T, Huynh K T. **A Graph-Based Framework for Complex Traffic Sign Arrangements in Vietnam**. *Sci. Tech. Dev. J.* 2025; 28(4):3863-3869.

ships, and learning spatial and semantic relationships between signs using AGNNs. Third, this approach is validated on a dataset of Vietnamese traffic signs, achieving high accuracy in both detecting individual signs as well as correctly grouping stacked signs, thus highlighting the feasibility of this current approach.

The remainder of this paper is organized into the following sections: Related work, Methodology, Experimental results, Discussion, and Conclusion.

RELATED WORKS

Research on traffic sign detection can be grouped into two broad categories: traditional methods and deep-learning approaches.

Traditional techniques rely on hand-crafted feature extraction combined with machine-learning classifiers such as support vector machines (SVM), k-nearest neighbors (k-NN), or decision trees. These methods train a model on a labeled dataset and then use the trained model to predict labels for new, unseen instances. These algorithms are attractive because of their simplicity and fast training, but they depend heavily on carefully designed features[10]. For example, a decision tree may struggle with variations in illumination, rotation, or perspective unless extensive preprocessing and feature manipulation are applied. Furthermore, traffic sign detection is a multi-class problem; consequently, these classifiers must be capable of handling such problems. For example, the core architecture of SVMs is inherently binary: extending this to multi-class problems would increase complexity and lower performance. Similarly, deepening a decision tree model to capture subtle variations in the input data may lead to larger models that overfit the data, thus compromising classification accuracy.

The Multi-scale Deconvolutional Network is a deep-learning approach that improves traffic sign detection and localization by integrating a Multi-scale Convolutional Neural Network with a deconvolution subnetwork[11]. The architecture comprises three stages: a Convolutional Residual Network, a modified Feature Pyramid Network (FPN), and a multiscale classifier and detector. This design can adapt to different datasets, including the Chinese Traffic Sign Dataset or the German Traffic Sign Recognition Benchmark, highlighting its versatility across a range of traffic sign systems.However, adding multiscale and deconvolution layers increases model complexity, leading to higher computational costs and slower inference times, which are an issue for real-time applications. The complex architecture also increases training time, especially when working with datasets with many sign types.

An FPN is a top-down architecture that integrates high-level semantic features across all scales to produce a multi-scale feature extractor[12]. It can identify both large overhead signs and small road signs within the same frame, making it well-suited to traffic sign recognition. Furthermore, combining FPN with detectors such as YOLO or Faster R-CNN can improve detection and localization across scales. However, FPN requires more computational resources because it extracts features at multiple resolutions; this makes it less appropriate for real-time applications on resource-constrained or low-power devices.

YOLO variants are frequently used for traffic sign detection because they can operate in real time, which is essential for applications such as autonomous driving[13]. YOLO models perform detection in a single forward pass by dividing the input image into a grid and predicting bounding boxes and class probabilities for each cell.This grid-based approach allows YOLO to detect objects at multiple scales as well as capture spatial relationships. However, YOLO treats each traffic sign independently; it does not recognize when signs form a cohesive group or serve a combined purpose. Furthermore, while this method incorporates some spatial context, it does not explicitly model semantic relations or hierarchical structures between detected objects.

Given the dense arrangement of traffic signs on Vietnamese highways and the current state of existing research, this study aimed to develop an approach to organize these signs into coherent instructions that enable drivers to quickly grasp their meaning and maintain safe vehicle control.The following section introduces a graph-based framework for modeling complex traffic sign arrangements in Vietnam.

METHODOLOGY

This section introduces a GNN approach to handling complex traffic sign arrangements involving stacked signs. The system comprises three stages, shown in Figure 1 .

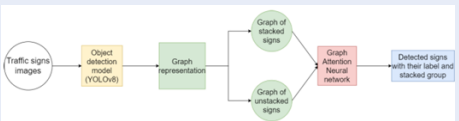

Figure 1: The proposed system.

Starting from user-supplied photos, a YOLO object detection model identifies traffic signs and assigns labels. Each detected sign becomes a graph node, with edges linking nodes that are vertically stacked. The node features of each sign include the bounding-box coordinates (x, y, width, height) and the sign label. Edge features contain a groupId indicating whether the connected nodes belong to the same vertical stack. In order to determine vertical alignment, the center of each box is calculated from its coordinates; this is then used to calculate the absolute difference between the x-centers of two boxes. If this horizontal distance exceeds a certain threshold, the boxes are not considered vertically aligned, as they would be too far apart horizontally, thus preventing misclassification of separate signs as a group.

Once the graph is constructed with appropriate node and edge features, an AGNN learns spatial relationships among nodes to identify stacked sign groups.

Message passing mechanism in Graph Neural Network

The selection of an AGNN over a conventional GNN stems from limitations inherent in the message-passing mechanism of standard GNNs. Before the development of attention blocks, the message-passing layer in GNNs—specifically in Graph Convolution Neural Networks (GCNs) [14] - utilized the process described in Figure 2.

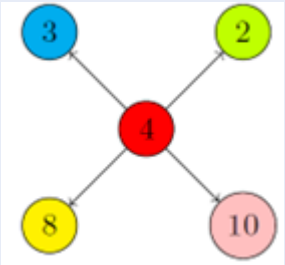

Figure 2: Diagram of a graph used to calculate message-passing.

In Figure 2 , each node is weighted by the number of nodes to which it is connected. The formula used to calculate message-passing rules is presented in Equation (1).

$$h_N = \sqrt{\frac{1}{d_v}} \sum_{u \in N(v)} \sqrt{\frac{1}{d_u}} h_u \quad (1)$$

As an example, applying this message-passing rule to the red node yields Equation (2):

$$h_r = \sqrt{\frac{1}{4}} \left(\frac{h_r}{\sqrt{4}} + \frac{h_b}{\sqrt{3}} + \frac{h_g}{\sqrt{2}} + \frac{h_y}{\sqrt{8}} + \frac{h_p}{\sqrt{10}} \right) \quad (2)$$

This weighting approach has a clear drawback for complex graphs such as social networks.Specifically, the red node is connected to several highly connected nodes; these nodes dominate the message flow because their large numbers of connections will be heavily weighted, even though they may not be relevant to the local neighborhood of the target node. Consequently, the message-passing values generated by this formula will be disproportionately influenced by these highly connected nodes; not much information will be provided about the neighborhood of the red node.

Attention Graph Neural Network in handle group of stacked traffic signs

To address the limitations of the previous weighting approach in GCNs, an attention mechanism is incorporated into the message-passing layer [15]. Instead of a fixed average weighted by the number of connections to a node, the layer computes dynamic attention weights that quantify each neighbor's contribution tothe representation of a target node (Equation (3)).

$$h_N = \sum_{u \in N(v)} \underbrace{softmax_u\left(a\left(h_u, h_v\right)\right)}_{\alpha\, u,v} h_u \quad (3)$$

The initial attention coefficients are derived by projecting input features into a higher-dimensional space so that every node can pay attention to every other node, temporarily disregarding graph structure. A softmax is applied to normalize these scores across each node's neighbors; the value of α u,v is calculated using Equation (4).

$$\alpha u, v = \frac{exp(e_i j)}{\sum_{k \in N} exp(e_i k)} = \frac{exp\left(LeakyReLU\left(W_\alpha^T [Wh_i \| Wh_j]\right)\right)}{\sum_{k \in N} exp(LeakyReLU(W_\alpha^T [Wh_i \| Wh_j]))} \quad (4)$$

where h_i represents the input features of node *i*, W is the learnable weight matrix, || is the concatenation operation, and LeakyReLU is the leaky rectified linear unit activation function. The final layer of the AGNNis described in Equation (5).

$$h_u = \sigma\left(\frac{1}{K} \sum_{k=1}^{K} \sum_{v \in N_u} \alpha_{uv} W^k x_v\right) \quad (5)$$

The model collects a target node's neighbors and its feature vector before applying a linear transformation by multiplying the node by the weight matrix. Two corresponding node states are selected and fed through a neural network layer, after which the computed attention coefficients are summed to produce the updated embedding for the target node (attention coefficient alpha). This describes a single basic layer; in practice, the number of main layers may be doubled by using a multi-head approach. This attention mechanism can help the model weigh the importance of different spatial relationships between signs, allowing it to naturally capture spatial relationships between stacked signs by representing each sign as a node and forming edges according to their relative positions.

EXPERIMENTAL RESULTS

Dataset

The dataset consists of Vietnam traffic sign images sourced from Roboflow and Kaggle. Rather than classifying individual sign types, this study focuses only on detecting the spatial arrangement of the signs, i.e., whether they are stacked or isolated. Figure 3 presents the preprocessing pipeline used in this study.

A total of 1,439 images containing 16,066 annotated bounding boxes were collected. The dataset comprises 82.3% stacked signs and 17.7% non-stacked signs, representative of typical urban settings where space constraints lead to frequent stacking. Furthermore, both sign types are present in 72.8 % of images, providing ample training examples to help the model distinguish between isolated and stacked configurations.

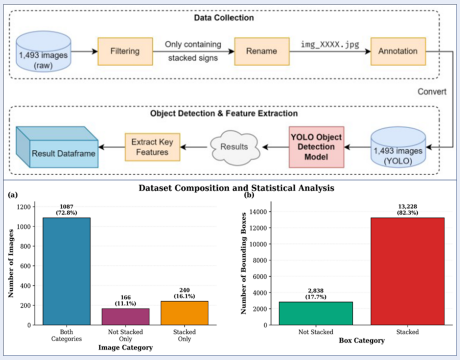

Figure 3: Overview of the data preparation pipeline and dataset characteristics. (a) Illustration of the data preprocessing pipeline. (b) Distribution of stacked and isolated traffic signs in the dataset.

After collection, only images containing stacked signs were retained, resulting in a dataset containing 1,493 samples. File names were standardized to the following format: "imgXXXX.jpg" (i.e., img0001.jpg to img1493.jpg) to simplify downstream processing. Each sign was manually annotated with bounding boxes. These annotations were then converted into a YOLO-compatible .yaml file. The resulting dataset was used to train a customized YOLO model tailored to Vietnamese traffic signs. For each detected traffic sign, the following details were filtered and stored for subsequent analysis.

- The filename was extracted from the results dictionary after YOLO object detection.
- The x_1, y_1 (top-left corner) and x_3, y_3 (bottom-right corner) coordinates of the bounding box as derived from the YOLO results.
- The x_{center} and y_{center} of the bounding box as calculated by averaging the x_1, x_3 coordinates and y_1, y_3 coordinates, respectively.
- The width and height of the bounding box as derived from the YOLO results.
- The confidence score of the detected bounding box as obtained from the "conf" value in the YOLO results.
- The class ID (cls_id) of the detected bounding box as obtained from the YOLO results. Used for reference only (i.e., not used in the next stage).
- The class name (cls_name) of the detected bounding box as obtained from the YOLO results. Used for reference only.
- The detection_id, a unique hash value created from the filename, x_1, y_1, x_3, y_3, and cls_id to ensure the traceability and consistency of the data.

The AGNN setup used in this study comprised 1,442 nodes and 1,911 edges for training and 2,880 nodes and 5,328 edges for testing. The graphs were sampled independently to allow for inductive analysis. Each node is described by ten features, including coordinates, size, confidence, and class, and edges are used to connect neighboring bounding boxes using a strength metric based on spatial proximity. The test graph was denser (an average degree of 3.70 compared to 2.65 for the training graph), reflecting the natural distribution of traffic signs in real-world settings. Table 1 summarizes the hyperparameters configured for each layer of the AGNN, while Table 2 presents the complete statistics of the training and test graphs.

InTable 1, the "Input shape" is the input feature dimensions for each layer, the "Output shape" defines feature dimension after applying the transformation in the layer. For this model layers, the output size is calculated as the product of hidden channels and attention heads. The number of attention heads used in the GATv2Conv layer, allows the network to focus on different aspects of neighborhood information, improving feature representation is shown at the "Heads" and "Dropout" defines the dropout rate applied to the GATv2Conv layers, prevents the model from becoming too dependent on any specific set of features or nodes.

DISCUSSION

The performance of the AGNN setup in classifying stacked traffic signs was evaluated using the F1 score metric and the ROC curve. These metrics are well-suited to the class imbalance present in this dataset,

Table 1: Parameters of each layer in the AGNN setup.

Layer	Input Shape	Output Shape	Heads	Dropout
Node Encoder	(Batchsize, 9)	(Batchsize, 96)		0.0
GATv2Conv (Layer 1)	(Batchsize, 96)	(Batchsize, 384)	4	0.6
Linear 1	(Batchsize, 384)	(Batchsize, 256)		0.0
Linear 2	(Batchsize, 256)	(Batchsize, 96)		0.0
GATv2Conv (Layer 2)	(Batchsize, 96)	(Batchsize, 384)	4	0.6
Linear 1	(Batchsize, 384)	(Batchsize, 256)		0.0
Linear 2	(Batchsize, 256)	(Batchsize, 96)		0.0
Classification layer	(Batchsize, 96)	(Batchsize, 2)		0.0

Table 2: Comparison of training and test graph datasets.

Metric	Training	Test	Difference
Nodes	1,442	2,880	+100% (2 x larger)
Edges	1,911	5,328	+179% (2.8 x more)
Node features	10	10	Same
Edge features	1	1	Same
Avg degree	2.65	3.70	+40% denser
Label shape	[1442,1]	[2880,1]	Same format

where the number of stacked signs significantly outnumber the number of isolated signs. The ROC plot and the F1-score chart are presented in Figure 4.The effectiveness of the AGNN was evaluated against three representative baselines: GCN (spectral convolution with uniform neighbor aggregation) [14], GraphSAGE (inductive spatial aggregation) [16], and APPNP (fixed propagation) [17]. These methods cover the main GNN paradigms, allowing for a comprehensive evaluation of whether adaptive attention better captures relationships among stacked signs compared to simpler aggregation schemes.

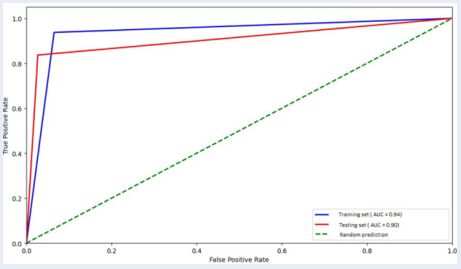

Figure 4: ROC curve and F1 score of the proposed model.

The ROC curve illustrates the trade-off between sensitivity (TPR) and specificity (1 – FPR); curves closer to the top-left corner are indicative of better model performance. The area under this curve (AUC) describes the ability of the model to distinguish between classes.Our model achieves an AUC of 0.94 on the training data and 0.90 on the test data, resulting in an ROC curve that falls near the upper left corner of the chart, highlighting the excellent performance of this model at distinguishing classes.

Table 3 compares the proposed AGNN approach with three baselines. The proposed AGNN achieves an AUC of 0.90 compared to an AUC of 0.72 for the best-performing baseline, a significant improvement of 25%. This gain underscores the importance of learned attention when neighbor relevance varies, as is the case in stacked sign detection. Consistent improvements in both AUC and F1 confirm that adaptive attention effectively addresses this challenging classification task. In summary, the proposed AGNN is proficient in identifying stacked signs.

Traffic sign detection and recognition was then evaluated across a wide range of scenes: a mix of stacked and individual signs, exclusively stacked signs on highways, and exclusively stacked signs in traffic typical of urban settings. A majority of images used

Table 3: Comparison of the performance of different GNN methods for stacked sign detection. Best results are shown in bold.

Method	AUC Score	F1 Score
YOLO + GraphSAGE	0.69 ± 0.023	0.78 ± 0.023
YOLO + APPNP	0.68 ± 0.018	0.65 ± 0.047
YOLO + GCN	0.72 ± 0.045	0.81 ± 0.038
YOLO + AttentionGNN	**0.09 ± 0.005**	**0.84 ± 0.041**

for this assessment were taken from highways in Ho Chi Minh City normal traffic examples were obtained from the same city (Figure 5).

Figure 5: Results of using the model on multiple images from different scenarios in Ho Chi Minh City.

Figure 6: Further processing on the results for better interpretation

In Figure 5, the blue labels correspond to isolated signs, while the red labels denote stacked signs. The model successfully clusters stacked signs using spatial cues, treating them as a single group, while individual signs are correctly identified and remain separated. Its output can be further processed to generate distinct color-coded groups of stacked signs, improving interpretability Figure 6.

Figure 6 demonstrates that the Attention Graph Neural Network can give the detection process context information that traditional methods cannot by extracting and analyzing the relationships between each entity in a large network. This adds more useful value to the way we use the results for downstream task in the ADAS system in the future.

CONCLUSIONS

This study demonstrates that constructing a graph that detects traffic signs and applying an AGNN to handle spatial relationships between each node allows for the reliable identification of stacked and isolated signs. This approach ensures accurate detection while capturing the collective intent of grouped signs, thereby providing commuters with more dependable and context-aware guidance.

Despite Vietnam's complex and often disorganized traffic environment, the proposed framework reliably and efficiently organizes traffic signs in a way that can be further processed to improve interpretability. Future work should explore community detection to automatically cluster a collection of stacked signs without manual labeling. The creation of large language models that can use these clustered signs to provide lane-specific guidelines should also be investigated.

This will enhance ADAS systems and provide commuters on Vietnamese highways with more precise and helpful instructions.

LIST OF ABBREVIATIONS

GNN - Graph Neural Networks.
GCN - Graph Convolution Neural Network.
YOLO - You Only Look Once.
SVM - Support Vector Machine.
AUC - Area under the Curves.

CONFLICT OF INTEREST

The authors declare that they have no competing interests.

AUTHORS' CONTRIBUTIONS

The authors declare that they have no competing interests.

ACKNOWLEDGEMENTS

We gratefully acknowledge AIoT Lab Vietnam for their invaluable support in this research.

REFERENCES

1. Masello L, Castignani G, Sheehan B, Murphy F, McDonnell K. On the road safety benefits of advanced driver assistance systems in different driving contexts. Transp Res Interdiscip Perspect. 2022;15.
2. Lim XR, Lee CP, Lim KM, Ong TS, Alqahtani A, Ali M. Recent advances in traffic sign recognition: approaches and datasets. Sensors (Basel). 2023;23(10):4674.
3. Escalera ADL, Moreno LE, Salichs MA, Armingol JM. Road traffic sign detection and classification. IEEE Trans Ind Electron. 1997;44(6):848–59.
4. Tabernik D, SkoD. Deep learning for large-scale traffic-sign detection and recognition. IEEE Trans Intell Transp Syst. 2019;21(4):1427–40.
5. Mathias M, Timofte R, Benenson R, Van Gool L. Traffic sign recognition—How far are we from the solution? In: and others, editor. In The 2013 international joint conference on Neural networks (IJCNN); 2013. p. 1–8.
6. Yurtsever E, Lambert J, Carballo A, Takeda K. A survey of autonomous driving: common practices and emerging technologies. IEEE Access. 2020;8:58443–69.
7. Huang Z, Li L, Krizek GC, Sun L. Research on traffic sign detection based on improved YOLOv8. Journal of Computer and Communications. 2023;11(7):226–32.
8. Zhu Y, Yan WQ. Traffic sign recognition based on deep learning. Multimedia Tools Appl. 2022;81(13):17779–91.
9. Chen C, Wu Y, Dai Q, Zhou HY, Xu M, Yang S. A survey on graph neural networks and graph transformers in computer vision: A task-oriented perspective. IEEE Trans Pattern Anal Mach Intell. 2024;46(12):10297–318.
10. Mogelmose A, Trivedi MM, Moeslund TB. Vision-based traffic sign detection and analysis for intelligent driver assistance systems: perspectives and survey. IEEE Trans Intell Transp Syst. 2012;13(4):1484–97.
11. Pei S, Tang F, Ji Y, Fan J, Ning Z. Localized traffic sign detection with multi-scale deconvolution networks. In: and others, editor. In 2018 IEEE 42nd annual computer software and applications conference (COMPSAC) 2018 . vol. 1; 2018. p. 355–360.
12. Lin TY, Dollár P, Girshick R, He K, Hariharan B, Belongie S. Feature pyramid networks for object detection. In: and others, editor. In Proceedings of the IEEE conference on computer vision and pattern recognition 2017; 2017. p. 2117–2125.
13. Redmon J, Divvala S, Girshick R, Farhadi A. You only look once: Unified, real-time object detection. In: and others, editor. In-Proceedings of the IEEE conference on computer vision and pattern recognition; 2016. p. 779–788.
14. Kipf TN, Welling M. Semi-Supervised Classification with Graph Convolutional Networks. In: Published as a conference paper at ICLR 2017; 2017.
15. Velić P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y. Graph Attention Networks. In: ICLR 2018; 2018.
16. Hamilton WL, Ying R, Leskovec J. Inductive representation learning on large graphs. Adv Neural Inf Process Syst. 2017;30:1024–34.
17. Gasteiger J, Bojchevski A, Günnemann S. Predict then propagate: Graph neural networks meet personalized PageRank. In: {I}nternational {C}onference on {L}earning {R}epresentations; 2019.