A Survey of GNN-Based Graph Similarity Learning

Nan Jiang, Bo Ning, Jingyang Dong
School of Information Science and Technology
Dalian Maritime University
Dalian, China
{jn1120211454, ningbo, djy1998}@dlmu.edu.cn

2023 8th International Conference on Image, Vision and Computing (ICIVC) | 979-8-3503-3523-1/23/$31.00 ©2023 IEEE | DOI: 10.1109/ICIVC58118.2023.10269885

***Abstract*—With the advancement of deep learning, the application of graphs has become increasingly widespread. Graph similarity learning has gained significant attention as a crucial component in various learning tasks, such as classification, clustering, and subgraph matching, making it a research hotspot in the field of graphs. Currently, graph similarity learning based on graph neural networks (GNNs) primarily involves mapping input graphs to a target space using deep learning models, aiming to approximate the structural distances between graphs in the output space. Firstly, this paper provided a comprehensive review of research on GNN-based graph similarity learning. Based on different graph representation learning approaches, it categorized them into three types: GNN-CNN hybrid models for graph similarity learning, Siamese GNN-based graph similarity learning, and GNN-based graph matching networks for similarity learning. Secondly this paper then provided a detailed analysis of these three types of models. Finally, it discusses the challenges and future research directions in GNN-based graph similarity learning.**

***Index Terms*—deep learning, graph similarity learning, Siamese GNN, graph matching networks**

I. Introduction

Graph, as a widely existing and ubiquitous data form, has been extensively applied in various industries. It possesses the characteristics of direct, natural relationship representation and ease of modeling, playing an important role in data representation and analysis. More and more data are being organized and represented in the form of graphs due to the ability of graph data to clearly express many structured data in the real world. For example, in the modeling process of complex social network data, users can be treated as nodes in the graph, while the relationships between users are represented as edges in the graph. This construction of a social network model allows for an intuitive visualization of the relationships between users and helps us understand and analyze various phenomena and behaviors in social networks. This modeling approach of graph data makes the structure and features of social networks more intuitive and clear [1] [2].

With the widespread application of deep learning in Euclidean data [3] [4], research on graph representation learning has become a new trend. It has also inspired various creative designs in the non-Euclidean domain, particularly in graph neural networks. Graph data is widely applied in domains such as chemistry [5] [6], medicine [7], and citation networks [8] [9]. Overall, the use of graph data is becoming increasingly popular in various fields because it enables more accurate representation and modeling of complex relationships and structures. By mapping real-world data onto the nodes and edges of graphs, we can better understand and analyze the data, thus driving the development of research and applications in various fields.

However, this gives rise to a fundamental problem of how to compute the similarity between two given graphs, namely the graph similarity calculation problem. Unlike NP-hard problems like graph edit distance [10] and maximum common subgraph [11], the powerful representation learning capabilities and scalability of Graph Neural Network (GNN) [12] can effectively handle large-scale data. The graph similarity learning method based on GNN involves learning representations of graphs using GNN, converting the edges and nodes in the graph into low-dimensional vector representations. The learning for a pair of graphs can be achieved through weight sharing and cross mechanisms between the two graphs to output embedded results. Then, a matrix or vector is generated for each graph to represent its embedding. Finally, a dot product layer or fully connected layer is added to predict the similarity score between the two graphs. Therefore, the graph similarity learning method based on GNN, with its strong modeling capabilities, adaptive representation learning, structural invariance, and multi-scale modeling, effectively learns and compares the similarity between graphs, providing strong support for various graph-related tasks.

The remaining organization of this paper is as follows. In Section II, we provide an overview of the development of GNN models and the numerous models built upon the GNN framework. In Section III, we provide a detailed introduction to three models used for graph similarity learning: the GNN-CNN hybrid model, Siamese GNN for graph similarity learning, and GNN-based graph matching networks, with relevant examples for each. In Section IV, we discuss the existing challenges in the field of graph similarity learning. Finally, in Section V, we summarize the content of this paper and express our expectations for the future development of the field of graph similarity learning.

II. Development of GNN

GNN stands for Graph Neural Network, which is a machine learning model for processing graph data. Unlike traditional neural networks, GNNs are inspired by graph theory and network science. The development of GNN is shown in Fig.1.

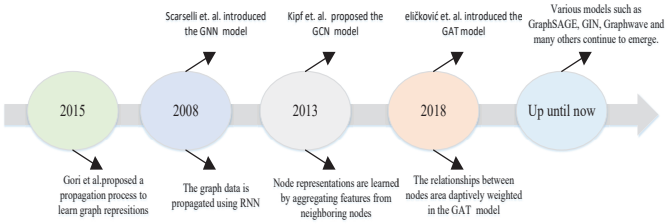

Fig. 1. The development of GNN

GNNs were first proposed by Gori [13] to learn graph representations by employing a process of propagation. In 2008, Scarselli et al. proposed the use of Recursive Neural Network (RN) to process graph structured data, but the idea did not attract much attention at the time due to limitations in computational efficiency and representational power. In 2013, Kipf and Welling proposed a graph neural network model based on convolutional operations, called graph convolutional network (GCN) [14]. GCN learns the representation of nodes by aggregating the features of neighbouring nodes with efficient computation and good representational power, laying the foundation for the development of GNN. In 2018, eličković et al. proposed the graph attention network (GAT) [15], which is based on a self-attentive mechanism. GAT further enhances the representation learning capability of graph data by adaptively weighting the relationships between nodes, which allows flexible learning of the importance and correlation between nodes. To date, GNNs have made rapid progress in the field of graph data analysis and processing. Many new GNN models and variants have been proposed, including GraphSAGE [16], GIN [17], GraphWave [18], GraphGAN [19], etc. Meanwhile, GNNs have been widely used in social network analysis [20], recommender systems [21], bioinformatics [22], computer vision [23] and other fields with impressive results.

III. GNN-BASED GRAPH SIMILARITY LEARNING

Based on the way graph similarity plays a role in representation learning, this paper divides existing research on GNN-based graph similarity learning into three categories, including hybrid GNN-CNN models for graph similarity learning, Siamese GNN for graph similarity learning, and GNN-based graph matching networks.

A. GNN-CNN models for graph similarity learning

GNN-CNN based hybrid networks for graph similarity learning mainly use GNNs to learn graph representations and transform the learned representations into CNNs for graph similarity computation. This model is typically used for classification or regression tasks and often a fully connected layer is added to do the final similarity score calculation.

SimGNN [24], a typical GNN-based method for computing graph similarity, is an improved model based on GSimCNN [25]. Firstly, the model uses the GCN method to obtain the neighbourhood information of nodes and generate the node embedding of the graph as the basis of the whole model. Then, instead of using a simple aggregation method to generate graph-level embeddings, a node attention mechanism is introduced to make each node aware of the entire graph. The formula for this process is as follows,

$$\begin{aligned} h &= \sum_{n=1}^{\mathrm{N}} f_2\left(u_n^{T_n} c\right) u_n \\ &= \sum_{n=1}^{\mathrm{N}} f_2\left(u_n^T \tanh\left(\left(\frac{1}{N}\sum_{m=1}^{N} u_m\right) W_2\right)\right) u_n \end{aligned} \tag{1}$$

where c is the result of a non-linear transformation of the average of the node embeddings, providing global structure and feature information of the graph by learning the weight matrix. f_2 is the sigmoid function. After generating the graph-level embeddings, the model did not use a simple inner product for the data representation, but used (Neural Tensor Network) NTN [26] to indicate the relationship between the two graph-level embeddings to obtain the graph-level interaction scores. However, directly reducing the dimensionality at this stage would neglect the influence of substructures on graph similarity. To address this, the authors bypass the NTN module and takes advantage of the simplicity and efficiency of the histogram to extract the histogram features embedded in the nodes.These histogram features, along with the graph interaction scores that have been normalized and quantified, are then fed into a fully connected layer. Finally, the scores are further reduced in dimensionality to obtain the final similarity scores. This approach captures both the coarse-grained global comparison information for graph-level comparison and the fine-grained comparison information for node embedding comparison, providing a comprehensive view for graph comparison models.

DGE-GSIM [27] is a framework for graph similarity learning that incorporates the learning of edge features in the process of graph embedding learning. Specifically, inspired by the Line Graph [28] in CensN et [29], a node-edge exchanged edge graph is constructed. And then a multi-task learning framework is introduced to fuse features for graph embedding learning. Node embeddings and edge embeddings are treated as two tasks in this framework. Then, node embeddings and edge embeddings are separately aggregated to obtain different graph-level embeddings. Based on this, a three-way graph similarity estimation method is proposed to measure the similarity between graphs. This method consists of three similarity calculation modules: graph-graph interaction, node-node interaction, and node-graph interaction. The following is the formulas for graph-graph interaction.

$$g_v\left(h_v^i, h_v^j\right) = \sigma\left(\left(h_v^i\right)^T W_v^{[1:K]} h_v^j + M_v \begin{bmatrix} h_v^i \\ h_v^j \end{bmatrix} + b_v\right) \tag{2}$$

$$g_{\mathrm{e}}\left(h_e^i, h_e^j\right) = \sigma\left(\left(h_e^i\right)^T W_e^{[1:K]} h^j + M_e \begin{bmatrix} h_e^i \\ h_e^j \end{bmatrix} + b_e\right) \tag{3}$$

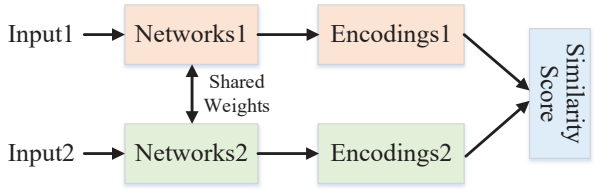

Fig. 2. The architecture of Siamese

where i and j denote different graphs, and h_v and he are node and edge embeddings, respectively, $W_e^{[1:K]}$ and $W_v^{[1:K]}$ denotes a weight tensor, M_v and M_e represents a weight vector, [] denotes a connection operation. Instead of using a simple inner product of h_i and h_j to model the interaction, the authors propose a more sophisticated method to capture the interaction between two graphs. The model connects g_v and g_e as the first part of the similarity embedding. The similarity matrix S between nodes encodes pairwise similarity scores. To ensure that the model is not affected by graph representations, the authors extract histogram features from S as the second part of the similarity embedding. For node-graph interaction, the model first obtains the graph-level embedding of another graph from each node embedding in one graph. Then, using a multi-perspective matching function, it updates the embedding of that node. Finally, the obtained node embeddings are aggregated to obtain the graph's similarity embedding. Lastly, the three parts of the similarity embeddings are concatenated, resulting in a complete similarity embedding. The dimension of the embedding is then reduced using a fully connected neural network to obtain the final similarity prediction score.

B. Siamese GNN models for graph similarity learning

Siamese, in simple terms, is a "connected neural network". Siamese networks typically consist of two neural networks with the same architecture, parameters, and weights. As shown in Fig.2, these two neural networks map their inputs to a new space to form an output encoding. Finally, the encoding is calculated to obtain the similarity between the two inputs.

Siamese GNN models employ a Siamese network architecture, which simultaneously learns representations from two graphs. The learned representations are then used to calculate the similarity based on the representations obtained from GNNs.

A specific encoding scheme for semantic graphs is defined in [30], which enables their utilization within neural networks for evaluating graph similarity. Since both edges and nodes carry rich semantic information that is crucial for assessing graph similarity, the author integrates this semantic information with the workflow structure as input data for the neural network. The NEST graph encoding scheme is employed to create digital vector space encodings. This encoding transforms knowledge with different emphases, such as nodes and edges, into vector representations. Nodes and edges are encoded using one-hot encodings, where a single element is set to 1 and the rest of the vector elements are set to 0, encoding the information in binary form. In this way, all the encodings can be clearly distinguished by the neural network, enabling appropriate processing of these vectors.

After the encoder processes the original graph data, the first embeddings of nodes and edges are obtained. During this process, the embedding components, which are linked to individual embedding vectors, are sufficiently trained by the neural network structure to generate appropriate embeddings. These embeddings are then input into the propagation layer, where the edge results of the graph are iteratively combined with the node embeddings to capture local neighborhood information for each node embedding. At each step, the embedding of an individual node is updated by combining it with the embeddings of the nodes linked by edges. Finally, the aggregator merges the embeddings of all nodes in the graph to form the overall graph embedding. The graph embeddings are then compared to obtain the final similarity score prediction.

C. GNN-based graph matching networks

GNN-based graph matching networks combines GNNs with matching mechanisms during the learning process, aiming to enhance Siamese GNNs using this approach. Furthermore, it incorporates cross-graph interactions during the graph representation learning process.

A Multilevel graph matching networks (MGMN) framework is using for end-to-end computation of similarity between any pair of graph-structured objects in [31]. And then a novel Node-Graph Matching Network that compares each contextual node embedding with the graph-level embedding of the other graph, is introduced. define a multi-perspective matching function that effectively captures rich cross-level interaction features.

First, a multi-layer Graph Convolutional Network (GCN) with a Siamese network architecture is used to generate node embeddings. Then, they compute the cross-graph attention coefficients between nodes and learn the graph-level embedding vector corresponding to another graph from the perspective of a node i. The formulas for this process are as follows.

$$\vec{\tilde{h}}_{G,\text{ avg}}^{2,i} = \sum_{j \in V^2} \alpha_{i,j} \vec{h_j^2}, v_i \in V^1, \tag{4}$$

$$\vec{\tilde{h}}_{G,\text{ avg}}^{1,i} = \sum_{i \in V^1} \beta_{j,i} \vec{h_i^1}, v_j \in V^2 \tag{5}$$

where $\vec{\tilde{h}}_{G,\text{ avg}}^{2,i}$ is the graph-level embedding vector of graph G_2, $\alpha_{i,j}$ is the cross-graph attention coefficient between node i in graph G_1 and the node j in graph G_2, $\vec{h_j^2}$ is the embedding of node j in graph G_2.

Furthermore, utilizing a multi-perspective matching function f_m, the embedding of node i is compared with its corresponding attended graph-level embedding. The resulting similarity feature vector is considered as the updated node embedding for node i. The formulas for this process are as follows.

$$\overrightarrow{\tilde{h}}_i^1 = f_m\left(\overrightarrow{h}_i^1, \overrightarrow{\tilde{h}}_{G,\text{ avg}}^{2,i}, W_m\right), v_i \in V^1 \tag{6}$$

$$\overrightarrow{\tilde{h}}_j^2 = f_m\left(\overrightarrow{h}_j^2, \overrightarrow{\tilde{h}}_{G,\text{ avg}}^{1,j}, W_m\right), v_j \in V^2 \tag{7}$$

Finally, the updated node embeddings are aggregated to obtain the final graph-level embedding. The obtained graph-level embeddings are then used to calculate the similarity score between the two graphs.

A contrastive graph matching network for self-supervised graph similarity learning is proposed in [32]. Combining GNN and Siamese based on contrast learning as the basic node embedding layer, CGMN then performs graph augmentation techniques on each of the two graphs to generate two related graph views. Then the graph view and original graph are embedded into a low-dimensional space through an encoder. By expanding the similarity between positive samples and reducing the similarity between negative samples, the consistency between embedded nodes is maximized, and a cross-view interaction method is proposed to enrich the self-supervised signal. And it helps to maintain the monotonically increasing correlation between node similarity and node consistency. Through the information interaction between the graph and the view of another graph, a self-supervised loss is constructed by maximizing the consistency of positive samples and minimizing the consistency of negative samples. Finally, mean pooling is used as the default aggregation function to obtain graph-level embeddings to calculate the final similarity.

IV. Research Challenges

Graph similarity learning has wide applications in computer science and interdisciplinary fields such as life sciences, drug discovery, and pattern recognition. However, as an important component in current AI-driven pharmaceutical technologies, GNN-based graph similarity learning still faces challenges in graph representation learning, graph invariance, scalability of graphs, and handling sparse graphs.

- **Graph representation learning:** GNNs aim to learn representation vectors for graphs, which can be used in downstream tasks. However, challenges arise in learning effective graph representations due to the complexity of graph structures and the diversity of node features. Therefore, further research is needed to explore more efficient approaches for graph representation learning.
- **Graph invariance:** Similar graphs may have different node orders or edge connections, yet they still represent the same underlying structure. Therefore, GNNs need to possess a certain level of invariance to recognize such specific similarities between graphs. However, existing GNN models face challenges in handling graphs with different orders or connections.
- **Graph scalability:** As the number of nodes and edges increases, graphs become more complex, and the learning complexity of graph similarity calculations also grows. The computational complexity of GNN-based graph similarity methods typically depends on the size of the graph. Therefore, for large-scale graph data, computing graph similarity becomes time-consuming and resource-intensive.
- **Handling sparse graphs:** Sparse graphs have fewer connections between nodes, and the neighbors of a node may not provide enough information to accurately reflect its context and surrounding environment. This can lead to longer propagation paths and slower information transfer within the graph. This not only hinders graph representation learning and feature propagation based on GNNs but also makes it challenging for GNN-based graph similarity learning methods to effectively capture global structures.

V. Summary and Outlooks

In summary, significant research progress has been made in the field of GNN-based graph similarity learning in recent years. By learning graph representation vectors and leveraging the powerful modeling capabilities of GNNs, researchers have proposed various methods to measure and compare the similarity between graphs. This paper provides an overview of existing work in GNN-based graph similarity learning, categorizing them into GNN-CNN models for graph similarity learning, Siamese GNN models for graph similarity learning, and GNN-based graph matching networks. While significant advancements have been made, challenges still remain in GNN-based graph similarity learning, such as graph representation learning, handling sparse graphs, and computational efficiency. In the face of these challenges, researchers are actively seeking solutions by improving GNN models, designing more accurate similarity measurement methods, and optimizing computational efficiency. It is hoped that future research will further explore and address these challenges to advance GNN-based graph similarity learning, enabling breakthroughs in a wider range of domains.

References

[1] KUMAR, Sanjay; MALLIK, Abhishek; PANDA, B. S. Influence maximization in social networks using transfer learning via graph-based LSTM. Expert Systems with Applications, 2023, 212: 118770.

[2] JIA, Peng, et al. SRFA-GRL: Predicting group influence in social networks with graph representation learning. Information Sciences, 2023, 638: 118960.

[3] Michael M Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. 2017. Geometric deep learning: going beyond euclidean data. IEEE Signal Processing Magazine 34, 4 (2017), 18–42.

[4] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and S Yu Philip. 2020. A comprehensive survey on graph neural networks. IEEE transactions on neural networks and learning systems 32, 1 (2020), 4–24.

[5] MOHAPATRA, Somesh; AN, Joyce; GÓMEZ-BOMBARELLI, Rafael. Chemistry-informed macromolecule graph representation for similarity computation, unsupervised and supervised learning. Machine Learning: Science and Technology, 2022, 3.1: 015028.

[6] YANG, Kevin, et al. Analyzing learned molecular representations for property prediction. Journal of chemical information and modeling, 2019, 59.8: 3370-3388.

[7] JAGTAP, Surabhi. Multilayer Graph Embeddings for Omics Data Integration in Bioinformatics. 2023. PhD Thesis. Université Paris-Saclay.

[8] DO, Thanh-Cong, et al. Rapid Response System Based on Graph Attention Network for Predicting In-Hospital Clinical Deterioration. IEEE Access, 2023, 11: 29091-29100.
[9] CUI, Zhaojian, et al. Intelligent Recommendation for Departments Based on Medical Knowledge Graph. IEEE Access, 2023, 11: 25372-25385.
[10] H Bunke. 1983. What is the distance between graphs. Bulletin of the EATCS 20 (1983), 35–39
[11] Horst Bunke and Kim Shearer. 1998. A graph distance metric based on the maximal common subgraph. Pattern recognition letters 19, 3-4 (1998), 255–259.
[12] Scarselli F, Gori M, Tsoi AC, Hagenbuchner M, Monfardini G (2008) The graph neural network model. IEEE Trans Neural Netw 20(1):61–80
[13] Gori M, Monfardini G, Scarselli F (2005) A new model for learning in graph domains. In: Proceedings. 2005 IEEE international joint conference on neural networks, 2005., IEEE, vol 2, pp 729–734
[14] Bruna J, Zaremba W, Szlam A, LeCun Y (2013) Spectral networks and locally connected networks on graphs. arXiv preprint arXiv:1312.6203
[15] VELIČKOVIĆ, Petar, et al. Graph attention networks. arXiv preprint arXiv:1710.10903, 2017.
[16] HAMILTON, Will; YING, Zhitao; LESKOVEC, Jure. Inductive representation learning on large graphs. Advances in neural information processing systems, 2017, 30.
[17] XU, Keyulu, et al. How powerful are graph neural networks?. arXiv preprint arXiv:1810.00826, 2018.
[18] DONNAT, Claire, et al. Learning structural node embeddings via diffusion wavelets. In: Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018. p. 1320-1329.
[19] WANG, Hongwei, et al. Graphgan: Graph representation learning with generative adversarial nets. In: Proceedings of the AAAI conference on artificial intelligence. 2018.
[20] XU, Shouzhi, et al. Rumor detection on social media using hierarchically aggregated feature via graph neural networks. Applied Intelligence, 2023, 53.3: 3136-3149.
[21] ALAGAPPAN, Jothi Kumar; VICTOR, Savaridoss Paul. Deep auto-encoder based clustering algorithm for graph-based web page recommendation system. Concurrency and Computation: Practice and Experience, 2023, e7505.
[22] ZHANG, Xiwen, et al. Learning representation for multiple biological networks via a robust graph regularized integration approach. Briefings in Bioinformatics, 2022, 23.1: bbab409.
[23] DONG, Li; LIANG, Zheng; WANG, Yue. Graph convolutional network-based image matting algorithm for computer vision applications. IET Image Processing, 2022, 16.10: 2817-2825.
[24] BAI, Yunsheng, et al. Simgnn: A neural network approach to fast graph similarity computation. In: Proceedings of the twelfth ACM international conference on web search and data mining. 2019. p. 384-392.
[25] Bai Y , Ding H, Sun Y , Wang W (2018) Convolutional set matching for graph similarity. arXiv preprint arXiv:1810.10866
[26] Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In NIPS. 926–934.
[27] TAN, Wenhui, et al. DGE-GSIM: A multi-task dual graph embedding learning for graph similarity computation. In: 2022 The 6th International Conference on Machine Learning and Soft Computing. 2022. p. 39-47.
[28] Frank Harary and Robert Z Norman. 1960. Some properties of line digraphs. Rendiconti del circolo matematico di palermo 9, 2 (1960), 161–168.
[29] Xiaodong Jiang, Pengsheng Ji, and Sheng Li. 2019. CensNet: Convolution with Edge-Node Switching in Graph Neural Networks.. In IJCAI. 2656–2662.
[30] HOFFMANN, Maximilian, et al. Using siamese graph neural networks for similarity-based retrieval in process-oriented case-based reasoning. In: Case-Based Reasoning Research and Development: 28th International Conference, ICCBR 2020, Salamanca, Spain, June 8–12, 2020, Proceedings 28. Springer International Publishing, 2020. p. 229-244.
[31] LING, Xiang, et al. Multilevel graph matching networks for deep graph similarity learning. IEEE Transactions on Neural Networks and Learning Systems, 2021.
[32] JIN, Di, et al. Cgmn: A contrastive graph matching network for self-supervised graph similarity learning. arXiv preprint arXiv:2205.15083, 2022.