IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received 7 April 2023, accepted 23 May 2023, date of publication 30 May 2023, date of current version 7 June 2023.

*Digital Object Identifier 10.1109/ACCESS.2023.3281338*

RESEARCH ARTICLE

# NFE-PCN: A Node Feature Enhanced Embedding Framework for Pattern Change in Dynamic Network

**TONGXIN ZHANG**[1], **QIANG WEI**[2], **AND LUXI LU**[2]
[1]College of Information System Engineering, PLA Strategic Support Force Information Engineering University, Zhengzhou 450000, China
[2]National Key Laboratory of Science and Technology on Blind Signal Processing, Chengdu 610041, China

Corresponding author: Tongxin Zhang (ztx1234561524@126.com)

This work was supported by the National Key Laboratory of Science and Technology on Blind Signal Processing, Chengdu, China.

**ABSTRACT** Dynamic networks are complex networks as their structures and node features change over time. However, they can better represent the real world, thus attracting the interest of researchers. Although realistic dynamic networks often exhibit changes in their patterns, the existing dynamic network models tend to classify all the snapshots as having the same pattern to learn during their embedding. These embedding models ignore a large amount of information about the patterns of dynamic networks. So, it is necessary to design a dedicated framework for learning the patterns of dynamic networks. Accordingly, this paper proposes a new framework, namely the NFE-PCN framework for effectively extracting information about the change in the patterns of networks. Specifically, the framework first determines the pattern in which the dynamic network snapshot is located, and then enhances the node information between networks by maintaining the same pattern. We conduct experiments with both real and artificial datasets for predicting links and classifying nodes. The obtained results show that the existing model under this framework decreases the computational effort in dynamic network embedding. The performance in the network embedding is improved by up to 29%, which is quite significant.

**INDEX TERMS** Dynamic network, node feature, snapshot, link prediction, graph neural network.

## I. INTRODUCTION

A wide variety of networks exist in the real world. With the proliferation of smart devices, network information can be better captured, allowing people to use network data to better reflect the implicit features lying behind a network, such as the social relationships between individuals in community networks [1], chemical structure of substances in chemical molecular networks [2], and assistance to merchants in recommending products to consumers [3]. Due to the presence of a large number of networks with complex and variable structures, we usually map a network onto a computationally convenient dense space. Network embedding converts a network into a regular machine learning problem by learning a low-dimensional spatial representation of the nodes of the network. Therefore, network embedding has become a matter of interest to researchers.

The existing network embedding methods are broadly classified into three categories: random wander-based network embedding, matrix decomposition-based network embedding and deep learning-based network embedding methods [4]. With the development of deep learning, graph neural networks (GNN) have gradually become the mainstream method for network embedding. These network models usually focus only on known and static networks. However, real-life networks have highly dynamic characteristics, involving frequent changes in the network structure and node features. The process of network changes leaves rich historical information that can be used in subsequent analysis. Therefore, static network embedding methods no longer meet realistic needs, and hence we need to focus our research on dynamic networks.

The associate editor coordinating the review of this manuscript and approving it for publication was Yongming Li.

With the continuous development of graph neural networks and increased attention to dynamic networks, academic research has put increasing attention on dynamic graph neural networks (DGNN). Among them, Kumar et al. [5] proposed the JODIE method, which can compute network embedding anytime without depending on the segmentation of snapshots, but the applied network scenarios are applicable only to bipartite graph networks and its generalization performance is weak. Goyal et al. [6] proposed the DynGEM method, which is a deep auto-encoder based on graph embedding that can handle increasing dynamic graphs. Pareja et al. [7] proposed EvolveGCN as a good blend of GCN and recurrent neural network approaches, where the previously studied dynamic embedding of GCNs trains the static networks in each snapshot separately and then concatenates them through the recurrent neural networks. EvolveGCN uses RNNs to evolve GNN, so that the dynamic information can be captured from the evolved network parameters, and the model can then be made flexible to handle dynamic data without the need for nodes to be present all the time. Sankar et al. [8] also proposed a new approach, namely, DySAT, where a self-attentive mechanism was introduced in dynamic graph neural networks for the first time and a multi-headed attention mechanism was used to parallelize the computation. DySAT is capable to capture certain dynamic periodic information with better embedding performance. In addition to the classical dynamic network embedding methods as described above, many other dynamic network embedding models exist [9], [10], [11]. The above dynamic graph neural network models can embed dynamic networks, but they do not take into account the changes in the network patterns. However, a real situation usually has multiple patterns with significant differences in the networks, and there are sudden transitions between those patterns. For example, a company communicates more internally on Fridays and less at the same time on the weekends. This situation presents a challenge to the existing dynamic graph neural network.

We propose the NFE-PCN framework to cope with the changes in the patterns of dynamic networks. NFE-PCN first discovers patterns of networks and then embeds snapshot networks in different patterns differently based on the discriminated results. The specific operation first determines the pattern to which the snapshot belongs and then passes the node features of the previous snapshot network to the following network of the same pattern in a chronological order. The features are passed for improving the difference of the same network pattern from other network patterns, so as to get a better network embedding effect.

We used three different datasets for predicting links and classifying nodes. The obtained results show that the method with the inclusion of the NFE-PCN framework yields a significant embedding improvement (about 9.2% on average). The main contributions of this paper are areas as follows:

- We illustrate the impact of changes in the patterns of dynamic networks on network embedding.
- We propose an embedding framework that can effectively exploit the changes in the patterns of networks in adapting to the existing dynamic graph neural models.
- We conduct extensive experiments with real datasets to demonstrate that the effectiveness of the existing dynamic network embedding models can be improved significantly by applying the NFE-PCN framework.

## II. PROBLEM DEFINITION

### A. DYNAMIC NETWORK

We denote the dynamic network as $G = (V, E, X)$, where $V$ represents the set of nodes, $E$ represents the set of links and $X$ represents the set of node features in the network, all of which change over time. The dynamic network considered in this paper consists of a series of snapshot networks, i.e., $G = \{g_0, g_1, \ldots, g_{N-1}\}$, where $N$ is the number of snapshots. Each snapshot carries its network structure $g_t = (V_t, E_t, X_t)$,where $V_t$ is the set of nodes, $E_t$ is the set of links and $X_t$ is the node features on snapshot $t$.

### B. NETWORK PATTERN

Network patterns in this paper are defined as the clusters inherent in dynamic networks, which reflect the structure and state of a network during the performance of different tasks and functions. Network patterns are often changed in real life, and one of their concrete manifestations is the regular changes of nodes and links in dynamic networks as shown in Figure 1. The pattern evolution of the network in Fig. 1 is as follows: there are a total of 20 nodes, and they belong to different communities at different moments. At time $T_0$, community 2 has the largest number of nodes and community 0 has the least number of nodes. So, the network is considered to be in pattern A. At time $T_i$, community 2 has many nodes. A large number of nodes from community 2 flow into community 0 and a small number into community 1. So, the network is considered in pattern B. At time $T_j$, community 2 has many nodes again. So, we can consider the network to be in pattern A again.

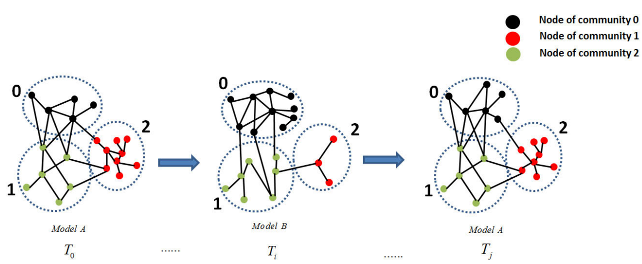


**FIGURE 1.** Schematic illustration of evolution of network pattern.

### C. IMPACT OF NETWORK MODEL ON DYNAMIC NETWORK EMBEDDING

Frequent changes in network patterns may degrade the performance of the existing dynamic graph embedding methods. In order to better illustrate the impact of network pattern changes on the existing dynamic graph embedding methods, we use Dancer [12] to generate a dynamic network with two patterns (pattern A and pattern B) and use the average clustering coefficient of the network as an indicator to indicate the extent of changes in the network as shown in Figure 2. We can visualize the changes in the network pattern, where it is seen that a significant change in the average clustering coefficient takes place at the end of the seventh snapshot, at which point the network is considered to get changed from Pattern A to Pattern B.

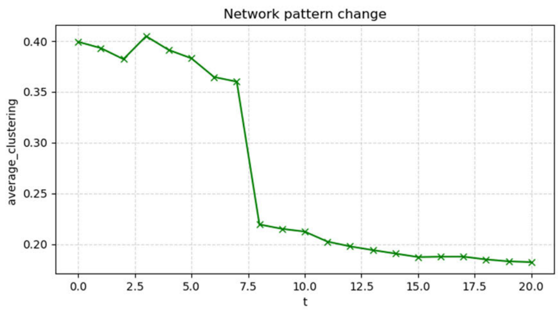


**FIGURE 2. Variation of the average clustering coefficient of the dynamic network generated by Dancer.**

After generating the two patterns of the network, we first train the models using the data of each network pattern separately. We then combine the data the two network patterns to train the models without distinguishing their pattern. Finally, we obtain three models (Model A, Model B, and Model AB). Model A is a DySAT model trained by 8 snapshot data of pattern A, model B is a DySAT model trained by 12 snapshot data of pattern B, and model AB is trained with 20 snapshot data of both patterns. Then, the three models are tested using the data of both network patterns A and B. Finally, the data of models A and B are subjected to a network reconstruction task under these three models, and the reduced network structure is compared with the original network structure. The distribution of the obtained results is shown in Figure 3.

Upon training with the data of pattern A only, model A exhibits the best performance and model B exhibits the worst one. The same results are obtained in the case of the data of pattern B also. The performance of model AB obtained upon training jointly with the data of both patterns A and B is found to be worse than those of model A under data of pattern A and model B under data of pattern B.

The reason for obtaining such results is that when the two patterns are trained jointly, their changes can interfere with the effect of dynamic network embedding. From this, we get that the distinction of network patterns is essential and can affect the performance of the network embedding to some extent. So, we propose the NFE-PCN framework for solving this problem.

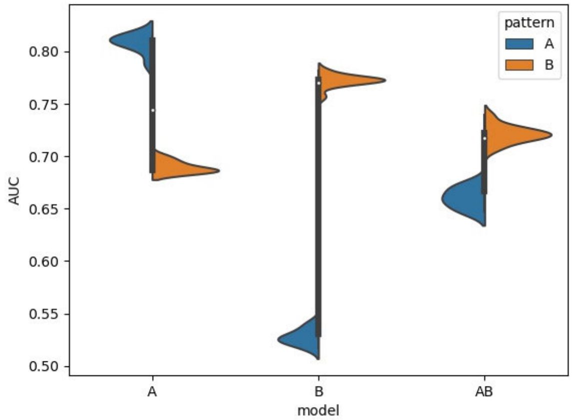


**FIGURE 3. Embedding performance of patterns A and B under different network models. The horizontal coordinates are the four trained models, and the vertical coordinates are the embedding effects (AUC) of patterns A and B. Orange color indicates pattern A, blue color indicates pattern B.**

## III. METHOD

In this section, we describe the NFE-PCN framework in detail, which has two main components: pattern discovery unit and enhancement unit as shown in Figure 4. The pattern discovery unit acts as a discriminator of patterns in the network before dynamic network embedding, and the enhancement unit enables snapshots with pattern labels to reinforce node features in the same pattern network through recurrent neural networks (RNNs). Next, we describe each unit component in detail.

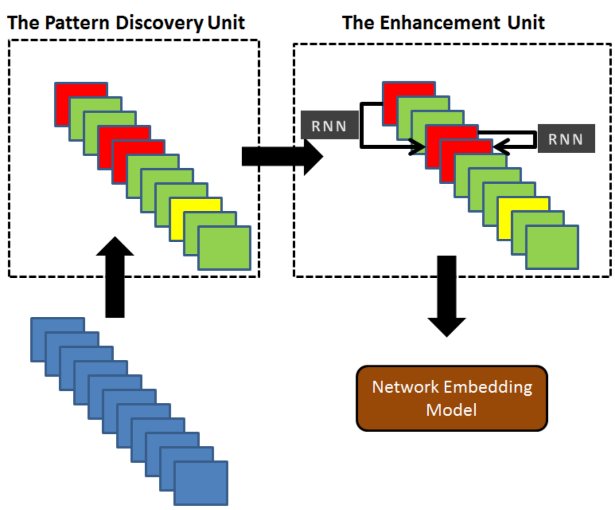


**FIGURE 4. Flow chart of NFE-PCN framework. The pattern labels of the snapshot are obtained after the pattern discovery unit clusters the snapshots. Then, the enhancement unit is used to enhance the node features in the snapshot according to different patterns. Finally, the snapshot is fed into the general dynamic network model.**

### A. PATTERN DISCOVERY UNIT

In this subsection, we discuss the use of the pattern discovery unit. The pattern discovery unit obtains a snapshot of the network in the same pattern by clustering. The unit consists of the following two main steps:

1) Pattern discrimination of network snapshot, and
2) Aggregate snapshots according to network patterns.

In the process of network pattern discrimination, GED (graph edit distance) [13] used in this paper first measures the distance between snapshot networks. Then, we use the DBSCAN clustering algorithm for obtaining snapshots with different patterns by clustering them. This process is defined as follows:

$$J_{ij} = GED(g_i, g_j) \tag{1}$$

$$\boldsymbol{J} = \begin{pmatrix} J_{00} & \cdots & J_{0j} \\ \vdots & \ddots & \vdots \\ J_{i0} & \cdots & J_{ij} \end{pmatrix} \tag{2}$$

$$(g_0^{P_0}, g_1^{P_1}, \ldots, g_{N-1}^{P_K}) = DBSCAN(\boldsymbol{J}) \tag{3}$$

where $g$ is the snapshot, N is the number of snapshots, $\boldsymbol{J}$ is the distance matrix between snapshots, P is the pattern to which the snapshot belongs after clustering, and K is the number of types of patterns. After getting snapshots with pattern labels, we can aggregate multiple adjacent snapshots of the same pattern into one. This process is defined as follow:

$$Agg(g_0^{P_0}, g_1^{P_1}, \ldots, g_{N-1}^{P_K}) = (G_0, G_1, \ldots G_M) \tag{4}$$

where $M$ is the number of snapshots after aggregation; and $Agg$ can be regarded as the maximum selection, minimum selection, specific selection, etc. The maximum selection is specifically represented by selecting all the nodes and links in these networks, which will be assigned to the aggregated snapshots as long as they exist in any of the networks. In order to better understand the process of the pattern discovery unit, we draw an example of pattern change over time, where 13 snapshots are aggregated by the clustering algorithm, and then distributed in time order to obtain Figure 5 showing the pattern change of the dynamic network, where the circular, rectangular, and triangular snapshots are different patterns, i.e., $(0 - T_1)$, $(T_1 - T_2)$, $(T_2 - T_3)$, $(T_3 - T_4)$ are pattern B, pattern C, pattern A, and pattern C, respectively.

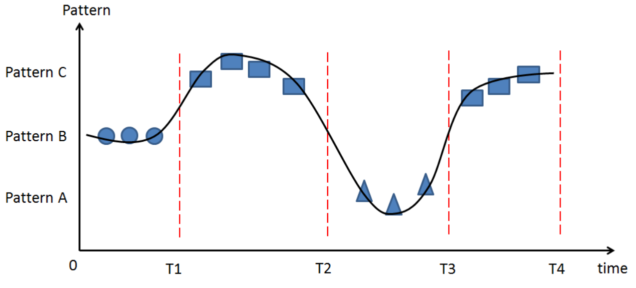


**FIGURE 5. Change in pattern with time.**

In this section, the pattern discovery unit also highlights the valid information in the snapshot network of the same pattern by aggregating neighboring snapshots of the pattern under certain circumstances as shown in Figure 6. The operation of aggregating the snapshots can also reduce the amount of data, which can improve the performance of the model under certain circumstances and significantly reduce the computational time during the model training (see Section IV-G for detail).

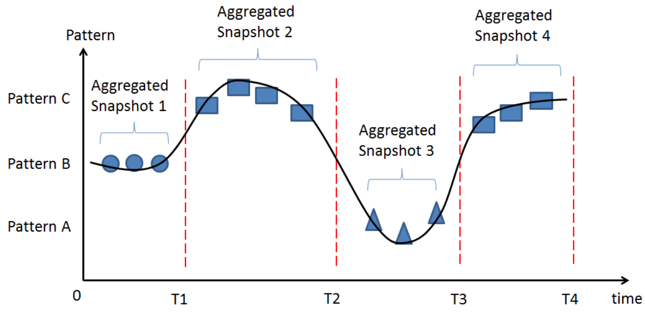


**FIGURE 6. Schematic diagram of aggregating snapshots by pattern.**

### B. ENHANCEMENT UNIT

After processing in the pattern discovery unit, the dynamic network is divided into multiple network patterns. It is worth noting that the snapshot networks in the same pattern often have similar network structures and node features. In order to capture the network pattern evolution, we use the enhancement unit. According to the pattern in which the current snapshot is located, the node features of the previous snapshot in the same pattern are passed to the current snapshot by a recursive neural unit (GRU [14] is used in this paper).

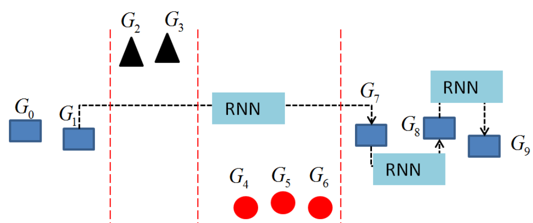


**FIGURE 7. Schematic diagram of node feature enhancement unit.**

In the snapshot without aggregation, the structure of the enhancement unit is schematically shown in Figure 7, which shows 10 snapshots divided into three patterns (rectangular, triangular and circular). We take the rectangular pattern as an example. Since networks $G_0$, $G_1$, $G_7$, $G_8$ and $G_9$ have the same pattern, the node features in $G_1$, $G_7$ and $G_8$ are passed through RNN, to $G_7$, $G_8$, $G_9$, respectively. The unit assigns the information of the snapshot features from the same pattern in the past to those in the current snapshot through an RNN structure, which enhances the node information of the current snapshot as shown in (3), where $X_t^A$ and $X_{t-1}^A$ are the node features in pattern A at moments $t$ and *t-1*, respectively.

$$X_t^A = \begin{cases} RNN(X_t^A, X_{t-1}^A)\mathrm{t} \neq 0 \\ X_t^A \mathrm{t} = 0 \end{cases} \tag{5}$$

In the case of snapshot aggregation as shown in Figure 8, we pass the feature information to two non-adjacent identical pattern snapshots through the RNN structure. In Figure 8, we aggregate by selecting the first snapshot in the adjacent identical pattern snapshots as the aggregated snapshot.

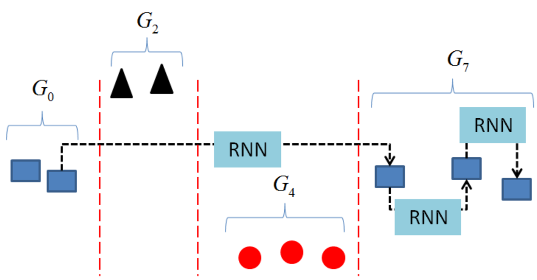


**FIGURE 8. Schematic diagram of feature enhancement unit after snapshot aggregation.**

We focus on the changes in the features of nodes. The nodes in a network have different features in different modes. Since there is some similarity between snapshots of the same pattern, the processing of the enhancement unit can use the pattern information of the network to enhance the node features at the current moment, so that the embedding performance of the model is improved.

## IV. RESULTS

### A. DATASET

In this paper, three datasets are used to test the model performance under the NFE-PCN framework, which includes two real-world datasets and a synthetic dataset generated by Dancer. Their basic information is shown in Table 1. We next present these three datasets separately.

**TABLE 1. Three datasets used in experiments.**

| | Number of nodes | Maximum number of links | Number of snapshots | Number of patterns |
|---|---|---|---|---|
| DAGG | 100 | 1348 | 150 | 4 |
| Core1 | 309 | 3031 | 75 | 3 |
| Core2 | 162 | 1772 | 76 | 2 |

Real-world datasets: We use two publicly available social network datasets, email-Eu-core-temporal-Dept1 and email-Eu-core-temporal-Dept2. These two networks are generated using email data from a large European research institution, and the two datasets are collected from different departments.

Synthetic dataset (DAGG): Dancer is a dynamic network generator that can generate many dynamic snapshot networks based on users' input, such as the number of nodes, number of communities, probability of node feature change, probability of node migration, etc. Since the probability of change between each snapshot generated in Dancer is always the same, we modify it based on Dancer to simulate the real world more accurately. We add two parameters: moment of network pattern mutation, and probability of node migration and node feature change. Once the network mutates, both features and labels of a large number of nodes in the network change (i.e., the community changes in which the nodes are located). We treat this situation as the generation of a new pattern.

### B. TASK ASSIGNMENT

Our proposed NFE-PCN framework can support various tasks. The experiments conducted in this paper use a link prediction task and a node classification task to verify the effectiveness of the framework.

#### 1) LINK PREDICTION

Here, the information before time $t$ is used to predict the network links at time $t + 1$. Since the historical information of a dynamic network is included in the model parameters, we obtain the probability of a link through a fully connected network.

#### 2) NODE CLASSIFICATION

The task of node classification is to predict the label information of the nodes in the network at time t. Due to the small number of publicly available datasets for node classification in dynamic networks, we use only the Dancer dataset for the purpose of demonstration.

### C. EXPERIMENTAL MODEL

In order to demonstrate the superiority of the NFE-PCN framework, three dynamic graph neural network embedding models are selected for validation in this paper, which are GCN-GRU [15], EGCN, DYSAT and DynGEM.

- GCN-GRU is a classical approach that uses recursive mechanisms to obtain dynamic network embedding by concatenating network features.
- EvolveGCN(EGCN) is a classical approach of the type of recursive dynamic graph neural network. Unlike GCN-GRU, it analyzes the network at the level of structure using an RNN to concatenate the parameter weights in the network.
- DYSAT is a typical approach for the dynamic graph neural network with attention mechanism, which uses the self-attention mechanism to learn the connections between snapshots for the dynamic embedding of the network.
- DynGEM is a graph self-encoder approach for node embedding, which uses the self-encoder parameters learned in the previous snapshot as the initialization parameters in the current snapshot, thus enabling dynamic embedding.

### D. EVALUATION METHODOLOGY

The area-AUC metric under the ROC curve is used in the experiments for predicting links, and the MAP (Mean Average Precision) metric is used for classifying nodes. MAP generally refers to the average AP value of all the categories in all networks, which in this paper can be understood as the classification accuracy of all kinds of nodes in all the snapshots. A higher MAP represents a better effect of the model.

### E. RESULTS OF LINK PREDICTION

In this experiment, the control method and NFE-PCN framework are used to test the merits of the model through the link prediction task. The performance results of AUC obtained on the three datasets are presented in Table 2, which shows that the dynamic network embedding model in the NFE-PCN framework obtains a great improvement in performance and achieves a more desirable result, where the EGCN model performance in dataset core2 could be improved by 29%.

**TABLE 2. Performance of link prediction.**

| Method | Core1 | Core2 | DAGG |
|---|---|---|---|
| DySAT | 0.936 | 0.939 | 0.843 |
| EGCN | 0.738 | 0.709 | 0.626 |
| GCN-GRU | 0.844 | 0.877 | 0.725 |
| DyGEM | 0.697 | 0.693 | 0.634 |
| DySAT + NFE-PCN | **0.963** | **0.959** | **0.873** |
| EGCN + NFE-PCN | **0.827** | **0.921** | **0.639** |
| GCN-GRU + NFE-PCN | **0.911** | **0.915** | **0.764** |
| DyGEM + NFE-PCN | **0.732** | **0.803** | **0.664** |

### F. RESULTS OF NODE CLASSIFICATION

The experiments for node classification are conducted on the Dancer dataset, and their comparative performance in the form of histograms is shown in Figure 9, where yellow bars represent the dynamic network model in the NFE-PCN framework and green bars represent the original dynamic network model. Performances of three control methods are improved after incorporating the NFE-PCN framework, thus confirming the effectiveness of the framework.

### G. EXPERIMENTAL ANALYSIS OF AGGREGATION

In order to enrich the performance of the NFE-PCN framework and to demonstrate the clustering role of the pattern discovery unit, we take the snapshots after clustering separately for conducting experiments. We set the experimental objectives to incorporate the original dynamic network model, pattern discovery clustering model, NFE-PCN model, and the NFE-PCN and pattern discovery clustering models. We use the dataset of pattern A for the experiments of link prediction. The obtained embedding results and model training time results are shown in Tables 3 and 4, respectively.

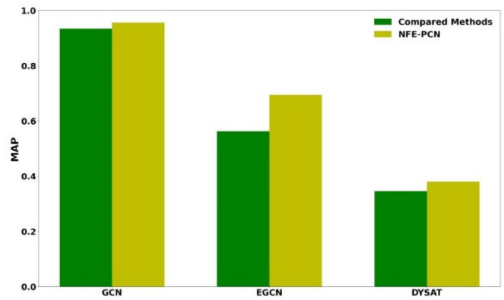


**FIGURE 9. Performance of node classification on Dancer dataset.**

**TABLE 3. Performance of link prediction on core1.**

| | DySAT | EGCN | GCN-GRU | DyGEM |
|---|---|---|---|---|
| Original Model | 0.936 | 0.738 | 0.844 | 0.697 |
| No pattern discovery unit | 0.963 | 0.827 | 0.911 | 0.726 |
| No enhancement unit | 0.956 | 0.724 | 0.873 | 0.703 |
| NFE-PCN | **0.965** | **0.870** | **0.942** | **0.732** |

**TABLE 4. Training time required by dynamic network models.**

| | DySAT | EGCN | GCN-GRU | DyGEM |
|---|---|---|---|---|
| Original Model | 13.44s | 13.11s | 12.23s | 18.34s |
| No pattern discovery unit | 157.51s | 15.56s | 15.93s | 34.43 |
| No enhancement unit | 4.62s | 2.5s | 3.87s | 7.53s |
| NFE-PCN | 24.2s | 4.65s | 5.40s | 15.32s |

The results reported in Table 3 show that the network embedding is generally enhanced when there is no pattern discovery unit, and it is enhanced to approximate the embedding performance of the NFE-PCN framework. The embedding performance of the network is not changed significantly when there is no enhancement unit. This indicates that the units, which play a role in improving the network embedding effect for dynamic networks, are mainly enhanced units. The results presented in Table 4 demonstrate that the training speed of the model decreases significantly after adding only the pattern discovery unit. Although the computation of the model increases after adding the augmentation unit, the training of the network model can be reduced after adding the pattern discovery unit. From Table 4, we can see that the pattern discovery unit can reduce the training time of the model. In summary, we can conclude that the pattern discovery unit reduces the model training time, and the enhancement unit improves the network embedding performance. The NFE-PCN framework that combines both of them has the best comprehensive performance.

### H. EXPERIMENTAL EXPANSION

As discussed in Section II. The three models were trained separately using data of pattern A and B. The corresponding results are shown in Figure 3. After designing the NFE-PCN framework, we put the AB model into the NFE-PCN framework for training and then repeat the same experiments as performed in Section II. The obtained corresponding results are shown in Figure 10.

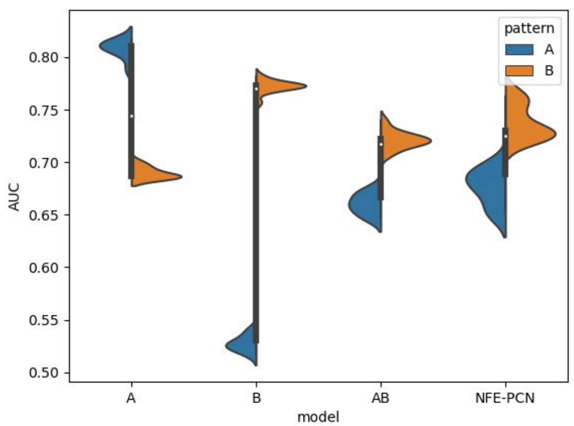


**FIGURE 10. Embedding performance of AB model in NFE-PCN framework.**

We can see from Figure 10 that AB model in the NFE-PCN framework get some degree of performance improvement compared to that of the AB model, but not yet to the same extent as done when exactly the same network pattern is used for training. Therefore, the research on network embedding using network pattern still needs to be explore.

### I. COMPLEXITY ANALYSIS

The NFE-PCN framework proposed in this paper is mainly divided into pattern discovery unit and enhancement unit. A set of a network data has $n$ nodes, $m$ edges, $d$ features, and $S$ snapshots. The framework first obtains the distance between the snapshots using the GED algorithm as it is the minimum operand to find the transformation between networks. The GED algorithm has a complexity of $O(m + n)$. The distance between S snapshots obtained by GED, and obtain the pattern information of the network using the DBSCAN algorithm. Since the complexity of DBSCAN is $O(S \log S)$, so the complexity of the pattern discovery unit is $O(n+m+S \log S)$.

The main role of the augmentation unit is to add the GRU model to the original model to convey the network pattern information. Since the complexity of GRU is equal to $O(nd^2)$, the complexity of enhancement unit also is $O(nd^2)$. So the total complexity of the NFE-PCN framework is $O(n + m + S \log S + nd^2)$. For large-scale networks, $n, m \gg d, S$, the above equation can be approximated as $O(n + m + nd^2)$, where $d$ is a constant. So, it can be concluded that the complexity of the NFE-PCN framework is linear, i.e., $O(n)$. Because of the low complexity of the framework, it has a good scalability.

## V. CONCLUSION

A large number of embedding models for dealing with dynamic networks were proposed and their effectiveness was demonstrated by many researchers. Since dynamic networks in real situations possess many regular implicit features, we propose the NFE-PCN framework. We present the pattern discovery unit and enhancement unit for addressing the pattern discovery of networks and enhancement of network information of the same pattern, respectively. We propose this concept of node feature enhancement as it is closer to real life and it will use more information about the node features in the network.

Our proposed approach currently has some limitations as well. Firstly, for changing the network patterns, we use only the enhancement of network embedding from the direction of node attributes. It does not consider the effect of the network structure level. So, another potential option for this topic is to enhance the network embedding through the network structure. Secondly, our NFE-PCN framework can be applied only to dynamic networks in the form of snapshots. It cannot be applied to continuous dynamic networks, which is a research breakthrough at present.

We note two directions for future work. Firstly, synthetic data is used in this paper as the existing dynamic network dataset is fewer, and the labels and features of nodes could not be changed over time. Therefore, we expect to improve the dynamic network generator (Dancer) to be closer to real life. Secondly, for the snapshot clustering method of the pattern discovery unit, we expect that more features in the network can be extracted in future research to make the clustering of network patterns more accurate.

## REFERENCES

[1] M. E. J. Newman, "Communities, modules and large-scale structure in networks," *Nature Phys.*, vol. 8, no. 1, pp. 25–31, Dec. 2011.

[2] D. K. Duvenaud, D. Maclaurin, J. Iparraguirre, R. Bombarell, T. Hirzel, A. Aspuru-Guzik, and R. P. Adams, "Convolutional networks on graphs for learning molecular fingerprints," in *Proc. Adv. Neural Inf. Process. Syst.*, Sep. 2015, pp. 2224–2232.

[3] L. Zhang, J. Li, Q. Zhang, F. Meng, and W. Teng, "Domain knowledge-based link prediction in customer-product bipartite graph for product recommendation," *Int. J. Inf. Technol. Decis. Making*, vol. 18, no. 1, pp. 311–338, Jan. 2019.

[4] J. Skarding, B. Gabrys, and K. Musial, "Foundations and modeling of dynamic networks using dynamic graph neural networks: A survey," *IEEE Access*, vol. 9, pp. 79143–79168, 2021.

[5] S. Kumar, X. Zhang, and J. Leskovec, "Predicting dynamic embedding trajectory in temporal interaction networks," in *Proc. 25th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2019, pp. 1269–1278.

[6] P. Goyal, N. Kamra, X. He, and Y. Liu, "DynGEM: Deep embedding method for dynamic graphs," *Social Inf. Netw.*, vol. 8, no. 3, pp. 25–31, May 2018.

[7] A. Pareja, G. Domeniconi, J. Chen, T. Ma, T. Suzumura, H. Kanezashi, T. Kaler, T. Schardl, and C. Leiserson, "EvolveGCN: Evolving graph convolutional networks for dynamic graphs," in *Proc. AAAI Conf. Artif. Intell.*, Apr. 2020, vol. 34, no. 4, pp. 5363–5370.

[8] A. Sankar, Y. Wu, L. Gou, W. Zhang, and H. Yang, "DySAT: Deep neural representation learning on dynamic graphs via self-attention networks," in *Proc. 13th Int. Conf. Web Search Data Mining*, Jan. 2020, pp. 519–527.

[9] D. Xu, J. Liang, W. Cheng, H. Wei, H. Chen, and X. Zhang, "Transformer-style relational reasoning with dynamic memory updating for temporal network modeling," in *Proc. Nat. Conf. Artif. Intell.*, vol. 35, Sep. 2021, pp. 4546–4554.

[10] W. Yu, W. Cheng, C. C. Aggarwal, K. Zhang, H. Chen, and W. Wang, "NetWalk: A flexible deep embedding approach for anomaly detection in dynamic networks," in *Proc. 24th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, Jul. 2018, pp. 2672–2681.

[11] W. Cheng, H. Chen, W. Yu, and D. Xu, "Adaptive neural networks for node classification in dynamic networks," in *Proc. IEEE Int. Conf. Data Mining*, Aug. 2020, pp. 1402–1407.

[12] C. Largeron, P. N. Mougel, O. Benyahia, and O. R. Zaïane, "DANCer: Dynamic attributed networks with community structure generation," *Knowl. Inf. Syst.*, vol. 53, no. 1, pp. 109–151, Oct. 2017.

[13] L. Yang and L. Zou, "Noah: Neural-optimized A* search algorithm for graph edit distance computation," in *Proc. IEEE 37th Int. Conf. Data Eng. (ICDE)*, Apr. 2021, pp. 576–587.

[14] K. Cho, B. Van Merriënboer, C. Gulcehre, D. Bahdanau, F. Bougares, H. Schwenk, and Y. Bengio, "Learning phrase representations using RNN encoder–decoder for statistical machine translation," *Comput. Sci.*, vol. 14, no. 1, pp. 1179–1189, Jun. 2014.

[15] J. Chen, X. Wang, and X. Xu, "GC-LSTM: Graph convolution embedded LSTM for dynamic network link prediction," *Appl. Intell.*, vol. 11, no. 8, pp. 1142–1152, Dec. 2018.


**TONGXIN ZHANG** received the B.S.E. degree in electronic science and technology from Beijing Jiaotong University, in 2020. He is currently pursuing the M.S. degree with PLA Information Engineering University. His research interests include dynamic networks and graph neural networks.

**QIANG WEI** received the Ph.D. degree from the University of Electronic Science and Technology of China, in 2022. He has been an Assistant Researcher with the National Key Laboratory of Science and Technology on Blind Signal Processing, since 2015. His research interests include network structure learning and graph neural networks.

**LUXI LU** received the Ph.D. degree from the University of EECS, Peking University of China, in 2010. He was a Visiting Student with CTVR, Trinity College Dublin, from 2008 to 2009. He has been an Associate Research Fellow with the National Key Laboratory of Science and Technology on Blind Signal Processing, since 2019. His research interests include cognitive radio and dynamic resource allocation.