



An integrated simplicial neural network with neuro-fuzzy network for graph embedding

Phu Pham¹

Received: 28 June 2023 / Accepted: 27 April 2024 / Published online: 22 May 2024

© The Author(s), under exclusive licence to Springer-Verlag GmbH Germany, part of Springer Nature 2024

Abstract

In recent years, graph neural network (GNN) has become the main stream for most of recent researches due to its powers in dealing with complex graph data learning problems. However, as most of the recent GNN-based architectures have been mainly designed to only evaluate direct relational structures between nodes. As the results, these techniques are unable to capture the sophisticated multi-way relationships in graph. The multi-way relationships can be represented by both explicit graph-based and complex topological structures. In general, the multi-way relationships in graph can be modelled as simplicial complexes, hyper-graphs, e.g., and can be efficiently preserved under the simplicial neural networks (SNN). There are several notable SNN-based architectures have been proposed recently, such as the well-known simplicial convolutional neural network (SCNN). The SNN-based techniques have shown the competitive performances in handling graph learning. However, most of recent proposed SNN-based architectures are designed upon the deep neural learning paradigm. Therefore, they still encountered several challenges with regard to the feature noise and data uncertainty. To overcome these limitations, in this paper, we proposed a novel integrated SNN and neuro-fuzzy network (NFN) graph embedding technique, called as: SFGE. Our SFGE model is designed to better capture the multi-way structural representation of graph by taking advances of different advanced graph-based and fuzzy-based neural learning techniques. By taking advances of neuro-fuzzy learning approach, our model can efficiently support to eliminate the feature uncertainty/ambiguity during the task-driven fine-tuning process. In addition, it also supports to better capture the rich multi-way relational structures of the input graphs under the topology-enhanced graph analysis approach. Extensive empirical studies within a real-world molecular graph dataset have effectiveness of our SFGE in dealing with graph classification task.

Keywords Deep learning · Graph neural network · Simplicial complexes · Neuro-fuzzy network · Graph embedding

1 Introduction

In recent years, the GNN [1, 2] has been considered as an important deep neural architecture for handling sophisticated graph-structured datasets. It has attracted a lot of attentions from our research communities due to its indispensable advantages in comparing with previous deep neural architectures. In fact, the recent proposed graph neural architectures have shown significant performances in multiple complex graph data representation as well as task-driven learning tasks. As the results, it has been widely adopted in multiple data analysis and mining problems, including: information

network analysis and mining (INAM) [3, 4], natural language process (NLP) [5], computer vision (CV) [6], bioinformatics [7, 8], networked data recommendation [9, 10], etc. Generally speaking, the GNN has shown remarkable performances for data mining problems that require in-depth evaluation upon the multi-viewed/complex relational structures of graph-based datasets. In recent years, there are various GNN-based techniques have been proposed to deal with sophisticated task-driven graph learning problems. These graph learning problems can be generally categorized from node/edge-level (e.g., node classification/clustering, link prediction, etc.) to complete/graph-level (e.g., molecular graph classification, graph generation, etc.). Especially, within the networked data analysis and mining research direction, graph neural networks have presented state-of-the-art performances in different learning tasks in both graph-modelled and non-graph-modelled datasets. In addition, they

✉ Phu Pham
pta.phu@hutech.edu.vn

¹ Faculty of Information Technology, HUTECH University, Ho Chi Minh, Vietnam

also enable the capability of dealing with multiple graph representation learning problems in the context of inductive and task-driven learning manners. In contrast with previous classical node embedding approach [3], the graph neural network can assist to capture the representations of nodes/graphs that are not involved within the model learning process. As the results, it can help to continuously learn the representations of unseen graph-structural data without any re-training effort. This inductive learning capability is considered as effective for handling graph data representation learning problem, especially within large-scaled datasets, like as: social networks (e.g., Facebook, Instagram, Twitter, etc.), e-commercial networks (e.g., Amazon, Alibaba, etc.).

In general, similar to the data analysis and mining tasks for other data structure's types (e.g., text, image, video, etc.), the feature engineering from graph-based data structure is always considered as an important process. In fact, the quality of extracted features from graph can directly influence to the overall accuracy performance of the given learning model. From the past, as most of the graph-based feature engineering process has been highly relied on the hand-crafted approaches in which expert knowledge is required to extract distinctive features from the given graph datasets. In fact, the hand-crafted feature engineering process for graph-based datasets is frequently considered as time-consuming and high-cost, especially within large-scaled networked datasets, as it requires the involvement of expert knowledge in selecting proper graph's structural features for different tasks. To overcome these limitations, there are several efforts which has adopted the automatic data representation learning paradigm (initially proposed in NLP domain [11, 12]), into graph learning area. This approach is frequently called as: node embedding (a.k.a. node representation learning). The node/graph embedding is an approach of transforming the nodes/graphs into fixed dimensional embedding vectors while still keeping the relational/graph-structural information of the given networked dataset. The idea of representing graph's nodes as embedding vectors mainly adopted from the well-known word/text embedding methods which are popular within the NLP domain (e.g., Word2Vec [11], GloVe [12], etc.). At that time, there are some notable node-embedding techniques have been proposed, such as: DeepWalk [13], Node2Vec [14], etc. These node embedding methods are used to deal with the automatic feature engineering from graph-structured data.

Even these node embedding techniques can support to capture the general local proximity structure of the given graph, they are still unable to deal with complex task-specific learning problem on graph. In more specifics, the previous node embedding methods are considered as transductive learning approach. Therefore, the given generic node embedding technique is unable to learn

the representations of unseen nodes/graphs. Thus, it required the extra re-training efforts in order to achieve the embeddings of new graph's nodes. To find the solution for this drawback, go along with the development of deep learning [15, 16] as well as achievements within the neural learning/optimization strategies [17–19], there are multiple deep neural architectures have been proposed. The contemporarily proposed neural learning techniques have been designed to effectively deal with task-specific graph-based data representation learning problem. Among recent advanced neural architectures which are proposed for graph embedding problem, GNN is considered as the most powerful one which have been extensively studied in recent years to deal with complex graph learning problems.

1.1 Contemporary achievements & remained challenges

In order to overcome challenges related to transductive learning-based and task-driven graph learning problems, there are several graph neural learning architectures have been proposed recently. These well-known GNN-based architectures, such as: GCN [20], GraphSAGE [21], GAT [22], GRAND [23], GATv2 [24] and GIN [25]/GINE [26]. They have remarkably demonstrated state-of-the-art performances within various graph's node-levelled as well as graph-levelled learning tasks. In fact, most of notable GNN-based architectures are designed to learn the representations of nodes/edges which is all considered as node-levelled embedding approach. Then, the node embedding aggregation/combination is applied to efficiently achieve the whole-graph embedding vector for dealing with graph-levelled tasks like as classification. This node feature vector aggregation process is frequently known under the name: graph pooling/readout process. For different graph-levelled representation learning technique, the readout functions are defined differently. There are several readout/pooling strategies have been proposed recently, such as: node vector summation, node embedding pooling strategies [27, 28], etc. To get better inside the node feature representation learning process, the core paradigm of GNN-based architecture is relied on the iteration message passing schema over neural layers. In other words, within a given k-layered graph neural architecture, the node embedding vector is generated by aggregating the information from its k-hop neighborhood nodes. The message passing (a.k.a. node feature aggregation) schema is assumed as be able to help to sufficiently capture the information which is shared between graph nodes. However, the GNN-based architectures which are designed upon this schema are still considered as unable to capture the rich and multi-scaled topological structures from the input graphs. Generally speaking, a graph-based data structure can be modelled as a topology object, as its connectivity

information and edge's weights can be considered as the topological properties. There are multiple studies [29] [30, 31] have recently presented the potentiality of learning topological information from graph-structural data. In fact, the graph-based topological information can be considered as important information sources as they can carry in-depth multi-scaled structural features of the input graphs. These multi-viewed structural features can help to improve both quality of node-levelled/graph-levelled representations as well as accuracy performances of various task-driven learning objectives. Recently, there are several attempts have dedicated to seek for proper integration between the topological structures (e.g., connected components, cycles, etc.) of graph and graph neural representation learning. It is needless to say that the recent topology-enhanced graph neural network (TopoGNN)-based models have shown the significant improvements in various graph representation and learning problems. There are multiple TopoGNN-based techniques have been introduced recently, such as: GFL [29], PEGN [30], TOGL [31], etc. These topology-driven graph learning techniques have significantly shown great performances while handling various graph-based data mining tasks. The achievements of these TopoGNN-based models also prove the potentiality of the integration between topological data analysis (TDA) (e.g., persistent homology, persistent image, etc.) and GNN. The extensive experiments [29–31] within benchmark graph datasets have proved the power of TopoGNN-based models in handling complex-structural graph representation learning problem. Especially, the TopoGNN-based models have outperformed classical GNN-based models in the case that the connectivity and global graph's structure are considered as highly critical, such as within the molecular graph classification-driven representation learning problems.

There are some recent studies within the topology-embraced graph learning area have pointed some drawbacks of traditional GNN-based architectures. These limitations are mainly related to ability of modelling the multi-way relationships between nodes in the given graph. In more specifics, most of existing GNN-based techniques (like as: GCN, GraphSAGE, etc.) are only capable for obtaining the direct/explicit pairwise relations between nodes within graph. As the results, they are considered as unaffordable for sufficiently handling graph structural representation learning task in the context of in-depth analyzing multi-way connectivity (e.g., molecular graph, social network graphs, etc.). To cope with this problem, in very recent times, the rapid development of topology-enhanced machine learning/deep learning (TDL) [32] has led to novel and interesting proposals for dealing with multi-way connectivity learning on graph. Among recent proposals within this domain, the simplicial neural network (SNN) is considered as a notable type of deep neural architecture. There are some SNN-based

models, e.g., SNN [33], SCONV [34], SCNN [35], SCNP [36], etc. have enabled to learn the multi-way connectivity information from graph. In general, the SNN-based techniques can help to model the multi-way relationships between nodes as simplicial complexes, for examples: hyper-graphs. By modelling multi-way connectivity information of graph as simplicial complexes, it can support to obtain high-order/view-varied relationships between nodes.

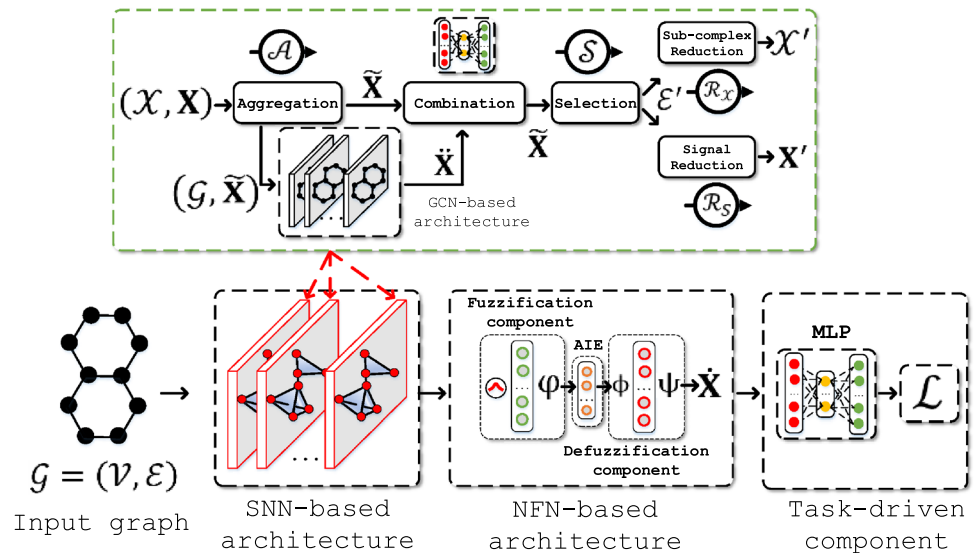
Major challenges related to graph learning & our research assumptions. Generally speaking, most of GNN/SNN-based architectures were recently developed as complex and multi-modal neural architectures. As the results, these techniques can easily encounter with problems with regard to the feature noise and data uncertainty occurrence during the data embedding as well as task-driven model fine-tuning processes. In deep graph learning approach, the feature noise/uncertainty is frequently happened. The root cause of this limitation is come from the high-complexity and explainability of the multiple neural layers and a large number of hyper-parameters/configurations are involved within the model learning process. In recent years, there are some remarkable researches [37–43] within the fuzzy neural learning area have been conducted. These recent studies have shed some lights for effectively dealing with the feature uncertainty and noise occurrence during the high-dimensional [44] data representation learning process, especially within sophisticated multi-layered graph neural architectures. As the results, there are still rooms for further studies and enhancements in this SNN-based graph learning approach.

1.2 Our motivations & contributions

Mainly motivated by recent achievements of recent studies within both GNN, TopoGNN and SNN for dealing with complex/multi-viewed graph representation learning approach, in this paper, we proposed a novel graph embedding technique, named as: SFGE. Figure 1 illustrates the overall architecture of SFGE model. In more specifics, the proposed SFGE model in this paper is an integration between simplicial neural and deep neuro-fuzzy network. Our model can support to deal with the multi-way and topology-enhanced molecular graph representation learning and classification problems. In the graph representation learning process of our model, in order to better model and learn the combined global-levelled structural and multi-way representations of graphs, we proposed a novel SNN-based graph learning component.

In more specifics, in our proposed SFGE model, we enhanced the classical simplicial complex-based graph learning component. It is designed to integrate with the global message passing learning paradigm of traditional GNN-based architecture. This global-levelled graph

Fig. 1 The general architecture of our proposed SFGE model within this paper



feature aggregation process is directly employed within the simplicial complex convolution and pooling learning processes. The graph simplicial complex and pooling processes are also designed under the SNN-based learning schema. To achieve better representation of graph through simplicial neural learning, we employ the simplicial pooling mapping and pooling strategies that have been recently introduced in the previous SCNP model [36]. In the previous approach of the conventional SCNP model, the simplicial complex structures are first modelled from the input graphs. Then, these structures are efficiently encoded through different processes, including: aggregation, selection and reduction. However, in order the better achieve the accumulated graph-based signals during the aggregation process, we implement an extra graph neural learning layer, which is specified as a GCN-based architecture, right after the local aggregation mechanism. In our approach, we use this extra graph learning to efficiently achieve better global-scoped graph-structural features as well as multi-way lower/upper connectivity information in the given graphs. Similar to the approach of previous SNN-based techniques, the lower/upper connectivity information of the graph in our approach are also modelled by using simplicial complexes. After, relevant to previous studies, the graph simplicial structure is passed through selection, reduction and readout steps. These steps support to achieve the final representation of the input graph. Then, we pass the generated graph representation vectors into an uncertainty reduction component before feeding the graph representations to task-driven component. This task-driven learning neural component can be designed as the multi-layered perceptron (MLP) or singular full-connect layer which supports for delivering graph classification outputs. The feature noise/uncertainty reduction mechanism is designed as a deep

fuzzy neural architecture. Specifically, within our approach, a deep fuzzy neural architecture with the adaptive inference mechanism is adapted. By doing like this, our model can significantly assist to reduce the feature noise/uncertainty which is occurred within the generated graph embeddings that have been learnt in previous stages. As the results, this extra fuzzy neural learning component help to enhance the overall accuracy performance of the graph categorization-driven fine-tuning process. In general, our contributions within this paper can be summarized as 3-folds, including:

- First of all, we adopted the simplicial neural aggregation and pooling strategy in previous work to efficiently learn multi-way interacting relational structures within the given graph. However, as an enhancement from previous SCNP model, we implement an extra global-levelled GNN-based layer. This layer can support to effectively capture the joined rich global graph-structural as well as lower/upper interacting relationships between graph nodes. This extensive graph neural learning layer is implemented at the output layer of the given simplicial aggregation mechanism. Then, similar to previous work, the simplicial complex subset selection/reduction and readout strategies are applied to achieve the graph-levelled representations.
- Secondly, we implemented an additional deep fuzzy neural architecture with adaptive inference mechanism in our model in order to achieve the noise/ambiguity-reduced graph-levelled representations. Then, the given deep fuzzy neural learning component is implemented at the output layer of the given simplicial neural architecture. In more specifics, for the of implementation of the given fuzzy neural component in our SFGE model, we mainly inherited from the architecture of

previous studies [42]. In general, the RFNN model [42] is considered as recent robustness deep neuro-fuzzy architecture. It applied learnable inference engine to efficiently improve the performance of the inference process. As the results, the adaptive inference engine within the given RFNN model is considered as sufficiently powerful which can help to transform and produce better uncertainty-reduced as well as task-driven friendly graph representations. Thus, it can directly support to enhance the performance of graph classification task in the after all.

- Finally, to prove the effectiveness as well as outperformance of our proposed SFGE model in comparing with previous GNN/TopoGNN-based techniques, we conducted extensive experiments within a real-world molecular graph dataset. The comprehensive comparative studies in our paper have shown not only the effectiveness but also the outperformance of our model in comparing with recent state-of-the-art graph learning baselines. The experimental results also proved our proposed ideas in this paper can be considered as a promising direction for enhancing the performance of existing SNN-based and topology-embraced graph embedding techniques. The integration between SNN with neuro-fuzzy network can help to achieve better data representation as well as task-driven learning objectives.

The novelty & difference between our model with previous baselines. Our main concentration in this paper is the capability of reducing the feature uncertainty and noise occurrence during the learning process of the SNN-based architecture. Therefore, in our proposed SFGE model, we intend to integrate advanced deep fuzzy neural architecture into the multi-way graph embedding process of existing SNN-based architectures. This combination can support to overcome limitations of previous studies. Generally, within recent studies related to deep fuzzy-neuro learning area, there are different adaptive inference strategies [42] have been proposed. Within the fuzzy neural learning approach, the inference engine can support to efficiently deal with the complex high-dimensional data representation learning tasks in the context of noise/uncertainty reduction manner. Last but not least, in order to supplement our model with the ability of sufficiently capture the joined explicit graph-structural and multi-way relational information from the input graph, we also set up an extra graph neural learning component within our model. As the result, our model can support to achieve multi-scope structural information from the input graphs and is promising for enhancing the performance of both molecular graph representation and categorization-task driven fine-tuning processes.

The overall structure of our paper. Generally speaking, the left contents of our paper are organized into 4 sections.

In the next section, we shortly present the discussions about recent studies which are related to the graph neural network, simplicial neural network and topological graph learning approaches. These research topics are considered as tightly regarding with our studies within this paper. In this section, we our extensively discuss about pros/cons of previous graph learning techniques. Then, within the methodological section, we provide the detailed descriptions about the methodology and implementation of different neural learning component within our proposed SFGE model. In addition, within our third section, we also generally provided important preliminaries/background concepts. These preliminaries are extensively applied within our studies. In the fourth section, we comprehensively demonstrate the empirical studies with extensive performance comparisons to prove the outperformances of our proposed SFGE model in comparing with state-of-the-art graph embedding baselines. In this section, the ablation studies about important parameters/configurations of our model are also provided. Finally, in the last section, we summarize about our achievements with the proposed SFGE as well as highlight several potential enhancements for this research direction.

2 Related works

In this section, we briefly reviewed about recent studies which are mainly related with the graph representation learning and simplicial neural network domains. It is needless to say that graph-based data analysis and mining has been extensively and continuously studied for many decades. There are primitive graph learning problems, such as: classification, clustering, link prediction, etc. These generic mining tasks have a wide range of application in multiple disciplines, such as: social network analysis, networked data recommendation, bio/cheminformatics, etc. In the last few years, graph feature engineering is always considered as the high-effort and challenging tasks. The challenges are come from many problems with related to the complexity in structure as well as large-size of real-world graph datasets (e.g., social networks, bibliographic networks, protein-drug interaction networks, etc.). There are some classical node-embedding techniques [13, 14, 45] have widely utilized to deal with the automatic feature extraction from graph-structural datasets. However, they still encountered several limitations which are related to the capability of dealing with inductive and task-driven learning problems. Go along with the raise of deep learning in multiple areas of computer science, there are multiple fundamental graph mining problems have been boosted by applying different advanced deep neural architectures.

Generally speaking, most of the recent proposed GNN-based architectures can be grouped into two approaches, the explicit graph-structured and topology-driven graph learning approaches.

2.1 Explicit graph-structural representation learning approach

Generally, within the canonical graph neural learning approach, most of GNN-based architectures are developed upon the local node structural message passing (a.k.a. shared neighborhood feature aggregation) schema. This learning strategy in which the representation of a target/ego node is achieved by aggregating feature/information from its neighbors. There are multiple types of message aggregation schemas which are designed in different GNN-based architectures. These various types of node aggregation are aimed to efficiently achieve the k-hop aggregated node-levelled embeddings of a given graph under the multi-layered neural representation learning process. There are some well-known GNN-based architectures, such as: GCN [20], GraphSAGE [21] and GRAND [23]. These models enable to handle complex task-driven node representation learning problems in forms of end-to-end neural learning manner. In addition, to better control the shared feature aggregation process during the graph propagation learning process, within each hidden layer, Veličković, P. et al. [22] and Brody, S. et al. [24] have applied the multi-headed self-attention mechanism [46]. Specifically, within the graph attention network, the attention weights for neighbors of the corresponding target/ego node are computed. Then, they are utilized to assist to control how much neighborhood information can be aggregated to it. Thus, it can adaptively achieve better task-driven node representations. Then, in order to achieve the graph-levelled representation, the readout function is applied to combine all node embeddings and produce the completed graph embedding. Recently, within the graph-levelled readout and pooling approaches, there are notable mechanisms have been defined, such as: vector summation, diff-pool strategy [27], sort-pooling strategy [28], self-attention graph pooling strategy (SAGPool) [47], etc. On the other hand, in order to achieve better representation for graph-levelled representation learning approach in which isomorphism aspect is taken in consideration, Xu, K. et al. have proposed a novel graph embedding technique, named as: GIN [25]. Within the GIN model, the graph propagation learning mechanism is designed upon the Weisfeiler-Lehman (WL) isomorphism test paradigm. As the results, the GIN model tends to produce similar graph-levelled representations for isomorphic graphs. In the same approach, within the recent studies, Hu, W. et al. have proposed a novel pre-training strategy for graph in forms of high-expressive graph learning

approach, within the GINE model [26]. These recent high-expressive graph embedding techniques have demonstrated significant enhancements in graph-levelled embedding and classification problems. Even recent graph learning techniques have demonstrated state-of-the-art performances in different graph mining problem, there are some limitations still existed with these GNN-based architectures. In fact, there are some in-depth studies [36, 48] have indicated some major drawbacks which are related to the classical GNN-based architectures. These limitations are mainly related to the capability of extracting the topological structures and the multi-way/multi-scaled relational information between nodes in sophisticated graph-based datasets. Therefore, there are still rooms for further studies within the deep neural graph learning problem.

2.2 Topology-driven graph representation learning approach

2.2.1 Topological graph filtration learning approach

In recent times, the topological structural feature analysis in graph data has been extensively studied by many researchers and it has emerged as an interesting research direction. In fact, the topology-driven graph learning is considered as a nascent research area, it has recently attracted a lot of attentions from our research community. Within this approach the multi-scaled topological properties from graphs are carefully taken in consideration. Within early attempts, Hofer, C. et al. [29] has proposed a novel graph learning which is relied on the persistent homology analysis on graph-based data, called as graph filtration learning (GFL). The proposed GFL assists to evaluate and learn the rich topological features from graph which are preserved within the persistent barcodes. In more specifics, within the GFL model, graph filtration process is conducted by different trainable filtration functions. By using self-adaptive filtration functions, the GFL model enables to thoroughly analyze the graph's multi-scaled topological structures which are represented as the simplicial complexes. Within the GFL model, the persistent homological features which are learnt from graph are represented by persistent barcodes. Similar to that, Zhao, Q. et al. proposed a novel topology-driven graph learning technique call as: PEGN model [30]. The proposed PEGN model applied the persistent image projection approach to capture and represent significant topological features in graph. The persistent images are used to reflect the changes in topological structures of the given graph during the filtration learning process. As the results, it can support to efficiently learn and extract the significant topological features from the given graph. Thus, it can directly support to enrich the obtained information within graph representation in which topological structures are

highly focused. Last but not least, going with the potentiality of topological data analysis in graph, in very recent times, Horn, M. et al. has proposed a novel subgraph-based filtration learning mechanism within the TOGL [31]. This enhanced subgraph-based filtration learning strategy has supported to effectively preserve multi-viewed and multi-scaled topological structures of the given graph. Similar to the GFL model, TOGL model also used different trainable filtration functions to handle the graph filtration learning process.

2.2.2 Simplicial neural network based graph learning approach.

On the other side, mainly inspired from the topological signal processing domain [49, 50], the simplicial deep learning has recently emerged and widely studied for graph learning problem. In general, the standard simplicial complex neural architecture is developed upon the topological data analysis and feature extraction paradigm. Within this approach, the SNN-based architectures are designed to learn the representation of input data by in-depth analyzing its multi-way connectivity information. To do this, first of all the given SNN-based neural network will try to model the input data as simplicial complexes. These simplicial complexes are assumed as specific instances of hyper-graphs. With this approach, it can help to evaluate the rich algebraic descriptions of the given data structure as well as efficiently preserve the multi-way relational information between instances. This multi-way connectivity information is normally hidden/not explicitly interpreted from graphs. Among notable studies within this area, the early attempt of Ebli, S. et al. is considered as the most popular SNN-based architecture [33]. Similar to that with studies of Bunch, E. et al. in the SCONV [34], which proposed novel approaches for generalizing the simplicial convolutional operations within the lower/upper neighborhoods of simplicial complexes. Going with this direction, Bodnar, C. et al. proposed MPSN model [51], proposed a novel aggregation and updating functions. These novel node feature aggregation mechanisms support to effectively exploit lower/upper neighborhoods within the previous extracted simplicial complexes of the given graph. By taking disadvantages of the attention mechanism [46], in recent time, Giusti, L. et al. proposed a novel simplicial attention network (SAN) model [52]. The proposed SAN model enables to efficiently weight the lower/upper neighborhoods of the given graph-modelled simplicial complexes by using the multi-headed self-attention mechanism, in a task-oriented fashion. Similar to that, in very recent times, Yang, M. et al. proposed the SCNN model [35]. The SCNN model assists to conduct the multi-hop information aggregation between nodes. Thus, it can support to effectively achieve

the graph representation upon the corresponding lower/upper neighborhoods. As the results, the proposed SCNN model can provide the capability of spectral interpretability through the definition of simplicial filters as well as the simplicial Fourier transformation [48]. To better presenting in different resolutions, recently, Cinque, D. M. et al. proposed a novel SCNP model [36] which introduced various simplicial pooling/mapping strategies for SCNN model. The simplicial mapping and pooling strategies which are proposed within SCNP model. Going with the topology-enhanced graph learning direction, these novel mapping/pooling strategies are also designed upon topological signal processing paradigm. As the results, these novel SNN-based architectures have demonstrated state-of-the-art performances in various graph learning tasks. However, within the simplicial neural learning for graph data area, there is few researches which have concentrated on the feature noise/uncertainty reduction aspect. The feature noise/uncertainty occurrence is a common problem within both general deep learning and deep graph learning areas. This challenge is mainly come from graph feature uncertainty which is frequently occurred within the representation learning process, especially within sophisticated multi-model deep neural architectures. In recent years, along with the progress of deep learning in multiple disciplines, deep fuzzy neural network has raised and become a powerful tool for dealing with high-dimensional and uncertainty data representation learning problems. There are notable studies [40–42] have presented the capability of effectively extracting high-qualified latent features from the complex data structures, including graph while dealing with task-driven data embedding tasks. Thus, there are still opportunities for further enhancements within the SNN-based graph learning direction. For further enhancements in this direction the integration between SNN-based graph learning and fuzzy neural network is considered as the most potential and promising direction.

Extensive comparative studies with different graph neural learning approaches. In order to prove the effectiveness of our proposed SFGE model, we extensively evaluated its performance with different graph neural architectures which are classified in both graph explicit-structural and topology-driven approaches. As different graph learning paradigm has its own advantages while handling different complex graph-based structures, therefore it is considered as necessary to compare the performance of our SFGE model with different approaches to ensure the capability of our model in sufficiently preserving the joined explicit-structural and topological information from the input graphs. Thus, it can later benefit the multi-viewed graph representation learning and classification-driven fine-tuning processes. Within the graph explicit-structural approach, we evaluate the performance of our proposed SFGE model with standard

Table 1 The list of common mathematic notations/symbols

Symbol	Explanation
$\mathcal{G} = (\mathcal{V}, \mathcal{E})$	A graph-based structure with its lists of nodes and edges/links
$\mathcal{H}_{ }$	The ($ $)-simplex which is defined as the subset of graph nodes ($\mathcal{V}$)
$\mathcal{N}_l$ and $\mathcal{N}_t$	The lower and upper neighborhoods of a ($ $)-simplex
$\mathcal{X}$	The simplicial complexes
$\mathcal{D}_0, \mathcal{D}_1$ and $\mathcal{D}_2$	The (0)-simplices (nodes), (1)-simplices (edges) and (2)-simplices (triangles) within the graph ($\mathcal{G}$)
$\ddagger_{ }$	The ($ $)-simplicial signal
$\mathbf{B}$	The incidence matrix
$\mathbf{L}$	The Laplacian matrix
$\mathbf{X}, \tilde{\mathbf{X}}$ and $\tilde{\tilde{\mathbf{X}}}$	The graph node feature, graph-structural representation and simplicial signal summarization matrices, respectively
$\mathcal{P}(\cdot)$	The mapping/pooling function
$\text{GCN}(\cdot)$	A generic GNN-based architecture as a mapping function
$\text{Linear}(\cdot)$	A linear neural layer as a mapping function
$\sigma(\cdot)$	The activation function of a specific neural layer
$\wp$ and ψ	The fuzzification and de-fuzzification components within a deep fuzzy neural network, respectively

graph neural architectures, like as: GCN [20], GraphSAGE [21], GAT [22], GIN [25], GINE [26], GRAND [23], SAG-Pool [47] and GATv2 [24]. These well-known GNN-based architectures have demonstrated great achievements within complex graph-structural embedding problems. However, they still suffered some limitations related to the capability of capturing the topological information from the input graphs. Within the multi-view/multi-way relational learning approach, we also compare our model with the state-of-the-art persistent homology/simplicial complex-based graph learning models, including: TOGL [31] and SCNP [36]. The extensive evaluations within benchmark graph datasets have proved the abilities of these topology-embraced graph learning techniques in handling rich-topological graph representation learning tasks. In addition, the detailed setups of these graph learning baselines within our comparative studies will be provide within the sub-Sect. 4.2.1.

3 Methodology & implementation

In this section, we formally present the methodology as well as detailed implementation of our proposed SFGE model for effectively dealing with multi-way and feature uncertain-minimized graph learning task. Generally speaking, our proposed SFGE model is composed by two components which are the simplicial convolutional and the deep neuro-fuzzy components. In our approach, the simplicial convolutional network with the mapping/pooling strategy is implemented with the inspiration from the previous SCNP model [36]. In our SFGE model, we have enhanced the aggregation

mechanism of the previous SCNP model in order to enable the capability of better capturing multi-viewed graph representations. The modification in our model's aggregation mechanism includes the implementation of an extra GCN-based layer. This graph neural layer is placed at the output layer of the given simplicial complex aggregation component. As the results, it can directly support to increase the capability of global shared feature interpretability.

Moreover, it also enables our model to jointly achieve the neighborhood structural information from the input graph. Later, the output embedded graphs from the SNN-based architecture are passed through a deep fuzzy neural architecture. This fuzzy neural learning component is designed upon learnable inference paradigm [42]. It assists to achieve the adaptive firing strengths for fuzzy rules which are applied for input graph embeddings. Therefore, this fuzzy neural component can directly support to deduce the feature noises/ambiguities which are frequently occurred during the graph embedding process. Finally, the output de-fuzzified graph embeddings are fed into a task-driven neural component. This classification-driven component is simply designed as an MLP-based architecture with softmax classifier to deal with graph classification task. In addition, within the initial contents of this section, we also present preliminaries/background concepts with regard to our studies. The Table 1 lists all common mathematic symbols/notations which are utilized within the left contents of our paper.

3.1 Preliminaries & background concepts

In this part, we briefly present several background concepts which are related to the baselines as well as learning paradigms of the SNN, simplicial complex/signal and deep neuro-fuzzy network. In this paper, our studies are mainly concentrated on graph representation learning and classification problems. Therefore, our target data sources are modelled as graph-based structures. Generally, a graph-structural dataset is denoted as: $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ —with the its sets of nodes ($\mathcal{V}$) and edges ($\mathcal{E}$).

3.1.1 Simplicial complex and signal processing on graph.

Given a finite set of nodes within the given graph ($\mathcal{G}$), a $(\parallel)$ -simplex, denoted as: $(\mathcal{H}_{\parallel})$, is defined as a subset of ($\mathcal{V}$), with the $(\parallel + 1)$ in cardinality. As a specific face of the given $(\parallel)$ -simplex, $(\mathcal{H}_{\parallel})$ is considered as a subset of $(\mathcal{H}_{\parallel})$, with $(\parallel)$ in cardinality. As the results, the $(\mathcal{H}_{\parallel})$ is considered as having $(\parallel + 1)$ faces. In addition, we also have the co-face of $(\mathcal{H}_{\parallel})$ which is defined as: $(\parallel + 1)$ -simplex. The $(\mathcal{H}_{\parallel+1})$ also includes the $(\mathcal{H}_{\parallel})$. The lower and upper neighborhoods of a $(\parallel)$ -simplex are denoted as: $(\mathcal{N}_{\downarrow})$ and $(\mathcal{N}_{\uparrow})$, respectively. They are defined as the set of simplices with which they share the face and co-face of $(\mathcal{H}_{\parallel})$, respectively. Simplex is considered as the core concept of simplicial complexes (definition 1) in algebraic topology and simplicial signal (definition 2) processing areas.

Definition 1 **Simplicial complexes** ($\mathcal{X}$): a simplicial complex of order ($\mathcal{K}$), denoted as: $(\mathcal{X}_{\mathcal{K}})$ is defined as a list of $(\parallel)$ -simplex, or: $(\mathcal{H}_{\parallel})$ with: $\parallel \in [0, \dots, \mathcal{K}]$. As the results, for any: $\mathcal{H}_{\parallel} \in \mathcal{X}_{\mathcal{K}}$, if: $\mathcal{H}_{\parallel-1} \subset \mathcal{H}_{\parallel}$, then: $\mathcal{H}_{\parallel-1} \in \mathcal{X}_{\mathcal{K}}$. We also can formulate the set of simplices within $(\mathcal{X}_{\mathcal{K}})$, as: $D_{\mathcal{K}} := H_{\mathcal{K}} : H_{\mathcal{K}} \in \mathcal{X}_{\mathcal{K}}$ and $\mathcal{D}_{\parallel} \in \mathcal{X}_{\mathcal{K}}$, the $(\mathcal{N}_{\parallel})$ presents for the cardinality of $(\mathcal{D}_{\parallel})$, or: $|\mathcal{D}_{\parallel}| = \mathcal{N}_{\parallel}$. In addition, we also can write: $\mathcal{H}_{\parallel-1} \sim \mathcal{H}_{\parallel}$, if: $\mathcal{H}_{\parallel-1} \subset \mathcal{H}_{\parallel}$, to indicate the orientation of $(\mathcal{H}_{\parallel-1})$ is coherent with the one of $(\mathcal{H}_{\parallel})$, whereas: $\mathcal{H}_{\parallel-1} \not\sim \mathcal{H}_{\parallel}$.

Definition 2 **$(\parallel)$ -Simplicial signal** ($\ddagger_{\parallel}$): is defined as the mapping of a simplicial complex (including its simplices) to real values. This mapping can be formulated as the following: $\ddagger_{\parallel} : \mathcal{D}_{\parallel} \rightarrow \mathbb{R}, \parallel \in [0, \dots, \mathcal{K}]$.

Same with the previous approaches within the simplicial complex-based neural learning area, our concentrations in this paper are the simplicial complexes that their order is up to two, or: $(\mathcal{X}_2)$. Given a graph-based dataset ($\mathcal{G}$), which is modelled as simplicial complexes. Specifically, we have its set of nodes ($\mathcal{V}$) as the (0) -simplices, or denoted as: $\mathcal{D}_0 = \mathcal{V}$. Then, its set of edges

are considered as the (1) -simplices, or denoted as: $\mathcal{D}_1 = \mathcal{E}$. and the set of triangles, denoted as: $(\mathcal{T})$ are the (2) -simplices, or: $\mathcal{D}_2 = \mathcal{T}$. We also have: $|\mathcal{V}| = \mathcal{V}$, $|\mathcal{E}| = \mathcal{E}$ and $|\mathcal{T}| = \mathcal{T}$. The algebraic representation can be used to describe the structure of a given simplicial complex $(\mathcal{X}_{\mathcal{K}})$. The structure of $(\mathcal{X}_{\mathcal{K}})$ is described by a set of incidence matrices, denoted as: $\mathbf{B}_{\parallel}, \parallel \in [0, \dots, \mathcal{K}]$. Each incidence matrix $(\mathbf{B}_{\parallel})$ describe the incidence between the $(\parallel)$ -simplices and $(\parallel - 1)$ -simplices, as:

$$[\mathbf{B}]_{ij} = \begin{cases} 0, \text{if: } \mathcal{H}_{-1} \not\subset \mathcal{H} \\ 1, \text{if: } \mathcal{H}_{-1,i} \subset \mathcal{H}_j \text{ and } \mathcal{H}_{-1,i} \sim \mathcal{H}_j \\ -1, \text{if: } \mathcal{H}_{-1,i} \subset \mathcal{H}_j \text{ and } \mathcal{H}_{-1,i} \not\sim \mathcal{H}_j \end{cases}.$$

Then, from the incidence information between simplices, we can construct the high-order combinatorial Laplacian matrices [53] of the given order: $\parallel \in [0, \dots, \mathcal{K}]$, as: $\mathbf{L}_0 = \mathbf{B}_1 \mathbf{B}_1^T$, $\mathbf{L}_{\parallel} = \mathbf{B}_{\parallel}^T \mathbf{B}_{\parallel} + \mathbf{B}_{\parallel+1} \mathbf{B}_{\parallel+1}^T = \mathbf{L}_{\parallel,d} + \mathbf{L}_{\parallel,u}$. The $(\mathbf{L}_{\parallel,d})$ and $(\mathbf{L}_{\parallel,u})$ present for the lower and upper Laplacian matrices, respectively. These matrices encode the lower and upper adjacency information of $(\parallel)$ -simplices. For example, within the given graph ($\mathcal{G}$), two edges are known as lower adjacency if they share common nodes. Moreover, two edges are considered as upper adjacency if they are faces of common triangles. From that, for the (0) -simplices ($\mathcal{D}_1$)—with the set of nodes of ($\mathcal{G}$), all the node pairs only have the upper adjacency—if they are incident to the same edges. As the results, its corresponding Laplacian matrix $(\mathbf{L}_0)$ contains only one term. Thus, it is quite similar with the normal graph Laplacian matrix. The Hodge decomposition [54] can be applied within these high-order Laplacian matrices. It supports to decompose the $(\parallel)$ -simplicial signal ($\ddagger_{\parallel} \in \mathbb{R}^{\mathcal{N}_{\parallel}}$). In general, the Hodge decomposition process can be formulated as: $\ddagger_{\parallel} = \mathbf{B}_{\parallel}^T \ddagger_{\parallel-1} + \mathbf{B}_{\parallel+1} \ddagger_{\parallel+1} + \ddagger_{\parallel,h}$. The $(\mathbf{B}_{\parallel}^T \ddagger_{\parallel-1})$, $(\mathbf{B}_{\parallel+1} \ddagger_{\parallel+1})$ and $(\ddagger_{\parallel,h})$ terms are utilized to present for the irrotational, solenoidal and harmonic components, respectively. To more focusing on the edge flow signal and complexes of order 2, or: $(\mathcal{X}_2)$. Some notations will be simplified as the following: $(\ddagger_1)$, as: $(\ddagger)$, $(\mathbf{L}_1)$, as: $(\mathbf{L})$, $(\mathbf{L}_{1,d})$, as: $(\mathbf{L}_d)$, $(\mathbf{L}_{1,u})$, as: $(\mathbf{L}_u)$ and $(\mathcal{X}_2)$ as: $(\mathcal{X})$. For the lower and upper neighborhoods of the given (i^{th}) edge, we simply denote they as: $(\mathcal{N}_{i,\downarrow})$ and $(\mathcal{N}_{i,\uparrow})$, respectively.

3.1.2 Simplicial convolutional network (SCN) with pooling operation.

In the recent studies, the Hodge decomposition paradigm has been applied. This decomposition strategy assists to design notable convolutional and filter strategies. Specifically, it can support to efficiently extract components of simplicial signals from the unstructured data collection. Within the recent works [36] of Cinque, D.

M. et al., the general convolutional and pooling strategies for the convolutional operation have been generally defined as the following (as shown in the Eq. 1):

$$\mathbf{X}' = \sigma \left(\mathcal{P} \left(\sum_{p=1}^{J_d} \mathbf{L}_d^p \mathbf{X} \mathbf{D}_p + \sum_{p=1}^{J_u} \mathbf{L}_u^p \mathbf{X} \mathbf{U}_p + \mathbf{X} \mathbf{H} \right) \right) \quad (1)$$

In this equation, the $(\mathbf{X} \in \mathbb{R}^{E \times G})$ is the input matrix of collected (G) edge signals. The $(\mathbf{D})$, $(\mathbf{U})$ and $(\mathbf{H})$ present the filter, residual weight and learnable parameters of the given simplicial convolutional architecture. The mapping function $(\mathcal{P})$ presents for different pooling strategies which are specified within this study [36].

3.1.3 Fuzzy neural network with adaptive inference engine (AIE).

Along with developments of deep learning, there are multiple advanced deep neural architectures have been proposed in recent years. The deep neuro-fuzzy network is considered as one of hot research topics in recent time. Recent studies have shown the advanced capability of the NFN-based architectures in eliminating feature noise/uncertainty related challenges. These challenges are frequently occurred during the data representation learning and task-driven fine-tuning processes of sophisticated multi-model neural architectures, such as: GNN and SNN. In recent studies related to the NFN, there are some notable inference mechanisms have been proposed to enhance the performance of fuzzy neural learning paradigm. Generally speaking, these recent proposed inference mechanisms can assist to obtain the proper firing strength values for different fuzzy rules. Therefore, they can significantly handle complex and multi-dimensional data representation learning problems with high-level uncertainty. Among notable works [40–42] within the deep fuzzy neural learning domain, Zhang, L. et al. proposed RFNN [42] which applied an adaptive inference mechanism which is designed as a learnable neural architecture that can support to achieve proper firing strengths for corresponding fuzzy rules within the fuzzification and de-fuzzification processes.

3.2 SFGE model: integrated SNN and NFN for graph learning

3.2.1 Multi-viewed graph learning with SNN and GCN.

First of all, for the multi-way relational graph learning paradigm of our model, we mainly inspired from the previous SCNP model. Our model's simplicial learning approach contains three separated processes, which are the aggregation, selection and reduction processes. These

operation steps support to achieve the multi-way relational representations of the given graph. Within this approach, the input graph $(\mathcal{G})$ is modelled as simplicial complexes. In more specifics, for the input of the given simplicial convolutional architecture as signals, they are denoted as: $\mathbf{X} = \mathbf{X}_d + \mathbf{X}_u + \mathbf{X}_h$. Then in the later steps, the extracted simplicial complex-based structures are encoded/represented by using the lower and upper neighborhood Laplacian matrices, denoted as: $\mathcal{X} = (\mathbf{L}_d, \mathbf{L}_u)$. The $(\mathcal{P})$ function presents for the mapping/pooling strategy is formulated as the following (as shown in Eq. 2):

$$\mathcal{P} : (\mathcal{X}, \mathbf{X}) \mapsto (\mathcal{X}', \mathbf{X}') \quad (2)$$

In this equation, the $(\mathcal{X}')$ and $(\mathbf{X}')$ present for the output sub-complex and hidden signal representation through the mapping/pooling $(\mathcal{P})$ mechanism. In the approach of the previous SCNP model, this pooling mechanism is composed as separated mapping procedures. There are three main types of procedures, include: aggregation—denoted as: $(\mathcal{A})$, selection—denoted as: $(\mathcal{S})$ and reduction—denoted as: $(\mathcal{R})$. In general, the local aggregation process is first applied to achieve the summarization of the input signals $(\mathbf{X})$, as: $\mathcal{A} : (\mathcal{X}, \mathbf{X}) \mapsto (\mathcal{X}, \tilde{\mathbf{X}})$. Then, the summary of input signals is passed through the selection step in order to properly choose subset of edges, denoted as: $(\mathcal{E}' \in \mathcal{E})$. Then the subset of edges are used to compose the output sub-complex $(\mathcal{X}')$. Generally, the selection process is formulated as the following: $\mathcal{S} : (\mathcal{X}, \tilde{\mathbf{X}}) \mapsto \mathcal{E}'$.

Finally, within the signal reduction process $(\mathcal{R})$, the selected edge subsets and summary of input signals which are achieved in previous staged are utilized to generate the output sub-complex and signals, denoted as: $\mathcal{R} : (\mathcal{X}, \tilde{\mathbf{X}}, \mathcal{E}') \mapsto (\mathcal{X}', \mathbf{X}')$. This reduction process is conducted by two sub-processes which are in charged for down-sampling sub-complex structure and signal. These sub-processes are formulated as the following: $\mathcal{R}_s : (\mathcal{E}', \tilde{\mathbf{X}}) \mapsto \mathbf{X}'$ and $\mathcal{R}_x : (\mathcal{E}', \mathcal{X}) \mapsto \mathcal{X}'$, respectively. Generally, the produced signal matrix $(\mathbf{X}')$ is known as the simplicial structural representation of the given graph. This signal matrix representation is considered as carrying rich informative multi-way relationships between nodes in the given graph. The detailed implementations of different pooling strategies for the local aggregation, selection and aggregation processes can be found within the original paper [36]. Then, in our model to simultaneously preserve the global shared structural information of the given graph, we implement an extra GCN-based layer. This approach is considered as the explicit graph-structural connectivity which can facilitate the simplicial signal learning processes. In more specifics, at the output part of the given SNN-based mapping and pooling component, we put a 1-layered GCN-based architecture. This graph neural learning component takes the

explicit graph structure ($\mathcal{G}$) and it corresponding summarized simplicial signals ($\tilde{\mathbf{X}}$) as the inputs. Then, this GCN-based architecture supports to generate the 1-hop structural representation of the given graph over the summarization of input signals, denoted as: ($\tilde{\mathbf{X}}$). Generally, this process is generally formulated as shown in the Eq. (3):

$$GCN : (\mathcal{G}, \tilde{\mathbf{X}}) \mapsto \ddot{\mathbf{X}} \quad (3)$$

$$Linear : (\ddot{\mathbf{X}} || \tilde{\mathbf{X}}) \mapsto \tilde{\mathbf{X}} \quad (4)$$

Then, we apply a linear mapping layer within this component to achieve the joint global shared and multi-way relational structural information of the given graph. The approach of combining multiple graphs which are learnt from different aspects are known as the multi-viewed representation learning approach. This mapping mechanism supports to softly combine the global graph structural information ($\tilde{\mathbf{X}}$), with the corresponding signal summarization ($\ddot{\mathbf{X}}$). The overall steps of this process are illustrated as shown in the Eq. (4). In our model, we simply implement this component as a full-connected linear neural network component. Within previous data embedding fusion studies, the linear mapping is considered as a simple and effective mechanism. It can support to softly align the multi-viewed informative features in both graph-explicit and simplicial complex-modelled structures.

3.2.2 Uncertainty-reduced graph embedding transformation with NFN.

Finally, we achieve the final representation of the given graph, denoted as: ($\mathbf{X}'$), after passing the graph representation through multiple simplicial convolutional layers with the corresponding pooling strategy. In order to transform the output ($\mathbf{X}'$) into an uncertain-minimized embedding, we implement a deep fuzzy neural architecture. This extended fuzzy neural architecture is employed at the output part of the simplicial convolutional architecture. Then noise/uncertainty-reduced graph embeddings can be later used to enhance the performance of graph classification-driven fine-tuning process. In more specifics, for the NFN-based component in our model, we mainly adopted from ideas of previous RFNN model [42]. First of all, the output signals ($\mathbf{X}'$) of the given graph are first passed through the fuzzification component, denoted as: (φ). This fuzzification component is responsible for fuzzing the inputs with membership functions. Then the AIE is applied to computed the appropriated fire strength values for different fuzzy rules associated with the inputs. Within the de-fuzzification component (ψ) of the RFNN model, the normalized firing

strength value for a specific (k^{th}) fuzzy rule. Then, the firing strength value for the fuzzy rule is generally computed as the following: $\overline{\phi_k(\mathbf{X}')} = \frac{\phi_k(\mathbf{X}')}{\sum_{k=1}^K \phi_k(\mathbf{X}')} [34]$. Then, it is later combined with the defuzzification units which is denoted as: $\}_k(\omega; \tilde{\mathbf{X}})$ in order to generate the corresponding crisp outputs ($\tilde{\mathbf{X}}$). In the original RFNN model, the defuzzification unit is designed as MLP-based architecture. As the results, the (ω) is represented for the set of learnable parameters within the given neural architecture. At the end, the de-fuzzified output signals, a.k.a. multi-viewed graph representation, are then fed into a graph classification task-driven component. Similar with the defuzzification mechanism of RFNN model, this component is also designed as an MLP-based architecture. As a task-driven layer, the softmax activation function is applied in this component in order to deliver class/label predictions for the input graphs. The overall processes of this step are generally illustrated within the Eq. (5a).

$$\begin{aligned} \hat{y} &= \text{softmax}(\tilde{\mathbf{X}})(a) \\ \mathcal{L} &= - \sum_{i \in D} \sum_{c \in C} y_i^c \ln(\hat{y}_i^c)(b) \end{aligned} \quad (5)$$

Then, to optimize trainable parameters of our proposed SFGE model for handling graph classification problem, we utilized the cross-entropy (CE) loss function (as illustrated within Eq. 5b). The CE-based graph learning objective supports to compute the gradient which are then used to update the corresponding model's parameters in different neural components during the training/back-propagation process. As shown in the Eq. (5), the (D) and (C) present for the sets of training data and unique labels/classes of the given dataset, respectively. The (y_i^c) and ($\hat{y}_i^c$) are the vector-encoded ground-truth and predicted labels of the given (i^{th}) input graph, respectively. The overall model is optimized with the defined graph categorization task by using Adam optimization strategy and predefined learning rate (η).

4 Experiments & discussions

In this section, we present the extensive empirical studies to show the effectiveness and outperformance of our proposed SFGE model in comparing with state-of-the-art GNN/TopoGNN-based graph embedding techniques. In our experiments, we mainly used a real-world molecular graph dataset which is considered imbalanced. Then, we implemented different graph representation techniques to achieve the graph embeddings. Then, the achieve graph embeddings are fine-tuned to deal with the molecular graph categorization problem. Additionally, within this section we also provide the ablation studies on different important

model's parameters/hyper-parameters. These extensive empirical studies support to evaluate their effects upon the stability and performance of our model while handling graph representation learning and classification problems.

4.1 Dataset descriptions & experimental setups

4.1.1 Dataset usage & pre-processing steps.

For all experiments within this paper, we use a real-world chemical molecular graph dataset which is recently introduced in this study [55]. The introduced BeeTox dataset in this paper contains a set of chemical compounds which are used/included within pesticides. In order to construct this dataset, Wang, F. have collected the information of different chemical/biological compounds which are used in pesticides from multiple public databases. These public databases, include: Pesticide Properties Database by Agriculture & Environment Research Unit (AERU) ^[1], University of Hertfordshire, the Pesticide Action Network (PAN) ^[2] database provided by Pesticide Action Network North America and ECOTOX ^[3] database provide by United States Environment Protection Agency (USEPA). Specifically, this dataset contains the information of chemical/biological compounds which are categorized into negative and positive compounds (not or harmful/toxic for bee). The public version of this dataset can be achieved at this repository ^[4]. In more details, the original version of this dataset contains 891 samples. Each compound in this dataset has 5 different information fields, including: name, CAS, the toxicity level, source, dataset/category and SMILES string. As this dataset is considered as imbalanced dataset in which the number of negative/positive samples are 654/237. Therefore, to use this dataset in our experiments, we have randomly picked up about 468 molecules with the balanced ratio, as: negative (233) / positive (235) labelled chemical compounds. This modified BeeTox dataset version is used for all experiments within our papers.

Data pre-processing steps and molecular graph construction. From the constructed class-balanced BeeTox, named as: B-BeeTox dataset version (as described in previous part), to achieve the graph-based structures from chemical compounds within this dataset, we mainly relied on the RDKit library ^[5]. For the dataset initial preparation and pre-processing steps, we mainly utilized the RDKit

library. This library can support to effectively extract the adjacency matrix/atom connectivity information from the given molecules/compounds in the given B-BeeTox dataset. Then, these extracted molecular graphs through RDKit library are utilized within our experiments for handling the representation learning and classification problems by applying different graph learning baselines. For the input node features of constructed molecular graphs, similar to the empirical studies within this study [31], we don't use any extra initial node/graph features (e.g., chemical/biological descriptive information, molecular fingerprint, etc.) as the inputs for deep graph learning models. The reason for this experimental scenario is that we just want to evaluate the performance of different graph learning techniques with only the connectivity structural information is taken in consideration. Therefore, in our experiments, the accuracy performances of different graph embedding techniques as well as our proposed SFGE model are reported might be lower than the original paper with the same baselines.

4.1.2 Experimental setup & evaluation metric usage.

For comparative experiments in this paper, we applied the same evaluation strategies of previous works [31, 36]. In more specifics, our proposed SFGE model as well as other comparative graph learning techniques are configured to analyze the connectivity information of molecular graphs only. Therefore, for the initial node features of constructed molecular graphs within the given B-BeeTox dataset, we set them as dummy values. Then, the graph learning techniques in our empirical studies are configured to ignore the input node features. For evaluating the accuracy performances of our proposed SFGE model as well as other comparative baselines, we utilized the same data splitting strategy of: [0.8, 0.1, 0.1]. This splitting strategy is similar the empirical studies in previous study [31]. It means that 80% of the B-BeeTox dataset will be used for the training purpose, the left parts are used or development/validation purpose (10%) and testing purpose (10%). Then, within the evaluation process, the accuracy performances of our SFGE model and other graph learning baselines while handling the molecular graph classification task are computed by using different methods, including: area under curve (AUC), accuracy (ACC) and F-measure (F1-Macro) standard evaluation metrics. The accuracy performances of different graph classification techniques under the ACC and F-measure evaluation metrics are computed as the following (as shown in the Eqs. 6, 7 and 8):

$$ACC = \frac{TP + TN}{TP + TN + FP + FN} \quad (6)$$

¹ AERU (University of Hertfordshire): <https://sitem.herts.ac.uk/aeru/>

² PAN: <https://www.pesticideinfo.org/>

³ ECOTOX: <https://cfpub.epa.gov/ecotox>

⁴ BeeTox dataset: <http://chemyang.ccnu.edu.cn/ccb/server/beetox/index.php/home/index>

⁵ RDKit: <https://www.rdkit.org/>

Table 2 The detailed configurations/hyper-parameters of our proposed SFGE model for the comparative experiments in this paper

Our proposed SFGE model's configurations	Value
The number of SNN-based hidden layers (k^{SNN})	2
The dimensionality of SNN-based hidden state (d^{SNN})	64
The number of hidden layers (k^{GCN}) and hidden state size (d^{GCN}) for the GCN-based architecture which is applied in SNN's aggregation step	1 and 64
The default pooling ratio for different pooling strategies within the given SNN/SCNP-based architecture	0.5
Maximum number of training epochs	150
Patience (number of epochs) for the early stopping mechanism	15
The default learning rate (η)	0.001
The default training batch size	128

$$P = \frac{TP}{TP + FP}, R = \frac{TP}{TP + FN} \quad (7)$$

$$F-1 = 2 \frac{PR}{P + R} \quad (8)$$

In these equations, the (TP), (TN), (FP) and (FN) terms present for the true-positive, true-negative, false-positive and false-negative. For the AUC metric, it is identified as the area under the receiver operating characteristics (ROC) curve which represents the true positive and false positive rates at different classification thresholds. Similar to the experimental scenario in previous studies, for each model, we conducted the experiments 5 times with different shuffled train/validation/test splits, the we take the average performances as the final outputs.

4.1.3 Model's configurations for experiments

For the implementation of our proposed SFGE model, we mainly utilized the Python programming language. For the implementation of deep neural architectures within our SFGE model, we mainly use the PyTorch and PyTorch-Geometric libraries. To setup our model for empirical studies in this paper, we set the number of hidden layers (k^{SNN}) (within the given SNN-based learning architecture), as: 2. The default dimensionality of node embedding vector/hidden state (d^{SNN}) is set as: 64. For the number of hidden layers (k^{GCN}) and hidden state size/dimensionality (d^{GCN}) of the GCN-based architecture which is applied for the local aggregation, are respectively set as: 2 and 64. For the configurations of deep fuzzy neural architecture which is used to achieve the better feature uncertainty/noise-reduced and classification-friendly graph representations. For this NFN-based architecture, we set the number of fuzzy rules as: 10. The other configurations of the given RFNN model, we kept them as the same as described within the original paper in which the RFNN model achieved the highest accuracy performance for the classification task. Additionally, within

the implementation and training processes of our proposed SFGE model, the early stopping mechanism is applied with the patience is set as: 15 (stopping if the model accuracy performance upon the validation set doesn't have any improvement after a specific number of epochs). The Table 2 lists all detailed information about hyper-parameters/configurations of our proposed SFGE model for experiments in this paper.

4.2 Experimental results & discussions

4.2.1 Comparative baselines

To demonstrate the effectiveness as well as outperformance of our proposed SFGE model for dealing with molecular graph representation learning and classification tasks within the B-BeeTox dataset, we also implemented several state-of-the-art graph learning techniques for the performance comparison purpose. The implemented graph learning methods in this section can be categorized into traditional GNN -based and TopoGNN-based and SNN-based approaches. In more specifics, within the conventional GNN-based approach, we implemented common techniques, like as: GCN [20], GraphSAGE [21], GAT [22], GIN [25], GINE [26], GRAND [23], SAGPool [47], GATv2 [24]. Within the TopoGNN-based approach, we implemented the TOGL model [31]. Finally, for the SNN-based approach, we implemented our main competitor in this paper which is the recent SCNP [36] model.

In order to implement the above comparative baselines for handling the molecular graph representation learning and classification tasks in the B-BeeTox dataset, we have configured them with the same configurations of our SFGE model. Specifically, for the basic configurations, like as general graph learning process, we configured the number of hidden layers and graph embedding vector dimensionality as the same with our SFGE model's configurations (listed within Table 2). Similar to that, the number of GNN-based hidden layers (k^{GNN}) and graph embedding dimensionality (d^{GNN}) for these baselines are also set as: 2 and 64, respectively.

Table 3 Accuracy performance in terms of AUC, ACC and F1-Macro evaluations of different techniques for handling graph classification task within B-BeeTox dataset

Metrics Models	AUC		ACC		F1-Macro	
	Performance	Improve (%)	Performance	Improve (%)	Performance	Improve (%)
GCN	0.4918	25.55↑	0.4894	20.21↑	0.3348	52.40↑
GraphSAGE	0.5000	23.49↑	0.5021	17.17↑	0.3342	52.67↑
GAT	0.5000	23.49↑	0.5021	17.17↑	0.3342	52.67↑
GATv2	0.5000	23.49↑	0.4979	18.16↑	0.3324	53.50↑
GRAND	0.4813	28.29↑	0.4809	22.33↑	0.3587	42.25↑
GIN	0.5163	19.59↑	0.5106	15.22↑	0.3785	34.81↑
GINE	0.5121	20.57↑	0.5064	16.17↑	0.3698	37.98↑
SAGPool	0.5246	17.70↑	0.5277	11.48↑	0.4239	20.37↑
TOGL	0.5000	23.49↑	0.5021	17.17↑	0.3342	52.67↑
SCNP	0.5788	6.68↑	0.5553	5.94↑	0.5065	0.73↑
SFGE	0.6174		0.5883		0.5102	

For the definitions of the significance of bold, as: within the first column: GCN, GraphSAGE, GAT, GATv2, GRAND, GIN, GINE, SAGPool, TOGL, SCNP and SFGE are comparative baselines and our proposed model. Within the last row: the highest accuracy performances in terms of AUC, ACC and F1-Macro measure

For the implementation of the topology-driven graph learning approach, like as the: TOGL model, we set the number of filtration/coordination functions which are used for the graph filtration learning process, as: 10. Similar to that, the hidden state/output vector embedding size of the filtration process is configured as: 64. For the implementation of the SCNP model, we set all its hyper-parameters/configurations similar as our SFGE model as listed in the above Table 2. Specifically, for different mapping/pooling strategies which are utilized within the SCNP (as well as our SFGE model), we conducted the experiments with all types of pooling strategies. Then, similar to our model's evaluation strategy, we collected all the performances of different mapping/pooling strategies of the SCNP model. Then, we reported the average accuracy performances as the final results for the comparison purpose. For the other configurations of different GNN-based, TopoGNN-based and SNN-based graph embedding techniques which are related to the training process, like as: training efforts/epochs, default learning rate, dropout rate, optimization strategy, etc., we configured them as the same with our model as listed in the Table 2.

4.2.2 Comparisons on the graph classification task

For the comparative studies, we implemented our proposed SFGE model and other comparative techniques to deal with the graph categorization problem within the real-world B-BeeTox molecular dataset. The accuracy performances in terms of ACC and F1-Macro based evaluation metrics of different techniques are reported as shown in Table 3. In general, as shown from the experimental outputs, our proposed SFGE model has clearly outperformed most of graph learning techniques while dealing with the molecular

graph representation learning and categorization problems. In more specifics, as shown from the results, in comparing with classical GNN-based techniques, such as: GCN, GraphSAGE, GAT, GATv2, GRAND, GIN and GINE—our proposed SFGE model significantly achieve better accuracy performances than previous techniques. In more details, it has improved the classification performances approximately: 25.54/20.2%/52.4%, 23.48%/17.16%/52.67, 23.48%/17.16%/52.67%, 23.48%/18.15%/53.5%, 28.28%/22.33%/42.24%, 19.59%/15.21%/34.8%, 20.57%/16.17%/37.97%, 17.69%/11.48%/20.36% in terms of AUC/ACC/F1-Macro evaluation metrics, respectively. Hence, these experimental outputs have showed the potentiality of simplicial neural network within graph representation learning and classification problems as the multi-way/hidden connectivity information from the input molecular graphs can be sufficiently captured during the representation learning process. In addition, as shown from the experimental outputs, we also can figure out the capability of topology-enhanced graph learning techniques such as the TOGL, SCNP and our proposed SFGE model have achieved better performances than previous conventional explicit-structural graph learning techniques. Therefore, it proved the potentiality of integrating topological structure extraction with the graph feature aggregation process. This integration has supported to enhance the quality of learnt graph representations as well as task-driven fine-tuning process. Last but not least, by looking into details of the experimental results, the overall accuracy performance of SNN-based architectures (including the SCNP and our proposed SFGL model) is significantly higher than the classical GNN-based techniques, averagely: 18.84%, 13.87% and 41.88% in terms of AUC, ACC and F1-Macro accuracy scores. In comparing

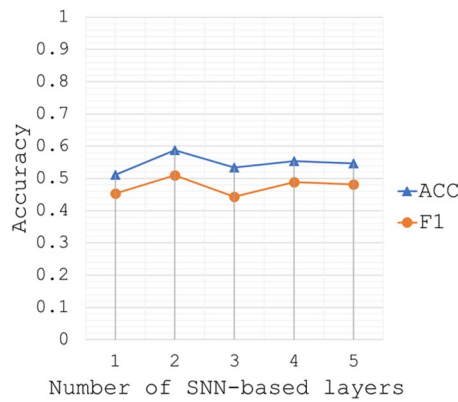


Fig. 2 Studies on the influence of the (k^{SNN}) parameter upon the graph classification task

with topology-enhanced GNN baseline, such as the notable TOGL model, our proposed SFGL model also gained better accuracy performances for the graph classification task, approximately: 23.48%, 17.16% and 52.67% in terms of AUC, ACC and F1-Macro evaluation metrics. Similar to that, the experimental results also indicated that the SFGE model gained better performance than our main competitor, the SCNP model. From the experimental results, it slightly enhanced the AUC/ACC/F1-Macro based accuracy performances about 6.68%/5.93%/0.72%, respectively.

To sum up, as shown from the accuracy performance comparative outcomes between our proposed SFGE model and other graph embedding baselines, it is clear about the effectiveness and outperformance of SFGE. In addition, the comprehensive performance comparisons between our proposed SFGE model with state-of-the-art graph learning techniques in multiple approaches (e.g., explicit graph-structural learning, topological graph filtration and SNN) have proved the necessary of our proposed ideas within this paper. Our model has also shown the fact is that the multi-aspect shared structural (from explicit graph structure) and multi-way relational (from simplicial complex) information of the given graph can be jointly captured during the embedding process. Last but not least, within our model, we also utilized an extra deep neuro-fuzzy network at the output layer of the given graph structure-enhance SNN-based graph learning architecture. This neural fuzzy architecture supports to effectively reduce the negative effects with regarded to the occurrence feature noises/ambiguities. The feature noise and ambiguity are normally occurred within sophisticated multi-modal neural architectures. This NFN-based component supports to effectively transform and reduce the feature uncertainty from graph representations. As the results, it can directly assist to improve the performance of learnt graph representation augmentation as well as graph classification-driven fine-tuning processes in the after all.

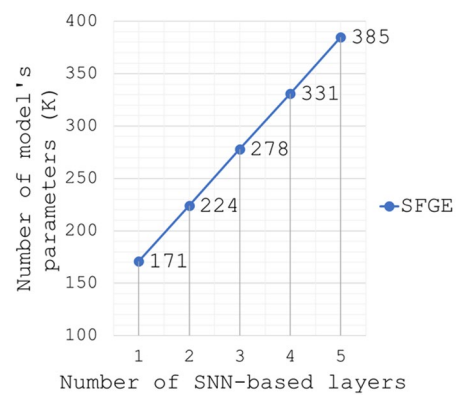


Fig. 3 Model's complexity with different number of SNN-based layers

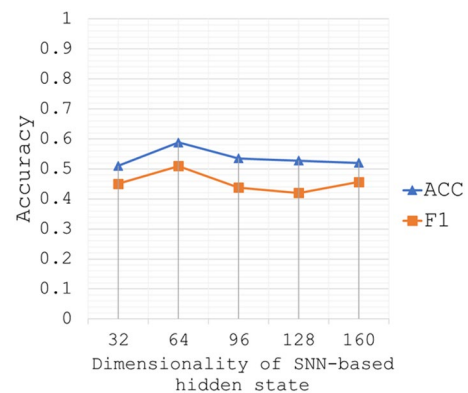


Fig. 4 Studies on the influence of the (d^{SNN}) parameter upon the graph classification task

4.3 Ablation studies

In this section, we conducted the extensive empirical studies in order to evaluate the sensitivity of some important model's configurations/parameters. These critical configurations can have thorough influences upon our model's stability as well as accuracy performance while dealing with graph classification task. The selected configurations/parameters for experiments within this section includes: SNN-based number of hidden layers (k^{SNN}), SNN-based hidden state size (d^{SNN}) and the effectiveness of different SNN-based pool strategies which are implemented within our SFGE model, including: none (no applying pooling) (SFGE-NP), max (SFGE-MAX), self-attention (SFGE-SA) and top-k (SFGE-TK) pooling strategies. First of all, with the number of hidden layers within the SNN-based component, to evaluate the influence of this configuration, we varied the value of (k^{SNN}) parameter within the range of [1, 5], the reported the changes in ACC/F1-Macro accuracy performances for the graph classification task. As shown from the experimental outputs in

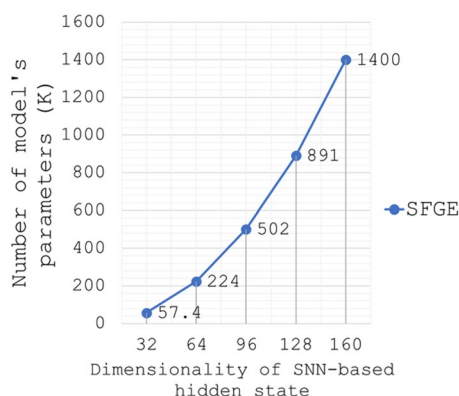


Fig. 5 Model's complexity with different value of SNN-based hidden state size

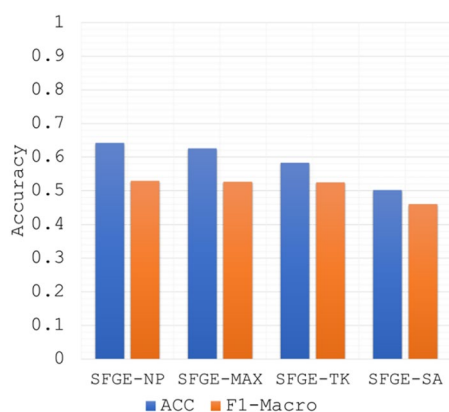


Fig. 6 Extensive evaluation on the performance of different pooling strategies which are used within our proposed SFGE model

Fig. 2, our proposed SFGE model is quite insensitive with this configuration/parameter. The experimental outputs have indicated that despite the changes in number of SNN-based hidden layers, the ACC/F1-Macro based accuracy performance lines of our proposed SFGE model are quite smooth. Thus, it showed the fact is that our model still can achieve acceptable performance with a small number of SNN-based hidden layers in which the model's complexity is linearly increased with this parameter (as shown in Fig. 3). Thus, it can help to efficiently reduce the time and computational cost for the training process.

Similar to that, with the extensive model's parameter evaluations on the dimensionality of SNN-based hidden state parameter (d^{SNN}), the outputs are also the same with experiments for the (k^{SNN}) parameter. Specifically, as shown from the experimental results within the Fig. 4, it seems that our model's accuracy performance lines (for both ACC and F1-Macro evaluation metrics) are quite stable with the changes of this parameter. For the experiments with this

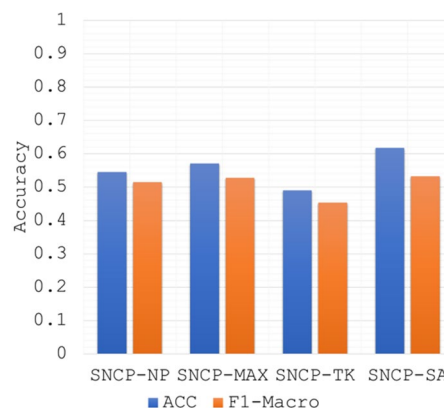


Fig. 7 Extensive analysis on the performances of different pooling strategies of previous SCNP model within B-BeeTox dataset

(d^{SNN}) parameter, we have varied the value of SNN-based hidden state size of our model with range: [32, 160]. As the increase in the value of the (d^{SNN}) parameter has led to dramatically escalation of model's complexity (as shown in Fig. 5). Therefore, choosing a reasonable value for this parameter should be also taken in consideration, especially while handling large-scaled graph datasets. Finally, as our proposed SFGE is mainly adopted the mapping/pooling strategies within the SNN-based architecture from the previous SCNP baseline, there are different pooling strategies have been implemented within the original work. However, within our empirical studies, we only evaluated three types of pooling strategies including: max (SFGE-MAX), top-k (SFGE-TK), self-attention (SFGE-SA) and no-pooling (SFGE-NP). Specifically, within our experimental studies, different pooling strategies are applied within our model and we reported the average ACC/F1-Macro based accuracy performances of these strategies while handling graph classification tasks. As shown from the experimental results in Fig. 6, different pooling strategies didn't show significant enhancements in accuracy performance of our model. Specifically, these pooling strategies didn't provide any clear change in the model's accuracy performance while handling the molecular graph representation learning and classification problems. The results are also similar with the previous SCNP model, as shown in Fig. 7. However, this experimental result doesn't guarantee all pooling types will show the same accuracy performances within different datasets. Therefore, extensive empirical studies about this aspect should be taken in consideration while implementing the model within the practical/realistic environment.

5 Conclusions & future works

In conclusion, in this paper we proposed a novel integration between simplicial neural network and neuro-fuzzy learning framework, called as: SFGE model. Our proposed SFGE model can support to effectively deal with the multi-way relational and uncertainty-reduced graph representation learning problem. In more details, within this paper, our studies mainly focused on designing a novel graph learning technique which adopt from the simplicial complex mapping/pooling and message passing paradigms. Thus, our SFGE model can assist to efficiently capture the multi-way relational information from the input graphs. In the context of simplicial complex-based neural learning approach, the input graphs are modelled as simplicial complexes then learnt through simplicial signal aggregation process. In the simplicial signal aggregation operations of our model, we also implemented an extra GCN-based architecture. This extensive graph learning layer can support to achieve the richer multi-viewed graph representation. These multi-viewed graph representations carry the hidden/multi-way relationships as well as explicit global shared features between graph nodes. This extra graph learning component takes the summarization of simplicial signals as well as the explicit graph's structure as the inputs. Then, it produces the corresponding k-hop relational features between nodes in forms of latent embedding vectors. After that, the different/multi-viewed representations of a graphs are combined through an embedding fusion component. This embedding fusion component is simply designed as a full-connected linear neural layer. In the end, the unified graph output embeddings which are produced by the SNN-based architecture are then fed into a deep fuzzy neural network. Through this approach, our SFGE model can support to transform the given graph representation into feature noise/uncertainty-minimized. In addition, it also supports to generate better classification-friendly graph embeddings which can utilize to improve the accuracy performance of graph classification tasks. In comparing with previous graph embedding techniques our model can directly support to achieve richer-structural multi-way graph representations. The extensive experiments within a real-world molecular graph dataset have demonstrate the effectiveness as well as outperformance of our proposed SFGE model in comparing with previous graph embedding and classification baselines. In comparing with state-of-the-art TopoGNN-based and SNN-based graph learning approaches, our proposed SFGE model has shown the capability of extracting rich multi-scaled graph-structural information in the context of feature noise/ambiguity reduction.

Remained challenges of SFGE model & future works. However, there is still a limitation of our proposed SFGE model related to the multi-view graph structural preserving

aspect. In more specifics, even the graph simplicial signal aggregation process has been improved by implementing an extra graph neural learning mechanism, there are still some limitations of this approach. Our model's major limitation is the capability of dealing with global-levelled graph structural preserving aspect in which the generated representations through our SFGE model for different graphs which are in different structures should be differentiable. This enhanced simplicial signal aggregation mechanism is still considered as unable to distinguish the differences between various structure-varied input graph. Thus, within our next enhancement, we intend to improve the multi-scaled graph learning component of our SFGE model, especially within our model feature aggregation process for node representation. Through this enhancement, we can supplement our model with the ability of efficiently capturing the global structural information from the sophisticated graph structures. This approach is considered as the high-expressive graph learning approach. The integration with recent high-expressive graph learning techniques, like as: GIN and GINE. These high-expressive graph learning architectures are considered as potential enhancements for our model. Last but not least, in our future work, we also plan to extend the architecture of our proposed SFGE model in order to make it capable for dealing with complex graph learning tasks, such as: molecular graph structure prediction and generation. These complex graph learning tasks such as graph-based data generation have plentiful applications in real-world problems, such as new material substance or new drug discoveries.

Acknowledgements This research is funded by HUTECH University, Ho Chi Minh City, Vietnam.

Funding This research was funded by HUTECH University.

Data availability There is no data associated with our studies in this paper.

Declarations

Conflicts of interest Employment: Phu Pham is currently working as full-time researcher and lecturer at HUTECH University. Financial interests: Phu Pham has received research support from HUTECH University. Non-financial interests: None.

References

1. Wu Z, Pan S, Chen F, Long G, Zhang C, Philip SY (2021) A comprehensive survey on graph neural networks. *IEEE Trans Neural Netw Learn Syst* 32(1):4–24
2. Han M, Wu H, Chen Z, Li M, Zhang X (2023) A survey of multi-label classification based on supervised and semi-supervised learning. *Int J Mach Learn Cybern* 14(3):697–724
3. Cui P, Wang X, Pei J, Zhu W (2018) A survey on network embedding. *IEEE Trans Knowl Data Eng* 31(5):833–852

4. Pham P, Do P (2020) W-Metagraph2Vec: a novel approval of enriched schematic topic-driven heterogeneous information network embedding. *Int J Mach Learn Cybern* 11(8):1855–1874
5. Pham P, Nguyen LT, Pedrycz W, Vo B (2022) Deep learning, graph-based text representation and classification: a survey, perspectives and challenges. *Artif Intell Rev* 56(6):4893–4927
6. Alzubi JA, Jain R, Nagraath P, Satapathy S, Taneja S, Gupta P (2021) Deep image captioning using an ensemble of CNN and LSTM based deep neural networks. *J Intell Fuzzy Syst* 40(4):5761–5769
7. Browne F, Wang H, Zheng H (2018) Investigating the impact human protein–protein interaction networks have on disease-gene analysis. *Int J Mach Learn Cybern* 9:455–464
8. Sharma A, Rani R (2020) Drug sensitivity prediction framework using ensemble and multi-task learning. *Int J Mach Learn Cybern* 11(6):1231–1240
9. Wei Y, Ma H, Wang Y, Li Z, Chang L (2023) Dual graph attention networks for multi-behavior recommendation. *Int J Mach Learn Cybern* 14(8):2831–846
10. Pham P, Nguyen LT, Nguyen NT, Pedrycz W, Yun U, Lin JCW, Vo B (2023) An approach to semantic-aware heterogeneous network embedding for recommender systems. *IEEE Trans Cybern* 53(9):6027–6040
11. Mikolov T, Chen K, Corrado G, Dean J (2013) Efficient estimation of word representations in vector space. *ICLR*
12. Pennington J, Socher R, Manning CD (2014) Glove: Global vectors for word representation. *EMNLP*, pp 1532–1543
13. Perozzi B, Al-Rfou R, Skiena S (2014) Deepwalk: Online learning of social representations. *SIGKDD*, pp 701–710
14. Grover A, Leskovec J (2016) Node2vec: Scalable feature learning for networks. *SIGKDD*, pp 855–864
15. Xu W, Guo D, Mi J, Qian Y, Zheng K, Ding W (2023) Two-way concept-cognitive learning via concept movement viewpoint. *IEEE Trans Neural Netw Learn Syst*. <https://doi.org/10.1109/TNNLS.2023.3235800>
16. Alzubi JA, Alzubi OA, Beseiso M, Budati AK, Shankar K (2022) Optimal multiple key-based homomorphic encryption with deep neural networks to secure medical data transmission and diagnosis. *Expert Syst* 39(4):e12879
17. Guo D, Jiang C, Sheng R, Liu S (2022) A novel outcome evaluation model of three-way decision: a change viewpoint. *Inf Sci* 607:1089–1110
18. Alzubi OA, Alzubi JA, Alweshah M, Qiqieh I, Al-Shami S, Ramachandran M (2020) An optimal pruning algorithm of classifier ensembles: dynamic programming approach. *Neural Comput Appl* 32:16091–16107
19. Movassagh AA, Alzubi JA, Gheisari M, Rahimi M, Mohan S, Abbasi AA, Nabipour N (2021) Artificial neural networks training algorithm integrating invasive weed optimization with differential evolutionary model. *J Ambient Intell Humaniz Comput*. <https://doi.org/10.1007/s12652-020-02623-6>
20. Kipf TN, Welling M (2017) Semi-supervised classification with graph convolutional networks. *ICLR*
21. Hamilton W, Ying Z, Leskovec J (2017) Inductive representation learning on large graphs, vol 30. *NeurIPS*
22. Veličković P, Cucurull G, Casanova A, Romero A, Lio P, Bengio Y (2018) Graph attention networks. *ICLR*
23. Feng W, Zhang J, Dong Y, Han Y, Luan H, Xu Q, Tang J (2020) Graph random neural networks for semi-supervised learning on graphs. *NeurIPS* 33:22092–22103
24. Brody S, Alon U, Yahav E (2022) How Attentive are Graph Attention Networks?. *ICLR*
25. Xu K, Hu W, Leskovec J, Jegelka S (2019) How Powerful are graph neural networks?. *ICLR*
26. Hu W, Liu B, Gomes J, Zitnik M, Liang P, Pande V, Leskovec J (2019) Strategies for pre-training graph neural networks. *ICLR*
27. Ying Z, You J, Morris C, Ren X, Hamilton W, Leskovec J (2018) Hierarchical graph representation learning with differentiable pooling. *NeurIPS* 31
28. Zhang M, Cui Z, Neumann M, Chen Y (2018) An end-to-end deep learning architecture for graph classification. *AAAI* 32(1)
29. Hofer C, Graf F, Rieck B, Niethammer M, Kwitt R (2020) Graph filtration learning. *ICML*, pp 4314–4323
30. Zhao Q, Ye Z, Chen C, Wang Y (2020) Persistence enhanced graph neural network. *AISTATS*, pp 2896–2906
31. Horn M, De Brouwer E, Moor M, Moreau Y, Rieck B, Borgwardt K (2022) Topological graph neural networks. *ICLR*
32. Hensel F, Moor M, Rieck B (2021) A survey of topological machine learning methods. *Front Artif Intell* 4:681108
33. Ebli S, Defferrard M, Spremann G (2020) Simplicial neural networks. *NeurIPS 2020 Workshop TDA and Beyond*
34. Bunch E, You Q, Fung G, Singh V (2020) Simplicial 2-complex convolutional neural nets. *NeurIPS 2020 Workshop TDA and Beyond*
35. Yang M, Isufi E, Leus G (2022) Simplicial convolutional neural networks. *ICASSP*, pp 8847–8851
36. Cinque DM, Battiloro C, Di Lorenzo P (2023) Pooling strategies for simplicial convolutional networks. *ICASSP*, pp 1–5
37. Xu W, Yuan K, Li W, Ding W (2022) An emerging fuzzy feature selection method using composite entropy-based uncertainty measure and data distribution. *IEEE Trans Emerg Topics Comput Intell* 7(1):76–88
38. Xu W, Guo D, Qian Y, Ding W (2022) Two-way concept-cognitive learning method: a fuzzy-based progressive learning. *IEEE Trans Fuzzy Syst*. <https://doi.org/10.1109/TFUZZ.2022.3216110>
39. Yuan K, Xu W, Li W, Ding W (2022) An incremental learning mechanism for object classification based on progressive fuzzy three-way concept. *Inf Sci* 584:127–147
40. Guo D, Xu W (2023) Fuzzy-based concept-cognitive learning: An investigation of novel approach to tumor diagnosis analysis. *Inf Sci* 639:118998
41. Guo D, Xu W, Qian Y, Ding W (2023) M-FCCL: Memory-based concept-cognitive learning for dynamic fuzzy data classification and knowledge fusion. *Inf Fusion* 100:101962
42. Zhang L, Shi Y, Chang YC, Lin CT (2023) Robust fuzzy neural network with an adaptive inference engine. *IEEE Trans Cybern*. <https://doi.org/10.1109/TCYB.2023.3241170>
43. Guo D, Xu W, Qian Y, Ding W (2023) Fuzzy-granular concept-cognitive learning via three-way decision: performance evaluation on dynamic knowledge discovery. *IEEE Trans Fuzzy Syst*. <https://doi.org/10.1109/TFUZZ.2023.3325952>
44. Pham P, Nguyen LT, Nguyen NT, Kozma R, Vo B (2023) A hierarchical fused fuzzy deep neural network with heterogeneous network embedding for recommendation. *Inf Sci* 620:105–124
45. Pham P, Nguyen LT, Vo B, Yun U (2022) Bot2Vec: A general approach of intra-community oriented representation learning for bot detection in different types of social networks. *Inf Syst* 103:101771
46. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez AN, Polosukhin, I (2017) Attention is all you need. *NeurIPS* 30
47. Lee J, Lee I, Kang J (2019) Self-attention graph pooling. *ICML*, pp 3734–3743
48. Yang M, Isufi E, Schaub MT, Leus G (2021) Finite impulse response filters for simplicial complexes. *EUSIPCO*, pp 2005–2009
49. Barbarossa S, Sardellitti S (2020) Topological signal processing over simplicial complexes. *IEEE Trans Signal Process* 68:2992–3007
50. Schaub MT, Zhu Y, Seby JB, Roddenberry TM, Segarra S (2021) Signal processing on higher-order networks: Livin' on the edge... and beyond. *Signal Process* 187:108149

51. Bodnar C, Frasca F, Wang Y, Otter N, Montufar GF, Lio P, Bronstein M (2021) Weisfeiler and lehman go topological: Message passing simplicial networks. ICML, pp 2005–2009
52. Giusti L, Battiloro C, Di Lorenzo P, Sardellitti S, Barbarossa S (2022) Simplicial attention networks. ICLR 2022 Workshop on Geometrical and Topological Representation Learning
53. Goldberg TE (2002) Combinatorial Laplacians of simplicial complexes. Senior Thesis, Bard College vol 6
54. Lim LH (2020) Hodge Laplacians on graphs. SIAM Rev 62(3):685–715
55. Wang F, Yang JF, Wang MY, Jia CY, Shi XX, Hao GF, Yang GF (2020) Graph attention convolutional neural network model

for chemical poisoning of honey bees' prediction. Sci Bull 65(14):1184–1191

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Springer Nature or its licensor (e.g. a society or other partner) holds exclusive rights to this article under a publishing agreement with the author(s) or other rightsholder(s); author self-archiving of the accepted manuscript version of this article is solely governed by the terms of such publishing agreement and applicable law.