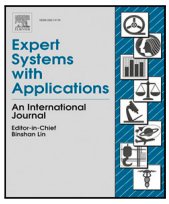
Expert Systems With Applications 271 (2025) 126676

Contents lists available at ScienceDirect

Expert Systems With Applications

journal homepage: www.elsevier.com/locate/eswa



# When graph neural networks meet deep nonnegative matrix factorization: An encoder and decoder-like method for community detection

Junwei Cheng [a], Chaobo He [a,*], Xuequan Lin [a], Weixiong Liu [a], Kunlin Han [b,c], Yong Tang [a]

[a] School of Computer Science, South China Normal University, Guangzhou China
[b] CMT US Holdings, LLC, Chicago, USA
[c] Computer Science Department, University of Southern California, Los Angeles USA

ARTICLE INFO

*Keywords:*
Graph neural networks
Deep nonnegative matrix factorization
Autoencoder
Community detection

ABSTRACT

Community detection is the best choice for accomplishing a variety of tasks in networks. It aims to find the optimal cluster structures, i.e., the high correlation between nodes in intra-cluster and low correlation between them in different clusters. Deep Nonnegative Matrix Factorization (DNMF) and Graph Neural Networks (GNNs) have attracted much attention because of their advantages. And they have been widely adopted for community detection. Unfortunately, the performance of these methods is limited by their respective shortcomings. Therefore, their performance is distant from satisfactory. Especially, the innate feature of DNMF makes it difficult to cope with complex networks and withstand topology noise. Additionally, the performance of GNNs is limited by over-smoothing problem. To overcome the aforementioned flaws, we propose a novel method, SADNG, which connects the DNMF and GNNs seamlessly. Specifically, the over-smoothing problem of GNNs is alleviated when DNMF participates in convolution operations. And the abilities of DNMF to withstand topology noise and to reconstruct non-linear networks are boosted when GNNs participates in reconstructing weighted networks. In all, this way of cooperation compensates for their respective disadvantages. Experimental results across multiple benchmark datasets, including Cora and Citeseer, demonstrate that SADNG is superior to most state-of-the-art methods in community detection.

## 1. Introduction

Networks are ubiquitous and play a major role in real-word. For example, in social networks (Teng et al., 2021), each node corresponds to a user and each edge denotes the friendship between them. To complete a variety of tasks (He et al., 2024; Xu et al., 2021) in social networks, a natural idea is to divide these nodes into communities, i.e., group of nodes that are well connected inside and weakly connected outside (Newman & Girvan, 2004). It is well known that analyzing these community structures is of importance to reveal the network functions.

A straightforward choice is community detection, which can be helpful to investigate community structures and understand the topology information to accomplish the aforementioned tasks. Community detection approaches usually assign each node to only one community. And it seeks to find the optimal community structures by certain methods, e.g., Nonnegative Matrix Factorization (NMF) (Wang et al., 2023), GNNs (Tsitsulin et al., 2023), Modularity (Schuetz & Caflisch, 2008), and Random Walk (Rosvall & Bergstrom, 2008). Therefore, how to find the optimal community structures has become a problem that all the above methods need to solve.

Especially, the characteristics of NMF, high interpretability and extensibility (He et al., 2022a), have drawn great attention. Sun et al. (2017) proposed a NMF-based method with encoder–decoder. By using encoder–decoder and nonnegative properties, they capture the low-dimensional representation of topology structure and reflect the community membership of nodes. Pompili et al. (2014a), Yuan and Oja (2005) and Psorakis et al. (2011) all contributed to the development of NMF. Despite the success of NMF, it may suffer from performance degradation when encountering complex networks (He et al., 2022b). A primary obstacle is the topology noise, such as networks with large hub nodes. And the other is the non-linear feature of complex networks (Ibrahim & Gleich, 2019). Inspired by deep learning, Trigeorgis et al. (2017a) proposed DNMF to develop the original NMF to form more high-accurate community membership, even though it is still a linear method. DNMF aims to alleviate topology noise by learning multiple layers of hierarchical mappings. Ye et al. (2018) further proposed

* Corresponding author.
*E-mail addresses:* jung@m.scnu.edu.cn (J. Cheng), hechaobo@m.scnu.edu.cn (C. He), xuequanlin@m.scnu.edu.cn (X. Lin), weixiongliu@m.scnu.edu.cn (W. Liu), kunlinha@usc.edu (K. Han), ytang@m.scnu.edu.cn (Y. Tang).

https://doi.org/10.1016/j.eswa.2025.126676
Received 9 July 2024; Received in revised form 3 November 2024; Accepted 24 January 2025
Available online 1 February 2025
0957-4174/© 2025 Elsevier Ltd. All rights are reserved, including those for text and data mining, AI training, and similar technologies.

DANMF with deep autoencoder characteristics. It is a clever architecture that combines DNMF and deep autoencoder. This architecture empowers their method to learn the hierarchical mappings, from high-dimensional to low-dimensional, between the original network and the final community membership.

Meantime, methods based on GNNs have made significant progress in community detection due to their non-linear feature. Cui et al. (2020) designed Laplacian smoothing filter and adaptive encoder to get better node embedding. Bo et al. (2020) designed a method with multiple structures. The subtlety of this method is that the delivery operator inside GNNs can learn the node embedding from the autoencoder.

Both DNMF-based methods and GNNs-based methods have their own advantages by analyzing the aforementioned method. Two natural questions are being asked: *How to construct a method that combines their respective advantages in a seamless way?* and further *How to alleviate the respective drawbacks of DNMF and GNNs by the method?* In this paper, inspired by Ye et al. (2018) and Bo et al. (2020), we answer these two questions. For the first question, we leverage the powerful representation of DNMF to obtain the hierarchical mapping. And then, we make it participate in the convolution operation of GNNs. For another question, we alleviate the flaws of GNNs and DNMF by mapping transmission and reconstructing networks. On one hand, the over-smoothing problem (Zhao & Akoglu, 2020) of GNNs is alleviated via DNMF. On the other hand, the abilities of DNMF to further withstand topology noise and to reconstruct non-linear networks are boosted via GNNs.

In conclusion, the work of this paper can be summarized as

- We propose a novel Structured Autoencoder-like method combining Deep Nonnegative matrix factorization and Graph neural networks (SADNG). It can be applied to topology networks. The proposed SADNG seamlessly combines the DNMF and GNNs and has a non-linear feature. To the best of our knowledge, this is the first time that GNNs and DNMF are fused into a unified method.
- We design a mechanism that combines mapping transmission and reconstruction networks. Firstly, this mechanism can boost the abilities of DNMF to withstand topology noise and to reconstruct non-linear networks. Additionally, it can alleviate the over-smoothing problem of GNNs. Thus, the performance of the unified method, SADNG, is remarkable.
- Our extensive experiments on a series of benchmark datasets indicate that the SADNG outperforms the representative related methods.

The rest of the paper is organized as follows. Section 2 introduces the related work of GNNs, NMF and DNMF. Section 3 provides a detailed description of preliminaries, notations and procedures of SADNG. Section 4 presents the experimental results in real topology networks. Finally, we draw conclusions in Section 5.

## 2. Related work

In this section, we will review the respective changes according to the development of GNNs, NMF and DNMF (He et al., 2022a; Su et al., 2024). Firstly, we introduce the application of GNNs in community detection. Then, we present the development of NMF and DNMF in topology networks.

### 2.1. GNNs-based methods

GNNs generally fall into two categories: spectral methods (Kipf & Welling, 2017; Zheng et al., 2024) and spatial methods (Ahmed et al., 2017; Zhuang & Ma, 2018). Methods based on the spectral domain theory all rely on the eigenvector vectors of Laplacian matrix. Further, Laplacian matrix is closely related to graph structure, which means that such methods cannot be generalized. Spatial domain-based methods compute adjacent domains via directly defining the convolution operation in graphs. Obviously, the challenge of such methods is to define the convolution operation for graphs of different scales. Recently, various derivative methods have been proposed because GNNs have a solid theoretical foundation. These derivative methods can be classified into four categories, including Graph Attention Networks-based methods (Jing et al., 2021; Luo et al., 2021), Graph Adversarial Networks-based methods (Chen et al., 2022; Wang et al., 2021), Graph Convolutional Networks-based methods (GCN) (Zhang et al., 2021; Zhao et al., 2021) and Graph Autoencoder-based methods (Salahian et al., 2023; Xia et al., 2022). For example, Tiwari et al. (2024) proposed a graph-regularized self-representation learning method, namely GRSSLFS, to address limitations in feature selection by reducing redundant and noisy features.

Although these GNNs-based methods have different architectures, they have two commonalities, i.e., non-linear feature and aggregating operation. Non-linear feature is extremely significant for GNNs, which helps GNNs to obtain the diversification node embedding in complex networks. For example, NOCD (Shchur & Günnemann, 2019) has effectively outperformed the others, although it looks very simple, because it leverages the non-linear feature of GNNs and probability-based loss function (Yang et al., 2013). Aggregating operations increase the coupling of nodes in networks, i.e., each node can learn more information from neighboring nodes. It looks like this way is very constructive because it allows node embedding to contain more topology information. However, there will be an over-smoothing problem if too much information of neighbor nodes is aggregated. Some works have attempted to solve this kind of problem, such as designing new filters (Bo et al., 2021; Wu et al., 2023).

### 2.2. NMF-based methods and DNMF-based methods

NMF has fully displayed some unique advantages compared with other methods (Liu et al., 2020), such as higher interpretability, simplicity and generality. These advantages benefit from an ingenious background of NMF, i.e., original networks could be reconstructed by using a linear operator of community membership and bias matrix. Therefore, a straightforward goal of NMF-based methods is to make the reconstructed network as close to the original network as possible. Yuan and Oja (2005) chose to project the original network to low-dimensional space to get accurate community membership. Pompili et al. (2014b) improved original NMF by imposing constraints for community membership. Deng et al. (2023) proposed a graph-regularized sparse NMF (GSNMF) to enhance traditional regularized NMF methods by addressing their susceptibility to noise from the square loss function in data reconstruction.

However, the topology noise in real networks makes it impossible for traditional NMF-based methods to achieve a pure community membership. Recently, inspired by deep learning, NMF is developed into a multi-level structure, i.e., DNMF (Trigeorgis et al., 2017b). DNMF not only has the advantages of NMF but also can get more high-accurate community membership by learning the hierarchical mappings from high-dimensional to low-dimensional. By learning hierarchical mappings of different dimensions, the impact of topology noise on the method is mitigated. Some representative methods are DANMF (Ye et al., 2018) and MDNMF (Huang et al., 2021). On this basis, Al-sharoa and Rahahleh (2023) proposed a deep robust autoencoder NMF to enhance community detection accuracy within noisy networks. Building on these advancements, Cheng et al. (2024) developed an adaptive deep NMF model to tackle challenges posed by topology and attribute noise in attributed networks, using a dual-DNMF with an autoencoder and adaptive attention mechanisms. Similarly, Li et al. (2024) introduced a contrastive deep NMF method to overcome limitations in capturing hierarchical structures and learning global network patterns in traditional NMF approaches. Further extending these concepts, Saberi-Movahed et al. (2024) introduced Dn$^2$MF$^{GL}$,

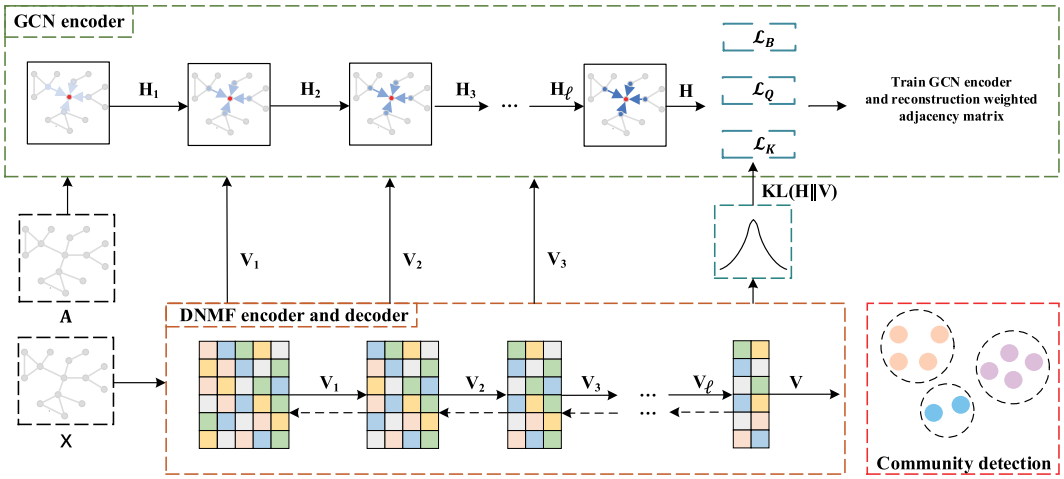



**Fig. 1.** SADNG is composed of GCN and DNMF. Overall, DNMF enables it to participate in convolution operation of each layer of GCN by transferring $\mathbf{V}_i$. And further guides the training process of GCN via Kullback–Leibler divergence. Moreover, GCN can be reconstructed weighted topology networks to make it a new input for DNMF. Finally, we can infer the communities from $\mathbf{V}$.

a deep NMF method that preserves both global and local geometric structures, addressing structural limitations of traditional DNMF models. Finally, Wang et al. (2024) presented an autoencoder-like deep NMF representation learning (ADNRL) model to improve traditional autoencoder-based deep NMFs, focusing on handling nonlinear mappings and gradient-based updates more effectively.

Apparently, both NMF-based methods and DNMF-based methods have immense potential in community detection. But, a feature will be found that limits their development when we review the background of NMF or DNMF, i.e., linear feature. In other words, the simple linear feature is inadequate to representations complex networks. This drawback, which is regarded as a double-edged sword from our perspective, will cause many methods to fail to achieve their expected results. In all, our proposed method, SADNG, integrates GCN and DNMF in a seamless way. What is more, SADNG inherits the advantages of DNMF and GCN and compensates for their respective disadvantages.

## 3. Methodology

In this section, we describe our method, i.e., SADNG, for community detection. Firstly, we give preliminaries of method and introduce notations. Then, we present the details of the various units in SADNG. Finally, we give a unified loss function and optimization rules of SADNG.

### 3.1. Preliminaries and notations

The parallel architecture of SADNG is shown in Fig. 1. It consists of three units: GCN encoder, DNMF encoder–decoder and reconstruction weighted topology networks.

- **GCN encoder**: The convolution operation of multi-layer GCN can be deemed as a high-dimensional to low-dimensional purification operation. We call this high- to low-dimensional operation as encoder. When the number of GCN layers is small, such as no more than two layers in some datasets, the GCN encoder provides good pre-data for subsequent community detection. However, when the number of convolutional layers of GCN is too large, the problem of over-smoothing is often prone to occur, which seriously affects the purity of the pre-data. Overall, the number of layers of GCN affects its performance. In order to alleviate the negative effect of over-smoothing problem on the GCN encoder, we incorporate the hierarchical mapping of DNMF-decoder in the convolution operation. And a hyperparameter is used to balance their linear combination. Notice, in linear combination, the matrix of GCN part is the GCN convolution result from the previous layer, while the other matrix comes from the DNMF hierarchical mapping in the current layer.
- **DNMF encoder–decoder**: DNMF is capable of alleviating topology noise by learning hierarchical mapping. At the encoder stage, each layer is a standard NMF with coupling relation, i.e., the output of the previous layer of NMF is the current layer's input (except for the first layer). The decoder stage can be deemed as a low- to high-dimensional restoration operation. Therefore, similar to Sun et al. (2017), we can update each matrix in DNMF from a holistic perspective and further obtain more high-accuracy community membership.
- **Adhesive**: We boost the stability of SADNG by using Kullback–Leibler divergence (KL divergence) and reconstructing weighted topology networks. On the one hand, we aim to guide the training of GCN by using the accurate community membership of DNMF with the help of KL divergence. On the other hand, we need to reconstruct weighted topology networks by using the finished training GCN. We use the non-linear feature of GCN to reconstruct the network so that the new network contains more information and makes up for non-linear flaw of DNMF.

Now, we introduce the notations that will be used to describe SADNG. Given an undirected network $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ over a set of $\mathcal{N}$ nodes $\mathcal{V} = \{v_1, v_2, \ldots, v_{\mathcal{N}}\}$, a set of $e$ edges $\mathcal{E} = \{(v_i, v_j)\}$. The other frequently used notations in this paper are summarized in Table 1.

**Problem statement**: Given an undirected network $\mathcal{G}$, dividing its nodes into $k$ communities by means of GCN and DNMF. It is very important to point out that the original attribute information of nodes is prohibited to be employed, and we need to detect communities in the unsupervised scenario.

### 3.2. DNMF encoder and decoder

**Encoder module**. Encoder module is aimed to transform hierarchical mapping from high- to low-dimensional, so that we can mitigate the effects of topology noise and obtain the purified community membership.

In this sense, the $\mathbf{X}$ need to be factored into $\ell+1$ nonnegative factor matrices to satisfy the objective. In simple terms, given $\mathcal{G}$, standard NMF approximates the $\mathbf{X}$ by the product of two nonnegative low-rank matrices, i.e., $\mathbf{X} \approx \mathbf{WV}$. Therefore, the goal of encoder module can be written as:

$$\mathbf{X} \approx \mathbf{W}_1\mathbf{W}_2 \cdots \mathbf{W}_\ell\mathbf{V}_\ell \tag{1}$$

**Table 1**
Notations.

| Notations | Descriptions |
|---|---|
| $\mathbf{A}$ | Adjacency matrix |
| $\mathbf{X}$ | Weighted adjacency matrix |
| $\ell$ | Number of SADNG layers |
| $\mathbf{V}_i$ | Hierarchical mapping at $i$th layer, where $0 < i \leq \ell$. When $i = \ell$, $\mathbf{V}_\ell$ is called community membership. |
| $\mathbf{W}_i$ | Bias matrix at $i$th layer where $0 < i \leq \ell$ |
| $\mathbf{H}_i$ | Node embedding at $i$th layer, where $0 < i \leq \ell$ |
| $\sigma(\cdot)$ | Activation function |
| $k$ | Number of communities |
| $I$ | Identity matrix |

where $\mathbf{X}$ will be initialized by $\mathbf{A}$ during training.

Obviously, the Eq. (1) does not have an explicit hierarchical relationship. It is essential to establish explicit coupling between layers. Thus, the hierarchical conditions need to be imposed on the formulation in Eq. (1) as follows:

$$\begin{aligned} \mathbf{X} &\approx \mathbf{W}_1\mathbf{V}_1 \\ \mathbf{V}_1 &\approx \mathbf{W}_2\mathbf{V}_2 \\ &\cdots \\ \mathbf{V}_{\ell-1} &\approx \mathbf{W}_\ell\mathbf{V}_\ell \end{aligned} \tag{2}$$

It is clear that each layer in Eq. (2) is standard NMF and the decomposition result, such as $\mathbf{V}_i$ where $0 < i < \ell$, is further decomposed in the next layer.

In all, this deep structure can enhance the ability to withstand topology noise and obtain more purified community membership. Finally, the objective of encoder can be rewritten as:

$$\begin{aligned} &\min_{\mathbf{W}_i,\mathbf{V}_i} \mathcal{L}_{en} = \|\mathbf{X} - \mathbf{W}_1\mathbf{W}_2\cdots\mathbf{W}_\ell\mathbf{V}_\ell\|_F^2, \\ &\quad s.t. \mathbf{W}_i \geq 0, \mathbf{V}_i \geq 0, i = 1, 2, \ldots, \ell \end{aligned} \tag{3}$$

**Decoder module**. Decoder strives to restore the original network, i.e., the low-dimensional hierarchical mapping is restored to high-dimensional. That is a critical stage due to the feature of NMF. On the one hand, decoder shrinks the reconstruction error. On the other hand, guaranteeing $\mathbf{V}_\ell$ to be capable of getting purified community membership.

To satisfy the above objective, the $\mathbf{V}$ can be derived as the product of $\mathbf{W}$ and $\mathbf{X}$, which can be expressed as $\mathbf{V} \approx \mathbf{W}^T\mathbf{X}$. Hence, the primary purpose of the decoder module can be formulated as follows:

$$\mathbf{V}_\ell \approx \mathbf{W}_\ell^T\mathbf{W}_{\ell-1}^T\cdots\mathbf{W}_1^T\mathbf{X} \tag{4}$$

Similar to the Eq. (2), the explicit hierarchical relationship of Eq. (4) need to be imposed as follows:

$$\begin{aligned} \mathbf{V}_\ell &\approx \mathbf{W}_\ell^T\mathbf{V}_{\ell-1} \\ \mathbf{V}_{\ell-1} &\approx \mathbf{W}_{\ell-1}^T\mathbf{V}_{\ell-2} \\ &\cdots \\ \mathbf{V}_1 &\approx \mathbf{W}_1^T\mathbf{X} \end{aligned} \tag{5}$$

Further, the object of decoder can be rewritten as:

$$\begin{aligned} &\min_{\mathbf{W}_i,\mathbf{V}_i} \mathcal{L}_{de} = \|\mathbf{V}_\ell - \mathbf{W}_\ell^T\mathbf{W}_{\ell-1}^T\cdots\mathbf{W}_1^T\mathbf{X}\|_F^2, \\ &\quad s.t. \mathbf{W}_i \geq 0, \mathbf{V}_i \geq 0, i = 1, 2, \ldots, \ell \end{aligned} \tag{6}$$

By combining Eqs. (3) and (6), we can get the unified objective function of DNMF as follows:

$$\begin{aligned} &\min_{\mathbf{W}_i,\mathbf{V}_i} \mathcal{L} = \mathcal{L}_{en} + \mathcal{L}_{de} = \\ &\|\mathbf{X} - \mathbf{W}_1\mathbf{W}_2\cdots\mathbf{W}_\ell\mathbf{V}_\ell\|_F^2 + \|\mathbf{V}_\ell - \mathbf{W}_\ell^T\mathbf{W}_{\ell-1}^T\cdots\mathbf{W}_1^T\mathbf{X}\|_F^2, \\ &\quad s.t. \mathbf{W}_i \geq 0, \mathbf{V}_i \geq 0, i = 1, 2, \ldots, \ell \end{aligned} \tag{7}$$

### 3.3. Optimization

Encoder of DNMF adapts standard NMF to train each layer. To establish explicit coupling between layers, we use former output as current input, i.e., Eq. (2). Decoder of DNMF needs to be fine-tuned by minimization of objective function in Eq. (7). Therefore, we give the updating rule of to minimize the objective function, as follows:

$$\begin{aligned} \mathcal{L} = &tr[(\mathbf{X} - \boldsymbol{\Phi}_{i-1}\mathbf{W}_i\boldsymbol{\Psi}_{i+1}\mathbf{V}_\ell)^T(\mathbf{X} - \boldsymbol{\Phi}_{i-1}\mathbf{W}_i\boldsymbol{\Psi}_{i+1}\mathbf{V}_\ell)] \\ &+tr[(\mathbf{V}_\ell - \boldsymbol{\Psi}_{i+1}^T\mathbf{W}_i^T\boldsymbol{\Phi}_{i-1}^T\mathbf{X})^T(\mathbf{V}_\ell - \boldsymbol{\Psi}_{i+1}^T\mathbf{W}_i^T\boldsymbol{\Phi}_{i-1}^T\mathbf{X})] \\ = &tr(\mathbf{X}^T\mathbf{X} - 2\mathbf{X}^T\boldsymbol{\Phi}_{i-1}\mathbf{W}_i\boldsymbol{\Psi}_{i+1}\mathbf{V}_\ell + \\ &\mathbf{V}_\ell^T\boldsymbol{\Psi}_{i+1}^T\mathbf{W}_i^T\boldsymbol{\Phi}_{i-1}^T\boldsymbol{\Phi}_{i-1}\mathbf{W}_i\boldsymbol{\Psi}_{i+1}\mathbf{V}_\ell) + \\ &tr(\mathbf{V}_\ell^T\mathbf{V}_\ell - 2\mathbf{V}_\ell^T\boldsymbol{\Psi}_{i+1}^T\mathbf{W}_i^T\boldsymbol{\Phi}_{i-1}^T\mathbf{X} + \\ &\mathbf{X}^T\boldsymbol{\Phi}_{i-1}\mathbf{W}_i\boldsymbol{\Psi}_{i+1}\boldsymbol{\Psi}_{i+1}^T\mathbf{W}_i^T\boldsymbol{\Phi}_{i-1}^T\mathbf{X}) \end{aligned} \tag{8}$$

where $\boldsymbol{\Phi}_{i-1} = \mathbf{W}_1\mathbf{W}_2\cdots\mathbf{W}_{i-1}$ and $\boldsymbol{\Psi}_{i+1} = \mathbf{W}_{i+1}\mathbf{W}_{i+2}\cdots\mathbf{W}_\ell$.

By fixing all the variables except for $\mathbf{W}_i$ $(0 < i \leq \ell)$, the partial derivative of Eq. (8) is deduced as:

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\mathbf{W}_i} = &-2\boldsymbol{\Phi}_{i-1}^T\mathbf{X}\mathbf{V}_\ell^T\boldsymbol{\Psi}_{i+1}^T + 2\boldsymbol{\Phi}_{i-1}^T\boldsymbol{\Phi}_{i-1}\mathbf{W}_i\boldsymbol{\Psi}_{i+1}\mathbf{V}_\ell\mathbf{V}_\ell^T\boldsymbol{\Psi}_{i+1}^T \\ &-2\boldsymbol{\Phi}_{i-1}^T\mathbf{X}\mathbf{V}_\ell^T\boldsymbol{\Psi}_{i+1}^T + 2\boldsymbol{\Phi}_{i-1}^T\mathbf{X}\mathbf{X}^T\boldsymbol{\Phi}_{i-1}\mathbf{W}_i\boldsymbol{\Psi}_{i+1}\boldsymbol{\Psi}_{i+1}^T \end{aligned} \tag{9}$$

From the multiplicative update rule of Oja (Pizzuti & Socievole, 2020), we can obtain:

$$\mathbf{W}_i = \mathbf{W}_i\frac{2\boldsymbol{\Phi}_{i-1}^T\mathbf{X}\mathbf{V}_\ell^T\boldsymbol{\Psi}_{i+1}^T}{\Delta} \tag{10}$$

where $\Delta$ as follows:

$$\begin{aligned} \Delta = &\boldsymbol{\Phi}_{i-1}^T\boldsymbol{\Phi}_{i-1}\mathbf{W}_i\boldsymbol{\Psi}_{i+1}\mathbf{V}_\ell\mathbf{V}_\ell^T\boldsymbol{\Psi}_{i+1}^T + \\ &\boldsymbol{\Phi}_{i-1}^T\mathbf{X}\mathbf{X}^T\boldsymbol{\Phi}_{i-1}\mathbf{W}_i\boldsymbol{\Psi}_{i+1}\boldsymbol{\Psi}_{i+1}^T \end{aligned} \tag{11}$$

By fixing all the variables except for $\mathbf{V}_i$ $(0 < i \leq \ell)$, the partial derivative of Eq. (8) is deduced as:

$$\begin{aligned} \frac{\partial\mathcal{L}}{\partial\mathbf{V}_i} = &-2\boldsymbol{\Psi}_{i+1}^T\mathbf{W}_i^T\boldsymbol{\Phi}_{i-1}^T\mathbf{X} + 2\mathbf{V}_i - 2\boldsymbol{\Psi}_{i+1}^T\mathbf{W}_i^T\boldsymbol{\Phi}_{i-1}^T\mathbf{X} + \\ &2\boldsymbol{\Psi}_{i+1}^T\mathbf{W}_i^T\boldsymbol{\Phi}_{i-1}^T\boldsymbol{\Phi}_{i-1}\mathbf{W}_i\boldsymbol{\Psi}_{i+1}\mathbf{V}_i \end{aligned} \tag{12}$$

Following similar derivation process of the updating rule for $\mathbf{W}_i$, the updating rule for $\mathbf{V}_i$ $(0 < i \leq \ell)$ is formulated as follows:

$$\mathbf{V}_i = \mathbf{V}_i\frac{2\boldsymbol{\Psi}_{i+1}^T\mathbf{W}_i^T\boldsymbol{\Phi}_{i-1}^T\mathbf{X}}{\mathbf{V}_i + \boldsymbol{\Psi}_{i+1}^T\mathbf{W}_i^T\boldsymbol{\Phi}_{i-1}^T\boldsymbol{\Phi}_{i-1}\mathbf{W}_i\boldsymbol{\Psi}_{i+1}\mathbf{V}_i} \tag{13}$$

### 3.4. GCN encoder and reconstruct networks

**Encoder module**. The Encoder–Decoder of DNMF is capable of learning high-quality hierarchical mappings from the data itself, such as $\mathbf{V}_1, \mathbf{V}_2, \ldots, \mathbf{V}_{\ell-1}$. However, due to its linear features, it falls short in reconstructing complex networks for complete community detection. This limitation affects the accuracy of the hierarchical mapping. The nonlinearity of GCN, on the other hand, addresses this issue by using node embeddings to reconstruct networks. Therefore, obtaining accurate node embeddings is crucial. Specifically, the node embedding of the $i$th layer of GCN, denoted as $\mathbf{H}_i$, can be obtained through the following convolution operation:

$$\mathbf{H}_i = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{H}_{i-1}\mathbf{B}_{i-1}) \tag{14}$$

where $\tilde{\mathbf{A}} = \mathbf{A} + \mathbf{I}$, $\tilde{\mathbf{D}}$ is diagonal matrix where $\tilde{\mathbf{D}}_{jj} = \sum_s \tilde{\mathbf{A}}_{js}$ and $\mathbf{B}$ is the weight matrix.

As we can see from Eq. (14), the node embedding $\mathbf{H}_{i-1}$ will propagate through the normalized adjacency matrix, $\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}$. And finally to obtain the new node embedding $\mathbf{H}_i$. Ideally, we can obtain high-accurate node embedding by continuous convolution operations. But the appearance of over-smoothing problem will seriously affect the accuracy of node embedding when the number of layers of GCN is greater

than two. Therefore, it is a clever way that transfer the $\mathbf{V}_i$ to convolution operations of GCN which relieves the impact of over-smoothing problem on node embedding. The transfer of $\mathbf{V}_i$ as follow:

$$\tilde{\mathbf{H}}_{i-1} = (1-\epsilon)\mathbf{H}_{i-1} + \epsilon\mathbf{V}_{i-1} \tag{15}$$

where $\epsilon$ is a balance coefficient.

Then, we use $\tilde{\mathbf{H}}_{i-1}$ as new input of the $i$-th layer in GCN to generate the $\mathbf{H}_i$ as follows:

$$\mathbf{H}_i = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{H}}_{i-1}\mathbf{B}_{i-1}) \tag{16}$$

where $\sigma(\cdot)$ denotes Softmax function when last layer, otherwise, denote ReLU function.

Notice, the input of the first layer GCN is described as:

$$\mathbf{H}_1 = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{A}\mathbf{B}_1) \tag{17}$$

**Reconstruct networks**. Limited by the linearity of DNMF, community membership cannot further improve its representation ability. That leads to the inability to improve the accuracy of community detection. Therefore, we make up the flaw of DNMF through $\mathbf{H}_\ell$. Specifically, the nonlinearity of GCN makes $\mathbf{H}_\ell$ contain more informative. Thus, the weighted network reconstructed by GCN is beneficial for DNMF to make up its the flaw of linearity and capture highly-accurate community membership.

The reconstruction of networks can be categorized based on whether community background is taken into consideration. When focusing on the situation that ignores the community background, we choose to add or remove some associations between nodes, regardless of whether they co-locate in the same community or not. The sole criterion for this operation is the level of similarity between nodes. Specifically, we add extra links to nodes with high similarity, while nodes with low similarity are not handled, effectively equivalent to removing links between them as DNMF filters out node associations with low weights. It is well-known that similar nodes tend to cluster together effortlessly in real networks, while dissimilar nodes are more dispersed. Therefore, removing and adding specific node associations can help DNMF withstand topology noise. The rule is as follows:

$$\mathbf{S} = \left(\frac{\mathbf{H}_\ell\mathbf{H}_\ell^T}{\|\mathbf{H}_\ell\|\|\mathbf{H}_\ell^T\|} - \mathbf{I}\right) \odot \mathbf{X} \tag{18}$$

$$\mathbf{X} \leftarrow \begin{cases} 0 + \mathbf{X}, \mathbf{S} < \dfrac{\sum\sum \mathbf{S}}{\sum\sum \mathbf{X}} \\ 1 + \mathbf{X}, otherwise \end{cases} \tag{19}$$

Similar to the aforementioned idea of adding or removing associations of nodes, we perform a like operation in think of the community background. The reason we do this is based on the high similarity of nodes within a community and a low similarity of nodes between communities. The rule as follows:

$$\mathbf{S} = \mathbf{U}\mathbf{U}^T \odot \left(\frac{\mathbf{H}_\ell\mathbf{H}_\ell^T}{\|\mathbf{H}_\ell\|\|\mathbf{H}_\ell^T\|} - \mathbf{I}\right) \odot \mathbf{X} \tag{20}$$

$$\mathbf{X} \leftarrow \begin{cases} 0 + \mathbf{X}, \mathbf{S} < \dfrac{\sum\sum \mathbf{S}}{\sum\sum \mathbf{U}} \\ 1 + \mathbf{X}, otherwise \end{cases} \tag{21}$$

where $\mathbf{U}$ is community affiliation matrix of nodes based on $\mathbf{H}_\ell$. The $\mathbf{U}_{jd}$ is 1 means the node $j$ affiliation community $d$, otherwise, 0.

### 3.5. Unified loss function

Traditional GCN applied to the supervised learning scenario, while cannot be directly applied to community detection in the unsupervised scenario. Here, we propose a triple loss module, which unified the DNMF and GCN in a framework and effectively trains the GCN end-to-end for node embedding.

Modularity maximization proposed by Newman (2006) which collocations other methods has been widely used in community detection. Formally, the modularity $Q$ is defined as follows:

$$Q = \frac{1}{4t}\sum_{m,n}(\mathbf{A}_{mn} - \frac{d_m d_n}{2t})(\mathbf{H}_m\mathbf{H}_n^T) \tag{22}$$

where $d_m$ is degree of node $m$ and $t$ is number of edges. Further, we can derive the trace form of $Q$ via defining the modularity matrix $\mathbf{B}$, as follows:

$$\begin{aligned} Q &= \frac{1}{4t}\sum_{m,n}(\mathbf{B}_{mn}\sum_p H_{mp}H_{jp}) \\ &= \frac{1}{4t}\sum_p\sum_{m,n}\mathbf{B}_{mn}\mathbf{H}_{mp}\mathbf{H}_{np} \\ &= \frac{1}{4t}\sum_p[\mathbf{H}^T\mathbf{B}\mathbf{H}]_{pp} \\ &= \frac{1}{4t}tr(\mathbf{H}^T\mathbf{B}\mathbf{H}) \end{aligned} \tag{23}$$

where $\mathbf{B}_{mn} = \mathbf{A}_{mn} - \frac{d_m d_n}{2m}$ and $\mathbf{H} = \mathbf{H}_\ell$.

Notice that the $\frac{1}{4t}$ has no effect on the maximum of $Q$. Therefore, the loss function of modularity maximization $\mathcal{L}_Q$ can be written as:

$$\mathcal{L}_Q = tr(\mathbf{H}_\ell^T\mathbf{B}\mathbf{H}_\ell) \tag{24}$$

Convolution operation like a filter that filters topology noise. Since $\mathbf{H}_\ell$ can well preserve the proximity between neighbor nodes for the encoder of GCN. Therefore, a natural idea is restore $\mathcal{G}$ via $\mathbf{H}_\ell$. Specifically, we pursue the restore adjacency matrix $\tilde{\mathbf{A}}$ to approximate the $\mathbf{A}$ as possible. Every element in $\tilde{\mathbf{A}}$ can be treated as probability that is connected or disconnected each node, i.e., 1 or 0. The construct of $\tilde{\mathbf{A}}$ as follows:

$$p(\tilde{\mathbf{A}} = 1) = \sigma(\mathbf{H}_\ell\mathbf{H}_\ell^T) \tag{25}$$

$$p(\tilde{\mathbf{A}} = 0) = 1 - \sigma(\mathbf{H}_\ell\mathbf{H}_\ell^T) \tag{26}$$

where $\sigma(\cdot)$ is sigmoid function.

Then, we employ the binary cross entropy function to measure $\tilde{\mathbf{A}}$ and $\mathbf{A}$. The loss function $\mathcal{L}_B$ can be write as:

$$\mathcal{L}_B = -\frac{1}{N^2}\sum\sum \mathbf{A}logp(\tilde{\mathbf{A}} = 1) + (1-\mathbf{A})logp(\tilde{\mathbf{A}} = 0) \tag{27}$$

Although $\mathcal{L}_Q$ and $\mathcal{L}_B$ guide the training of GCN together, GCN still fails to adapt in the unsupervised scenario. This tricky question can be solve by adding extra the loss function containing DNMF. DNMF has powerful representation that make node membership have high confidence. And we believe that the convergent DNMF can guide the training of GCN toward best performance. Hence, we choose to minimize the Kullback–Leibler divergence to complete the guidance, as follows:

$$\mathcal{L}_K = KL(\mathbf{H}_\ell \parallel \mathbf{V}_\ell) = \sum p(\mathbf{H}_\ell)log\frac{p(\mathbf{H}_\ell)}{p(\mathbf{V}_\ell)} \tag{28}$$

where $p(\cdot)$ means probability function.

In all, we combine the $\mathcal{L}_Q$, $\mathcal{L}_B$ and $\mathcal{L}_K$ as uniform loss to training GCN together. That not only enabled GCN to adapt to the unsupervised scenario, but also further parallel connected GCN and DNMF seamlessly. The uniform loss function as follows:

$$\mathcal{L} = \mathcal{L}_B - \alpha\mathcal{L}_Q + \mathcal{L}_K \tag{29}$$

where $\alpha$ is hyperparameter for balancing the contributions of the corresponding loss function. The term $\mathcal{L}_Q$ preserves modularity by aligning detected communities with structural clusters within the network, while $\mathcal{L}_B$ maintains consistency with the adjacency matrix, thereby reducing noise and enhancing robustness. Finally, $\mathcal{L}_K$ integrates the embeddings of DNMF and GCN, guiding the GCN to incorporate community-centric information derived from DNMF. This integrated framework allows SADNG to achieve both interpretability through the hierarchical community mappings of DNMF and accuracy through the non-linear feature extraction of GCN in community detection tasks.

### 3.6. Time complexity

The complete procedure of SADNF is outlined in Algorithm 1, where main computational cost is on the encoder module, the decoder module and computing the loss value. The computational cost for the DNMF encoder module is of order $O(T_{max}T_1\ell(n^2k+k^2n))$, and standard NMF iterates $T_1$ times until convergence. The computational cost for the GCN encoder module is of order $O(T_{max}\ell(|\mathbf{A}|n+n^3))$ for $T_{max}$ iterations to converge, where $|\mathbf{A}|$ is number of nonzeros in the adjacency matrix. The computational cost for the decoder module is of order $O(T_{max}\ell(n^2k+k^2n+k^3))$ for $T_{max}$ iterations to converge. For all loss functions, the total computational cost is of order $O(T_{max}(n^2k+k^2n))$ In general, $k \ll n$, thus theoretically the total time complexity is $O(T_{max}\ell(T_1n^2k+n^3))$. In fact, because many matrices are sparse, the total computational cost is probably lower than the theoretical value. Meanwhile, SADNG can converge in a short time, thus keeping the time overhead cost of the method within an acceptable range.

**Algorithm 1:** SADNG

**Input:** Graph $\mathcal{G}$, Number of layers $\ell$, Number of communities $k$, Maximum iterations $T_{max}$;
**Output:** Community membership $\mathbf{V}_\ell$;

Initialize $\mathbf{X} \leftarrow \mathbf{A}$;
Initialize $\mathbf{B}_i$ randomly;
Initialize the attribute matrix of GCN needed via $\mathbf{I}$;
**while** *iter* $\in 0, 1, \cdots, T_{max}$ **do**
  ▶**Training DNMF encoder**:
  **for** $i \in 1, 2, \cdots, \ell$ **do**
    $\mathbf{W}_i, \mathbf{V}_i \leftarrow$ Standard NMF with hierarchical relations, i.e., Eq. (2);
  ▶**Training GCN encoder**:
  **for** $i \in 1, 2, \cdots, \ell$ **do**
    **if** $i = 1$ **then**
      Generate the first GCN layer node embedding $\mathbf{H}_1 = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\mathbf{A}\mathbf{B}_1)$;
    **else**
      Generate the next GCN layer node embedding $\tilde{\mathbf{H}}_{i-1} = (1-\epsilon)\mathbf{H}_{i-1} + \epsilon\mathbf{V}_{i-1}$; $\mathbf{H}_i = \sigma(\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{A}}\tilde{\mathbf{D}}^{-\frac{1}{2}}\tilde{\mathbf{H}}_{i-1}\mathbf{B}_{i-1})$;
  ▶**Training DNMF decoder**:
  **while** *not converged* **do**
    **for** $i \in 1, 2, \cdots, \ell$ **do**
      Update $\mathbf{W}_i$ via Eq. (10) and Eq. (11);
      Update $\mathbf{V}_i$ via Eq. (13);
  Calculate $\mathcal{L}_Q$, $\mathcal{L}_B$ and $\mathcal{L}_K$ via Eq. (24), Eq. (27) and Eq. (28) respectively;
  ▶**Reconstructing networks**:
  Update $\mathbf{X}$ via Eq. (19) and Eq. (21) respectively;
**return** $\mathbf{V}_\ell$.

## 4. Experiments

In this section, we will conduct community detection experiments on real topology networks. First of all, we will introduce some benchmarks, metrics, and setting for this experiment. And then, we will analyze the experiment results in detail to show that SADNG has better performance. Finally, we will analyze the uniqueness and excellence of SADNG from two perspectives, i.e., reconstruction threshold and different propagation layers.

### 4.1. Datasets

Our SADNG is evaluated on eight datasets. And the detailed descriptions of these datasets are the followings:

**Table 2**
Layers configuration of SADNG and other baselines.

| Datasets | Layers configuration |
|---|---|
| Cora | 128-64-7 |
| Citeseer | 128-64-32-6 |
| Wisconsin | 32-16-5 |
| Washington | 32-16-5 |
| Cornell | 32-16-5 |
| Texas | 32-16-5 |
| Email | 128-64-42 |
| Wiki | 128-64-32-17 |

- **Cora**: A citation network involving 2708 scientific publications classified into one of 7 classes and 7835 links.
- **Citeseer**: A citation network consisting of 3312 nodes classified into one of 6 classes and with 7793 links.
- **Wisconsin**: A web page association network with 260 pages and 673 links. Each node is divided into one of 5 communities.
- **Washington**: A web page association network with 230 pages and 567 links. Each node is divided into one of 5 communities.
- **Cornell**: A web page association network with 195 pages and 460 links. Each node is divided into one of 5 communities.
- **Texas**: A web page association network with 187 pages and 437 links. Each node is divided into one of 5 communities.
- **Email**: A communication network contains 1005 researchers and 8204 relationships from them. These researchers came from 42 departments.
- **Wiki**: A document network with 2405 pages from 17 categories and 10173 edges.

Note that these datasets all contain self-loops

### 4.2. Baselines

We compare SADNG with seven state-of-the-art methods as baselines. These baselines mainly involve NMF-based methods and DNMF-based methods.

- **DANMF** (Ye et al., 2018): DANMF is the first algorithm that introduce deep NMF to community detection. The method learns the hierarchical mappings which enables it to get a more accurate community membership.
- **NMF** (Mankad & Michailidis, 2013): NMF is the fundamental method for other NMF-based derived methods. And its purpose is to minimize $\|\mathbf{A}-\mathbf{W}\mathbf{V}\|_F^2$.
- Sun et al. (2017): Sun et al. proposed integrating decoder and encoder into a unified framework in NMF background and achieved the sparsity of community membership because of its symmetry. This method aims to minimize $\|\mathbf{A}-\mathbf{W}\mathbf{V}\|_F^2 + \|\mathbf{V}-\mathbf{W}^T\mathbf{A}\|_F^2$.
- **DNMF**: The DNMF retains the decoder component of DANMF, and ignores the encoder component of it.
- **PNMF** (Yuan & Oja, 2005): PNMF directly minimizes $\|\mathbf{A}-\mathbf{V}\mathbf{V}^T\mathbf{A}\|_F^2$ to map the original networks to a part-based subspace.
- **MDNMF** (Huang et al., 2021): MDNMF fuses DNMF and modularity to preserves both the topology information and properties of community structure.
- **MNMF** (Wang et al., 2017): MNMF is a variant of NMF by additional modularity.
- **VGAE** (Kipf & Welling, 2016): VGAE is based on variational Bayes theory, and takes the lead in extending VAE to graph domain via GCN.
- **GCC** (Fettal et al., 2022): GCC is founded on a variant of simple graph convolutional networks, which perform clustering by minimizing the dissimilarity between convolutional node representations and their respective reconstructed cluster representations.

- **CDBNE** (Zhou et al., 2023): CDBNE is an unsupervised attributed network embedding method that considers cluster-oriented information for community detection.
- **CVGAE** (Mrabah et al., 2024): CVGAE, a clustering method based on VGAE, addresses the issues of feature randomness and feature drift.
- **CDRS** (Zhu et al., 2023): CDRS incorporates supervision information, which guides the representation learning of nodes, to enhance community detection performance.

**Metrics**. We employ three popular metrics to evaluate SADNG and baselines. These metrics are Accuracy (AC), Adjusted Rand Index (ARI) and Normalized Mutual Information (NMI). For these metrics, a larger value indicates better performance.

**Parameter setting**. For methods requiring layer settings, we provide the layer size configuration in Table 2, based on empirical practices and considering the dataset size. For larger datasets, such as Cora and Citeseer, higher initial dimensions (e.g., 128–64) are set to effectively capture the complexity and density of these networks, gradually reducing to the final community count levels. Conversely, smaller datasets like Texas and Wisconsin use lower initial dimensions (e.g., 32–16). We train SADNG and baselines using all data points with 100 epochs and the learning rate is $10^{-2}$ if needed. For all baseline methods, **if the input requires node attributes matrix, we set it to the identity matrix**, while other configurations adopt the ones described in their respective papers. As for the representation learning method VGAE, we obtain community detection results in an end-to-end manner. In SADNG, the $\alpha$, we set it to $10^{-4}$. The $\epsilon$ is an important parameter for SADNG due to it can alleviate the over-smoothing problem of GCN. Therefore, we set $\epsilon$ by answering the following questions. Firstly, *Why over-smoothing occurs?* Secondly, *What role does DNMF play in alleviating over-smoothing?* And finally, *What is the effect of $\epsilon$ on over-smoothing problem?* In simple terms, GCN possesses great potential because the inherent convolution operations and each node is able to learn unknown information from neighbor nodes. However, nodes are too similar when they learn too much unknown information. This is an over-smoothing problem and it emerges gradually when convolution layers of GCN are more than two. DNMF has a powerful representation to capture the hierarchical mapping for each layer, and we think this hierarchical mapping is credible and highly-accurate. So we use hierarchical mapping to alleviate over-smoothing problem. In general, DNMF does not need to be overly involved in the convolution operation of GCN when convolution layers are less than two. Therefore, $\epsilon$ of Eq. (15) can be less than 0.5. But DNMF needs to dominate gradually in the convolution operation when convolution layers are more than two. In all, we set $\epsilon$ to 0.3 for the first fusion between $\mathbf{V}_i$ and $\mathbf{H}_i$, otherwise, 0.9.

### 4.3. Analysis of community detection results

Table 3 shows the community detection results on eight datasets. Based on the experimental results, we will analyze the outperformance of SADNG. Furthermore, we demonstrate that the seamless way to combine NMF and DNMF is reasonable through ablation experiments.

For each metric, SADNG mostly outperforms other baselines across seven datasets. It is well known that the WebKb datasets, containing Wisconsin, Washington, Cornell, and Texas, are dense networks with hub nodes, leading to the creation of many connected components when hub nodes are removed. This results in topology noise in these networks, and many baseline methods struggle to handle such noise due to the lack of an explicit community structure in the networks. However, SADNG achieves the best results in these datasets by leveraging the non-linear features of GCN to reconstruct networks. We employ a reconstruction operation on the original adjacency matrix to counteract the influence of hub nodes effectively. The results show that SADNG performs well on Washington, Wisconsin, Cornell and Texas for AC, ARI and NMI. Particularly, it achieves the largest relative improvements of 52.9%, 35.6%, and 24.2% on Texas, and improves by at least 8.1%, 15.6% and 1.9% over contemporaneous methods. These results confirm that SADNG performs better in handling topology noise by using the nonlinearity of GCN to reconstruct networks.

On the remaining four datasets, Cora, Citeseer, Email and Wiki, SADNG outperforms other baseline methods in most cases, except for some metrics in Email and Wiki where it achieves the runner-up or shows slight disadvantages. Notably, for AC, ARI and NMI, SADNG achieves maximum relative improvements of 36.5% 46.7% and 49.5%, and certain boosts of 9%, 8% and 2.7% over contemporaneous methods. The above experimental data demonstrates the effectiveness of SADNG.

Ablation experiments further show that the way to seamlessly connect GCN and DNMF is correct. Compared with DANMF, SADNG gives up the graph regularizer in it and is connected with GCN. Thus, DANMF and SADNG can form an ablation relationship. On Cora, Citeseer, Email, and Wiki, SADNG outperforms DANMF slightly in each metric. To be specific, SADNG get a maximum increase by 8% (in Cora), 21% (in Citeseer), and 20% (in Citeseer) on AC, ARI, and NMI respectively. And it achieved somewhat increase in Wiki, i.e., 2.6%, 1.2%, and 2.7% on AC, ARI, and NMI respectively. In WebKb, SADNG achieves the greatest performance improvement in Texas, i.e., 45%, 33% and 24% on AC, ARI, and NMI respectively, and slightly improves by 13%, 13% and 7% in Wisconsin respectively. There are mainly the following reasons for outperformance of SADNG. Firstly, like most NMF-based methods, DANMF has intrinsic linearity, which makes it hard to reconstruct complex networks and withstand topology noise, e.g., in WebKb datasets. Therefore, we come up with the idea to reconstructing networks by using non-linear features of GCN. In this way, we ingeniously bypasses the flaw of DNMF. By using the weighted networks, we mitigate the impact of topology noise and accurately capture hierarchical mappings. On the other hand, the powerful representation of DNMF provides a guarantee for the performance of GCN. We know that the earliest GCN only adapts in the supervised scenario and has an over-smoothing problem when the convolution layers are more than two, which makes it hard to suit the unsupervised scenario and further improve its performance. However, with the intervention of DNMF, GCN not only adapts the unsupervised scenario but also alleviates the over-smoothing problem. In addition, the performance of DNMF is improved by using GCN.

In summary, SADNG provides a simple yet effective approach by seamlessly integrating GCN with DNMF, enabling each method to compensate for the limitations of the other. This collaborative design allows SADNG to perform exceptionally well, especially on structurally complex datasets such as Washington and Wisconsin, where the nonlinear transformation capabilities of GCN help address topological inconsistencies. Additionally, on datasets like Cora, Citeseer and Wiki, SADNG demonstrates strong performance in metrics such as AC, ARI and NMI, highlighting its ability to detect communities in complex network environments.

### 4.4. Convergence analysis

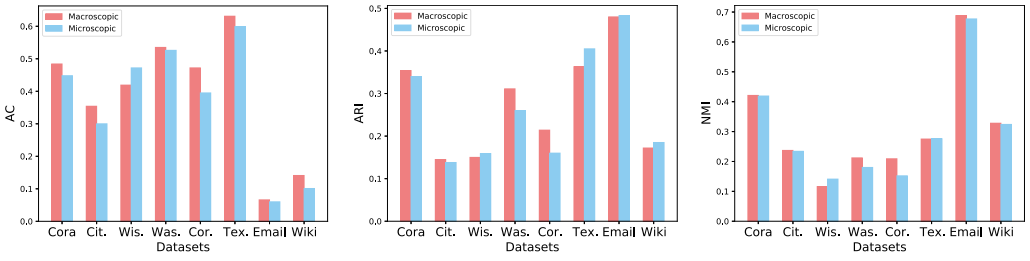
The convergence of the updating rules is guaranteed by the following two theorems.

**Theorem 4.1.** *The updating rules (DNMF component) satisfy the Karush–Kuhn–Tucker (KKT) optimality condition.*

**Proof 4.1.** Either Eq. (10) or Eq. (13) is result that to solve the partial derivative of corresponding variable, i.e., Eq. (9) and (12). And let partial derivative be zero. Finally, we simplified the calculation with Oja. Clearly, the process satisfies the KKT optimality condition.

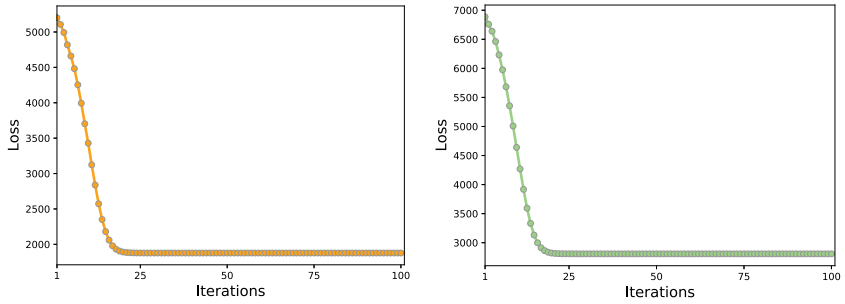
**Table 3**
Performance evaluation based on AC, ARI and NMI. The bold numbers represent the best results and the numbers with asterisk are runner-up.

| Dataset | Metric | DANMF | NMF | Sun et al. | DNMF | PNMF | MDNMF | MNMF | VGAE | GCC | CDBNE | CVGAE | CDRS | SADNG |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| Cora | AC | 0.401 | 0.130 | 0.319 | 0.294 | 0.119 | 0.302 | 0.322 | 0.424 | 0.395 | 0.475 | 0.448 | 0.441 | **0.484** |
| | ARI | 0.274 | 0.225 | 0.056 | 0.112 | 0.068 | 0.027 | 0.079 | 0.181 | 0.187 | 0.209 | 0.267 | 0.266 | **0.354** |
| | NMI | 0.363 | 0.335 | 0.154 | 0.177 | 0.224 | 0.094 | 0.237 | 0.270 | 0.344 | 0.361 | 0.324 | 0.318 | **0.421** |
| Citeseer | AC | 0.296 | 0.100 | 0.191 | 0.183 | 0.202 | 0.198 | 0.173 | 0.287 | 0.338 | 0.309 | 0.238 | 0.249 | **0.361** |
| | ARI | 0.104 | 0.079 | 0.010 | 0.067 | 0.015 | 0.001 | 0.011 | 0.098 | 0.220 | 0.223 | 0.201 | 0.271 | **0.320** |
| | NMI | 0.173 | 0.141 | 0.084 | 0.057 | 0.093 | 0.036 | 0.115 | 0.115 | 0.231 | 0.230 | 0.223 | 0.241 | **0.378** |
| Wisconsin | AC | 0.283 | 0.117 | 0.208 | 0.166 | 0.166 | 0.355 | 0.189 | 0.219 | **0.469** | 0.359 | 0.341 | 0.339 | 0.419* |
| | ARI | 0.013 | 0.067 | 0.071 | 0.054 | 0.087 | 0.049 | 0.037 | 0.042 | 0.015 | 0.037 | 0.042 | 0.050 | **0.150** |
| | NMI | 0.043 | 0.109 | 0.078 | 0.081 | 0.060 | 0.070 | 0.080 | 0.061 | 0.043 | 0.066 | 0.070 | 0.072 | **0.116** |
| Washington | AC | 0.143 | 0.104 | 0.200 | 0.161 | 0.117 | 0.261 | 0.217 | 0.117 | 0.476 | 0.430 | 0.426 | 0.413 | **0.535** |
| | ARI | 0.077 | 0.153 | 0.027 | 0.087 | 0.182 | 0.116 | 0.039 | 0.135 | 0.215 | 0.293 | 0.214 | 0.212 | **0.311** |
| | NMI | 0.079 | 0.090 | 0.081 | 0.108 | 0.107 | 0.151 | 0.059 | 0.155 | 0.147 | 0.177 | 0.123 | 0.093 | **0.212** |
| Cornell | AC | 0.082 | 0.215 | 0.426 | 0.446 | 0.344 | 0.426 | 0.374 | 0.174 | 0.449 | 0.451 | 0.462 | 0.423 | **0.472** |
| | ARI | 0.071 | 0.006 | 0.073 | 0.067 | 0.040 | 0.008 | 0.042 | 0.053 | 0.132 | 0.178 | 0.195 | 0.182 | **0.214** |
| | NMI | 0.078 | 0.073 | 0.117 | 0.100 | 0.071 | 0.039 | 0.075 | 0.105 | 0.170 | 0.200 | 0.201 | 0.198 | **0.209** |
| Texas | AC | 0.181 | 0.102 | 0.396 | 0.193 | 0.123 | 0.182 | 0.235 | 0.133 | 0.550 | 0.412 | 0.423 | 0.429 | **0.631** |
| | ARI | 0.029 | 0.221 | 0.104 | 0.070 | 0.251 | 0.007 | 0.161 | 0.139 | 0.207 | 0.193 | 0.172 | 0.235 | **0.363** |
| | NMI | 0.033 | 0.122 | 0.169 | 0.047 | 0.135 | 0.058 | 0.086 | 0.124 | 0.256 | 0.174 | 0.189 | 0.181 | **0.275** |
| Email | AC | 0.015 | 0.004 | 0.050 | 0.091 | 0.033 | 0.049 | 0.022 | 0.082 | **0.124** | 0.020 | 0.019 | 0.022 | 0.066 |
| | ARI | 0.294 | 0.286 | 0.034 | 0.149 | **0.521** | 0.013 | 0.401 | 0.179 | 0.201 | 0.295 | 0.175 | 0.194 | 0.480* |
| | NMI | 0.543 | 0.492 | 0.278 | 0.401 | **0.698** | 0.193 | 0.675 | 0.309 | 0.481 | 0.578 | 0.533 | 0.597 | 0.688* |
| Wiki | AC | 0.126 | 0.044 | 0.080 | 0.074 | 0.115 | 0.068 | 0.072 | 0.019 | **0.175** | 0.061 | 0.072 | 0.066 | 0.152* |
| | ARI | 0.168 | 0.077 | 0.025 | 0.084 | 0.122 | 0.012 | 0.064 | 0.004 | 0.071 | 0.155 | 0.050 | 0.076 | **0.180** |
| | NMI | 0.308 | 0.159 | 0.108 | 0.206 | 0.290 | 0.052 | 0.249 | 0.219 | 0.224 | 0.240 | 0.220 | 0.241 | **0.335** |




(a) AC (b) ARI (c) NMI

**Fig. 2.** Threshold analysis. Cit., Wis., Cor. and Tex. are abbreviations for Citeseer, Wisconsin, Cornell and Texas respectively.




(a) Cora (b) Citeseer

**Fig. 3.** Convergence analysis.

**Theorem 4.2.** *The uniform loss function is nonincreasing under the convolution operation of GCN and the operation of DNMF.*

**Proof 4.2.** Fig. 3 plots the loss trend of SADNG and Fig. 3(a) and Fig. 3(b) reveal Cora and Citeseer loss trend respective. Either Cora or Citeseer has obvious declining trends of loss when iterations are less than 25. And finally, the loss trend is nearly not changing. Obviously, we find SADNG satisfies the nonincreasing of loss with empirical observation

### 4.5. Analysis of reconstruction threshold

The thresholds of reconstruct networks are proposed based on what nodes should be in the community. Obviously, this threshold is obtained from a macro perspective. A natural question is: *Whether microscopic thresholds will affect the performance of SADNG?* In order to explore this question, we propose microscopic thresholds. These thresholds are proposed based on which community the node should belong to.

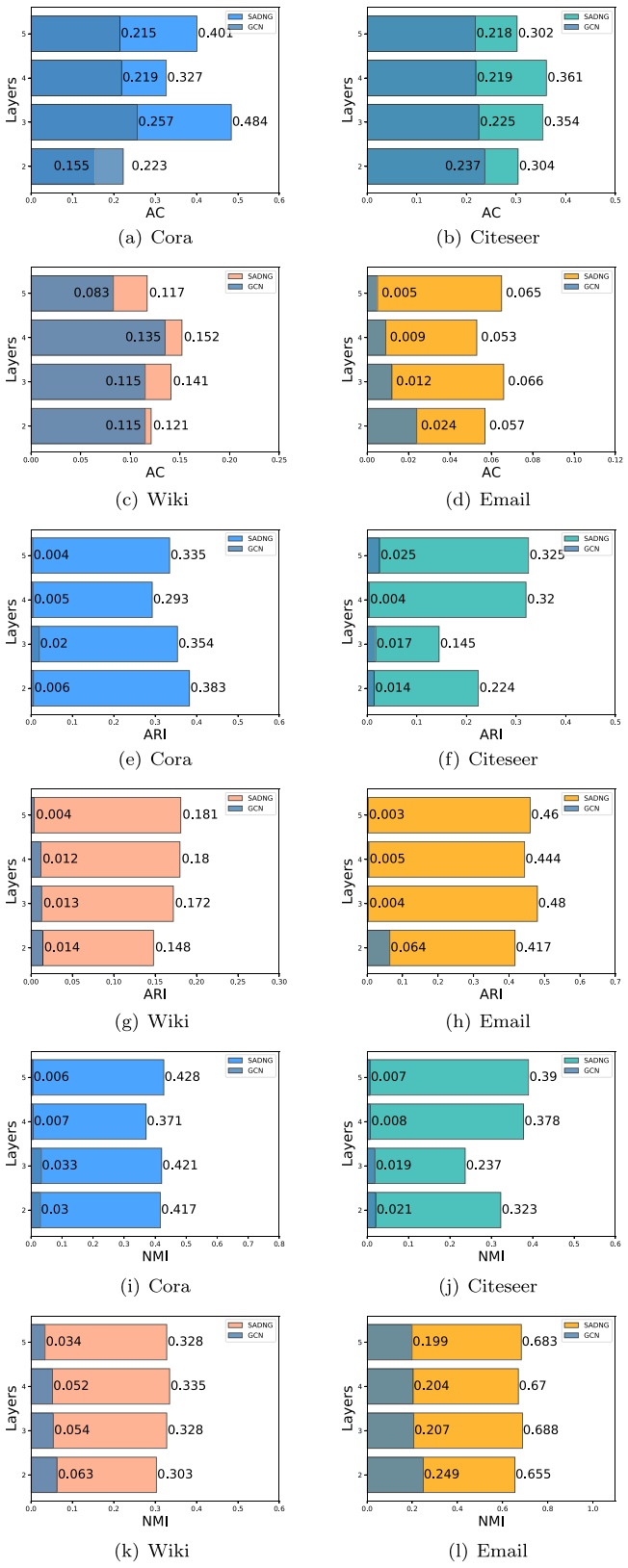



**Fig. 4.** Propagation layers analysis.

Similar to the aforementioned analysis, we consider whether take community background into consideration or not. If we ignore the community background, the thresholds are as follows:

$$\mathbf{E} = \begin{cases} 0, \mathbf{S}_m \leq \dfrac{\sum_m \mathbf{S}}{\sum_m \mathbf{X}} \\ 1, otherwise \end{cases} \tag{30}$$

where $\mathbf{S}$ is obtained from Eq. (18).

$$\mathbf{X} \leftarrow \begin{cases} 1 + \mathbf{X}, \mathbf{E} + \mathbf{E}^T \geq 1 \\ 0 + \mathbf{X}, otherwise \end{cases} \tag{31}$$

If we consider the community background as following:

$$\mathbf{E} = \begin{cases} 0, \mathbf{S}_m \leq \dfrac{\sum_m \mathbf{S}}{\sum_m \mathbf{X} \odot \mathbf{U}\mathbf{U}^T} \\ 1, otherwise \end{cases} \tag{32}$$

where $\mathbf{S}$ is obtained from Eq. (20). Notice that the diagonal element of $\mathbf{X}$ is zero when calculating Eq. (32) and the update rule of $\mathbf{X}$ is the same as the Eq. (31) to guarantee symmetry of each node.

Fig. 2 reveals the effect of macroscopic and microscopic thresholds on SADNG. We conclude firstly that macroscopic thresholds are better than microscopic because they can be resistant to severe topology noise. Recall the previous analysis, WebKb datasets have explicit topology noise because there are hub nodes. And Figs. 2(a)–2(c) indicate that SADNG with macroscopic threshold performs better, generally speaking. The main reason is that we consider the affiliation of nodes from the perspective of the community. Specifically, either Eq. (19) or Eq. (21) is thresholds for directly calculating the network not for each node (e.g. Eq. (30)). This will make it difficult for specific nodes' noise to affect the macroscopic thresholds. So it improves the capability of SADNG to withstand topology noise. In other datasets, we can find that the macroscopic thresholds have a slight advantage over the microscopic. In the end, we chose macroscopic thresholds because they can help SADNG to withstand topology noise and achieve prominent performance.

### 4.6. Analysis of different propagation layers

We conduct some experiments on Cora, Citeseer, Wiki, and Email datasets to reveal the effect of the number of convolution layers on SADNG. We know from past experiments, over-smoothing will be a serious problem as the number of convolution layers increases. If we cannot alleviate this situation, it will greatly reduce the performance of GCN, and it will further affect the performance of DNMF by reconstructing untrusted networks. As a result, the performance of SADNG will perhaps decrease. In the configurations, we set the number of convolution layers from 2 to 5, and set $\epsilon$ to 0.3 when the number of convolution layers is 2, and to 0.9 otherwise. At the same time, we plot the performance change of GCN at different layers to observe the trend of performance change. In this paper, we focus on topology networks without any node attributes and approach community detection in an end-to-end manner using an unsupervised approach. In order to match our work objectives, we choose to retain part of the loss functions, i.e., $\mathcal{L}_Q$ and $\mathcal{L}_B$, to drive the training of GCN without the need for well-designed new loss functions to guide the training. Furthermore, we employ real attributes of nodes and k-means for community detection on GCN. The analysis data is shown in Fig. 4.

We can observe that, in most cases, the performance of GCN has a gradual decline trend as the number of convolutional layers increases. Surprisingly, the performance change trend of SADNG is significantly different from that of GCN. We find that as the convolution layers increases, the performance of SADNG does not decrease. In addition, in some datasets, the performance of SADNG are the best in some metrics and still has an improving tendency, for example in Email dataset. From Figs. 4(g) and 4(k), we know that the performance of SADNG increases as the layer of convolution increases. In addition, in Figs. 4(f), 4(i) and 4(j), the performance of five-layer convolution is even better

**Table 4**
Analysis of $\mathcal{L}_{de}$ and $\mathcal{L}_K$. (SADNG-$\mathcal{L}_{de}$: SADNG without loss function $\mathcal{L}_{de}$. SADNG−$\mathcal{L}_K$: SADNG without loss function $\mathcal{L}_K$.).

| | Metric | SADNG | SADNG-$\mathcal{L}_{de}$ | SADNG−$\mathcal{L}_K$ |
|---|---|---|---|---|
| Cora | AC | **0.484** | 0.338 | 0.459 |
| | ARI | **0.354** | 0.201 | 0.309 |
| | NMI | **0.421** | 0.377 | 0.397 |
| Citeseer | AC | **0.361** | 0.23 | 0.325 |
| | ARI | **0.32** | 0.164 | 0.267 |
| | NMI | **0.378** | 0.246 | 0.347 |
| Wisconsin | AC | **0.419** | 0.326 | 0.357 |
| | ARI | **0.15** | 0.083 | 0.128 |
| | NMI | **0.116** | 0.057 | 0.084 |
| Washington | AC | **0.535** | 0.378 | 0.448 |
| | ARI | **0.311** | 0.154 | 0.207 |
| | NMI | **0.212** | 0.144 | 0.191 |
| Cornell | AC | **0.472** | 0.321 | 0.426 |
| | ARI | **0.214** | 0.059 | 0.194 |
| | NMI | 0.209 | 0.193 | **0.228** |
| Texas | AC | **0.631** | 0.486 | 0.511 |
| | ARI | **0.363** | 0.141 | 0.264 |
| | NMI | **0.275** | 0.198 | 0.257 |
| Email | AC | **0.066** | 0.023 | 0.038 |
| | ARI | **0.48** | 0.316 | 0.394 |
| | NMI | **0.688** | 0.612 | **0.668** |
| Wiki | AC | **0.152** | 0.059 | 0.105 |
| | ARI | **0.18** | 0.096 | 0.101 |
| | NMI | **0.335** | 0.275 | 0.325 |

than the others. The main reason is the cooperation of DNMF and GCN. As we mentioned before, the DNMF has powerful representation and can alleviate the over-smoothing problem of GCN. At the same time, the non-linearity of GCN makes the reconstructed networks more flexible and further promotes the performance of DNMF. This perfect closed-loop cooperation promises SADNG a remarkable performance.

Additionally, we find that adjusting the balance coefficient $\epsilon$ based on the depth of the GCN layers allows SADNG to amplify the impact of DNMF in deeper layers. Specifically, when the number of layers is small, a lower $\epsilon$ (e.g., 0.3) is applied to reduce the risk of over-smoothing. This approach helps retain high-resolution neighborhood information. As the GCN layer depth increases, a higher $\epsilon$ (e.g., up to 0.9) is used. This enables DNMF to actively contribute to the convolution process, effectively mitigating excessive homogenization of node embeddings while preserving essential local information. Experimental results confirm that this adaptive integration provides robustness against over-smoothing. It maintains neighborhood relevance in community detection tasks, thereby improving the overall performance of SADNG.

### 4.7. Analysis of $\mathcal{L}_{de}$ and $\mathcal{L}_K$

We compare SADNG with its two variants on all benchmark datasets. From the results in Table 4, we can draw the following conclusions:

(1) SADNG performs significantly better than SADNG-$\mathcal{L}_{de}$ on all benchmark datasets. Among them, for NMI, AC and ARI, SADNG achieves maximum improvements of 13.2%, 15.7% and 22.2% on Citeseer, Washington and Texas respectively. Obviously, these results show that $\mathcal{L}_{de}$ plays a vital role in SADNG, which helps our method extract features from the data and remove background noise. Therefore, the experimental results show that the decoding module is indispensable in our method.

(2) Comparing the result of $\mathcal{L}_K$ and SADNG, for NMI, AC and ARI, SADNG achieves average boosts of 5.6%, 6.4% and 1.7%. Although the node representation H is a linear combination of the respective representations from the GCN module and the DNMF module, it is undeniable that $\mathcal{L}_K$ has a positive impact on our method. As an unsupervised method, DNMF obtains a node representation $\mathbf{V}_\ell$ that contains community information with high confidence. Although $\mathbf{H}_\ell$ partially contains the information from $\mathbf{V}_\ell$, the training process of GCN cannot effectively guide $\mathbf{H}_\ell$ to incorporate more community information. Therefore, we need to adjust the training objective of GCN by emphasizing the distribution distance between $\mathbf{H}_\ell$ and $\mathbf{V}_\ell$.

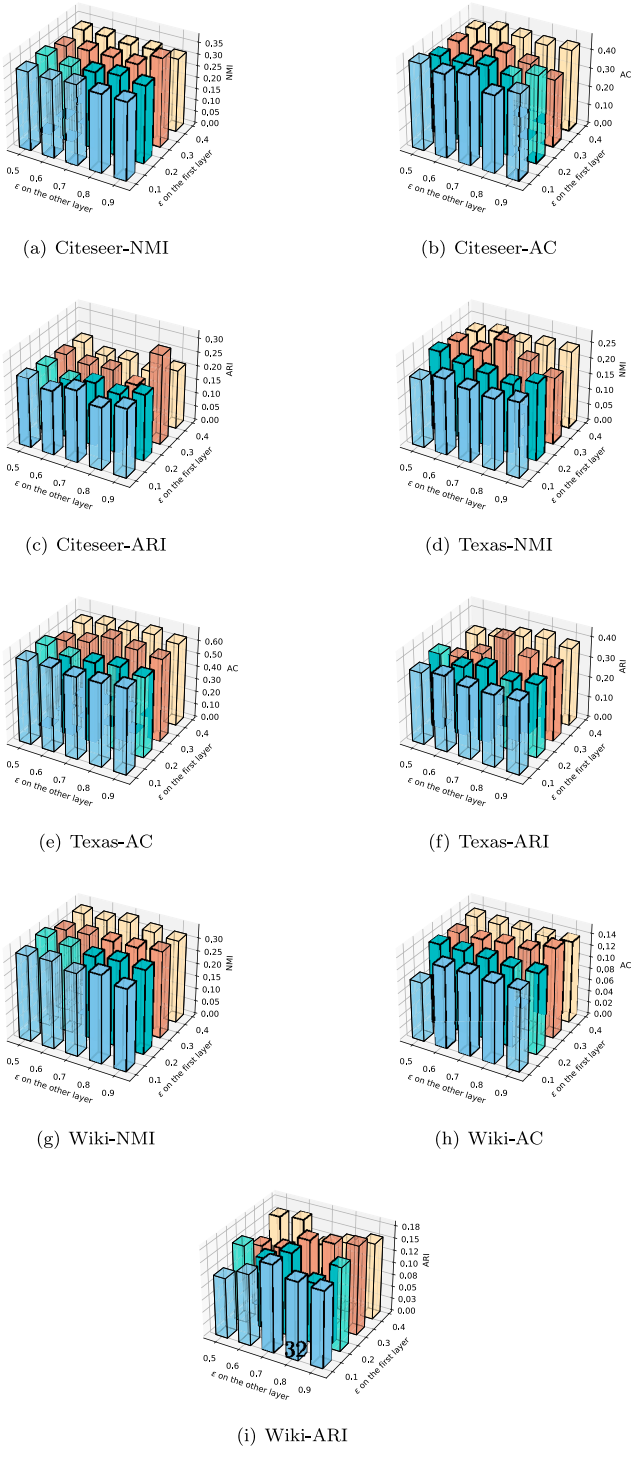



**Fig. 5.** Sensitivity analysis of parameter $\epsilon$.

### 4.8. Analysis of parameter $\epsilon$

We take Citeseer, Texas and Wiki as examples to analyze the potential impact of parameters $\epsilon$ on method performance. As we mentioned

at the beginning of this section, for the first fusion between $\mathbf{V}_i$ and $\mathbf{H}_i$, we recommend that the value of $\epsilon$ is not greater than 0.4, otherwise it should not be less than 0.5. The sensitivity experiment is shown in Fig. 5. Overall, different $\epsilon$ do not cause great fluctuations in method performance. Specifically, for NMI, AC and ARI, the performance variation of SADNG is in the range of 7.1%, 11.2%, and 15.4%, respectively, and is mainly reflected in the Texas dataset. The main reason for the performance fluctuation in the different fusion process of $\mathbf{H}_\ell$ and $\mathbf{V}_\ell$ is the presence of large hub nodes in datasets like Texas. As a result, the value of $\epsilon$ can cause a certain degree of variation in performance. For Citeseer and Wiki, although the method's performance also exhibits some fluctuations, the degree of fluctuation is notably lower compared to that observed in the Texas, particularly for Wiki. Nevertheless, SADNG still can achieve good performance with a fixed value of $\epsilon$, eliminating the need for prudent adjustments to its value.

### 4.9. Discussion

We propose the SADNG framework for general networks that only contain single topological information. This framework integrates DNMF with GCN, demonstrating significant advantages in community detection tasks. By combining the interpretability and noise resilience of DNMF with the nonlinear feature extraction capabilities of GCN, SADNG effectively addresses common challenges in complex network analysis, such as the over-smoothing issue in GCNs and the linearity limitations of DNMF. However, the application of SADNG to certain specialized scenarios may bring some challenges. In the following, we discuss SADNG from four perspectives: (1) the challenges it faces in large-scale, complex networks, (2) its compatibility with multimodal and heterogeneous networks, (3) its robustness in dynamic community detection tasks, and (4) its inherent limitations.

(1) Scalability in large-scale and complex networks. The scalability of SADNG depends primarily on the GCN encoder module. For instance, related studies, such as Bo et al. (2021), Wu et al. (2023), have enhanced the performance of the encoder by modifying the frequency response function of GCN, allowing for a more precise capture of smooth information. These findings can be seamlessly integrated into the SADNG framework to further enhance its flexibility. Although SADNG demonstrates adaptability in handling complex networks through improvements to the GCN encoder module, computational bottlenecks may still arise when processing large-scale networks. Specifically, due to the intrinsic computational complexity of DNMF, SADNG may encounter high computational overhead in large datasets. To alleviate this issue, it is possible to reduce the dimensionality of intermediate matrices in the DNMF encoder and decoder modules, thereby retaining essential features while minimizing the data volume required by subsequent layers.

(2) Compatibility with multimodal and heterogeneous networks. Although SADNG is primarily designed for unimodal and homogeneous networks, many real-world networks exhibit multimodal or heterogeneous characteristics, where different types of nodes and edges represent diverse entity relationships. Extending SADNG to such networks would require modality-specific adaptations. A promising approach would be to incorporate heterogeneous layers within the GCN, allowing each node and edge type to receive individualized attention through tailored convolutional filters. This adjustment would enable SADNG to capture interactions across various modalities and could be combined with DNMF to maintain structural coherence within each modality. Further development and evaluation of SADNG in multimodal contexts present a valuable direction for future research, particularly for applications in social or biological networks where diverse data types coexist.

(3) Robustness in dynamic networks. Although the current SADNG design primarily addresses static topologies, it can be adapted to accommodate dynamic network updates. For instance, employing GraphSAGE in place of GCN allows for incremental training to efficiently manage evolving data. Additionally, by incorporating a variant of GCN (Pareja et al., 2020), SADNG can capture temporal evolution patterns within network structures, thereby enhancing its capability to dynamically adapt to changing community structures.

(4) Limitation. The integration of DNMF and GCN, despite its demonstrated efficacy, imposes substantial computational demands, particularly for dense or large-scale networks, due to the multi-layered decomposition of high-dimensional matrices. This computational challenge remains a focal concern within the field of non-negative matrix factorization research. Future work will investigate alternative strategies to strengthen the synergy between GCN and DNMF, aiming to reduce matrix decomposition requirements while preserving the performance levels achieved by the SADNG framework.

## 5. Conclusion

In this paper, we introduce a novel method called SADNG, which seamlessly connects GCN and DNMF, to address the problem of community detection. Unlike traditional single DNMF-based and GCN-based methods, SADNG offers significant advantages. Particularly, its unique way of cooperation enhances the capability of DNMF to withstand topology noise and compensates for its linearity flaw. Additionally, SADNG alleviates the serious problem of over-smoothing in GCN through the integration with DNMF. This mutual cooperation and enhancement lead to significant performance improvements.

## CRediT authorship contribution statement

**Junwei Cheng:** Conceptualization, Investigation, Methodology, Software, Writing – original draft, Writing – review & editing, Data curation, Funding acquisition. **Chaobo He:** Writing – review & editing, Funding acquisition, Supervision. **Xuequan Lin:** Writing – review & editing. **Weixiong Liu:** Writing – review & editing. **Kunlin Han:** Writing – review & editing. **Yong Tang:** Supervision, Funding acquisition.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant 62477016 and 62077045, Guangdong Basic and Applied Basic Research Foundation under Grant 2024A1515011758, and Scientific Research Innovation Project of Graduate School of South China Normal University under Grant 2024KYLX067.

## Data availability

Data will be made available on request.

## References

Ahmed, N. K., Rossi, R. A., Zhou, R., Lee, J. B., Kong, X., Willke, T. L., & Eldardiry, H. (2017). Inductive representation learning in large attributed graphs. arXiv preprint arXiv:1710.09471.

Al-sharoa, E., & Rahahleh, B. (2023). Community detection in networks through a deep robust auto-encoder nonnegative matrix factorization. *Engineering Applications of Artificial Intelligence*, *118*, Article 105657. http://dx.doi.org/10.1016/J.ENGAPPAI.2022.105657.

Bo, D., Wang, X., Shi, C., & Shen, H. (2021). Beyond low-frequency information in graph convolutional networks. In *Thirty-fifth AAAI conference on artificial intelligence, AAAI 2021* (pp. 3950–3957). AAAI Press, http://dx.doi.org/10.1609/AAAI.V35I5.16514.

Bo, D., Wang, X., Shi, C., Zhu, M., Lu, E., & Cui, P. (2020). Structural deep clustering network. In *WWW '20: the web conference 2020, taipei, Taiwan, April 20–24, 2020* (pp. 1400–1410). ACM / IW3C2, http://dx.doi.org/10.1145/3366423.3380214.

Chen, J., Gong, Z., Mo, J., Wang, W., Wang, W., Wang, C., Dong, X., Liu, W., & Wu, K. (2022). Self-training enhanced: Network embedding and overlapping community detection with adversarial learning. *IEEE Transactions on Neural Networks and Learning Systems*, *33*(11), 6737–6748. http://dx.doi.org/10.1109/TNNLS.2021.3083318.

Cheng, J., Tang, Y., He, C., Han, K., Li, Y., & Wei, J. (2024). Community detection in attributed networks via adaptive deep nonnegative matrix factorization. *Neural Computing and Applications*, *36*(2), 897–912. http://dx.doi.org/10.1007/S00521-023-09066-Y.

Cui, G., Zhou, J., Yang, C., & Liu, Z. (2020). Adaptive graph encoder for attributed graph embedding. In *KDD '20: the 26th ACM SIGKDD conference on knowledge discovery and data mining, virtual event, CA, USA, August 23–27, 2020* (pp. 976–985). ACM, http://dx.doi.org/10.1145/3394486.3403140.

Deng, P., Li, T., Wang, H., Wang, D., Horng, S.-J., & Liu, R. (2023). Graph regularized sparse non-negative matrix factorization for clustering. *IEEE Transactions on Computational Social Systems*, *10*(3), 910–921. http://dx.doi.org/10.1109/TCSS.2022.3154030.

Fettal, C., Labiod, L., & Nadif, M. (2022). Efficient graph convolution for joint node representation learning and clustering. In *WSDM '22: the fifteenth ACM international conference on web search and data mining, virtual event/tempe, AZ, USA, February 21–25, 2022* (pp. 289–297). ACM, http://dx.doi.org/10.1145/3488560.3498533.

He, C., Cheng, J., Chen, G., Guan, Q., Fei, X., & Tang, Y. (2024). Detecting communities with multiple topics in attributed networks via self-supervised adaptive graph convolutional network. *Information Fusion*, *105*, Article 102254. http://dx.doi.org/10.1016/J.INFFUS.2024.102254.

He, C., Fei, X., Cheng, Q., Li, H., Hu, Z., & Tang, Y. (2022). A survey of community detection in complex networks using nonnegative matrix factorization. *IEEE Transactions on Computational Social Systems*, *9*(2), 440–457. http://dx.doi.org/10.1109/TCSS.2021.3114419.

He, C., Zheng, Y., Fei, X., Li, H., Hu, Z., & Tang, Y. (2022). Boosting nonnegative matrix factorization based community detection with graph attention auto-encoder. *IEEE Transactions on Big Data*, *8*(4), 968–981. http://dx.doi.org/10.1109/TBDATA.2021.3103213.

Huang, J., Zhang, T., Yu, W., Zhu, J., & Cai, E. (2021). Community detection based on modularized deep nonnegative matrix factorization. *International Journal of Pattern Recognition and Artificial Intelligence*, *35*(2), 2159006:1–2159006:17. http://dx.doi.org/10.1142/S0218001421590060.

Ibrahim, R., & Gleich, D. F. (2019). Nonlinear diffusion for community detection and semi-supervised learning. In *The world wide web conference, WWW 2019, san francisco, CA, USA, May 13–17, 2019* (pp. 739–750). ACM, http://dx.doi.org/10.1145/3308558.3313483.

Jing, B., Park, C., & Tong, H. (2021). HDMI: high-order deep multiplex infomax. In *WWW '21: the web conference 2021, virtual event/ljubljana, Slovenia, April 19–23, 2021* (pp. 2414–2424). ACM/IW3C2, http://dx.doi.org/10.1145/3442381.3449971.

Kipf, T. N., & Welling, M. (2016). Variational graph auto-encoders. In *NIPS Workshop on Bayesian Deep Learning*.

Kipf, T. N., & Welling, M. (2017). Semi-supervised classification with graph convolutional networks. In *International conference on learning representations*.

Li, Y., Chen, J., Chen, C., Yang, L., & Zheng, Z. (2024). Contrastive deep nonnegative matrix factorization for community detection. In *IEEE international conference on acoustics, speech and signal processing, ICASSP 2024, seoul, Republic of Korea, April 14–19, 2024* (pp. 6725–6729). IEEE, http://dx.doi.org/10.1109/ICASSP48485.2024.10446107.

Liu, F., Xue, S., Wu, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Yang, J., & Yu, P. S. (2020). Deep learning for community detection: Progress, challenges and opportunities. In *Proceedings of the twenty-ninth international joint conference on artificial intelligence, IJCAI 2020* (pp. 4981–4987). ijcai.org, http://dx.doi.org/10.24963/IJCAI.2020/693.

Luo, L., Fang, Y., Cao, X., Zhang, X., & Zhang, W. (2021). Detecting communities from heterogeneous graphs: A context path-based graph neural network model. In *CIKM '21: the 30th ACM international conference on information and knowledge management, virtual event, queensland, Australia, November 1–5, 2021* (pp. 1170–1180). ACM, http://dx.doi.org/10.1145/3459637.3482250.

Mankad, S., & Michailidis, G. (2013). Structural and functional discovery in dynamic networks with non-negative matrix factorization. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, *88*, Article 042812. http://dx.doi.org/10.1103/PhysRevE.88.042812.

Mrabah, N., Bouguessa, M., & Ksantini, R. (2024). A contrastive variational graph auto-encoder for node clustering. *Pattern Recognition*, *149*, Article 110209. http://dx.doi.org/10.1016/J.PATCOG.2023.110209.

Newman, M. (2006). Newman mej.. Modularity and community structure in networks. Proc natl acad sci USA 103: 8577–8582. *Proceedings of the National Academy of Sciences of the United States of America*, *103*, 8577–8582. http://dx.doi.org/10.1073/pnas.0601602103.

Newman, M., & Girvan, M. (2004). Finding and evaluating community structure in networks. *Physical Review E*, *69*(2), 26113.

Pareja, A., Domeniconi, G., Chen, J., Ma, T., Suzumura, T., Kanezashi, H., Kaler, T., Schardl, T. B., & Leiserson, C. E. (2020). EvolveGCN: Evolving graph convolutional networks for dynamic graphs. In *The thirty-fourth AAAI conference on artificial intelligence, AAAI 2020* (pp. 5363–5370). AAAI Press, http://dx.doi.org/10.1609/AAAI.V34I04.5984.

Pizzuti, C., & Socievole, A. (2020). Multiobjective optimization and local merge for clustering attributed graphs. *IEEE Transactions on Cybernetics*, *50*(12), 4997–5009. http://dx.doi.org/10.1109/TCYB.2018.2889413.

Pompili, F., Gillis, N., Absil, P., & Glineur, F. (2014). Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *Neurocomputing*, *141*, 15–25. http://dx.doi.org/10.1016/J.NEUCOM.2014.02.018.

Pompili, F., Gillis, N., Absil, P., & Glineur, F. (2014). Two algorithms for orthogonal nonnegative matrix factorization with application to clustering. *Neurocomputing*, *141*, 15–25. http://dx.doi.org/10.1016/J.NEUCOM.2014.02.018.

Psorakis, I., Roberts, S., Ebden, M., & Sheldon, B. (2011). Overlapping community detection using Bayesian non-negative matrix factorization. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, *83*, Article 066114. http://dx.doi.org/10.1103/PhysRevE.83.066114.

Rosvall, M., & Bergstrom, C. T. (2008). Maps of random walks on complex networks reveal community structure. *Proceedings of the National Academy of Sciences*, *105*(4), 1118–1123. http://dx.doi.org/10.1073/pnas.0706851105.

Saberi-Movahed, F., Biswas, B., Tiwari, P., Lehmann, J., & Vahdati, S. (2024). Deep nonnegative matrix factorization with joint global and local structure preservation. *Expert Systems with Applications*, *249*, Article 123645. http://dx.doi.org/10.1016/J.ESWA.2024.123645.

Salahian, N., Tab, F. A., Seyedi, S. A., & Chavoshinejad, J. (2023). Deep autoencoder-like NMF with contrastive regularization and feature relationship preservation. *Expert Systems with Applications*, *214*, Article 119051. http://dx.doi.org/10.1016/J.ESWA.2022.119051.

Schuetz, P., & Caflisch, A. (2008). Efficient modularity optimization by multistep greedy algorithm and vertex mover refinement. *Physical Review. E, Statistical, Nonlinear, and Soft Matter Physics*, *77*, Article 046112. http://dx.doi.org/10.1103/PhysRevE.77.046112.

Shchur, O., & Günnemann, S. (2019). Overlapping community detection with graph neural networks. In *Deep Learning on Graphs Workshop, KDD*.

Su, X., Xue, S., Liu, F., Wu, J., Yang, J., Zhou, C., Hu, W., Paris, C., Nepal, S., Di Jin, Sheng, Q. Z., & Yu, P. S. (2024). A comprehensive survey on community detection with deep learning. *IEEE Transactions on Neural Networks and Learning Systems*, *35*(4), 4682–4702. http://dx.doi.org/10.1109/TNNLS.2021.3137396.

Sun, B., Shen, H., Gao, J., Ouyang, W., & Cheng, X. (2017). A non-negative symmetric encoder–decoder approach for community detection. In *Proceedings of the 2017 ACM on conference on information and knowledge management, CIKM 2017, Singapore, November 06–10, 2017* (pp. 597–606). ACM, http://dx.doi.org/10.1145/3132847.3132902.

Teng, X., Liu, J., & Li, M. (2021). Overlapping community detection in directed and undirected attributed networks using a multiobjective evolutionary algorithm. *IEEE Transactions on Cybernetics*, *51*(1), 138–150. http://dx.doi.org/10.1109/TCYB.2019.2931983.

Tiwari, P., Movahed, F. S., Karami, S., Saberi-Movahed, F., Lehmann, J., & Vahdati, S. (2024). A self-representation learning method for unsupervised feature selection using feature space basis. *Transactions on Machine Learning Research*, *2024*, URL https://openreview.net/forum?id=LNvbgBFPMt.

Trigeorgis, G., Bousmalis, K., Zafeiriou, S., & Schuller, B. W. (2017). A deep matrix factorization method for learning attribute representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(3), 417–429. http://dx.doi.org/10.1109/TPAMI.2016.2554555.

Trigeorgis, G., Bousmalis, K., Zafeiriou, S., & Schuller, B. W. (2017). A deep matrix factorization method for learning attribute representations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *39*(3), 417–429. http://dx.doi.org/10.1109/TPAMI.2016.2554555.

Tsitsulin, A., Palowitch, J., Perozzi, B., & Müller, E. (2023). Graph clustering with graph neural networks. *Journal of Machine Learning Research*, *24*, 127:1–127:21.

Wang, J., Cao, J., Li, W., & Wang, S. (2021). CANE: community-aware network embedding via adversarial training. *Knowledge and Information Systems*, *63*(2), 411–438. http://dx.doi.org/10.1007/S10115-020-01521-9.

Wang, X., Cui, P., Wang, J., Pei, J., Zhu, W., & Yang, S. (2017). Community preserving network embedding. In *Proceedings of the thirty-first AAAI conference on artificial intelligence, February 4–9, 2017, San francisco, California, USA* (pp. 203–209). AAAI Press, http://dx.doi.org/10.1609/AAAI.V31I1.10488.

Wang, D., Li, T., Deng, P., Zhang, F., Huang, W., Zhang, P., & Liu, J. (2023). A generalized deep learning clustering algorithm based on non-negative matrix factorization. *ACM Transactions on Knowledge Discovery from Data*, *17*(7), 99:1–99:20. http://dx.doi.org/10.1145/3584862.

Wang, D., Zhang, P., Deng, P., Wu, Q., Chen, W., Jiang, T., Huang, W., & Li, T. (2024). An autoencoder-like deep NMF representation learning algorithm for clustering. *Knowledge-Based Systems*, [ISSN: 0950-7051] *305*, Article 112597. http://dx.doi.org/10.1016/j.knosys.2024.112597.

Wu, Z., Pan, S., Long, G., Jiang, J., & Zhang, C. (2023). Beyond low-pass filtering: Graph convolutional networks with automatic filtering. *IEEE Transactions on Knowledge and Data Engineering*, *35*(7), 6687–6697. http://dx.doi.org/10.1109/TKDE.2022.3186016.

Xia, W., Wang, Q., Gao, Q., Zhang, X., & Gao, X. (2022). Self-supervised graph convolutional network for multi-view clustering. *IEEE Transactions on Multimedia*, *24*, 3182–3192. http://dx.doi.org/10.1109/TMM.2021.3094296.

Xu, S., Yang, C., Shi, C., Fang, Y., Guo, Y., Yang, T., Zhang, L., & Hu, M. (2021). Topic-aware heterogeneous graph neural network for link prediction. In *CIKM '21: the 30th ACM international conference on information and knowledge management, virtual event, queensland, Australia, November 1–5, 2021* (pp. 2261–2270). ACM, http://dx.doi.org/10.1145/3459637.3482485.

Yang, J., McAuley, J., & Leskovec, J. (2013). Community detection in networks with node attributes. In *2013 IEEE 13th international conference on data mining* (pp. 1151–1156). http://dx.doi.org/10.1109/ICDM.2013.167.

Ye, F., Chen, C., & Zheng, Z. (2018). Deep autoencoder-like nonnegative matrix factorization for community detection. In *Proceedings of the 27th ACM international conference on information and knowledge management, CIKM 2018, torino, Italy, October 22–26, 2018* (pp. 1393–1402). ACM, http://dx.doi.org/10.1145/3269206.3271697.

Yuan, Z., & Oja, E. (2005). Projective nonnegative matrix factorization for image compression and feature extraction. *3540*, (pp. 333–342). ISBN: 978-3-540-26320-3, http://dx.doi.org/10.1007/11499145.35,

Zhang, X., Liu, H., Wu, X., Zhang, X., & Liu, X. (2021). Spectral embedding network for attributed graph clustering. *Neural Networks*, *142*, 388–396. http://dx.doi.org/10.1016/J.NEUNET.2021.05.026.

Zhao, L., & Akoglu, L. (2020). PairNorm: Tackling oversmoothing in GNNs. In *8th international conference on learning representations, ICLR 2020, addis ababa, ethiopia, April 26–30, 2020*. OpenReview.net.

Zhao, H., Yang, X., Wang, Z., Yang, E., & Deng, C. (2021). Graph debiased contrastive learning with joint representation clustering. In *Proceedings of the thirtieth international joint conference on artificial intelligence, IJCAI 2021, virtual event / montreal, Canada, 19–27 August 2021* (pp. 3434–3440). ijcai.org, http://dx.doi.org/10.24963/IJCAI.2021/473.

Zheng, S., Zhu, Z., Liu, Z., Li, Y., & Zhao, Y. (2024). Node-oriented spectral filtering for graph neural networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *46*(1), 388–402. http://dx.doi.org/10.1109/TPAMI.2023.3324937.

Zhou, X., Su, L., Li, X., Zhao, Z., & Li, C. (2023). Community detection based on unsupervised attributed network embedding. *Expert Systems with Applications*, *213*(Part), Article 118937. http://dx.doi.org/10.1016/J.ESWA.2022.118937.

Zhu, P., Li, J., Wang, Y., Xiao, B., Zhao, S., & Hu, Q. (2023). Collaborative decision-reinforced self-supervision for attributed graph clustering. *IEEE Transactions on Neural Networks and Learning Systems*, *34*(12), 10851–10863. http://dx.doi.org/10.1109/TNNLS.2022.3171583.

Zhuang, C., & Ma, Q. (2018). Dual graph convolutional networks for graph-based semi-supervised classification. In *Proceedings of the 2018 world wide web conference on world wide web, WWW 2018, lyon, France, April 23–27, 2018* (pp. 499–508). ACM, http://dx.doi.org/10.1145/3178876.3186116.