POLITECNICO DI TORINO
Repository ISTITUZIONALE

Entropy-Guided Graph Clustering via Rényi Optimization

Original
Entropy-Guided Graph Clustering via Rényi Optimization / Begga, A., Beretta, G., Vascon, S., Escolano, F., Lozano, M.A., Pelillo, M.. - ELETTRONICO. - 15727:(2025), pp. 189-199. (14th IAPR-TC15 International Workshop on Graph-Based Representations in Pattern Recognition (GbRPR 2025) Caen (FRA) June 25–27, 2025) [10.1007/978-3-031-94139-9_18].

Availability:
This version is available at: 11583/3001666 since: 2025-07-09T05:56:35Z

Publisher:
Springer

Published
DOI:10.1007/978-3-031-94139-9_18

Terms of use:

This article is made available under terms and conditions as specified in the corresponding bibliographic description in the repository

Publisher copyright
Springer postprint/Author's Accepted Manuscript (book chapters)

This is a post-peer-review, pre-copyedit version of a book chapter published in Graph-Based Representations in Pattern Recognition. The final authenticated version is available online at: http://dx.doi.org/10.1007/978-3-031-94139-9_18

(Article begins on next page)

18 November 2026

Entropy-Guided Graph Clustering via Rényi Optimization

Ahmed Begga[1[0009−0000−8733−2072]⋆], Guglielmo Beretta[2,3[0000−0002−4757−1965]], Sebastiano Vascon[2,5[0000−0002−7855−1641]], Francisco Escolano[1[0000−0003−3238−4021]], Miguel Angel Lozano[1[0000−0002−4757−5587]], and Marcello Pelillo[2,4,5[0000−0001−8992−9243]]

[1] University of Alicante, Alicante, Spain
[2] Ca' Foscari University of Venice, Venice, Italy
[3] Politecnico di Torino, Turin, Italy
[4] Zhejiang Normal University, Jinhua, Zhejiang Province, China
[5] ECLT, Venice, Italy

Abstract. Graph clustering is a fundamental task in network analysis, with applications ranging from community detection to protein complex identification. While Graph Neural Networks (GNNs) have shown promising results in this domain, they often struggle to balance local structure preservation with global cluster separation. We present a novel information-theoretic framework that enhances graph clustering through differentiable Rényi entropy optimization. Our approach introduces a computationally efficient masked entropy loss that encourages informative node representations while respecting graph topology. By integrating this framework with state-of-the-art GNN architectures, we achieve significant improvements in clustering quality across multiple benchmark datasets.

Keywords: Graph Clustering · Graph Neural Networks · Unsupervised learning · Information Theory.

1 Introduction

Graph clustering remains a fundamental challenge in network analysis, with applications ranging from community detection in social networks [8,20] to protein complex identification in biological networks [5]. Modern networks often contain rich node features alongside their topological structure, necessitating methods to effectively leverage both sources of information [16,21]. Unlike traditional approaches, relying on structural properties or attribute information independently [14], Graph Neural Networks (GNNs) have emerged as powerful tools for learning representations that capture both structural and attribute information [11]. Recent advances in graph clustering have leveraged these representations within various approaches, including spectral methods [2], attention mechanisms [22], and hierarchical pooling strategies [24]. However, these methods often lack

⋆ Corresponding author: ahmed.begga@ua.es

a principled way to deal with the entropy of the embedding, which affects the quality of the model's learned representation [24]. We present an approach to enhance graph clustering through entropy optimization, guiding graph representations toward more informative and well-separated clusters. Our method introduces a differentiable loss function based on Rényi entropy [17] that can be seamlessly integrated into any GNN architecture's training process. Unlike existing methods relying on a loss function that focus solely on the clustering task, our approach aims at producing more informative embeddings, within a theoretically grounded framework, while preserving both structural and attribute information. The key insight of our work is that while GNNs excel at capturing node-level information, the quality of the resulting clusters can be significantly improved by explicitly optimizing the entropy of the embedding space. Indeed, leveraging Rényi entropy optimization, we create an adaptive learning process that enhances cluster separation through information-theoretic principles. Our approach enriches the current graph clustering methods aiming to address one fundamental issue in graph clustering: bridging the gap between representation learning and clustering quality [8]. Our main contributions are:

- A novel, differentiable loss function based on Rényi entropy that can be integrated into any GNN architecture to enhance clustering performance while preserving both structural and attribute information;
- A theoretical foundation for integrating Rényi entropy optimization into existing GNN-based clustering approaches, providing a principled way to enhance the discriminative power of graph embeddings;
- Experiments showing significant improvements in clustering quality across multiple real-world datasets, particularly in scenarios where traditional methods struggle to produce informative embeddings.

Our experimental results demonstrate that this approach consistently outperforms state-of-the-art methods while maintaining computational efficiency. The method's ability to work with any GNN architecture makes it a versatile tool for enhancing graph clustering pipelines, particularly in scenarios where both network structure and node attributes play crucial roles in defining meaningful clusters [23].

2 Related Work

GNNs have revolutionized the processing of graph-structured data by enabling deep learning on irregular domains [4,18]. The key idea behind GNNs is the message passing framework [9], which iteratively updates node representations by aggregating information from their neighborhoods as follows:

$$\mathbf{h}_v^{(k+1)} = \text{UPDATE}\left(\mathbf{h}_v^{(k)}, \text{AGGREGATE}\left(\mathbf{h}_u^{(k)} : u \in \mathcal{N}(v)\right)\right) \tag{1}$$

where $\mathbf{h}_v^{(k)}$ represents the features of node v at layer k, whereas $\mathcal{N}(v)$ denotes the neighbors of v, and UPDATE is a learnable transformation. Through this

iterative process, GNNs learn representations capturing local graph structure *and* node features [11,12].

Graph clustering aims to partition nodes into groups with high intra-cluster and low inter-cluster connectivity [8]. While traditional approaches focused on optimizing structural metrics like modularity [14] or normalized cut [19], recent works have begun to explore the fundamental connection between clustering and information theory [25,13]. The key insight is that effective clustering should maximize cluster cohesion while minimizing inter-cluster similarity [10,16].

A significant advancement in this direction came with DiffPool [24], which introduced a differentiable hierarchical pooling framework that learns to progressively coarsen graphs. DiffPool generates soft cluster assignments through a learned assignment matrix $\mathbf{S} \in \mathbb{R}^{N \times C}$, where N is the number of nodes and C is the number of clusters given by the data labels. Each element $S_{ij} \in [0, 1]$ represents the probability of assigning node i to cluster j, thus $\sum_{j=1}^{C} S_{ij} = 1$. This method combines a link prediction loss function $\mathcal{L}_{\text{link}}$ with an entropy regularization term $\mathcal{L}_{\text{entropy}}$ to prevent degenerate solutions, given by

$$\mathcal{L}_{\text{link}} = \|\mathbf{A} - \mathbf{S}\mathbf{S}^\top\|_F, \qquad \mathcal{L}_{\text{entropy}} = -\sum_i \sum_j S_{ij} \log(S_{ij}), \tag{2}$$

with $\mathbf{A}$ being the adjacency matrix of the graph. The term $\mathcal{L}_{\text{entropy}}$ plays a crucial role in encouraging informative and well-distributed cluster assignments.

Building on these information-theoretic principles, MinCutPool [2] proposed a method merging spectral clustering theory with deep learning. Specifically, it minimizes a normalized-cut objective function $\mathcal{L}_{\text{cut}}$ leveraging a penalty function $\mathcal{L}_{\text{ortho}}$ to enforce orthogonality constraints. These functions are defined as follows:

$$\mathcal{L}_{\text{cut}} = -\frac{\text{Tr}(\mathbf{S}^\top \mathbf{A}\mathbf{S})}{\text{Tr}(\mathbf{S}^\top \mathbf{D}\mathbf{S})}, \qquad \mathcal{L}_{\text{ortho}} = \|\mathbf{S}^\top \mathbf{S} - \mathbf{I}\|_F, \tag{3}$$

where $\mathbf{D}$ is the degree matrix of the graph. This method naturally maximizes information flow between clusters while ensuring their distinctiveness through the orthogonality constraint.

Taking a different approach, DMoN [21] focuses on optimizing community structure through a loss function depending on the modularity matrix $\mathbf{B}$ of the graph, i.e., through the following differentiable version of modularity:

$$\mathcal{L}_{\text{DMoN}} = -\frac{1}{2E}\text{Tr}(\mathbf{S}^\top \mathbf{B}\mathbf{S}) + \frac{\sqrt{k}}{n}\Big\| \sum_i \mathbf{S}_i^\top \Big\|_F - 1. \tag{4}$$

This model balances community detection with cluster size regularization, implicitly promoting informative graph partitions.

While each of these approaches offers unique strengths — hierarchical representations (DiffPool), spectral guarantees (MinCutPool), and community-aware clustering (DMoN) — they all rely on indirect ways to improve their node representations to address the clustering task. In contrast, our work introduces a direct, principled approach to address the embedding entropy through Rényi

entropy optimization, thereby providing a flexible framework that can enhance any of these existing methods.

3 Preliminaries

Let $G = (V, E)$ be an undirected graph with node set $V = \{v_1, \ldots, v_n\}$ and edge set $E \subseteq V \times V$. Furthermore, we assume that each node v_i is associated with a feature vector $\mathbf{x}_i \in \mathbb{R}^d$. Let $\mathbf{A} \in \mathbb{R}^{n \times n}$ be the adjacency matrix of G, which satisfies $A_{ij} = 1$ if $(v_i, v_j) \in E$ and $A_{ij} = 0$ otherwise, and let $\mathbf{D} \in \mathbb{R}^{n \times n}$ be the degree matrix of G, i.e., the diagonal matrix defined by $D_{ii} = \sum_{j=1}^{n} A_{ij}$.

Information theory provides powerful tools for analyzing and quantifying the complexity of data distributions, particularly in the context of clustering aiming to minimize the intra-cluster variance while maximizing inter-cluster separation. The Rényi entropy family [17,6] offers a particularly elegant framework for this purpose, providing a notion of entropy generalizing Shannon entropy and depending on a real parameter α that modulates how the entropy responds to probability variations and that satisfies $\alpha > 0$ and $\alpha \neq 1$. For a probability distribution $P(x)$, the Rényi entropy of order α is defined as [6]:

$$H_\alpha(X) = \frac{1}{1-\alpha} \log \int P^\alpha(x) dx \tag{5}$$

This family includes several important special cases, each offering unique properties for optimization. Notably, as $\alpha \to 1$, it converges to the Shannon entropy. For $\alpha = 2$, it yields the so-called quadratic Rényi entropy [17], which has a key advantage in the context of clustering: it can be efficiently estimated via kernel density estimation [6]. This proves particularly valuable for our approach, providing a differentiable objective function that naturally encourages cluster formation. As mentioned in [6], when using Gaussian kernels $G(\cdot, \sigma)$ and $\alpha = 2$, the integral in (5) can be expressed in terms of the "information potential" $V(X)$ between samples, given by

$$V(X) = \frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} G(\|\mathbf{x}_i - \mathbf{x}_j\|^2, \sigma), \tag{6}$$

so that we may write $H(X) = -\log(V(X))$. Notably, there exists a physical interpretation which provides more insight into $V(X)$ and that directly relates to our clustering objectives. According to this interpretation, samples can be viewed as particles that interact through a force field, where $V(X)$ represents the potential energy of these interactions [6]. Crucially, descending the gradient of this potential naturally encourages similar nodes to cluster together, and if the function $V(X)$ is minimized in conjunction with other losses that maintain separation between dissimilar ones, this directly aligns the minimization of $V(X)$ with our goal of forming compact, well-separated clusters.

Building on this framework, this leads us to consider the following function:

$$\mathcal{L} = -\log\Big(\frac{1}{n^2} \sum_{i=1}^{n} \sum_{j=1}^{n} G(\|\mathbf{h}_i - \mathbf{h}_j\|^2, \varepsilon)\Big) \tag{7}$$

where $\mathbf{h}_i$ are the learned node representations, ε controls the kernel bandwidth. Minimizing this function results in representations where nodes have a more uniform spatial distribution in the embedding space.

By incorporating a modified version of this function into our loss, we explicitly encourage the network to learn embeddings where connected nodes sharing the same cluster have more similar representations. This leads to lower entropy within clusters and better separation between them, effectively leveraging the entropy minimization principle to create more compact and distinguishable clusters.

4 Methodology

Our approach leverages the powerful properties of Rényi entropy optimization to enhance state-of-the-art graph clustering methods through a carefully designed differentiable loss function. The key insight driving our method is that by strategically minimizing the entropy between connected nodes we can naturally guide the formation of well-defined, compact clusters. This process, which is visualized in Figure 1, demonstrates how entropy minimization serves as a powerful regularizer that encourages clearer cluster boundaries and more cohesive cluster structures.

4.1 Differentiable Rényi Loss for Cluster Enhancement

We formulate our Rényi entropy-based objective as a fully differentiable loss function that can be seamlessly integrated into any GNN architecture, specifically designed to enhance cluster formation:

$$\mathcal{L}_{\text{Rényi}} = -\log\left(n^{-2}\sum_{i,j} k(\mathbf{h}_i, \mathbf{h}_j)\cdot A_{ij}\right) \tag{8}$$

where $\mathbf{h}_i$ is the i-th column of the assignment matrix $\mathbf{S}$, which we regard as the learned representation of node i, and $k(\cdot,\cdot)$ is the Gaussian kernel function:

$$k(\mathbf{h}_i, \mathbf{h}_j) = \exp\left(-\frac{\|\mathbf{h}_i - \mathbf{h}_j\|^2}{2\varepsilon^2}\right), \tag{9}$$

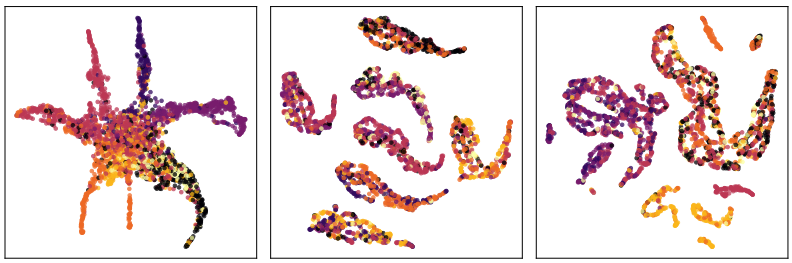
which measures the similarity between node embeddings, whereas A_{ij} is the adjacency matrix. A crucial aspect of Equation (8) is the strategic masking of the kernel matrix with the adjacency matrix. This masking serves two key purposes that directly enhance our clustering objectives:

1. It can significantly reduce computational complexity, since only existing edges are considered, hence resulting in a complexity of $\mathcal{O}(\mathsf{E})$ operations and memory, where E is the number of edges in the graph, whereas considering all possible node pairs would result in a complexity of $\mathcal{O}(n^2)$. For sparse graphs, where $\mathsf{E} \ll n^2$, this yields a substantial reduction in both computational and memory requirements while maintaining the effectiveness of the entropy-based clustering enhancement;
2. It focuses the entropy optimization specifically on structurally connected nodes,[6] ensuring that the learned representations respect the graph topology

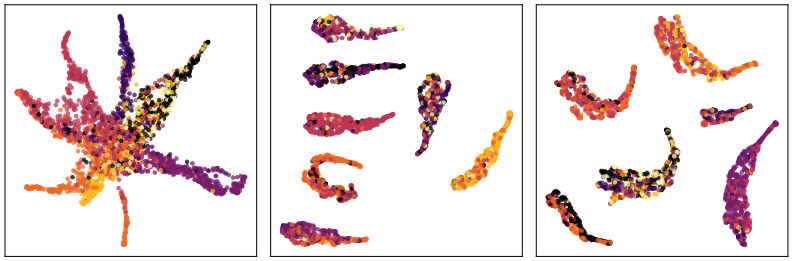
[6] However, this may be undersirable when working with heterophilic graphs [25].

while encouraging the formation of compact, coherent clusters. This targeted approach ensures that nodes within the same natural community are pulled closer together in the embedding space.

This formulation effectively leverages entropy minimization as a natural clustering mechanism: by minimizing the Rényi entropy between connected nodes, we implicitly encourage the formation of tight, well-separated clusters in the embedding space. The exponential nature of the Gaussian kernel ensures that this effect is strongest for nodes that are already similar, leading to a positive feedback loop that enhances cluster formation and separation.



(a) DiffPool (b) MinCutPool (c) DMoN



(d) DiffPool+ (e) MinCutPool+ (f) DMoN+

Fig. 1: Visualization of learned node embeddings using UMAP dimensionality reduction on the Cora dataset. Colors represent different ground truth classes. Top row shows baseline models. Bottom row shows models enhanced with Rényi entropy optimization.

4.2 Integration with Graph Neural Networks

We integrate our Rényi loss into several state-of-the-art graph clustering architectures by augmenting their original loss function as follows:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{original}} + \lambda \mathcal{L}_{\text{Rényi}}, \tag{10}$$

where the paremeter $\lambda > 0$ balances the contribution of the entropy-based term.

This combined loss creates a powerful synergy: while the original objective function preserves essential graph properties, the Rényi entropy term actively

works to enhance cluster formation by pulling similar nodes closer together in the embedding space. As demonstrated in Figure 1, this leads to significantly improved cluster separation and compactness across all architectures. Taking MinCutPool as an example, the total loss function becomes:

$$\mathcal{L}_{\text{total}} = \underbrace{-\frac{\text{Tr}(\mathbf{S}^\top \mathbf{A}\mathbf{S})}{\text{Tr}(\mathbf{S}^\top \mathbf{D}\mathbf{S})}}_{\text{Cut loss}} + \underbrace{\|\mathbf{S}^\top \mathbf{S} - \mathbf{I}\|_F}_{\text{Orthogonality}} \underbrace{- \lambda \log\left(n^{-2} \sum_{i,j} k(\mathbf{h}_i, \mathbf{h}_j) \cdot A_{ij}\right)}_{\text{Rényi entropy}} \tag{11}$$

This formulation demonstrates how the Rényi entropy term works in concert with traditional clustering loss functions: while the cut loss encourages community structure and the orthogonality term ensures cluster distinctness, our entropy-based regularization actively promotes the formation of compact, well-defined clusters by minimizing the information content within natural graph communities.

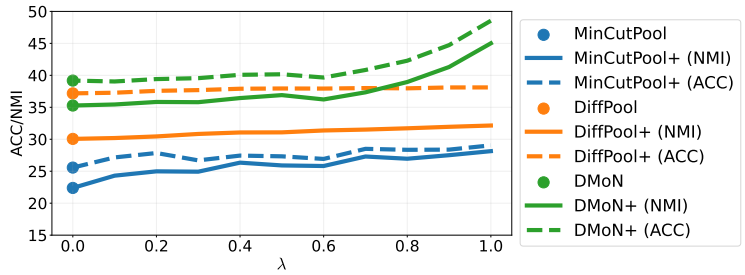
5 Experiments and Discussions

We evaluated our method on three widely-used citation network datasets: Cora, CiteSeer, and PubMed [15]. Each dataset consists of scientific publications as nodes, with undirected citation links as edges. Node features are bag-of-words representations of the document content, and each node is assigned to one of several predefined research topics that serve as ground truth clusters. We

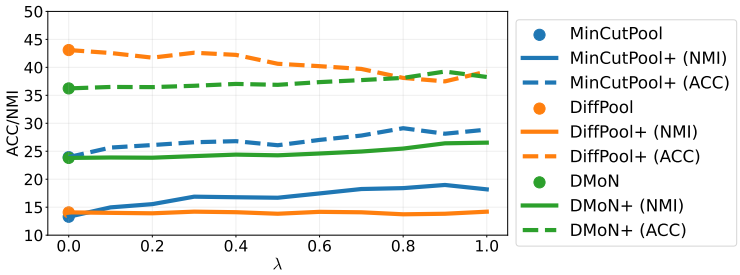
	Cora		**CiteSeer**		**PubMed**	
	NMI	ACC	NMI	ACC	NMI	ACC
DiffPool	30.06±0.98	37.18±1.42	14.07±2.32	42.59±2.33	18.63±4.73	45.99±4.98
DiffPool+	**31.06±0.87**	**37.90±0.96**	**14.08±2.33**	**43.09±2.88**	**18.87±4.80**	**46.02±5.12**
MinCutPool	22.39±2.61	25.57±1.96	13.29±1.40	23.96±2.52	15.36±3.12	39.14±8.26
MinCutPool+	**24.93±2.92**	**26.70±2.49**	**16.88±2.44**	**26.61±3.15**	**20.36±1.20**	**46.14±6.80**
DMoN	35.31±3.79	48.54±3.89	24.77±3.29	38.28±3.00	18.23±6.80	54.38±6.82
DMoN+	**45.01±3.17**	**50.26±4.08**	**26.54±2.50**	**42.56±3.95**	**19.08±6.36**	**55.11±5.62**

Table 1: Clustering performance comparison on citation networks. For each method, we report the average NMI and ACC (in %) with standard deviations over 10 runs. The + suffix indicates models enhanced with Rényi entropy optimization. Best results for each architecture are shown in bold.

implemented our approach using PyTorch and PyTorch Geometric [7]. All models were trained using the *Adam* optimizer, with a learning rate of 0.001 for 1000 epochs. The training was performed using graphs structure and node features, but *not* ground-truth node labels, since our method is meant to operate in an unsupervised setting. Ground-truth labels were only used post-training for evaluation purposes. For fair comparison, we maintained consistent architecture



(a) Cora dataset.



(b) CiteSeer dataset.

Fig. 2: Ablation study: Rényi entropy coefficient λ effect across pooling methods (NMI and ACC metrics).

across all methods: a two-layer GNN with 512 hidden units, followed by the respective pooling layer. To evaluate clustering quality [3], we first converted soft cluster assignments into hard assignments by assigning each node to its highest-probability cluster. Then, we evaluated the resulting clustering with the following metrics: 1) Normalized Mutual Information (NMI), which quantifies the mutual dependence between predicted clusters and ground truth labels and provides an information-theoretic measure of clustering quality; 2) Accuracy (ACC), which measures the proportion of correctly assigned nodes after finding the optimal one-to-one mapping between predicted clusters and ground truth. To ensure robust evaluation, all experiments were repeated 10 times with different random seeds, and we computed each metric's mean and standard deviation.

An ablation study was conducted to analyze the impact of λ ranging in the set $\Lambda = \{0.1 \times i : \ i = 0, 1, \ldots, 10\}$. As shown in Figure 2, as λ increases, the effect varies significantly across different architectures and datasets. On Cora dataset (Figure 2a), DMoN shows remarkable improvement, with ACC rising from 39% to nearly 48%. MinCutPool and DiffPool demonstrate more moderate but consistent improvements. However, the CiteSeer results (Figure 2b) reveal that increasing λ can be detrimental, particularly for DiffPool, where ACC decreases from 42% to 39%. This emphasizes the importance of proper hyperparameter tuning.

For each enhanced model and dataset combination, we searched for the associated value of $\lambda \in \Lambda$ yielding the best clustering performance. We reported the

corresponding scores in Table 1, whereby we compare the enhanced models against their baseline counterparts. The results demonstrate consistent improvements across all architectures and datasets. Particularly, enhanced DMoN achieves impressive gains on Cora dataset, improving NMI from 35.31% to 45.01% and ACC from 48.54% to 50.26%, a significant advancement in clustering quality.

The impact of Rényi loss varies across different architectures. DiffPool shows moderate but consistent improvements also with lower values of λ, suggesting that entropy optimization complements its hierarchical structure. MinCut exhibits more substantial gains, especially notable in PubMed dataset where accuracy improves by 7%. DMoN demonstrates the most dramatic improvements, particularly with higher λ values, indicating a strong synergy between modularity optimization and entropy maximization principles.

All hyperparameters configurations, including optimal λ values for each method and dataset combination, are available in our repository [1], along with the complete implementation of our approach.

6 Conclusions and Future Work

We introduced a differentiable Rényi entropy optimization framework for graph clustering that consistently improves state-of-the-art GNN architectures while maintaining computational efficiency. Our method enhances clustering quality across multiple datasets, demonstrating the effectiveness of information-theoretic principles in graph representation learning.

We defer to future work the study of our approach on diverse graphs structures, also to test the scalability to larger networks, and with additional clustering techniques. Furthermore, it would be interesting to consider also topology-based metrics that are independent of ground truth labels, e.g., modularity, conductance.

Acknowledgement

The authors acknowledge the support from the Spanish Government under project PID2022-142516OB-I00. G. B. was funded jointly by Univesità Ca' Foscari di Venezia and Politecnico di Torino.

References

1. Begga, A.: EGGC2025 (2025), `https://github.com/AhmedBegggaUA/EGGC2025`
2. Bianchi, F.M., Grattarola, D., Alippi, C.: Spectral clustering with graph neural networks for graph pooling. Paper presented at ICLR 2020, 13–18 July 2020
3. Bo, D., Wang, X., Shi, C., Zhu, M., Lu, E., Cui, P.: Structural deep clustering network. In: WWW'20. pp. 1400–1410. ACM, New York (2020)
4. Bruna, J., Zaremba, W., Szlam, A., LeCun, Y.: Spectral networks and locally connected networks on graphs. Paper presented at ICLR 2014, Banff, Canada, 14–16 April 2014. `http://arxiv.org/abs/1312.6203`

5. Cabreros, I., Abbe, E., Tsirigos, A.: Detecting community structures in Hi-C genomic data. In: 2016 Annual Conference on Information Science and Systems (CISS). pp. 584–589. IEEE (2016)
6. Escolano, F., Suau, P., Bonev, B.: Information Theory in Computer Vision and Pattern Recognition. Springer, London (2009)
7. Fey, M., Lenssen, J.E.: Fast graph representation learning with PyTorch Geometric. Presented at RLGM Workshop, ICLR 2019, New Orleans, USA, 6–9 May 2019
8. Fortunato, S., Hric, D.: Community detection in networks: A user guide (2016), http://arxiv.org/abs/1608.00163
9. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. Paper presented at ICML 2017, Sydney, Australia, 6–11 August 2017
10. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: KDD'16. pp. 855–864. ACM, New York (2016)
11. Hamilton, W.L., Ying, R., Leskovec, J.: Representation learning on graphs: Methods and applications. IEEE Data Eng. Bull. **40**(3), 52–74 (2017)
12. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. Paper presented at ICLR 2017, Toulon, France, 24–26 April 2017
13. Micheli, A., Tortorella, D.: Designs of graph echo state networks for node classification. Neurocomput. **597**(C) (2024)
14. Newman, M.E.J.: Finding community structure in networks using the eigenvectors of matrices. Phys. Rev. E **74**, 036104 (2006)
15. Pei, H., Wei, B., Chang, K.C., Lei, Y., Yang, B.: Geom-GCN. Paper presented at ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020
16. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: online learning of social representations. In: KDD'14. pp. 701–710. ACM, New York (2014)
17. Rényi, A.: On measures of entropy and information. In: Proceedings of the Fourth Berkeley Symposium on Mathematical Statistics and Probability. vol. 1, pp. 547–562. Univ. of California Press (1961)
18. Scarselli, F., Gori, M., Tsoi, A.C., Hagenbuchner, M., Monfardini, G.: The graph neural network model. IEEE Trans. Neural Netw. **20**(1), 61–80 (2009)
19. Shi, J., Malik, J.: Normalized cuts and image segmentation. In: Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR). pp. 731–737. IEEE (1997)
20. Su, X., Xue, S., Liu, F., et al.: A comprehensive survey on community detection with deep learning. IEEE Trans. Neural Netw. Learn. Syst. **35**(4), 4682–4702 (2024)
21. Tsitsulin, A., Palowitch, J., Perozzi, B., Müller, E.: Graph clustering with graph neural networks (2020), https://arxiv.org/abs/2006.16904
22. Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y.: Graph attention networks. Paper presented at ICLR 2018, Vancouver, Canada, 30 April – 3 May 2018
23. Wang, C., Pan, S., Hu, R., Long, G., Jiang, J., Zhang, C.: Attributed graph clustering. Paper presented at IJCAI-19, Macao, China, 10–16 August 2019
24. Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W.L., Leskovec, J.: Hierarchical graph representation learning with differentiable pooling. Paper presented at NeurIPS 2018, Montréal, Canada, 2–8 December 2018
25. Zhong, Z., Gonzalez, G., Grattarola, D., Pang, J.: Unsupervised network embedding beyond homophily. Trans. Mach. Learn. Res. **2022**, submission no. 467 (2022)