Review Article

Review of article "Graph Clustering with Graph Neural Networks"

Aymane Masrour[a,1] **and Mohamed Badi**[a,2]

[a]*Department of Mathematics, Université d'Orsay*

Evaluated by Prof.: Karteek Alahari

Abstract—This report provides a critical review of the article "Graph Clustering with Graph Neural Networks." By reimplementing key methods using PyTorch and building scratch implementations of related baselines, we examine the theoretical foundations of spectral relaxation and modularity optimization in graph clustering. Our comparative analysis, based on both synthetic ADC-SBM data and real-world graphs, highlights differences in conclusions and offers insights into the robustness and scalability of the proposed approach.

Keywords—*Graph Neural Networks, Graph Clustering, Modularity, Spectral Clustering, ADC-SBM*

1. Introduction

Graph clustering plays a pivotal role in uncovering hidden community structures within complex networks, with applications ranging from social network analysis to bioinformatics. Graph Neural Networks (GNNs) have demonstrated remarkable success in supervised tasks such as node classification and link prediction. However, leveraging GNNs for unsupervised tasks like graph clustering remains challenging due to the intricacies of optimizing discrete clustering objectives in an end-to-end manner.

The article under review, "Graph Clustering with Graph Neural Networks," addresses these challenges by introducing Deep Modularity Networks (DMoN), a method that utilizes a modularity-inspired objective to learn soft cluster assignments through GNNs. In addition to DMoN, the original work evaluates baseline methods such as DiffPool and MinCutPool and employs the ADC-SBM model for synthetic data generation.

In this report, we revisit the original article's and **our main contributions** are reimplementing the DMoN model in PyTorch, along with from scratch implementations of DiffPool, MinCutPool, and the ADC-SBM synthetic model generator. See the github repo for full code with a notebook for deeper analysis [4]. Our goal is to compare the theoretical expectations and empirical findings of the original work with our own results, and to discuss any discrepancies in conclusions.

2. Theoretical Framework and Methodology

Graph clustering fundamentally aims to partition the nodes of a graph into communities characterized by dense intra-group connections and sparse inter-group links. A commonly used metric for evaluating the quality of such partitions is *modularity*, defined as

$$Q = \frac{1}{2m}\sum_{i,j}\left(A_{ij} - \frac{d_i d_j}{2m}\right)\delta(c_i, c_j), \tag{1}$$

where A is the adjacency matrix, d_i is the degree of node i, and $\delta(c_i, c_j)$ equals 1 if nodes i and j belong to the same cluster. Given that optimizing modularity is NP-hard, spectral relaxation techniques are employed to transform the discrete clustering problem into a continuous optimization problem. In this relaxation, the top eigenvectors of the modularity matrix provide soft assignments that can later be discretized.

Deep Modularity Networks (DMoN) extend this concept by integrating GNNs to learn a soft cluster assignment matrix $\mathbf{C}$ via a softmax function, thereby enabling end-to-end training. The soft assignments are computed as follows:

$$\mathbf{C} = \text{softmax}\big(\text{GCN}(\tilde{\mathbf{A}}, \mathbf{X})\big) \tag{2}$$

Where :

- $\tilde{\mathbf{A}} = \mathbf{D}^{-\frac{1}{2}}\mathbf{A}\mathbf{D}^{-\frac{1}{2}}$ is the normalized adjacency matrix
- $\mathbf{X}$ is the matrix of node features.

The objective function in DMoN is formulated as:

$$\mathcal{L}_{\text{DMoN}} = \underbrace{-\frac{1}{2m}\text{Tr}\left(\mathbf{C}^\top\left[\mathbf{A} - \frac{\mathbf{d}\mathbf{d}^\top}{2m}\right]\mathbf{C}\right)}_{\text{Modularity}} + \underbrace{\frac{\sqrt{k}}{n}\left\|\sum_i \mathbf{C}_i^\top\right\|_F}_{\text{Collapse Regularization}} - 1 \tag{3}$$

where the first term represents the negative modularity and the second term is a collapse regularization that prevents the trivial solution of assigning all nodes to a single cluster by enforcing balanced cluster sizes.

2.1. Theoretical Guarantees

The original article presents two key theorems to justify the proposed objective:

Non-Trivial Clustering

If the clustering solution encoded by the matrix $\mathbf{C}$ has positive modularity, then the DMoN objective $\mathcal{L}_{\text{DMoN}}$ is smaller than that of the trivial clustering solution (where all nodes are assigned to a single cluster).

This is true since the trivial clustering yields a modularity of zero, and any positive modularity contributes to a lower loss value. Moreover, the collapse regularization is minimized when cluster sizes are balanced. For completeness, we provide our own proof to this results in the appendix.

Consistency Under DC-SBM

Under the conditions specified in Theorem 3.1 from Zhao et al. (2012), and assuming equal cluster sizes in the Degree-Corrected Stochastic Block Model (DC-SBM), the DMoN objective achieves:

- Strong consistency when $\lambda_n / \log(n) \to \infty$.
- Weak consistency when $\lambda_n \to \infty$.

In other words, as the average node degree λ_n increases, the optimal soft assignment matrix converges to the true clustering, under the DC-SBM model with equal cluster size, ensuring that the clustering error vanishes in the large-graph limit.

In addition, the GNN architecture enhances the clustering process by integrating node features and local graph structure via a modified graph convolutional update:

$$X^{(t+1)} = \text{SeLU}\left(\tilde{A}X^{(t)}W^{(t)} + X^{(t)}W_{\text{skip}}\right), \tag{4}$$

This combination of spectral relaxation, collapse regularization, and GNN-based feature extraction forms the backbone of the theoretical framework evaluated in our review.

3. Comparative Baselines

In addition to the DMoN model, we compare our approach against two prominent baselines that employ soft assignments and end-to-end training for both embedding and clustering. However, each baseline uses a distinct training objective:

3.1. MinCutPool

MinCutPool [2] follows a mincut perspective, aiming to maximize intra-cluster connectivity relative to the degree distribution. Its objective function is given by:

$$\mathcal{L}_{\text{MC}} = -\frac{\text{Tr}(\mathbf{C}^\top \mathbf{A}\mathbf{C})}{\text{Tr}(\mathbf{C}^\top \mathbf{D}\mathbf{C})} + \|\mathbf{C}^\top\mathbf{C} - \mathbf{I}\|_F,$$

where:

- $\text{Tr}(\mathbf{C}^\top\mathbf{A}\mathbf{C})$ is a soft approximation of the total intra-cluster edge weight.
- $\text{Tr}(\mathbf{C}^\top\mathbf{D}\mathbf{C})$ represents the total cluster volume (with $\mathbf{D}$ being the degree matrix).
- The regularization term $\|\mathbf{C}^\top\mathbf{C} - \mathbf{I}\|_F$ enforces soft orthogonality, inspired by spectral clustering, where the optimal solution is given by a set of orthogonal eigenvectors yielding clear separation between clusters..

3.2. DiffPool

DiffPool [1] focuses on graph reconstruction, an approach that is particularly beneficial for node-level tasks such as classification, though it may sometimes compromise on capturing strong community structures. Its loss function is formulated as:

$$\mathcal{L}_{\text{LP}} = \|\mathbf{A} - \mathbf{C}\mathbf{C}^\top\|_F + \text{Entropy}(\mathbf{C}),$$

where:

- The reconstruction loss $\|\mathbf{A} - \mathbf{C}\mathbf{C}^\top\|_F$ encourages the learned cluster assignments to accurately represent the original graph structure.
- The entropy term, $\text{Entropy}(\mathbf{C})$, acts as a regularizer to push the soft assignments in $\mathbf{C}$ towards one-hot encodings, thereby facilitating more distinct clusters.

4. Experimental Evaluation

Our experimental evaluation is designed to validate the theoretical principles and compare empirical results with the original article's findings. We conduct experiments on both synthetic data generated via the ADC-SBM model and on real-world datasets.

4.1. Synthetic Data Experiments

Synthetic experiments using the ADC-SBM model allow us to test the limits of spectral clustering and modularity optimization. By varying parameters such as inter-cluster connectivity and feature variance, we simulate challenging conditions near the theoretical detectability limits.

4.1.1. ADC-SBM Model: Graph and Feature Generation

The synthetic data is generated using an attributed Degree-Corrected Stochastic Block Model (ADC-SBM), which extends the classical SBM by simultaneously generating the graph connectivity and node features.

Graph Generation (DC-SBM) The expected value of the adjacency matrix is defined as:

$$\mathbb{E}[A_{ij}] = \theta_i \theta_j D_{b_i,b_j},$$

where:

- $b_i \in \{0, \dots, k-1\}$ denotes the cluster assignment for node i, with approximately n/k nodes per cluster.
- θ_i is a power-law degree correction factor that accounts for node heterogeneity.
- D_{b_i,b_j} is a symmetric matrix where the diagonal entries D_{ii} control the average intra-cluster edge count (d_{in}) and the off-diagonal entries D_{ij} (for $i \neq j$) control the average inter-cluster edge count (d_{out}).
- The overall average degree is given by $d = d_{\text{in}} + d_{\text{out}}$.

Feature Generation Node features are generated according to:

$$\mathbf{x}_i \sim \mathcal{N}(\boldsymbol{\mu}_{z_i}, \sigma^2 I),$$

where:

- The feature cluster centers are drawn from $\boldsymbol{\mu}_z \sim \mathcal{N}(0, \sigma_c^2 I)$.
- $z_i \in \{0, \dots, k_f - 1\}$ denotes the feature cluster assignment, with k_f being the number of feature clusters.
- The relationship between graph clusters (b_i) and feature clusters (z_i) can be controlled to simulate different scenarios:
  - **Matched:** $z_i = b_i$
  - **Nested:** $z_i \subset b_i$
  - **Grouped:** $b_i \subset z_i$

By controlling the relationship between graph clusters (b_i) and feature clusters (z_i), we assess the sensitivity of clustering methods to the alignment or divergence of signals from edges and node features.

4.1.2. Experimental Scenarios and Metrics

To systematically explore the performance of our model, we vary key parameters across multiple synthetic scenarios. Table 1 summarizes each scenario and its associated parameter sweep:

Table 1. Parameter Variations for Synthetic ADC-SBM Scenarios

Scenario	Parameter
1	$d_{\text{out}} \in [2, 5.0]$
2	$\sigma_c \in [10^{-2}, 10^{1}]$
3	$\sigma_c \in [10^{-2}, 10^{1}]$ (nested)
4	$\sigma_c \in [10^{-2}, 10^{1}]$ (grouped)
5	$d \in [2^2, 2^7]$
6	$d_{\text{max}} \in [2^2, 2^{10}]$

Each scenario highlights a different aspect of structural or feature-based complexity:

- **Scenario 1:** As d_{out} grows, inter-cluster edges increase, challenging the detectability of distinct communities.
- **Scenarios 2–4:** Changing σ_c controls the separation of feature clusters. Matched, nested, or grouped configurations test whether the clustering method can handle alignment or divergence between graph structure and node features.
- **Scenario 5:** Increasing d (the total degree) tends to strengthen within-cluster connectivity, potentially simplifying the clustering task.
- **Scenario 6:** Adjusting d_{max} in the power-law distribution tests robustness against degree heterogeneity in large graphs.

By measuring Normalized Mutual Information (NMI) under each scenario, we assess how well each clustering method recovers the true block assignments despite varying levels of structural and feature-based challenges.

4.1.3. Results and Discussion

The experimental results (see Figure 1 in annex B) under various synthetic scenarios indicate that:

- **DMoN Performance:** DMoN achieves high and stable NMI performance across different parameter configurations, validating its design.
- **Robustness to Structural Changes:** Variations in parameters such as d_{out} and $d_{\max}$ demonstrate that DMoN is robust to changes in graph structure. The NMI values (often scaled by 100 for readability) and cluster size metrics show consistent behavior.
- **Robustness to Feature-Edge Signal Alignment:** The experiments reveal that DMoN maintains high performance even when the alignment between node features and graph structure varies. In scenarios where feature clusters are matched, nested, or grouped relative to graph clusters, DMoN consistently recovers the underlying community structure, highlighting its ability to leverage both edge and feature signals effectively.

4.2. Real-World Data Experiments

For real-world experiments, we consider three citation network datasets: Cora, Citeseer, and PubMed. In these datasets, each node represents a paper, edges represent citation links, and node features are derived from bag-of-words representations. These networks are characterized by imbalanced class distributions, power-law edge distributions, and diverse intra- and inter-class connectivity patterns.

4.2.1. Experimental Setup

In our real-world experiments, we use a uniform architecture across all datasets to ensure consistency. Specifically, we employ a single GNN layer with dropout for generating node embeddings. The same architecture is applied across Cora, Citeseer, and PubMed, and the outcomes are evaluated using three metrics:

- **Normalized Mutual Information (NMI):** Measures the correspondence between the learned clusters and the ground truth.
- **F1-score:** Assesses how well clusters match the actual node labels.
- **Modularity:** Evaluates the strength of community structures in the graph.

This combination of metrics enables us to determine whether a method is effective in both conventional community detection (via NMI and modularity) and in node-level classification tasks (via F1-score).

4.2.2. Quantitative Results: NMI Scores

Baseline	Citeseer	Cora	PubMed
DMoN	0.283 ± 0.038	0.467 ± 0.023	0.084 ± 0.079
DiffPool	0.032 ± 0.065	0.042 ± 0.084	0.036 ± 0.072
MinCutPool	0.290 ± 0.019	0.396 ± 0.042	0.198 ± 0.018

Table 2. NMI Scores for Citeseer, Cora, and PubMed.

Comment: The NMI results indicate that DiffPool struggles to achieve high NMI scores, suggesting that it does not effectively group nodes into strongly modular communities. In contrast, both DMoN and MinCutPool yield higher NMI values, which is consistent with expectations that modularity-focused losses can better recover dense, label-aligned subgraphs.

4.2.3. Quantitative Results: F1 Scores

Baseline	Citeseer	Cora	PubMed
DMoN	0.430 ± 0.027	0.513 ± 0.049	0.478 ± 0.030
DiffPool	0.488 ± 0.026	0.463 ± 0.002	0.572 ± 0.052
MinCutPool	0.392 ± 0.053	0.411 ± 0.064	0.400 ± 0.020

Table 3. F1 Scores for Citeseer, Cora, and PubMed.

Comment: With regard to F1 scores, DiffPool outperforms both DMoN and MinCutPool on Citeseer and PubMed, implying that its reconstruction-based approach better captures label boundaries. However, on Cora, DMoN achieves the highest F1 score, which suggests that the graph structure in Cora is particularly well-aligned with the node labels.

4.2.4. Quantitative Results: Modularity Scores

Baseline	Citeseer	Cora	PubMed
DMoN	0.711 ± 0.024	0.711 ± 0.016	0.266 ± 0.220
DiffPool	0.088 ± 0.177	0.079 ± 0.158	0.067 ± 0.134
MinCutPool	0.711 ± 0.040	0.705 ± 0.007	0.548 ± 0.008

Table 4. Modularity Scores for Citeseer, Cora, and PubMed.

Comment: Both DMoN and MinCutPool achieve high modularity scores on Citeseer and Cora, demonstrating their effectiveness in capturing strong community structures. However, on PubMed, DMoN shows higher variability in modularity, which may indicate overfitting or sensitivity to specific graph properties, while MinCutPool maintains a more stable performance.

4.2.5. Analysis and Discussion

In summary, our experiments show that:

- DMoN and MinCutPool excel in achieving high NMI and modularity scores, reflecting their effectiveness in recovering the underlying community structures.
- DiffPool, while providing better F1-scores on certain datasets, does not achieve the same level of modularity or NMI, suggesting that reconstruction-based approaches may focus more on label boundaries rather than on robust community detection.
- The variability observed in DMoN's performance on PubMed highlights potential limitations, which might be attributed to the absence of dataset-specific model fine-tuning.

5. Comparison with Original Work

While our assessment on the synthetic data match the conclusions in the reference article [3], we came up to different conclusions on the real dataset. In fact, the author of the article claims that DMoN outperforms all other methods across every metric (NMI, modularity, F1) on real-world data. However, our experiments reveal a more nuanced picture. While DMoN demonstrates superior performance in community detection, DiffPool achieves higher F1 scores in node classification, outperforming DMoN in this metric, while MinCutPool consistently delivers balanced results, which are somewhat expected behaviors given the trained objective in each case as explained in section 3.

6. Conclusion

In this review, we revisited the article "Graph Clustering with Graph Neural Networks" by reimplementing its key components using PyTorch. The theoretical framework—based on spectral relaxation and modularity optimization—provides a solid foundation for understanding the clustering process in GNNs. The experimental evaluation, spanning both synthetic and real-world data, confirms the effectiveness and efficiency of the approach while highlighting subtle differences in performance and conclusions. These insights pave the way for future research aimed at further refining unsupervised graph clustering techniques.

References

[1] R. Ying, J. You, C. Morris, X. Ren, W. Hamilton, and J. Leskovec, "Hierarchical graph representation learning with differentiable pooling", in *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

[2] F. M. Bianchi, D. Grattarola, and C. Alippi, "Mincutpool: Graph clustering and partitioning with differentiable pooling", in *Proceedings of the 37th International Conference on Machine Learning (ICML)*, 2020.

[3] A. Tsitsulin, J. Palowitch, B. Perozzi, and E. Müller, "Graph clustering with graph neural networks", *Journal of Machine Learning Research*, vol. 24, pp. 1–21, 2023, Submitted 9/20; Revised 4/23; Published 5/23.

[4] A. Masrour and M. Badi, "Github repo : Graph clustering with graph neural networks", *https://github.com/Masrour-Aymane/Unsupervised_Clustering_Using_GNNs*, 2025.

Proof of Theorem 1 (Collapse Penalty Comparison)

For soft assignment matrix $C \in \mathbb{R}^{n\times k}$ with rows summing to 1 (obtained via softmax), define $v = \sum_{i=1}^{n} C_{i,\cdot} \in \mathbb{R}^k$ where $\sum_{r=1}^{k} v_r = n$.

1. **Trivial clustering** $(v = (n, 0, ..., 0))$:

$$\|v\|_2 = n \Rightarrow \text{Penalty} = \sqrt{k} - 1$$

2. **Maximization problem**:

$$\text{maximize } \|v\|_2 \text{ s.t. } v_r \geq 0, \sum v_r = n$$

3. **Key observations**:
   - $\|v\|_2^2$ is strictly convex
   - Feasible set is convex and compact
   - Maximum must occur at extreme points (trivial clusterings)

A. Results on synthetic data

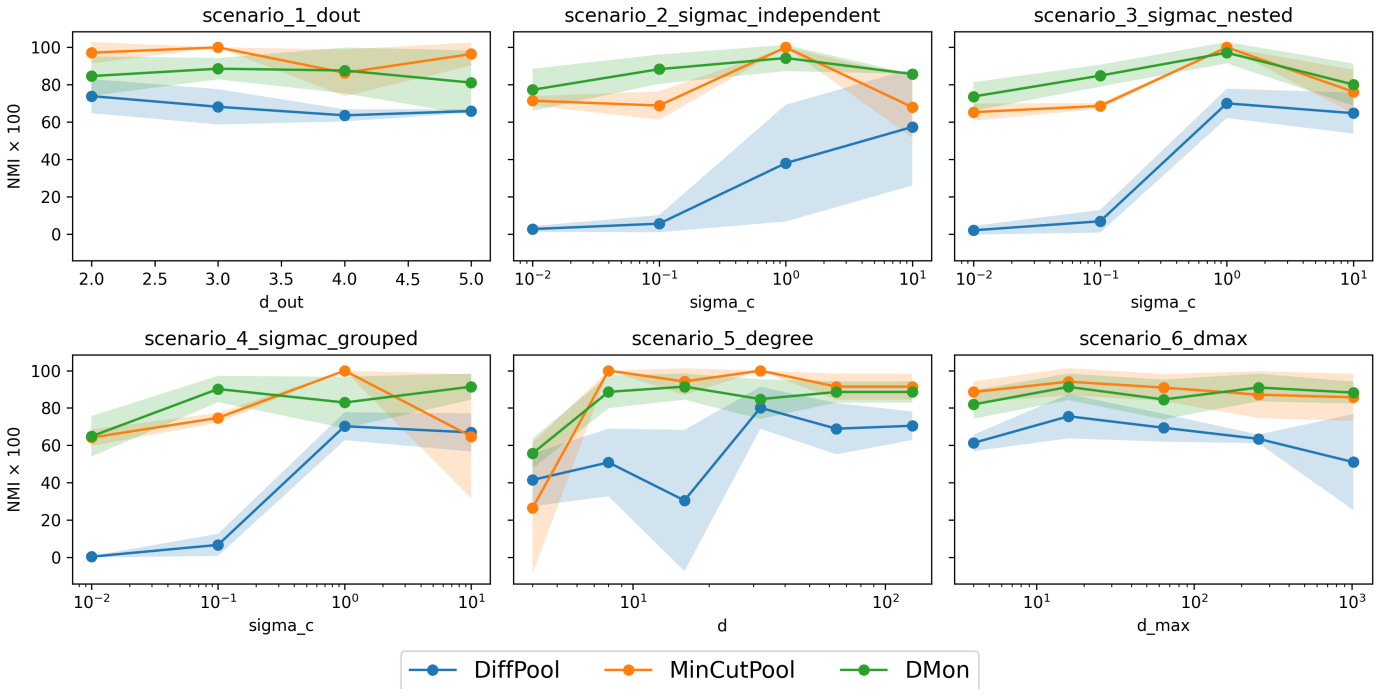

Figure 1. Results on synthetic data.