

GBGC: Efficient and Adaptive Graph Coarsening via Granular-ball Computing

Shuyin Xia¹, Guan Wang¹, Gaojie Xu¹, Sen Zhao^{2*} and Guoyin Wang³

¹ Chongqing Key Laboratory of Computational Intelligence, Key Laboratory of Big Data Intelligent Computing, Key Laboratory of Cyberspace Big Data Intelligent Security, Ministry of Education, School of Computer Science and Technology, Chongqing University of Posts and Telecommunications, Chongqing, China

² Chongqing Key Laboratory of Computational Intelligence, Key Laboratory of Big Data Intelligent Computing, Chongqing University of Posts and Telecommunications, Chongqing, China

³ National Center for Applied Mathematics in Chongqing, Chongqing Normal University, Chongqing 401331, China

xiasy@cqupt.edu.cn, wangguan66@foxmail.com, s230201132@stu.cqupt.edu.cn,
zhaosen@cqupt.edu.cn, wanggy@cqnu.edu.cn

Abstract

The objective of graph coarsening is to generate smaller, more manageable graphs while preserving key information of the original graph. Previous work were mainly based on the perspective of spectrum-preserving, using some predefined coarsening rules to make the eigenvalues of the Laplacian matrix of the original graph and the coarsened graph match as much as possible. However, they largely overlooked the fact that the original graph is composed of subregions at different levels of granularity, where highly connected and similar nodes should be more inclined to be aggregated together as nodes in the coarsened graph. By combining the multi-granularity characteristics of the graph structure, we can generate coarsened graph at the optimal granularity. To this end, inspired by the application of granular-ball computing in multi-granularity, we propose a new multi-granularity, efficient, and adaptive coarsening method via granular-ball (GBGC), which significantly improves the coarsening results and efficiency. Specifically, GBGC introduces an adaptive granular-ball graph refinement mechanism, which adaptively splits the original graph from coarse to fine into granular-balls of different sizes and optimal granularity, and constructs the coarsened graph using these granular-balls as supernodes. In addition, compared with other state-of-the-art graph coarsening methods, the processing speed of this method can be increased by tens to hundreds of times and has lower time complexity. The accuracy of GBGC is almost always higher than that of the original graph due to the good robustness and generalization of the granular-ball computing, so it has the potential to become a standard graph data preprocessing method.

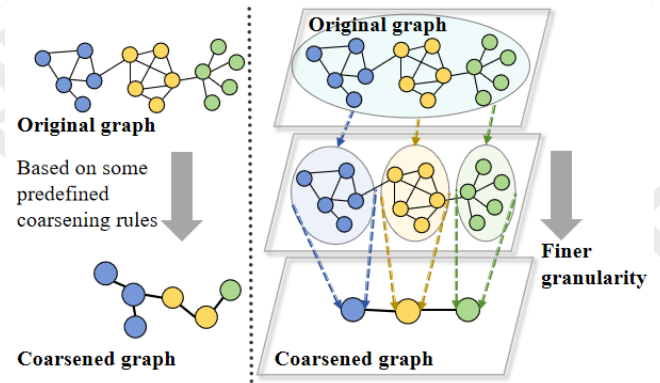


Figure 1: The differences between traditional graph coarsening methods and our proposed method.

1 Introduction

Graphs are crucial in various fields [Hamilton *et al.*, 2017; Song *et al.*, 2023; Xv *et al.*, 2023; Fout *et al.*, 2017; Hamaguchi *et al.*, 2017]. Their growing size offers more information but requires substantial computational resources [Kumar *et al.*, 2023; Xu *et al.*, 2018; Duan *et al.*, 2022]. Efficient processing methods, such as graph coarsening, are essential for processing large-scale graphs [Hashemi *et al.*, 2024].

Graph coarsening aims to simplify graphs while preserving their key properties and improving the efficiency of downstream tasks [Hashemi *et al.*, 2024]. Most existing works have focused on spectrum-preserving [Loukas, 2019; Jin *et al.*, 2020; Chen *et al.*, 2023]. This involves ensuring that the eigenvalues of the original graph closely match those of the coarsened graph. Although spectrum-preserving methods have many advantages, they involve the eigenvalue problem of the Laplacian matrix, which overly relies on the global properties of the graph and may hinder their application in processing large graph data due to high computational costs. Graph coarsening also finds applications in deep learn-

*Corresponding Author

ing [Huang *et al.*, 2021; Pang *et al.*, 2021], with its challenges such as excessive information loss when coarsening is not executed properly, potentially resulting in a decline in model performance. Some works suggest that training Graph Neural Networks (GNNs) is often computationally expensive. A potential solution to alleviate the mentioned problems is dataset condensation or distillation [Jin *et al.*, 2021; Jin *et al.*, 2022]. The goal is to construct a small synthetic set to train the GNN, thereby reducing costs to some extent. However, these studies overlook the granularity characteristics of the graph structure.

The process of graph coarsening does not eliminate any of the original nodes, but instead merges the nodes into supernodes according to some coarsening rules, and aggregates the original edges into superedges. Taking the right half of Figure 1 as an example, the graph typically consists of subregions with different levels of granularity, where nodes within the same subregion show a high degree of connectivity and similarity, and these nodes should be aggregated into one supernode. Initially, the original graph can be viewed as a coarsest structure, and then further refine through finer granularity until the final coarsening results are obtained. Ignoring the multi-granularity characteristics of the graph structure, as shown in the left half of Figure 1, can affect the quality of the coarsened graph and the efficiency of subsequent tasks. By taking advantage of the multi-granularity characteristics of graph structure, the various structural information of the graph can be more fully understood, which can lead to more precise generation of supernodes while improving efficiency.

Actually, it is not easy to coarsen the original graph and model granularity characteristics of the graph structure, which mainly faces two challenges: **(1) How to efficiently and effectively aggregate the nodes in the original graph into the supernodes in the coarsened graph?** Nodes with a high degree of connectivity and similarity should be more inclined to be aggregated into a supernode, which is a key consideration in graph coarsening. Inspired by the advantages of granular-ball computing in distinguishing multi-granularity of discrete data [Xia *et al.*, 2023], and since granular-ball computing has not yet been applied in the field of graphs, we further explore its potential in graph coarsening and propose an adaptive granular-ball graph refinement mechanism. **(2) How to generate coarsening results according to the multi-granularity characteristics of the graph structure?** The graph structure has multiple levels of granularity features, which exhibit unique topological structures at different levels. Using this feature to guide the process of graph coarsening not only improves the efficiency of graph coarsening, but also enhances the model’s ability to understand complex graph structure. Therefore, it is very important to develop a coarsening method that can capture the relationship between the nodes in the original graph and the supernodes in the coarsening graph from a multi-granularity perspective.

To this end, we propose a new multi-granularity, efficient, and adaptive coarsening method via granular-ball. Specifi-

cally, **for challenge (1)**, the original graph is initialized as coarse-grained granular-balls, which are then iteratively split by a fine-grained binary splitting mechanism. In the initial stage, the method only uses global calculation, and the subsequent steps rely on local calculation for iterative splitting, and the computational complexity is low. **For challenge (2)**, according to the multi-granularity characteristics of the graph structure, our proposed method can adaptively use multiple granular-balls of different sizes to generate coarsening results. Extensive experiments conducted on several graph datasets show that our proposed method can outperform the state-of-the-art methods. Our contributions are summarized as follows:

- We emphasize the importance of multi-granularity characteristics of the graph structure in graph coarsening.
- We propose a new graph coarsening method called **granular-ball graph coarsening (GBGC)**, which adaptively uses different sizes of granular-balls to represent supernodes in the coarsened graph. GBGC involves global calculation (calculating the distance between all pairs of nodes in a graph) only in the initialization, thus reducing the time complexity.
- Extensive testing and comparison of datasets confirms the validity and accuracy of GBGC, and we provide a new perspective on graph coarsening methods.

2 Related Work

The spectrum of the Laplacian matrix reflects the graph’s structure, and spectral distance serves as a measure of structural similarity between graphs [Wilson and Zhu, 2008]. Loukas *et al.* introduced “restricted spectral similarity” (RSS), where the eigenvalues and eigenvectors of the coarsened graph approximate those of the original graph within the main feature subspace [Loukas and Vandergheynst, 2018; Loukas, 2019]. Jin *et al.* used spectral distance to evaluate spectral preservation and developed coarsening algorithms to minimize this distance [Jin *et al.*, 2020]. Imre *et al.* proposed a near-linear time-preserving sparse algorithm for large graphs [Imre *et al.*, 2020]. Chen *et al.* applied Gromov-Wasserstein (GW) distance to limit the spectral difference between graphs and their coarsenings, proving an upper bound related to spectral difference [Chen *et al.*, 2023].

Graph coarsening methods have gained popularity in deep learning in recent years. Xie *et al.* proposed MLC-GCN, a multilevel coarsening classification method for GCNs that retains both local and global graph information and learns representations at multiple levels [Xie *et al.*, 2020b]. Fahrbach *et al.* introduced an efficient coarsening method based on the Schur complement, which reduces computational time by embedding related vertices on the coarsened graph [Fahrbach *et al.*, 2020]. Huang *et al.* used graph coarsening for scalable GNN training, reducing the number of graph nodes by up to 10 times without significant accuracy loss [Huang *et al.*, 2021]. Pang *et al.* proposed InfomaxPooling (CGIPool), which maximizes mutual information between each pooling layer’s input and the coarsened graph to preserve graph-level dependencies [Pang *et al.*, 2021].

¹Code is available from <https://anonymous.4open.science/r/GBGC>.
Supplementary materials are available from <https://github.com/Wangwangguanguan/Supplementary-materials.git>.

Training GNNs is computationally expensive, and dataset condensation [Fang *et al.*, 2024] or distillation [Wang *et al.*, 2018] can help reduce costs by creating smaller synthetic graph datasets for training. Zhao *et al.* proposed a condensation method called DC, which matches network parameter gradients between a small synthetic and a large real dataset [Zhao *et al.*, 2020]. Jin *et al.* introduced GCOND, a graph compression method that reduces nodes by compressing both node features and structure in a supervised way [Jin *et al.*, 2021]. They also proposed DosCond, a framework that uses gradient matching to effectively compress real graphs into smaller, discrete information graphs [Jin *et al.*, 2022].

Above methods have significantly enhanced the efficiency of various downstream tasks, making them a focal point of ongoing research and development. Despite these advancements, many of the existing approaches tend to overlook the inherent granularity of the graph structure. This oversight can limit their effectiveness in capturing the complex relationships between nodes. In contrast, our work addresses this gap by explicitly considering the granularity characteristics of the graph. We propose a novel approach that captures and utilizes the relationships between nodes from a multi-granularity perspective, enabling more effective graph coarsening.

3 Methodology

3.1 Notations and problem statement

We denote an original graph as $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, whose node set is \mathcal{V} with size $|\mathcal{V}| = N$, and whose edge set is \mathcal{E} with size $|\mathcal{E}| = E$. The adjacency matrix of \mathcal{G} is denoted as $\mathbf{A} \in \{0, 1\}^{N \times N}$, and $\mathbf{A}_{ij} = 1$ if there is an edge between nodes v_i and v_j . The Laplacian matrix is denoted as $\mathbf{L} = \mathbf{D} - \mathbf{A}$, where \mathbf{D} is the degree matrix with $\mathbf{D}_{ii} = \sum_j \mathbf{A}_{ij}$. The normalized Laplacian matrix is $\hat{\mathbf{L}} = \mathbf{D}^{-\frac{1}{2}} \mathbf{L} \mathbf{D}^{-\frac{1}{2}}$. Given an original graph \mathcal{G} , graph coarsening methods aim to find a smaller coarsened graph denoted as $\bar{\mathcal{G}} = (\bar{\mathcal{V}}, \bar{\mathcal{E}})$ to approximate \mathcal{G} . The number of nodes in $\bar{\mathcal{G}}$ is denoted as \bar{N} , the number of edges is denoted as \bar{E} , and $\bar{N} < N$, $\bar{E} < E$. The adjacency matrix of $\bar{\mathcal{G}}$ is denoted as $\bar{\mathbf{A}} \in \{0, 1\}^{\bar{N} \times \bar{N}}$, and $\bar{\mathbf{A}}_{ij} = 1$ if there is an edge between supernodes \bar{v}_i and \bar{v}_j . In the matrix form, we use $\mathbf{C} \in \mathbb{R}^{N \times \bar{N}}$ to denote a projection matrix that maps nodes in \mathcal{V} to supernodes in $\bar{\mathcal{V}}$, and $\mathbf{C}_{ij} = 1$ if node i in \mathcal{V} is mapped to supernode j in $\bar{\mathcal{V}}$. The Laplacian matrix of the coarsened graph can be denoted as $\bar{\mathbf{L}} = \mathbf{C}^T \mathbf{L} \mathbf{C}$. This computation ensures that the spectral properties of the original graph are preserved in the coarsened graph.

Our proposed GBGC seeks to obtain a coarsened graph $\bar{\mathcal{G}}$ to approximate \mathcal{G} . The original graph \mathcal{G} is refined into multiple non-overlapping subdomains as follows:

$$\mathcal{G} = \{\tilde{\mathcal{G}}_1, \tilde{\mathcal{G}}_2, \dots, \tilde{\mathcal{G}}_t\} (t = \bar{N}), \quad (1)$$

$$\tilde{\mathcal{G}}_i = (\tilde{\mathcal{V}}_i, \tilde{\mathcal{E}}_i), i = 1, 2, \dots, t, \quad (2)$$

where $\tilde{\mathcal{V}}_i$ and $\tilde{\mathcal{E}}_i$ represent the set of nodes and edges of the graph $\tilde{\mathcal{G}}_i$ contained inside the granular-ball \mathcal{GB}_i , with $\tilde{\mathcal{V}}_i \subset \mathcal{V}$ being a subset of nodes from \mathcal{G} . \bar{N} is the number of supernodes in $\bar{\mathcal{G}}$, and each \mathcal{GB} corresponds to a supernode in $\bar{\mathcal{G}}$. In

the process of graph coarsening implementation, the information loss between the original graph and the coarsened graph should be minimized, the number of granular-balls should be as few as possible, and the adaptive quality requirements should be met.

3.2 Model introduction

Figure 2 illustrates the overall framework of GBGC, which consists of two stages: (I) *the process of granular-balls formation*; (II) *the graph coarsening results*.

(I) The process of granular-balls formation

This stage consists of four steps. **Step 1** is based on the law of “global precedence” [Chen, 1982], the original graph as the coarsest granular-ball. **Step 2** generates \sqrt{N} initial granular-balls. Mathematically,

$$\mathcal{GB}_{init} = \{\mathcal{GB}_1, \mathcal{GB}_2, \dots, \mathcal{GB}_s\} (s = \sqrt{N}). \quad (3)$$

The value of \sqrt{N} is an empirical choice, referenced from previous work [Xie *et al.*, 2020a; Yu and Cheng, 2001], which has been shown to balance the trade-off between size and quantity of granular-balls.

The procedure for generating \mathcal{GB}_{init} is as follows. Firstly, we select the node with the highest degree $v_{center} = \arg \max_{v \in \mathcal{V}} \deg(v)$ as the initial center. Subsequently, breadth-first search (BFS) starts with the v_{center} , traversing layer by layer until the number of nodes at layer i exceeds \sqrt{N} :

$$|\mathcal{N}_i| > \sqrt{N}, \quad (4)$$

where $|\mathcal{N}_i|$ denotes the number of nodes at layer i . Then delete these nodes from \mathcal{V} :

$$\mathcal{V}' = \mathcal{V} \setminus \mathcal{N}_i. \quad (5)$$

Continue to find the next center with the highest degree $v'_{center} = \arg \max_{v \in \mathcal{V}'} \deg(v)$ in \mathcal{V}' , and so on through the entire graph to determine \mathcal{GB}_{init} .

All nodes participate in the calculation, so global calculation is involved in this step. This approach not only accelerates subsequent granular-ball splitting processes but also provides an effective starting point for granular-ball coarsening in graphs with a small number of nodes. **Step 3** is to calculate the quality of each granular-ball, which is defined as follows.

Definition 1 For a granular-ball \mathcal{GB} , its quality is formulated as follows:

$$quality(\mathcal{GB}) = \frac{\tilde{E}}{\tilde{N}} + \frac{\sum_{\tilde{i} \neq \tilde{j} \neq \tilde{k}} \tilde{\mathbf{A}}_{\tilde{i}\tilde{j}} \tilde{\mathbf{A}}_{\tilde{j}\tilde{k}} \tilde{\mathbf{A}}_{\tilde{k}\tilde{i}}}{\sum_{\tilde{i} \neq \tilde{j} \neq \tilde{k}} \tilde{\mathbf{A}}_{\tilde{i}\tilde{j}} \tilde{\mathbf{A}}_{\tilde{j}\tilde{k}}}, \quad (6)$$

where \tilde{E} and \tilde{N} represent the number of edges and nodes within the graph contained inside \mathcal{GB} , respectively. $\tilde{\mathbf{A}}_{mn}$ indicates the presence ($\tilde{\mathbf{A}}_{mn} = 1$) or absence ($\tilde{\mathbf{A}}_{mn} = 0$) of an edge between nodes \tilde{m} and \tilde{n} ($\tilde{m}, \tilde{n} = \tilde{i}, \tilde{j}, \tilde{k}$) inside \mathcal{GB} .

Quality is composed of two components: the average degree and the global clustering coefficient (also known as the transitivity). The first component quantifies the average number of edges connected to each node in the graph, serving as

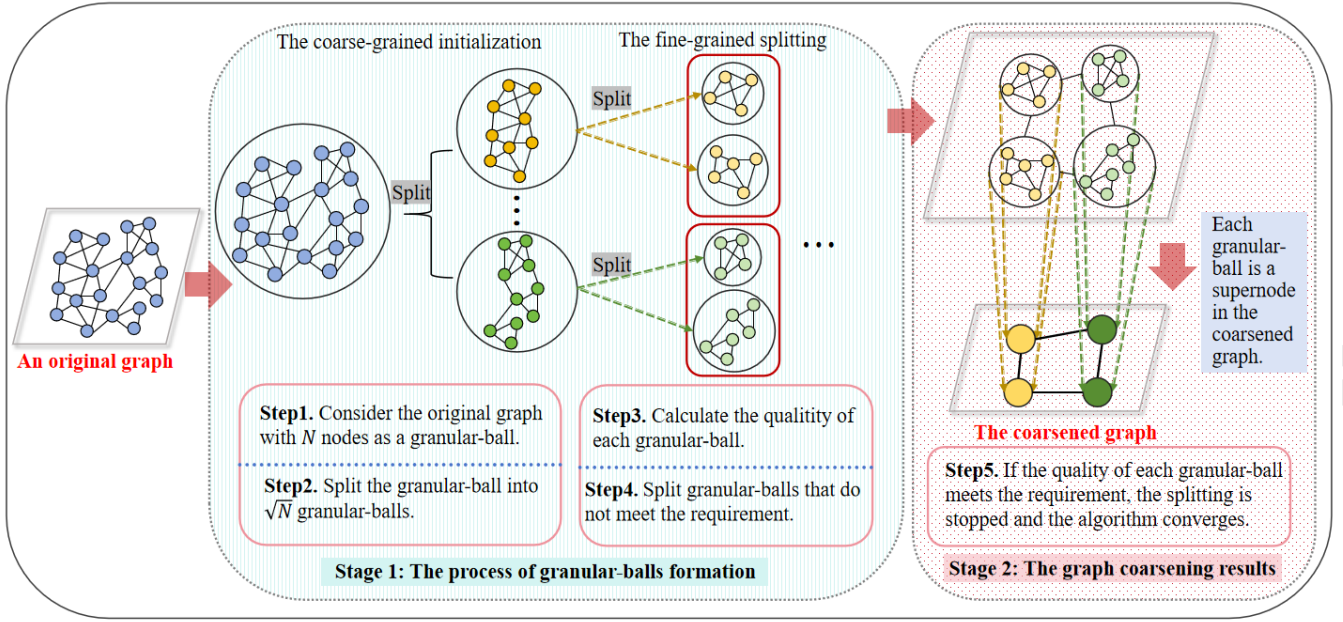


Figure 2: The framework of our method. GBGC consists of two main stages. As shown in this figure, the first stage is the process of granular-balls formation, which consists of four steps. Specifically, Steps 1 and 2 depict the initialization of coarse-grained granular-balls, while Steps 3 and 4 demonstrate the fine-grained binary splitting process. The second stage consists of Step 5, where GBGC generates the final coarsening results.

an intuitive measure of the connectivity within the graph. In the second component, $\sum_{\tilde{i} \neq \tilde{j} \neq \tilde{k}} \tilde{\mathbf{A}}_{ij} \tilde{\mathbf{A}}_{jk} \tilde{\mathbf{A}}_{ki}$, represents the number of actual triangles present in the graph. In contrast, $\sum_{\tilde{i} \neq \tilde{j} \neq \tilde{k}} \tilde{\mathbf{A}}_{ij} \tilde{\mathbf{A}}_{ik}$ denotes the total number of possible triples in the graph, where a triple is defined as a central node connected to two other nodes, regardless of whether the two peripheral nodes are directly connected. The second component measures the local clustering of nodes within the graph, and its computation relies on the number of triangles and triples.

The higher the quality of the granular-ball, the more tightly connected the nodes inside the granular-ball. This method of quantifying the quality of the granular-ball ensures that the structural properties of the graph are considered and balanced in subsequent splitting of the granular-ball.

Step 4 is to perform a fine-grained binary splitting for each granular-ball \mathcal{GB}_j in \mathcal{GB}_{init} . The fine-grained splitting identifies the two nodes with the highest degree of parent granular-ball (\mathcal{GB}_j) as the splitting centers of the two child granular-balls (\mathcal{GB}_{jA} and \mathcal{GB}_{jB}), which are denoted as

$$v_A = \arg \max_{v \in \tilde{\mathcal{V}}_j} \deg(v), v_B = \arg \max_{v \in \tilde{\mathcal{V}}_j \setminus v_A} \deg(v), \quad (7)$$

where $\tilde{\mathcal{V}}_j$ represents the set of nodes within the granular-ball \mathcal{GB}_j . Each center node traverses the remaining nodes layer by layer through BFS, and other non-center nodes are assigned to the node according to who was first traversed by a center node's BFS. Only the nodes within granular-balls are involved in the calculation, so Steps 3 and 4 involve local calculation. The adaptive splitting condition of parent granular-ball splitting is given by the following formula:

$$quality(\mathcal{GB}_{jA} + \mathcal{GB}_{jB}) > quality(\mathcal{GB}_j), \quad (8)$$

where $quality(\mathcal{GB}_{jA} + \mathcal{GB}_{jB})$ denotes the sum of the qualities of the two child granular-balls, and $quality(\mathcal{GB}_j)$ denotes the quality of the parent granular-ball. This formula is employed to decide whether to proceed with splitting granular-balls. If there is a notable difference in quality between the parent granular-ball and its child granular-balls, suggesting that further splitting improves the overall system performance, the splitting operation will continue. This strategy ensures both the precision and efficiency of the splitting process, preventing unnecessary computational complexity from arising.

(II) The graph coarsening results

This stage consists of **Step 5**, which is to generate the coarsening results. In the process of graph coarsening, GBGC splits the nodes of the original graph from coarse-grained to fine-grained, and generates the supernodes of the coarsened graph. Therefore, each granular-ball \mathcal{GB} corresponds to a supernode in the coarsened graph $\bar{\mathcal{G}}$. Mathematically,

$$\bar{\mathcal{V}} = \{\mathcal{GB}_1, \mathcal{GB}_2, \dots, \mathcal{GB}_t\} (t = \bar{N}), \quad (9)$$

which covers all nodes in \mathcal{G} . If there is already an edge between the nodes contained within two different granular-balls, then an edge is created between the two granular-balls. The process can be represented as follows:

$$\bar{\mathcal{E}} = \{(\mathcal{GB}_i, \mathcal{GB}_j) | \exists e = (u, v) \in \mathcal{E}, u \in \tilde{\mathcal{V}}_i, v \in \tilde{\mathcal{V}}_j, \mathcal{GB}_i \neq \mathcal{GB}_j\}. \quad (10)$$

This methodology enhances the representation of graph by focusing on broader connectivity patterns, while maintaining the critical properties of the graph.

3.3 Theoretical analysis of GBGC

Due to the difference in the number of nodes between the original graph \mathcal{G} and the coarsened graph $\bar{\mathcal{G}}$, we introduce appropriate intrinsic functionals such as the Rayleigh quotient (which is a quantity related to the structure of the graph and does not vary with the arrangement of nodes). The projection matrix \mathbf{C} , where $\mathbf{C}_{ij} = 1$ if v_i belongs to \mathcal{GB}_j , ensures that the spectral information of the two graphs becomes comparable [Cai *et al.*, 2021]. Therefore, we first give the definitions of the Rayleigh quotient of the original graph and the coarsened graph respectively. Then the coarsening effect of GBGC is proved by the Rayleigh quotient.

Definition 2 For an original graph \mathcal{G} , its Rayleigh quotient is defined as:

$$R_o(\mathbf{L}, \mathbf{x}) = \frac{\mathbf{x}^T \mathbf{L} \mathbf{x}}{\mathbf{x}^T \mathbf{x}}, \quad (11)$$

where \mathbf{L} is the Laplace matrix of \mathcal{G} , \mathbf{x} is the eigenvector of \mathbf{L} . According to the definition of the projection matrix \mathbf{C} in Section 3.1, the Rayleigh quotient of the coarsened graph obtained by GBGC is given, which is defined as follows:

Definition 3 For the coarsened graph $\bar{\mathcal{G}}$, its Rayleigh quotient is defined as:

$$R_c(\bar{\mathbf{L}}, \bar{\mathbf{x}}) = \frac{(\mathbf{C}^T \mathbf{x})^T \bar{\mathbf{L}} (\mathbf{C}^T \mathbf{x})}{(\mathbf{C}^T \mathbf{x})^T (\mathbf{C}^T \mathbf{x})}, \quad (12)$$

where $\bar{\mathbf{L}}$ is the Laplace matrix of $\bar{\mathcal{G}}$.

Theorem 1 Assuming \mathbf{C} is a generalized orthogonal matrix (i.e. $\mathbf{C}^T \mathbf{C} \approx \mathbf{I}$, where \mathbf{I} is the identity matrix), we have:

$$R_c(\bar{\mathbf{L}}, \bar{\mathbf{x}}) \approx R_o(\mathbf{L}, \mathbf{x}). \quad (13)$$

Proof. Considering the assumption that \mathbf{C} is a generalized orthogonal matrix. For the denominator of $R_c(\bar{\mathbf{L}}, \bar{\mathbf{x}})$, using the orthogonality of \mathbf{C} , we get $(\mathbf{C}^T \mathbf{x})^T (\mathbf{C}^T \mathbf{x}) \approx \mathbf{x}^T \mathbf{C} \mathbf{C}^T \mathbf{x} = \mathbf{x}^T \mathbf{I} \mathbf{x} = \mathbf{x}^T \mathbf{x}$. And then, for the numerator of $R_c(\bar{\mathbf{L}}, \bar{\mathbf{x}})$, we have $(\mathbf{C}^T \mathbf{x})^T \bar{\mathbf{L}} (\mathbf{C}^T \mathbf{x}) \approx \mathbf{x}^T \mathbf{C} \bar{\mathbf{L}} \mathbf{C}^T \mathbf{x}$. According to $\bar{\mathbf{L}} = \mathbf{C}^T \mathbf{L} \mathbf{C}$, we obtain $\mathbf{C} \bar{\mathbf{L}} \mathbf{C}^T \approx \mathbf{L}$. Thus, $\mathbf{x}^T \mathbf{C} \bar{\mathbf{L}} \mathbf{C}^T \mathbf{x} \approx \mathbf{x}^T \mathbf{L} \mathbf{x}$. Finally, using these approximations, the Rayleigh quotient of the coarsened graph obtained by GBGC can be approximated as $R_c(\bar{\mathbf{L}}, \bar{\mathbf{x}}) \approx R_o(\mathbf{L}, \mathbf{x})$. \square

4 Experiments

In this section, we design experiments to evaluate our GBGC and answer the following research questions (RQs): **RQ1:** Does our method achieve better performance compared to the state-of-the-art methods? **RQ2:** How does GBGC perform with respect to time consume? **RQ3:** What impact do the key components have on GBGC’s performance? **RQ4:** How do hyperparameters effect GBGC’s performance? **RQ5:** How does GBGC perform in visualization?

Dataset	Size	$ \mathcal{V} $	$ \mathcal{E} $
MUTAG [Debnath <i>et al.</i> , 1991]	188	17.93	19.79
PROTEINS [Borgwardt <i>et al.</i> , 2005]	1113	39.06	72.82
IMDB-BINARY [Cai and Wang, 2018]	1000	19.77	96.53
NCI109 [Morris <i>et al.</i> , 2020b]	4127	29.68	32.13
DHFR [Sutherland <i>et al.</i> , 2003]	756	42.43	44.54
BZR [Vincent-Cuaz <i>et al.</i> , 2021]	405	35.75	38.36
Tox21_ARLBDtesting [Cooper and Schürer, 2019]	253	21.85	22.73
OVCAR-8H [Morris <i>et al.</i> , 2020a]	39253	46.67	48.70
P388H [Morris <i>et al.</i> , 2020a]	40651	40.45	41.89
SF-295H [Morris <i>et al.</i> , 2020a]	39030	46.65	48.68
DD [Morris <i>et al.</i> , 2020a]	1178	284.32	715.66

Table 1: Summary of dataset information. $|\mathcal{V}|$ and $|\mathcal{E}|$ denote the average number of nodes and edges, respectively. All graphs are undirected in this paper.

4.1 Experimental setup

Datasets: We use several standard graph classification datasets, which are shown in Table 1. See Appendix for implementation details.

Baselines: We compare with six baseline methods also for graph-level tasks: Variation Neighborhood Graph Coarsening (VNGC) [Loukas, 2019]; Variation Edge Graph Coarsening (VEGC) [Loukas, 2019]; Multilevel Graph Coarsening (MGC) [Jin *et al.*, 2020]; Spectral Graph Coarsening (SGC) [Jin *et al.*, 2020]; Dataset One-Step Condensation (DosCond) [Jin *et al.*, 2022]; Kernel Graph Coarsening (KGC) [Chen *et al.*, 2023].

Evaluation index of coarsening effect: In this paper, we mainly use accuracy (ACC) and spectral distance (SD) to measure the effect of coarsening of GBGC. According to the reference [Wilson and Zhu, 2008], since the spectrum of the Laplacian distills information about the graph structure, one may interpret the spectral distance as a proxy for the structural similarity of the two graphs. The spectral distance between the original graph and the coarsened graph is defined by calculating the Euclidean distance between the eigenvalue vectors of the two graphs, i.e., $SD(\mathcal{G}, \bar{\mathcal{G}}) = \sqrt{\sum_i (\lambda_i^{(\mathcal{G})} - \lambda_i^{(\bar{\mathcal{G}})})^2}$, where $\lambda_i^{(\mathcal{G})}$ and $\lambda_i^{(\bar{\mathcal{G}})}$ are the i -th eigenvalues of the Laplace matrix of the original and coarsened graphs, respectively. When the spectra have different sizes, then the smaller one can be padded with zeros (while maintaining the magnitude ordering). This is equivalent to adding disjoint nodes to the smaller graph to make both graphs have the same number of nodes.

4.2 Performance of coarsening method for graph classification (RQ1)

Graph classification is a key data mining task, however, the complexity and diversity of large-scale graph data bring great challenges to classification. Graph coarsening can reduce graph size, speeding up training and inference for graph classification models. However, improper coarsening may cause information loss, reducing accuracy. Thus, we evaluate various coarsening methods by comparing their performance with the same classifier applied to the original graph.

Our GBGC is free of randomness. To evaluate comparison models thoroughly, we used multiple random seeds and applied cross-validation with repeated runs. The DosCond method, which doesn’t use a ratio for computation, averages

Datasets	MUTAG	PROTEINS	IMDB-BINARY	NCI109	DHFR	BZR	Tox21_AR-LBD	OVCAR-8H	P388H	SF-295H	DD
r_a	0.35	0.24	0.19	0.38	0.35	0.40	0.34	0.38	0.37	0.38	0.20
VNGC	77.75 ± 2.15	63.45 ± 0.92	49.60 ± 0.78	61.68 ± 0.22	62.03 ± 1.66	77.74 ± 1.45	96.60 ± 0.88	91.44 ± 0.07	91.35 ± 0.08	91.79 ± 0.06	65.70 ± 1.25
VEGC	81.93 ± 2.24	63.90 ± 1.00	61.50 ± 1.88	62.03 ± 0.40	68.12 ± 1.04	78.52 ± 2.35	94.93 ± 1.23	91.88 ± 0.08	91.86 ± 0.12	92.09 ± 0.11	69.19 ± 1.28
MGC	84.56 ± 1.75	65.88 ± 1.28	65.70 ± 1.18	65.09 ± 0.72	71.42 ± 1.63	76.06 ± 2.01	95.49 ± 1.56	91.80 ± 0.11	91.51 ± 0.13	91.93 ± 0.07	OOM
SGC	83.54 ± 2.27	64.81 ± 1.58	<u>66.60 ± 1.29</u>	62.95 ± 0.59	63.21 ± 1.83	78.04 ± 0.93	95.46 ± 0.72	91.45 ± 0.14	91.24 ± 0.15	91.59 ± 0.06	67.39 ± 1.26
DosCond	70.00 ± 1.68	63.89 ± 1.28	59.17 ± 0.21	56.75 ± 1.08	71.66 ± 0.70	66.50 ± 1.29	87.43 ± 0.74	92.11 ± 1.35	92.93 ± 1.05	85.49 ± 1.03	70.33 ± 1.83
KGC	77.60 ± 2.01	64.62 ± 1.32	63.10 ± 1.40	57.75 ± 0.70	64.28 ± 1.81	78.04 ± 1.96	94.90 ± 0.95	91.31 ± 0.12	90.91 ± 0.13	91.44 ± 0.08	OOM
GBGC	90.94 ± 2.12	65.34 ± 0.98	66.10 ± 1.15	66.53 ± 0.72	72.62 ± 1.69	80.01 ± 0.80	97.19 ± 1.19	92.53 ± 0.06	<u>92.33 ± 0.11</u>	92.63 ± 0.09	69.44 ± 1.65
FULL	85.12 ± 2.44	66.61 ± 0.99	69.30 ± 1.34	64.91 ± 0.81	71.42 ± 1.26	79.78 ± 2.05	94.31 ± 0.79	<u>92.35 ± 0.14</u>	92.28 ± 0.14	92.54 ± 0.08	66.72 ± 1.01

Table 2: Comparison of classification performance with the baseline methods, FULL indicates the performance of the dataset without coarsening. The best results are bolded, the second results are underlined, and “OOM” means out of memory.

Datasets	MUTAG	PROTEINS	IMDB-BINARY	NCI109	DHFR	BZR	Tox21_AR-LBD	OVCAR-8H	P388H	SF-295H	DD
VNGC	3.40	41.10	<u>18.31</u>	109.37	33.66	15.87	4.00	2005.11	2065.66	2111.72	427.02
VEGC	<u>2.80</u>	<u>25.00</u>	19.85	<u>58.92</u>	<u>14.53</u>	<u>6.95</u>	<u>2.57</u>	<u>946.53</u>	<u>1020.55</u>	<u>973.16</u>	<u>359.26</u>
MGC	3.36	2227.51	38.90	344.16	138.92	43.07	6.14	15661.21	11146.01	15837.83	OOM
SGC	27.04	1336.21	41.67	1375.79	443.26	206.77	32.83	38465.95	27621.03	38165.46	163677.65
KGC	7.13	81.38	23.97	208.59	55.03	27.03	6.98	3517.12	2934.62	3256.30	OOM
GBGC	0.44	11.70	3.63	17.74	9.37	2.43	0.96	350.46	281.59	333.02	44.61

Table 3: Running time comparison table (Unit: Second). The best results are bolded, the second results are underlined, and “OOM” means out of memory.

results from ten repeated experiments to ensure more reliable accuracy metrics. This reduces the impact of chance errors on final results, making them more robust. To ensure a fair comparison, the identical coarsening parameter r_a , as determined by the adaptive coarsening process of GBGC, is utilized across the VNGC, VEGC, MGC, SGC, and KGC. The performance results are shown in Table 2. From Table 2, we have the following observations:

- **It is evident that GBGC achieves higher graph classification accuracy compared to baseline methods on most datasets.** GBGC performs better on most datasets than comparison baseline methods and FULL. For the P388H dataset, the accuracy of GBGC was not significantly different from that of DosCond. However, GBGC consistently excels in graph classification, especially in maintaining stable performance across multiple datasets. This indicates that GBGC better preserves the original graph’s properties while reducing its size, effectively capturing structural information and improving classification accuracy.
- **GBGC demonstrates high accuracy in graph classification after coarsening the original graph.** GBGC outperforms the full dataset in classification accuracy on most datasets. This is due to the coarsened graph reducing noise and the granular-ball effectively capturing data distribution. Similar to other work related to granular-ball computing, GBGC is robust and generalizable. It uses multi-granularity representation and granular-ball splitting to generate supernodes, preserving key graph structure information for better classification. However, GBGC performs worse than FULL on IMDB-BINARY and PROTEINS datasets, likely because these datasets have better distribution and less noise, reducing the benefits of coarsening.

We refer to references [Loukas, 2019] and [Wilson and Zhu, 2008] and use SD to evaluate the performance of the

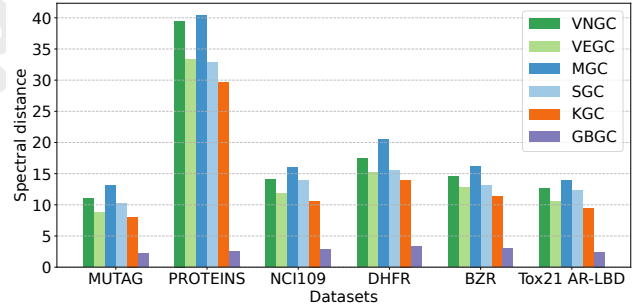


Figure 3: Comparison diagram of SD .

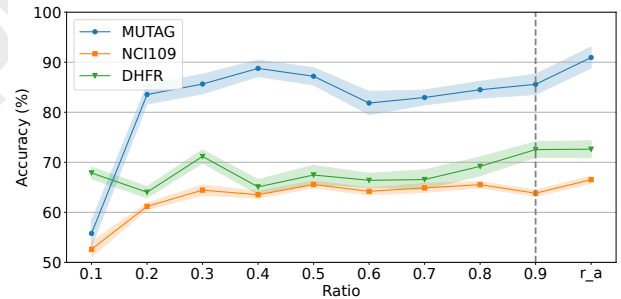


Figure 4: Comparison of different coarsening ratios and adaptive coarsening ratio (r_a) of GBGC.

coarsening methods. Figure 3 shows an intuitive comparison of SD . It can be seen that the spectral distance of the GBGC is better than that of comparison methods.

4.3 How does GBGC perform with respect to time consume?(RQ2)

This section evaluates the efficiency of GBGC and other coarsening methods. Since DosCond’s runtime depends on

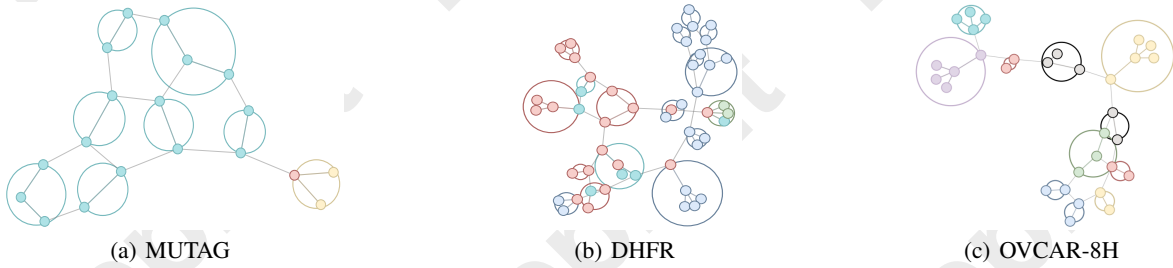


Figure 5: Visualization of GBGC in different datasets.

GNN training, we exclude it from the comparison for fairness and focus on the runtime of GBGC and the other five methods. The runtime comparison is shown in Table 3. From the results in Table 3, we observe that:

- **The running time of GBGC is tens to hundreds of times faster than comparison methods, which is significantly lower than other methods.** Especially on the P388H dataset, the advantages of GBGC are even more significant.
- **The high efficiency of GBGC can be attributed to its unique granular-ball generation and splitting strategies.** These strategies use global computation for coarse-grained granular-ball initialization and local computation for fine-grained binary splitting, ensuring low time complexity and high efficiency. Importantly, GBGC maintains accuracy while achieving efficiency. By using granular-balls of varying sizes, GBGC effectively captures the original data, often outperforming the FULL in classification accuracy.

4.4 Ablation studies (RQ3)

In this section, we perform ablation experiments on several datasets to examine GBGC, including **GBGC-w/o fine-grained binary splitting** and **GBGC-w/o coarse-grained initialization**. The results are shown in Table 4. We have the following observations:

- Removing the fine-grained binary splitting mechanism in GBGC significantly reduces the model’s performance, emphasizing its importance. This mechanism is crucial for capturing the graph’s inherent granularity and preserving node relationships. Without this mechanism, the model struggles to effectively represent the multi-granularity features of the graph, underscoring its necessity for achieving optimal performance in tasks that rely on precise graph coarsening and representation.
- Removing the coarse-grained initialization from GBGC leads to a significant performance drop, highlighting its crucial role in the model. It provides an initial representation that captures the graph’s broader structure, enabling the model to handle higher-level node relationships. Its absence highlights the substantial contribution it makes to the model’s ability to process and represent the graph effectively, making it an indispensable component to achieve strong performance.

Datasets	MUTAG	PROTEINS	NCI109	DHFR	DD	Avg
GBGC	90.94	65.34	66.53	72.62	69.44	72.97
-w/o binary splitting	89.36	63.42	63.44	72.56	68.85	71.53
-w/o initialization	79.21	63.90	63.55	70.09	67.66	68.88

Table 4: Results of the ablation studies.

4.5 Hyper-parameter sensitivity analysis (RQ4)

In this section, we design a non-adaptive GBGC (see Algorithm 4 in Appendix for algorithm design) to evaluate the performance of different r . Following the experimental settings of [Chen *et al.*, 2023], we studied the effect of different $r \in (0, 1)$ with a step size of 0.1 on GBGC performance, as shown in Figure 4. The model performs best when GBGC adaptively generates the coarsening rate, r_a . This is because GBGC uses a mix of large granular-balls for simple data distributions and many smaller ones for complex distributions, effectively representing the data and improving accuracy.

4.6 Case study (RQ5)

GBGC retains essential graph structure information, enabling classifiers to use it more efficiently for accurate graph classification. This section focuses on the MUTAG, DHFR, and OVCA8-8H datasets. For MUTAG, GBGC identifies molecular structures related to toxicity; for DHFR, it captures correlations between molecular fragments and drug activity; and for OVCA8-8H, it highlights key molecular features influencing drug response, aiding drug screening and personalized treatment prediction. Figure 5 illustrates GBGC’s analysis, where colored circles represent granular-balls (supernodes), showing its ability to capture node correlations.

5 Conclusion

In this work, we propose a new multi-granularity, efficient, and adaptive coarsening method via granular-ball(GBGC). The experimental results show that GBGC has higher accuracy compared with several state-of-the-art methods. At the same time, the speed of GBGC is increased by tens to hundreds of times, showing a low time complexity $O(N^{\frac{3}{2}} + E\sqrt{N})$. Additionally, an intriguing observation is that the accuracy of GBGC is almost consistently higher than that achieved on the original datasets, attributable to its robustness and generalizability. While our current approach focuses on graph classification tasks, other downstream tasks can be explored in the future to further broaden the applicability of our coarsening approach.

Acknowledgments

This work was supported in part by the National Natural Science Foundation of China under Grant Nos. 62221005, 62450043, 62222601, and 62176033.

References

- [Borgwardt *et al.*, 2005] Karsten M Borgwardt, Cheng Soon Ong, Stefan Schöner, SVN Vishwanathan, Alex J Smola, and Hans-Peter Kriegel. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl_1):i47–i56, 2005.
- [Cai and Wang, 2018] Chen Cai and Yusu Wang. A simple yet effective baseline for non-attributed graph classification. *arXiv preprint arXiv:1811.03508*, 2018.
- [Cai *et al.*, 2021] Chen Cai, Dingkan Wang, and Yusu Wang. Graph coarsening with neural networks. *arXiv preprint arXiv:2102.01350*, 2021.
- [Chen *et al.*, 2023] Yifan Chen, Rentian Yao, Yun Yang, and Jie Chen. A gromov–wasserstein geometric view of spectrum-preserving graph coarsening. pages 5257–5281, 2023.
- [Chen, 1982] L. Chen. Topological structure in visual perception. *Science*, page 699–700, Nov 1982.
- [Cooper and Schürer, 2019] Daniel J Cooper and Stephan Schürer. Improving the utility of the tox21 dataset by deep metadata annotations and constructing reusable benchmarked chemical reference signatures. *Molecules*, 24(8):1604, 2019.
- [Debnath *et al.*, 1991] Asim Kumar Debnath, Rosa L Lopez de Compadre, Gargi Debnath, Alan J Shusterman, and Corwin Hansch. Structure-activity relationship of mutagenic aromatic and heteroaromatic nitro compounds. correlation with molecular orbital energies and hydrophobicity. *Journal of medicinal chemistry*, 34(2):786–797, 1991.
- [Duan *et al.*, 2022] Keyu Duan, Zirui Liu, Peihao Wang, Wenqing Zheng, Kaixiong Zhou, Tianlong Chen, Xia Hu, and Zhangyang Wang. A comprehensive study on large-scale graph training: Benchmarking and rethinking. *Advances in Neural Information Processing Systems*, 35:5376–5389, 2022.
- [Fahrbach *et al.*, 2020] Matthew Fahrbach, Gramoz Goranci, Richard Peng, Sushant Sachdeva, and Chi Wang. Faster graph embeddings via coarsening. In *international conference on machine learning*, pages 2953–2963. PMLR, 2020.
- [Fang *et al.*, 2024] Junfeng Fang, Xinglin Li, Yongduo Sui, Yuan Gao, Guibin Zhang, Kun Wang, Xiang Wang, and Xiangnan He. Exgc: Bridging efficiency and explainability in graph condensation. In *Proceedings of the ACM on Web Conference 2024*, pages 721–732, 2024.
- [Fout *et al.*, 2017] Alex Fout, Jonathon Byrd, Basir Shariat, and Asa Ben-Hur. Protein interface prediction using graph convolutional networks. *Advances in neural information processing systems*, 30, 2017.
- [Hamaguchi *et al.*, 2017] Takuo Hamaguchi, Hidekazu Oiwa, Masashi Shimbo, and Yuji Matsumoto. Knowledge transfer for out-of-knowledge-base entities: A graph neural network approach. *arXiv preprint arXiv:1706.05674*, 2017.
- [Hamilton *et al.*, 2017] William L Hamilton, Rex Ying, and Jure Leskovec. Representation learning on graphs: Methods and applications. *arXiv preprint arXiv:1709.05584*, 2017.
- [Hashemi *et al.*, 2024] Mohammad Hashemi, Shengbo Gong, Juntong Ni, Wenqi Fan, B Aditya Prakash, and Wei Jin. A comprehensive survey on graph reduction: Sparsification, coarsening, and condensation. *arXiv preprint arXiv:2402.03358*, 2024.
- [Huang *et al.*, 2021] Zengfeng Huang, Shengzhong Zhang, Chong Xi, Tang Liu, and Min Zhou. Scaling up graph neural networks via graph coarsening. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, Aug 2021.
- [Imre *et al.*, 2020] Martin Imre, Jun Tao, Yongyu Wang, Zhiqiang Zhao, Zhuo Feng, and Chaoli Wang. Spectrum-preserving sparsification for visualization of big graphs. *Computers & Graphics*, 87:89–102, 2020.
- [Jin *et al.*, 2020] Yu Jin, Andreas Loukas, and Joseph JaJa. Graph coarsening with preserved spectral properties. In *International Conference on Artificial Intelligence and Statistics*, pages 4452–4462. PMLR, 2020.
- [Jin *et al.*, 2021] Wei Jin, Lingxiao Zhao, Shichang Zhang, Yozen Liu, Jiliang Tang, and Neil Shah. Graph condensation for graph neural networks. *arXiv preprint arXiv:2110.07580*, 2021.
- [Jin *et al.*, 2022] Wei Jin, Xianfeng Tang, Haoming Jiang, Zheng Li, Danqing Zhang, Jiliang Tang, and Bing Yin. Condensing graphs via one-step gradient matching. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 720–730, 2022.
- [Kumar *et al.*, 2023] Manoj Kumar, Anurag Sharma, Shashwat Saxena, and Sandeep Kumar. Featured graph coarsening with similarity guarantees. pages 17953–17975, 2023.
- [Loukas and Vandergheynst, 2018] Andreas Loukas and Pierre Vandergheynst. Spectrally approximating large graphs with smaller graphs. In *International Conference on Machine Learning*, pages 3237–3246. PMLR, 2018.
- [Loukas, 2019] Andreas Loukas. Graph reduction with spectral and cut guarantees. *J. Mach. Learn. Res.*, 20(116):1–42, 2019.
- [Morris *et al.*, 2020a] Christopher Morris, Nils M Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. *arXiv preprint arXiv:2007.08663*, 2020.
- [Morris *et al.*, 2020b] Christopher Morris, Gaurav Rattan, and Petra Mutzel. Weisfeiler and leman go sparse: Towards scalable higher-order graph embeddings. *Advances*

- in *Neural Information Processing Systems*, 33:21824–21840, 2020.
- [Pang *et al.*, 2021] Yunsheng Pang, Yunxiang Zhao, and Dongsheng Li. Graph pooling via coarsened graph info-max. In *Proceedings of the 44th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2177–2181, 2021.
- [Song *et al.*, 2023] Xiran Song, Jianxun Lian, Hong Huang, Zihan Luo, Wei Zhou, Xue Lin, Mingqi Wu, Chaozhuo Li, Xing Xie, and Hai Jin. xgc: An extreme graph convolutional network for large-scale social link prediction. In *Proceedings of the ACM Web Conference 2023*, pages 349–359, 2023.
- [Sutherland *et al.*, 2003] Jeffrey J Sutherland, Lee A O’Brien, and Donald F Weaver. Spline-fitting with a genetic algorithm: A method for developing classification structure- activity relationships. *Journal of chemical information and computer sciences*, 43(6):1906–1915, 2003.
- [Vincent-Cuaz *et al.*, 2021] Cédric Vincent-Cuaz, Titouan Vayer, Rémi Flamary, Marco Corneli, and Nicolas Courty. Online graph dictionary learning. In *International conference on machine learning*, pages 10564–10574. PMLR, 2021.
- [Wang *et al.*, 2018] Tongzhou Wang, Jun-Yan Zhu, Antonio Torralba, and Alexei A Efros. Dataset distillation. *arXiv preprint arXiv:1811.10959*, 2018.
- [Wilson and Zhu, 2008] Richard C Wilson and Ping Zhu. A study of graph spectra for comparing graphs and trees. *Pattern Recognition*, 41(9):2833–2841, 2008.
- [Xia *et al.*, 2023] Shuyin Xia, Guoyin Wang, and Xinbo Gao. Granular ball computing: an efficient, robust, and interpretable adaptive multi-granularity representation and computation method. *arXiv preprint arXiv:2304.11171*, 2023.
- [Xie *et al.*, 2020a] Jiang Xie, Zhong-Yang Xiong, Qi-Zhu Dai, Xiao-Xia Wang, and Yu-Fang Zhang. A new internal index based on density core for clustering validation. *Information Sciences*, 506:346–365, 2020.
- [Xie *et al.*, 2020b] Yu Xie, Chuanyu Yao, Maoguo Gong, Cheng Chen, and A Kai Qin. Graph convolutional networks with multi-level coarsening for graph classification. *Knowledge-Based Systems*, 194:105578, 2020.
- [Xu *et al.*, 2018] Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? *arXiv preprint arXiv:1810.00826*, 2018.
- [Xv *et al.*, 2023] Guipeng Xv, Chen Lin, Wanxian Guan, Jinping Gou, Xubin Li, Hongbo Deng, Jian Xu, and Bo Zheng. E-commerce search via content collaborative graph neural network. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2885–2897, 2023.
- [Yu and Cheng, 2001] Jian Yu and Qiansheng Cheng. The upper bound of the optimal number of clusters in fuzzy clustering. *Science in China Series: Information Sciences*, 44:119–125, 2001.
- [Zhao *et al.*, 2020] Bo Zhao, Konda Reddy Mopuri, and Hakan Bilen. Dataset condensation with gradient matching. *arXiv preprint arXiv:2006.05929*, 2020.