Bi-Directional Multi-Scale Graph Dataset Condensation via Information Bottleneck

Xingcheng Fu[1, 2, *†], **Yisen Gao**[3, *], **Beining Yang**[5], **Yuxuan Wu**[3], **Haodong Qian**[1, 2], **Qingyun Sun**[4], **Xianxian Li**[1, 2]

[1]Guangxi Key Lab of Multi-source Information Mining & Security, Guangxi Normal University, Guilin, China
[2]Key Lab of Education Blockchain and Intelligent Technology, Ministry of Education, Guangxi Normal University, China
[3]Institute of Artificial Intelligence, Beihang University, Beijing, China
[4]School of Computer Science and Engineering, Beihang University, Beijing, China
[5]University of Edinburgh, Edinburgh, UK
{fuxc, qianhaodong, lixx}@gxnu.edu.cn {yisengao, buaawyx, sunqy}@buaa.edu.cn, B.Yang-32@sms.ed.ac.uk

Abstract

Dataset condensation has significantly improved model training efficiency, but its application on devices with different computing power brings new requirements for different data sizes. Thus, condensing multiple scale graphs simultaneously is the core of achieving efficient training in different on-device scenarios. Existing efficient works for multi-scale graph dataset condensation mainly perform efficient approximate computation in scale order (large-to-small or small-to-large scales). However, for non-Euclidean structures of sparse graph data, these two commonly used paradigms for multi-scale graph dataset condensation have serious *scaling down degradation* and *scaling up collapse* problems of a graph. The main bottleneck of the above paradigms is whether the effective information of the original graph is fully preserved when consenting to the primary sub-scale (the first of multiple scales), which determines the condensation effect and consistency of all scales. In this paper, we proposed a novel GNN-centric **Bi**-directional **M**ulti-**S**cale **G**raph Dataset **C**ondensation (**BiMSGC**) framework, to explore unifying paradigms by operating on both large-to-small and small-to-large for multi-scale graph condensation. Based on the mutual information theory, we estimate an optimal "*meso-scale*" to obtain the minimum necessary dense graph preserving the maximum utility information of the original graph, and then we achieve stable and consistent "*bi-directional*" condensation learning by optimizing graph eigenbasis matching with information bottleneck on other scales. Encouraging empirical results on several datasets demonstrates the significant superiority of the proposed framework in graph condensation at different scales.

1 Introduction

With the rapid expansion of applications like social media, e-commerce, and recommendation systems on mobile and edge devices, the core challenge is efficiently training on graph data at an increasingly large scale. The multi-size dataset condensation approach is designed to enhance training efficiency by condensing graph datasets into various small yet informative graphs. These condensed graphs can accommodate the diverse computational needs of low-resource, on-device downstream users.

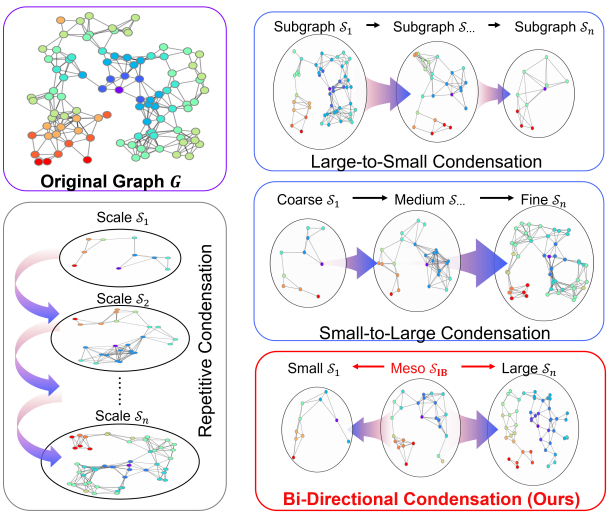


Figure 1: Comparison of existing paradigms

However, dataset condensation typically requires a computationally intensive learning-based process, and the multi-size property exacerbates the challenge in balancing efficiency and effectiveness. Based on the recent studies(He et al. 2024; Fang et al. 2024), there can be summarized three intuitive paradigms (illustrated in the Fig.1): (1) *Re-condensation*, which involves selecting an efficient and effective dataset condensation method to redistribute the data for each scale; (2) *Large-to-Small*, where the original dataset is condensed into multiple size/scales in one process by first creating a relatively large subset and then further condensing it step by step from large to small; and (3) *Small-to-Large*, which starts with the creation of smaller subsets that are then iteratively expanded to larger scales, preserving essential patterns at each stage. These paradigms offer different strategies for achieving efficient dataset condensation.

[*]These authors contributed equally.
[†]Corresponding author

As illustrated in Fig.2(b), while the Large-to-Small and Small-to-Large paradigms achieve a reduction in time by nearly 10 times. However, there is still a significant effectiveness gap remains compared to the naive re-condensation paradigm at certain scales. Generally, the Large-to-Small approach struggles at relatively small scales, whereas the Small-to-Large approach falters at larger scales. This raises a question of *whether we can combine the advantages of both paradigms to achieve a better balance between efficiency and effectiveness.* In order to better understand the effect of scale variation on the condensation performance, we further analyze the experimental phenomenon and attribute it to two key issues: (1) **Scaling Down Degradation**: Similar to the subset degradation problem observed in image data (He et al. 2024), graph data suffer from a severe "subgraph degradation problem." When we sample a subgraph from a condensed graph, its performance is much lower than a graph directly condensed to the target scale. In the large-to-small method, the distribution of effective information across a large number of nodes during the initial training process causes a more rapid degradation in performance when the sampling target scale is significantly reduced. (2) **Scaling Up Collapse**: as demonstrated in Fig. 2(b), because graph condensation uses fixed neural network model parameters across all scales, the small-to-large method tends to overfit long-tail information to adjust the fine-grained decision boundary at larger scales. This will lead to a large of noisy information during the scale-growing training process, thus limiting the condensation performance at larger scales.

To address these challenges, we first analyze the multi-scale graph condensation through the view of mutual information(Fig.2(c)). We observe a crucial finding: a mesoscopic scale (meso-scale) exists that balances scaling variation and condensation performance. Further, we propose starting at a 'meso-scale' and conducting bi-directional scaling to achieve optimal results on both ends. Inspired by recent research on information bottlenecks in graphs, we first estimate the exact size of the meso-scale and first condense a meso-scale subgraph of size, which is used to retain the maximum amount of valid information while minimizing the scale. Further, we introduce the information bottleneck principle into the optimization of bi-directional condensation to retain more effective information during the scale-changing. As shown in the Fig. 2(b), BiMSGC gains nearly same performance to the naive re-condensation performance at the similar time cost to the Large-to-small and Small-to-Large paradigms. We summarize our contributions as follows:

- For multi-scale graph dataset condensation, we first explore the limitations of existing work and propose a novel bi-directional multi-scale graph condensation paradigm.
- For the problem of retaining valid information in scale variation, we introduce the idea of subgraph condensation information bottleneck to minimize the scaling down degradation and scaling up collapse problems.
- Extensive experiments have demonstrated that our BiMSGC outperforms the baselines by a large margin in five datasets. For example, BiMSGC has 20.85 speedup compared with the baselines on Citeseer dataset.

2 Related Works

2.1 Graph Dataset Condensation

Graph dataset condensation (Sun et al. 2024) distills the data of graph structures to achieve the same effect of the original graph for graph neural network training. GCond (Jin et al. 2022) distills the graph by matching the model's gradient at each training step, while SFGC (Zheng et al. 2023) extends this to match the entire training trajectory. GDEM (Liu, Bo, and Shi 2024) offers a different perspective by eliminating spectral domain differences between the synthetic and original graphs through feature basis matching. On the other hand, GCDM (Liu et al. 2022)focuses on synthesizing the final compressed graphs by minimizing the distributional differences in the GNN embeddings.

2.2 Information Bottleneck

The information bottleneck (IB) (Lewandowsky and Bauch 2024) originates from rate distortion theory and aims to compress the source variable X into a compact representation Z while retaining the necessary information to predict the target variable Y. IB is used to evaluate the rate-distortion trade-off for the source variable and is widely applied in interpretable deep learning theories to enhance robustness (Saxe et al. 2018). In parallel, graph information bottleneck (Wu et al. 2020) has been introduced to characterize and manage the flow of information in graph-structured data.

3 Problem Formulation and Analysis

3.1 Notations

Given a large graph dataset $\mathbf{G} = (\mathbf{X}, \mathbf{A}, \mathbf{Y})$, where $\mathbf{X} \in \mathbb{R}^{N \times D}$ denotes the number of N nodes with D-dimensional characteristics, $\mathbf{A} \in \{0,1\}^{N \times N}$ denotes the adjacency matrix of the graph, and $\mathbf{Y} \in \{0,1,...,C-1\}^{N \times 1}$ denotes the label of each node in the C classes. The purpose of graph dataset condensation is to condense a large graph $\mathbf{G}$ into a synthetic graph $\mathbf{G}' = (\mathbf{X}', \mathbf{A}', \mathbf{Y}')$ while maintaining similar model training results, where $\mathbf{X}' \in \mathbb{R}^{N' \times D}$, $\mathbf{A}' \in \{0,1\}^{N' \times N'}$, $\mathbf{Y}' \in \{0,1,...,C-1\}^{N' \times 1} (N' \ll N)$. For the condensed graphs at different scales, we use $\mathbf{G}'_s$, $\mathbf{G}'_m$, $\mathbf{G}'_l$ to represent the small-scale, middle-scale, and large-scale condensed dataset respectively.

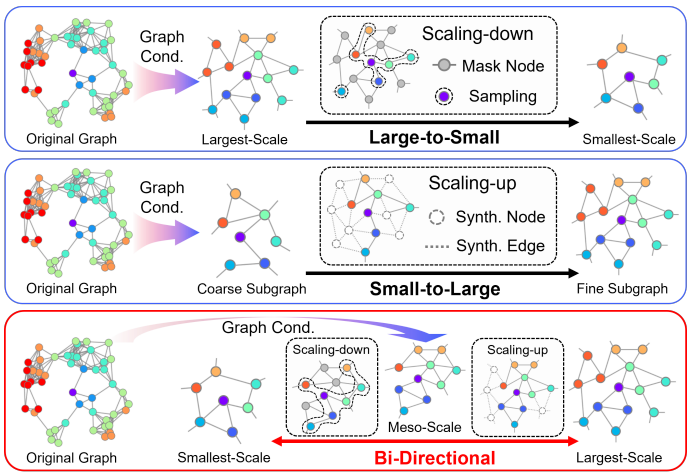
3.2 Problem Formulation

From the perspective of mutual information, the objective of graph dataset condensation can be uniformly expressed as:

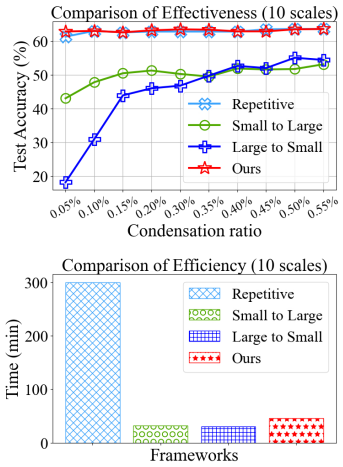
$$\max I\left(G'; H(G, Y)\right), \tag{1}$$

where I denotes the mutual information and $H(G, Y)$ represents the relevant information extracted from the original graph during model training for node classification task. Specifically, the relevant information refers to the gradients or the trajectory of the GNN training process for gradient matching methods, and feature distribution for distribution matching methods.

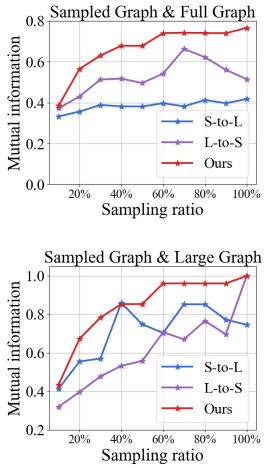
The task of multi-scale graph dataset condensation involves generating multiple condensed graphs at various



(a) The training strategies of three different directional paradigms.



(b) Performance.



(c) Mutual Information.

Figure 2: (a) Three different paradigm-specific training strategies: Large-to-Small is trained by mask sampling from a large graph to obtain a small graph. Small-to-Large is trained by considering the small graph as a subgraph expansion of the large graph. Our method obtains meso-scale subgraphs and then trains them separately to both sides; (b) Condensation performance on each scale (top) and time efficiency (bottom) obtained by training four different multi-scale compression strategies on Ogbn-arxiv using GCond as a backbone method; (c) The magnitude of mutual information between the different target condensation scales of the three paradigms and the original graph (top), as well as the magnitude of mutual information with the largest scale condensed graph (bottom), where S-to-L represents Small-to-Large and L-to-S represents Large-to-Small.

scales, each capable of achieving the same training effectiveness as the original dataset. Concerning requirements of efficiency, recondensing the graph at every desired scale is impractical, as it would result in significant time and space inefficiencies. A more effective approach is to generate condensed graphs at different scales by selecting subsets from the largest condensed graph. In other words, $\mathbf{G}'_s$, $\mathbf{G}'_m$, and $\mathbf{G}'_l$ are all subgraphs of $\mathbb{G}'$. Our objective then becomes:

$$\max I(G'_{sub}; H(G, Y)); \forall G'_{sub} \in Sub(G'). \quad (2)$$

where G'_{sub} represents a subgraph of the condensed graph G', $Sub(G')$ denotes the set of all subgraphs of G'.

3.3 Problem Analysis

Understanding scaling degeneracy and collapse problems. Based on experimental observations, we identify two key issues in multi-scale graph dataset condensation: scaling down degradation and scaling up collapse. We further analyze the primary factors contributing to these problems using mutual information theory. Following the approach of GMI (Peng et al. 2020)., we estimate the mutual information $I(G'_{sub}; G)$ between the sampled subgraph G'_{sub} and the original graph G and its mutual information $I(G'_{sub}; G')$ with the condensed graph G' at different scales. The results are shown in Fig 2(c).

Large-to-Small: This method is designed to preserve the mutual information between the large-scale subset G_l and the optimization objective, while progressively enhancing the mutual information for the small-scale subset G_s. Since it has a very high value of $I(G'_{sub}; G)$ with the original graph, this allows it to approach the results of re-condensation at larger scales. However, it is observed that the method yields very low mutual information $I(G'_{sub}; G')$ values at smaller scales, which explains the pronounced subgraph degradation problem in Fig2(b) at target condensation ratio below 0.30%.

Small-to-Large: This approach begins by preserving mutual information within small-scale graphs G_s, gradually introducing new training nodes to expand the scale and incorporate effective mutual information into the larger graph G_l. $I(G'_{sub}; G')$ using the small-to-large method is higher than that of the large-to-small method. It demonstrates that this method effectively maintains a lower bound of mutual information retention at the smallest scale. However, it does not fully address the issue of scale discrepancy. It has the smallest mutual information value $I(G'_{sub}; G)$ among the three condensation methods. This may be due to the introduction of a large number of new nodes in the later stages of training, which makes the node features contain a large amount of noisy information, thus affecting the upper limit of the expressive power of the condensed graph. This explains why the small-to-large method does not test as well as the other methods after the target condensation ratio is greater than 0.35% in Fig2(b).

Fortunately, we have identified a mesoscopic-scale condensation graph G_m that effectively mitigates the loss of mutual information caused by scale differences. In addressing the scaling down degradation problem, this approach retains as much mutual information as possible while minimizing scale reduction. Conversely, in the case of scaling up collapse, it reduces the redundancy of extra information that often accompanies scale increases, thereby maintaining a more balanced and efficient condensation process.

4 Methodology

In this section, we elaborate the proposed BiMSGC, a novel bi-directional and information bottleneck principle guided multi-scale graph condensation framework. Our work mainly consists of three parts. First of all, we present how we use the information bottleneck principle to guide meso-scale selection. Then, we introduce the bi-directional multi-scale graph dataset condensation optimization method based on information bottleneck. Finally, we combine the bi-directional condensation framework with a concrete graph condensation method based on eigenbasis matching. The architecture is shown in Figure 3, and the overall process of BiMSGC is described in Algorithm 1.

4.1 Bi-directional Multi-Scale Graph Dataset Condensation Framework

One indispensable step of multi-scale condensation is to generate a subgraph that best preserves mutual information despite scale difference. We start from distilling a meso-scale dataset, and then adjust it to target scales as needed, minimizing the impact of excessive scale differences while maintaining high computational efficiency.

Step 1: Information Bottleneck Guided Meso-Scale Selection.

To achieve a more stable condensation performance across multiple scales, we first distill a 'middle scale' subgraph G_m, referred to as *Meso-Scale* subgraph. This subgraph captures as much valid information as possible, but minimize redundancies and noise. Naturally, we associate it with the subgraph information bottleneck, which aims to compress the data from the original version while preserving key valid information. Specifically, the formulation is as follows:

$$G'_m = \underset{G'_{sub} \in Sub(G')}{\operatorname{argmax}} I\left(G'_{sub}; Y'\right) - \beta I\left(G'; G'_{sub}\right), \quad (3)$$

where β is a hyperparameter balancing informativeness and compression.

It is worthy to note that this step is only a preliminary estimation rather than a final optimization goal. Therefore, it can be approximated using MINE (Belghazi et al. 2018) or GIB (Wu et al. 2020). In practice, the meso-scale is selected by pre-setting several subgraph scales for calculation and then comparing them. Having the meso-scale determined, we obtain the condensed graph G'_m by training on this scale.

Step 2: Bi-directional Graph Condensation with IB.

After obtaining the meso-scale subgraph G'_m by initial condensation, we further train the synthesized graphs in both directions. Specifically, we view the optimization problem for both training processes uniformly as a variant of the subgraph information bottleneck and call it the **S**ubgraph **C**ondensation **I**nformation **B**ottleneck(SCIB). It takes the following specific form:

$$\max_{G'_{sub} \in Sub(G')} I\left(G'_{sub}; H(G,Y)\right) - \beta I\left(G'; G'_{sub}\right), \quad (4)$$

Considering the difficulty to compute and optimize the mutual information directly, VIB (Abdelaleem, Nemenman, and Martini 2023) optimized the objective by using a variational approach to compute the under-error session of the information bottleneck.

To be specific, for the first part $I\left(G'_{sub}; H(G,Y)\right)$, we replace $P(H(G,Y)|G'_{sub})$ with a parameterized variational approximation $Q(H(G,Y)|G'_{sub})$ and infer its approximate lower bound:

$$I(G'_{sub}; H(G,Y)) \geq \mathbb{E}\left[\log Q(H(G,Y)|G'_{sub})\right], \quad (5)$$

Simultaneously, for the second part $I(G', G'_{sub})$, we estimate it by introducing the variational approximation R for the marginal distribution P, following common practice. With Kullback-Leibler (KL) divergence, the mutual information can be approximated as:

$$\begin{aligned} I\left(G'_{sub}; G'\right) &\leq \mathbb{E}\left[\mathrm{KL}\left(P\left(G'_{sub}\middle| G'\right) \| R(G'_{sub})\right)\right] \\ &= \mathbb{E}\left[f(G'_{sub}, G')\right], \end{aligned} \quad (6)$$

By combining these two equations together, we derive the optimization objective L_{SCIB}:

$$\begin{aligned} L_{SCIB}(G'_{sub}; G') = &\mathbb{E}\left[\log Q(H(G,Y)|G'_{sub})\right] \\ &- \beta \mathbb{E}\left[f(G'_{sub}, G')\right]. \end{aligned} \quad (7)$$

For condensation from meso-scale to smaller scales, we consider the small-scale graphs G'_l to be subgraphs of the meso-scale graph G'_m. In this case, its optimization objective would be $L_{SCIB}(G'_l; G'_m)$. Similarly, for the process of expanding from meso-scale to large scales, we treat G'_m as a subgraph of large-scale graphs, with $L_{SCIB}(G'_m; G')$ as the corresponding optimization. The detailed derivation is reported in the Appendix.

Step 3: Instantiation of Multi-Scale Condensed Graph.

We instantiate the distribution Q assigning importance score to each node's impact on the training loss function, and then define the distribution R as a Bernoulli distribution with parameter θ. For $P\left(G_{sub}|G\right)$, we assign relevant scores to different scales by continuously adjusting global masks. While transitioning from meso to smaller scales, we gradually increase the global masks to reduce the training scale. Conversely, the global masks are decreased while expanding the training scale.

4.2 Application on Graph Condensation

In this section, we integrate the above framework with specific techniques for graph condensation. Theoretically, our approach can be combined with any graph condensation method to represent $H(G,Y)$. In order to retain more important information, we use eigenbasis matching method, which leverages key structural information along with gradient information.

First, we compute the Laplacian matrix $\mathbf{L}$ from the adjacency matrix $\mathbf{A}$ of the original graph and decompose it as $\mathbf{L} = \mathbf{U}\mathbf{\Lambda}\mathbf{U}^\top = \sum_{i=1}^{N} \lambda_i \mathbf{u}_i \mathbf{u}_i^\top$. We then initialize the corresponding eigenbasis $\mathbf{U}'_K = [\mathbf{u}'_1, \cdots, \mathbf{u}'_{N'}] \in \mathbb{R}^{N' \times K}$, where K is a hyperparameter.

For a simple GNN model, the loss function can be expressed as:

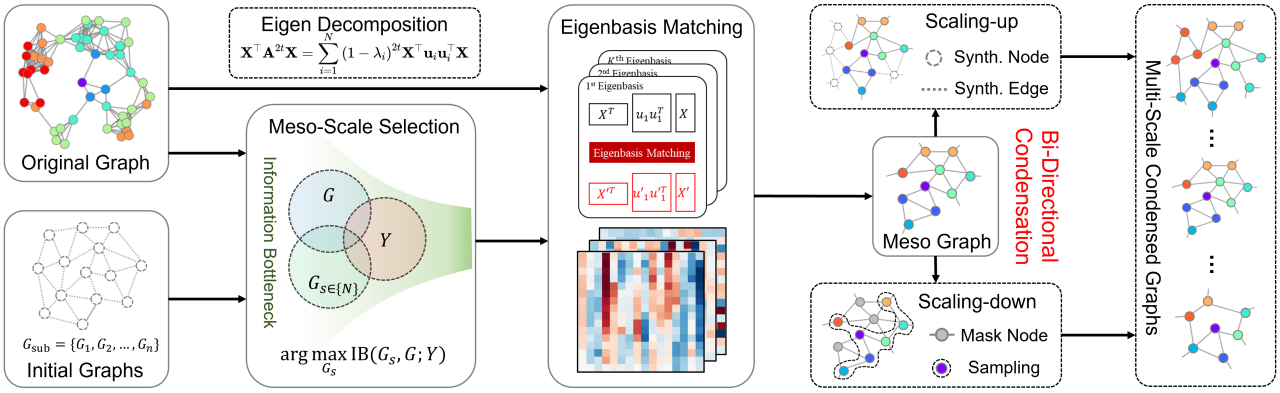


Figure 3: An illustration of BiMSGC architecture.

$$\begin{aligned}\mathcal{L}_{GM} &= \|\nabla_{\mathbf{W}} - \nabla'_{\mathbf{W}}\|_F^2 \\ &\leq \|\mathbf{W}\|\|\mathbf{X}^\top\mathbf{A}^{2t}\mathbf{X} - \mathbf{X'}^\top\mathbf{A'}^{2t}\mathbf{X'}\| \\ &\quad + \|\mathbf{X}^\top\mathbf{A}\mathbf{Y} - \mathbf{X'}^\top\mathbf{A'}\mathbf{Y'}\| \\ &\approx \|\mathbf{W}\| \underbrace{\textstyle\sum_{i=1}^{N'}(\mathbf{X}^\top\mathbf{u}_i\mathbf{u}_i^\top\mathbf{X} - \mathbf{X'}^\top\mathbf{u}'_i\mathbf{u}_i'^\top\mathbf{X'})}_{\mathcal{L}_e} \\ &\quad + \underbrace{\|\mathbf{X}^\top\mathbf{A}\mathbf{Y} - \mathbf{X'}^\top\mathbf{A'}\mathbf{Y'}\|}_{\mathcal{L}_o},\end{aligned} \tag{8}$$

Due to the orthogonality of eigenvector matrices, the representation space is constrained by a regularization loss:

$$\mathcal{L}_o = \left\|{\mathbf{U'}_K}^\top\mathbf{U'}_K - \mathbf{I}_K\right\|_F^2, \tag{9}$$

Generally, the optimization objective for eigenbasis matching is:

$$\mathcal{L}_{total} = \alpha\mathcal{L}_e + \beta\mathcal{L}_d + \gamma\mathcal{L}_o. \tag{10}$$

where α, β, γ are hyperparemeters.

As for condensing multi-scale graphs, we first take the eigenvectors corresponding to the first M eigenvalues and distill a meso-scale subgraph utilizing the equation. After that, we still match with the first M feature values for further training of small-scale graphs from meso-scale ones. For going from meso to large scales, the matching is shifted to the feature vectors corresponding to the first N' feature values.

4.3 Complexity Analysis

First, we decompose the K largest eigenvalues of the original image, of which the time complexity is $\mathcal{O}(KN^2)$. Complexity of calculating the loss function (10) is $\mathcal{O}(KN'd + Kd^2 + KN'^2)$. Since the training process consists of two phases, the time complexity of the whole process is a total of $\mathcal{O}(2(KN'd+Kd^2+KN'^2))$. However, since the first stage training already contains some of the optimization objectives for the second stage training, the corresponding training time can be reduced. The actual time complexity of our method can be approximated as $\mathcal{O}(KN'd+Kd^2+KN'^2)$, while re-condensation requires to train at each scale separately. To sum up, it has a complexity of $\mathcal{O}(N'(KN'd + Kd^2 + KN'^2))$ which is an order of magnitude higher in complexity than our method.

Algorithm 1: Bi-directional Condensation Algorithm

Input: Original Graph $\mathbf{G} = \{\mathbf{X}, \mathbf{A}\}$; Number of training meso-scale epochs E_1; Number of training bi-directional scale epochs E_2, the learning rate η.
Parameter: The relevant parameters θ about synthesized graph G'.
Output: The condensed graph G'.
Init a meso-scale G_M using Eq (3).
for $e = 1$ **to** E_1 **do**
  Train G_M using Eq (10).
  Update $\theta_m \leftarrow \theta_m - \eta\nabla\theta_m$.
end for
for $e = 1$ **to** E_2 **do**
  Train G_s using Eq (10) and Eq (7);
  Train G_l using Eq (10) and Eq (7);
  Update $\theta \leftarrow \theta - \eta\nabla\theta$.
end for

5 Experiment

5.1 Experimental Setup

Datasets. To evaluate the performance of our BiMSGC [1], we choose five node classification benchmark graphs, including three transductive graphs, Cora, Citeseer (Kipf and Welling 2017), Ogbn-Arxiv (Hu et al. 2021a) and two inductive graphs, Flickr and Reddit (Zeng et al. 2020).

Baselines. Based on GC-Bench (Sun et al. 2024) [2], we select gradients-based methods: GCond (Jin et al. 2022), SGDD (Yang et al. 2023), EXGC (Fang et al. 2024); trajectory-based methods: SFGC (Zheng et al. 2023),GEOM (Zhang et al. 2024); other method: GCDM (Liu et al. 2022), GC-SNTK (Liu et al. 2022), GDEM (Liu, Bo, and Shi 2024).

Settings and Hyperparameters. We followed the default values of parameters for baselines. All models were trained and tested on a single Nvidia A800 80GB GPU. The remaining details are given in the Appendix.

[1] https://github.com/RingBDStack/BiMSGC.
[2] https://github.com/RingBDStack/GC-Bench.

Dataset	Reduction rate(%)	Condensation Methods									Whole Dataset
		GCond	SFGC	SGDD	EXGC	GC-SNTK	GCDM	GDEM	GEOM	Ours	
Cora	0.50	$58.0_{\pm 5.4}$	$66.9_{\pm 7.0}$	$61.6_{\pm 6.3}$	$61.5_{\pm 7.2}$	$47.4_{\pm 13.3}$	$53.2_{\pm 2.7}$	$\underline{68.7}_{\pm 3.6}$	$39.9_{\pm 10.4}$	$\mathbf{78.2}_{\pm 1.3}$	$80.9_{\pm 0.1}$
	1.00	$73.4_{\pm 4.6}$	$70.3_{\pm 0.6}$	$\underline{74.4}_{\pm 2.0}$	$71.9_{\pm 2.5}$	$65.4_{\pm 10.0}$	$61.2_{\pm 4.8}$	$69.9_{\pm 1.7}$	$66.0_{\pm 6.6}$	$\mathbf{80.5}_{\pm 1.0}$	
	1.50	$79.2_{\pm 1.2}$	$\underline{79.3}_{\pm 0.5}$	$75.5_{\pm 2.3}$	$77.9_{\pm 1.2}$	$71.6_{\pm 4.9}$	$56.0_{\pm 3.0}$	$71.1_{\pm 2.9}$	$77.6_{\pm 2.1}$	$\mathbf{79.5}_{\pm 1.4}$	
	2.00	$80.0_{\pm 0.3}$	$80.8_{\pm 0.4}$	$77.9_{\pm 1.0}$	$79.8_{\pm 0.4}$	$74.8_{\pm 2.6}$	$79.0_{\pm 3.0}$	$74.1_{\pm 2.9}$	$\mathbf{83.6}_{\pm 0.2}$	$\underline{80.9}_{\pm 0.6}$	
CiteSeer	0.50	$51.9_{\pm 1.0}$	$33.3_{\pm 2.0}$	$41.7_{\pm 9.1}$	$45.9_{\pm 8.2}$	$40.3_{\pm 13.0}$	$51.8_{\pm 2.6}$	$\underline{68.1}_{\pm 2.5}$	$32.3_{\pm 5.9}$	$\mathbf{73.7}_{\pm 0.9}$	$71.1_{\pm 0.1}$
	1.00	$53.4_{\pm 1.4}$	$32.9_{\pm 1.8}$	$52.2_{\pm 7.2}$	$59.4_{\pm 7.1}$	$47.8_{\pm 8.2}$	$58.2_{\pm 0.2}$	$\underline{70.9}_{\pm 1.0}$	$47.5_{\pm 8.0}$	$\mathbf{73.6}_{\pm 0.6}$	
	1.50	$62.6_{\pm 0.7}$	$54.5_{\pm 0.6}$	$51.0_{\pm 7.0}$	$61.5_{\pm 8.1}$	$55.6_{\pm 8.9}$	$62.9_{\pm 0.8}$	$\underline{71.8}_{\pm 1.1}$	$66.0_{\pm 4.2}$	$\mathbf{73.6}_{\pm 1.8}$	
	2.00	$69.8_{\pm 0.3}$	$60.2_{\pm 0.6}$	$70.9_{\pm 0.3}$	$71.0_{\pm 0.6}$	$65.4_{\pm 5.1}$	$65.7_{\pm 1.1}$	$72.6_{\pm 0.3}$	$\mathbf{74.3}_{\pm 0.1}$	$\underline{73.6}_{\pm 0.2}$	
Ogbn -Arxiv	0.05	$48.8_{\pm 4.0}$	$45.3_{\pm 3.0}$	$52.2_{\pm 0.9}$	$44.1_{\pm 3.9}$	$25.1_{\pm 5.6}$	$51.9_{\pm 1.7}$	$\underline{61.0}_{\pm 1.9}$	$44.9_{\pm 4.4}$	$\mathbf{62.7}_{\pm 0.4}$	$71.8_{\pm 0.1}$
	0.10	$54.0_{\pm 0.7}$	$52.6_{\pm 0.8}$	$60.2_{\pm 0.3}$	$51.0_{\pm 3.0}$	$25.1_{\pm 5.6}$	$60.3_{\pm 2.9}$	$\underline{60.5}_{\pm 2.2}$	$50.5_{\pm 3.3}$	$\mathbf{63.2}_{\pm 0.2}$	
	0.30	$60.2_{\pm 1.1}$	$62.3_{\pm 1.2}$	$\underline{63.4}_{\pm 0.1}$	$59.2_{\pm 1.9}$	$46.1_{\pm 3.8}$	$62.4_{\pm 2.7}$	$60.6_{\pm 1.4}$	$\mathbf{63.8}_{\pm 0.9}$	$62.7_{\pm 0.2}$	
	0.50	$\underline{64.5}_{\pm 0.1}$	$\mathbf{66.2}_{\pm 0.4}$	$63.4_{\pm 0.3}$	$62.8_{\pm 0.5}$	$55.4_{\pm 1.2}$	$64.2_{\pm 2.9}$	$62.1_{\pm 1.0}$	$63.8_{\pm 0.2}$	$63.7_{\pm 0.2}$	
Flickr	0.10	$\underline{46.2}_{\pm 2.1}$	$44.1_{\pm 1.5}$	$44.0_{\pm 2.1}$	$42.2_{\pm 1.2}$	$28.3_{\pm 5.7}$	$32.9_{\pm 2.5}$	$42.6_{\pm 2.5}$	$43.0_{\pm 1.7}$	$\mathbf{50.3}_{\pm 0.5}$	$47.2_{\pm 0.1}$
	0.30	$\underline{46.8}_{\pm 0.1}$	$43.7_{\pm 0.7}$	$45.6_{\pm 0.2}$	$44.6_{\pm 0.9}$	$29.1_{\pm 3.9}$	$36.2_{\pm 0.8}$	$45.4_{\pm 2.6}$	$45.6_{\pm 0.5}$	$\mathbf{50.4}_{\pm 0.1}$	
	0.70	$47.1_{\pm 0.2}$	$44.3_{\pm 0.3}$	$\underline{48.0}_{\pm 0.2}$	$46.3_{\pm 0.4}$	$32.2_{\pm 6.0}$	$36.2_{\pm 1.3}$	$46.3_{\pm 2.4}$	$46.7_{\pm 0.2}$	$\mathbf{50.6}_{\pm 0.3}$	
	1.00	$47.1_{\pm 0.1}$	$44.2_{\pm 0.1}$	$47.9_{\pm 0.1}$	$46.9_{\pm 0.1}$	$31.2_{\pm 3.3}$	$35.9_{\pm 0.5}$	$\underline{49.4}_{\pm 3.4}$	$47.3_{\pm 0.1}$	$\mathbf{50.6}_{\pm 0.1}$	
Reddit	0.05	$73.0_{\pm 5.2}$	$64.4_{\pm 0.8}$	$81.0_{\pm 0.9}$	$71.4_{\pm 2.9}$	-	$78.0_{\pm 3.4}$	$\underline{83.6}_{\pm 0.9}$	$69.0_{\pm 5.7}$	$\mathbf{93.3}_{\pm 0.7}$	$93.9_{\pm 0.1}$
	0.10	$84.2_{\pm 1.5}$	$67.6_{\pm 0.2}$	$82.7_{\pm 0.2}$	$82.7_{\pm 2.1}$	-	$\underline{88.9}_{\pm 1.6}$	$86.6_{\pm 1.0}$	$83.9_{\pm 1.2}$	$\mathbf{93.6}_{\pm 0.2}$	
	0.15	$87.4_{\pm 0.8}$	$80.3_{\pm 0.3}$	$81.9_{\pm 0.7}$	$87.1_{\pm 1.0}$	-	$85.9_{\pm 0.4}$	$86.5_{\pm 0.3}$	$\underline{88.9}_{\pm 0.6}$	$\mathbf{92.9}_{\pm 0.1}$	
	0.20	$90.9_{\pm 0.2}$	$84.6_{\pm 1.9}$	$88.5_{\pm 0.2}$	$89.4_{\pm 0.3}$	-	$89.2_{\pm 0.5}$	$\mathbf{93.1}_{\pm 0.1}$	$91.5_{\pm 0.1}$	$\underline{93.1}_{\pm 0.1}$	

Table 1: Node classification performance of different condensation methods,. (Result: average score $\pm$ standard deviation. **Bold**: best; Underline: runner-up. -: cuda out of memory.)

5.2 Node Classification

The node classification performance is reported in Table 1, where we reviewed the performance of the node classification task at different scales. For the needs of the multi-scale graph dataset condensation task, we distilled all graphs to the largest scale and then tested them by random sampling the subgraph according to the target reduction rate.

First, our model demonstrates strong performance across all scales. Notably, even with a subgraph containing just 10% of the nodes from the largest condensed graph—a scenario that usually leads to a significant performance drop in other baseline models—our method maintains nearly lossless training performance. This suggests that our approach effectively mitigates challenges related to scale differences.

Second, our model not only maintains the training performance at different scales, but in some cases exceeds the performance of the original dataset. This remarkable result can be attributed to the information bottleneck refining process, in which we effectively retain the vast majority of low-frequency valid information while eliminating redundant data that may cause interference, which leads to improved generalization ability of the model.

5.3 Cross-architecture Generalization

Next, we evaluated the ability of the model's different scale condensation effects to generalize across models. For modeling, we used MLP (Hu et al. 2021b), SGC (Wu et al. 2019), APPNP (Klicpera, Bojchevski, and Günnemann 2019) and ChebNet (He, Wei, and Wen 2022). We tested the performance of the different compression models by sampling at four different size scales, and their means and variances are presented in the Table 2. For their specific performance at each scale, we have reported in the Appendix.

First, our approach effectively eliminates scale differences across various model architectures, achieving the best average performance with the lowest variance, especially on large-scale graphs like Reddit. Even with drastic scale reductions, training performance remains consistent, proving the method's adaptability and universal applicability for multi-scale graph dataset condensation.

Second, when it comes to model generalization, our approach consistently achieves optimal results across various models. This indicates that, even while preserving the multi-scale effect, our method remains highly competitive at every scale. It is a good illustration of the advantages of our method in terms of generalization ability.

5.4 Ablation and Sensitivity Study

Given that our model selects the initial meso-scale for multi-scale condensation process guided by information bottleneck(IB), we evaluated the necessity of IB and our model's sensitivity towards initial meso-scale selection. Here, we set the scales considered by meso-scale to 0.2,0.5,0.8.

Ablation study. For the ablation study on the impact of using IB, we set the maximum scale to 1.0% and condensed the Citeseer dataset to various target reduction scales. Results for meso-scales of 0.2 and 0.8 are shown in Figure 4, indicating that IB optimization effectively mitigates degradation caused by scale differences. Without IB, subgraph degradation occurs at a meso-scale of 0.8 with small compression rates, and larger scales fluctuate at a meso-scale of 0.2. With IB, performance across scales stabilizes.

Dataset	Models	Condensation Methods								
		GCond	SFGC	SGDD	EXGC	GC-SNTK	GCDM	GDEM	GEOM	Ours
Citeseer	GCN	59.4±8.4	45.2±14.2	54.0±12.2	59.5±10.4	52.3±10.8	59.7±6.1	70.9±2.0	55.0±18.8	**73.6**±0.1
	MLP	57.9±6.5	37.8±18.1	55.8±4.1	60.3±1.8	52.3±12.8	50.6±12.7	72.8±5.0	55.0±18.9	**73.2**±0.4
	SGC	50.6±6.9	40.1±14.1	55.0±10.4	59.3±12.1	53.2±15.0	28.2±3.0	72.3±0.1	55.5±18.2	**73.4**±0.1
	APPNP	52.0±19.1	35.3±10.1	56.2±16.4	55.3±16.3	38.8±13.9	62.5±3.8	71.5±0.4	43.6±15.3	**73.3**±0.3
	ChebNet	47.3±8.5	49.8±15.0	55.5±11.1	64.3±3.8	60.5±10.6	37.4±5.5	71.9±0.8	57.7±15.6	**73.2**±0.2
Cora	GCN	72.6±10.2	74.3±6.8	72.3±7.3	72.8±8.3	64.8±12.3	62.4±11.6	70.9±2.3	66.8±19.4	**79.8**±1.2
	MLP	70.6±2.4	77.5±4.6	71.9±7.5	71.1±5.6	63.4±14.3	52.6±9.0	68.8±6.9	66.8±19.4	**79.1**±0.6
	SGC	64.9±16.2	76.6±3.4	71.0±9.1	71.0±10.5	65.4±7.1	33.8±15.2	74.5±0.6	66.5±19.3	**79.7**±0.9
	APPNP	62.6±15.2	77.1±3.6	69.8±11.0	71.5±9.8	39.6±11.9	68.0±11.8	66.1±10.5	45.6±14.8	**79.9**±0.6
	ChebNet	69.1±7.3	**75.7**±4.2	73.00±5.2	72.9±5.9	67.7±9.2	39.8±6.9	68.0±9.0	67.00±15.2	75.3±0.8
Ogbn-Arxiv	GCN	56.9±6.9	56.6±9.4	59.8±5.3	54.3±8.3	38.0±15.2	59.7±5.4	61.1±0.7	57.0±11.1	**63.1**±0.4
	MLP	54.8±5.9	57.4±8.4	60.4±4.1	54.0±8.3	37.9±12.5	50.3±3.7	58.4±1.6	57.0±11.1	**62.4**±0.8
	SGC	55.8±8.8	56.1±9.2	61.7±2.5	53.2±9.9	40.1±16.7	58.1±5.3	**63.2**±0.1	54.2±11.2	60.2±2.2
	APPNP	57.3±7.3	54.8±7.3	59.6±3.3	55.2±7.4	36.0±17.9	54.4±3.0	60.5±1.7	53.1±10.4	**61.8**±1.6
	ChebNet	50.8±8.2	52.4±8.6	51.7±5.5	48.4±8.9	22.9±11.6	39.2±3.6	**57.4**±1.0	50.4±11.0	56.9±0.2
Flickr	GCN	46.8±0.4	44.1±0.2	46.4±1.9	45.0±2.1	30.2±1.7	35.3±1.5	45.9±2.8	45.7±1.8	**50.5**±0.1
	MLP	42.9±0.5	41.0±0.6	43.1±1.2	41.9±1.2	25.4±3.3	42.8±1.4	39.4±0.6	45.7±1.8	**49.7**±0.4
	SGC	46.6±0.3	45.0±0.9	45.5±2.3	44.9±1.6	32.4±3.9	34.1±0.9	39.8±0.2	45.3±1.7	**49.4**±0.3
	APPNP	31.1±3.9	36.0±2.3	41.8±2.6	31.6±3.5	21.6±3.4	42.8±1.4	31.5±0.1	36.8±3.4	**49.9**±0.1
	ChebNet	42.1±1.7	40.8±0.5	41.6±1.0	40.8±1.0	27.1±1.8	41.6±1.0	38.2±0.6	42.5±3.0	**46.0**±0.5
Reddit	GCN	83.9±7.7	74.2±9.7	83.5±3.3	82.7±7.9	-	85.5±5.2	87.4±4.0	83.3±10.0	**93.2**±0.3
	MLP	49.5±6.4	57.4±8.4	50.3±6.7	42.8±4.1	-	77.7±8.7	54.8±7.4	83.3±10.0	**93.2**±0.1
	SGC	86.5±6.0	56.1±9.2	84.0±2.9	83.3±8.2	-	83.8±3.1	86.5±2.5	82.4±10.4	**89.4**±0.6
	APPNP	81.9±6.8	66.2±12.8	82.8±7.1	81.9±7.1	-	82.6±6.7	77.7±14.1	81.7±9.2	**92.3**±0.2
	ChebNet	72.3±8.0	52.4±8.6	71.3±4.4	68.0±8.8	-	77.6±4.3	74.0±13.3	73.6±11.5	**86.7**±0.8

Table 2: Generalization of different condensation methods across GNNs. (Result: average score ± standard deviation. **Bold**: best; Underline: runner-up. -: cuda out of memory.)

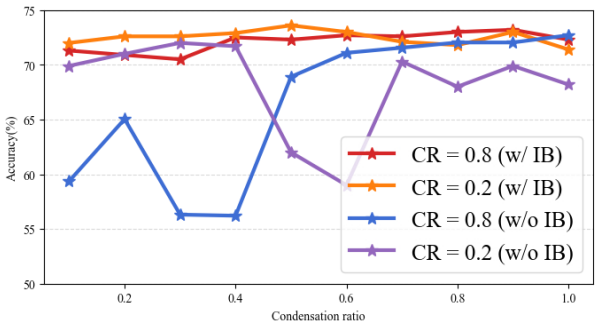


Figure 4: Ablation study for IB with different meso-scales.

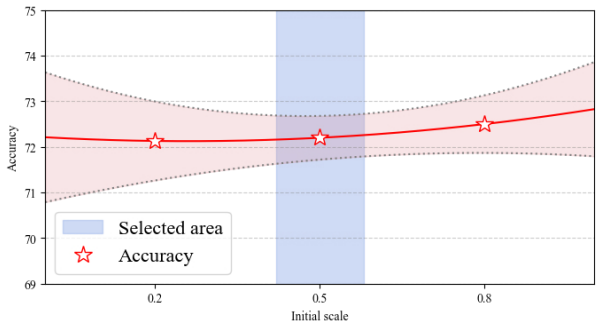


Figure 5: Sensitivity study on meso-scale selection.

Meso-scale selection analysis. To evaluate how different initial scales affect the performance of multi-scale condensation, we calculated the average values and standard deviation based on the accuracy of node classification at all scales, forming Fig.5. The results indicate that as the initial meso-scale increases, the average performance varies by less than 0.5%, with the lowest standard error observed at meso scales. This suggests that initial scale influences the consistency of bi-directional condensation, and our model maintains strong performance across all initial scales. We conclude that our IB-based adaptive selection of meso-scales ensures consistent performance across multiple scales with minimal average performance differences.

6 Conclusion

In this paper, we propose a GNN-centric Bi-directional Multi-scale Graph Dataset Condensation framework (BiMSGC) devised to achieve effective and efficient multi-scale graph condensation by unifying small-to-large and large-to-small paradigms. BiMSGC starts with generating a meso-scale subgraph under the guidance of Informational Bottleneck principles, and then synthesize graphs at other scales by pruning or expanding based on the meso-scale subgraph. Sufficient experiments were conducted to evaluate the performance of BiMSGC in node classification and cross-architecture generalization tasks, with results demonstrating the clear superiority of BiMSGC.

Acknowledgments

The corresponding author is Xingcheng Fu. Authors of this paper are supported by the National Natural Science Foundation of China through grants No.U21A20474 and No.62462007, and the Research Fund of Guangxi Key Lab of Multi-source Information Mining & Security (24-A-02-01). We extend our sincere thanks to all authors for their valuable efforts and contributions.

References

Abdelaleem, E.; Nemenman, I.; and Martini, K. M. 2023. Deep Variational Multivariate Information Bottleneck - A Framework for Variational Losses. *Arxiv*, abs/2310.03311.

Belghazi, M. I.; Baratin, A.; Rajeshwar, S.; Ozair, S.; Bengio, Y.; Courville, A.; and Hjelm, D. 2018. Mutual Information Neural Estimation. In Dy, J.; and Krause, A., eds., *ICML 2021*, volume 80 of *Proceedings of Machine Learning Research*, 531–540. PMLR.

Fang, J.; Li, X.; Sui, Y.; Gao, Y.; Zhang, G.; Wang, K.; Wang, X.; and He, X. 2024. EXGC: Bridging Efficiency and Explainability in Graph Condensation. In Chua, T.; Ngo, C.; Kumar, R.; Lauw, H. W.; and Lee, R. K., eds., *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*, 721–732. ACM.

He, M.; Wei, Z.; and Wen, J. 2022. Convolutional Neural Networks on Graphs with Chebyshev Approximation, Revisited. In Koyejo, S.; Mohamed, S.; Agarwal, A.; Belgrave, D.; Cho, K.; and Oh, A., eds., *NeurIPS 2022*.

He, Y.; Xiao, L.; Zhou, J. T.; and Tsang, I. W. 2024. Multi-size Dataset Condensation. In *ICLR 2024*. OpenReview.net.

Hu, W.; Fey, M.; Ren, H.; Nakata, M.; Dong, Y.; and Leskovec, J. 2021a. OGB-LSC: A Large-Scale Challenge for Machine Learning on Graphs. In Vanschoren, J.; and Yeung, S., eds., *NeurIPS Datasets and Benchmarks 2021*.

Hu, Y.; You, H.; Wang, Z.; Wang, Z.; Zhou, E.; and Gao, Y. 2021b. Graph-MLP: Node Classification without Message Passing in Graph. *Arxiv*, abs/2106.04051.

Jin, W.; Zhao, L.; Zhang, S.; Liu, Y.; Tang, J.; and Shah, N. 2022. Graph Condensation for Graph Neural Networks. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.

Kipf, T. N.; and Welling, M. 2017. Semi-Supervised Classification with Graph Convolutional Networks. In *ICLR 2017,*. OpenReview.net.

Klicpera, J.; Bojchevski, A.; and Günnemann, S. 2019. Predict then Propagate: Graph Neural Networks meet Personalized PageRank. In *ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net.

Lewandowsky, J.; and Bauch, G. 2024. Theory and Application of the Information Bottleneck Method. *Entropy*, 26(3): 187.

Liu, M.; Li, S.; Chen, X.; and Song, L. 2022. Graph Condensation via Receptive Field Distribution Matching. *CoRR*, abs/2206.13697.

Liu, Y.; Bo, D.; and Shi, C. 2024. Graph Distillation with Eigenbasis Matching. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Peng, Z.; Huang, W.; Luo, M.; Zheng, Q.; Rong, Y.; Xu, T.; and Huang, J. 2020. Graph Representation Learning via Graphical Mutual Information Maximization. In Huang, Y.; King, I.; Liu, T.; and van Steen, M., eds., *WWW 2020*, 259–270. ACM / IW3C2.

Saxe, A. M.; Bansal, Y.; Dapello, J.; Advani, M.; Kolchinsky, A.; Tracey, B. D.; and Cox, D. D. 2018. On the Information Bottleneck Theory of Deep Learning. In *ICLR 2018*. OpenReview.net.

Sun, Q.; Chen, Z.; Yang, B.; Ji, C.; Fu, X.; Zhou, S.; Peng, H.; Li, J.; and Philip, S. Y. 2024. GC-Bench: An Open and Unified Benchmark for Graph Condensation. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Wu, F.; Jr., A. H. S.; Zhang, T.; Fifty, C.; Yu, T.; and Weinberger, K. Q. 2019. Simplifying Graph Convolutional Networks. In Chaudhuri, K.; and Salakhutdinov, R., eds., *ICML 2019*, volume 97 of *Proceedings of Machine Learning Research*, 6861–6871. PMLR.

Wu, T.; Ren, H.; Li, P.; and Leskovec, J. 2020. Graph Information Bottleneck. In Larochelle, H.; Ranzato, M.; Hadsell, R.; Balcan, M.; and Lin, H., eds., *NeurIPS 2020*.

Yang, B.; Wang, K.; Sun, Q.; Ji, C.; Fu, X.; Tang, H.; You, Y.; and Li, J. 2023. Does Graph Distillation See Like Vision Dataset Counterpart? In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.

Zeng, H.; Zhou, H.; Srivastava, A.; Kannan, R.; and Prasanna, V. K. 2020. GraphSAINT: Graph Sampling Based Inductive Learning Method. In *ICLR 2020*. OpenReview.net.

Zhang, Y.; Zhang, T.; Wang, K.; Guo, Z.; Liang, Y.; Bresson, X.; Jin, W.; and You, Y. 2024. Navigating Complexity: Toward Lossless Graph Condensation via Expanding Window Matching. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Zheng, X.; Zhang, M.; Chen, C.; Nguyen, Q. V. H.; Zhu, X.; and Pan, S. 2023. Structure-free Graph Condensation: From Large-scale Graphs to Condensed Graph-free Data. In Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; and Levine, S., eds., *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*.