Enhancing Graph Condensation via Key Information Reconstruction

Hongjia Xu, Sheng Zhou*, Zhuonan Zheng, Qiaoyu Tan, Jiawei Chen, Jiajun Bu, *Member, IEEE*

***Abstract*—Graph data mining techniques in real-world scenarios often encounter significant computational challenges, especially when the graph contains a large number of nodes and edges. Recently, Graph Condensation (GC) has emerged to offer data-centric solutions that address the challenge of graph volume, enhancing the efficiency of graph data mining and storage. Current methods in GC rely solely on optimizing heuristic metrics of one-way maintenance of key information in the condensed graph. However, the maintenance of key information may be insufficient due to the significant condensation ratio, yet these methods lack an effective mechanism to verify and compensate for that. To this end, this paper aims to enhance the maintenance of key information through a reconstruction-based alignment mechanism. More specifically, inspired by the Kolmogorov Complexity, we revisit the theoretical foundations of GC and propose a way-back mechanism that introduces a feedback loop of learning to reconstruct the original graph from the condensed graph, with the objective of key information alignment, namely the WbGC. We modify several GC methods with our mechanism, and the experiments show that our approach provides an enhanced solution for GC. Code is available at https://anonymous.4open.science/r/WbGC-1842.**

***Index Terms*—Graph condensation, data-centric AI, graph neural network.**

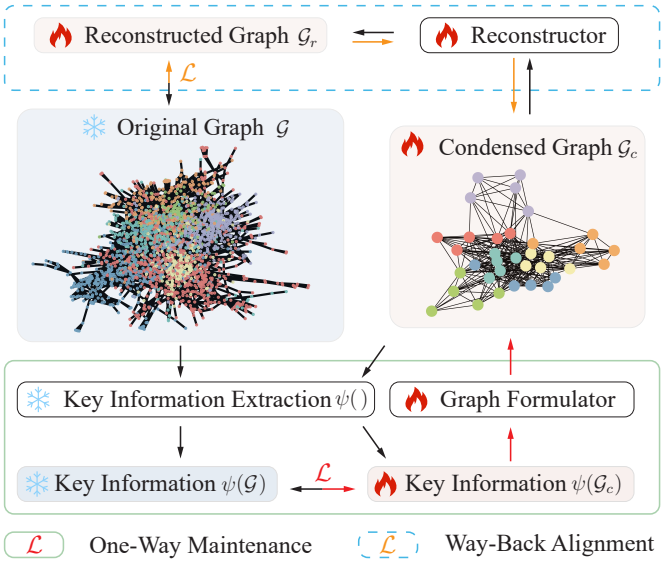


Fig. 1. Illustration of the proposed framework: an enhancement to existing GC methods that leverages a reconstruction mechanism to improve the alignment of key information. ❄ indicates that the corresponding module or data is fixed, while 🔥 signifies that it will be updated during the learning process.

I. Introduction

INFORMATION and patterns in biological systems [1], social networks [2], recommendation systems [3] and many other scenarios have been modeled as nodes and edges within a graph, and significant progress has been made in the development of techniques for mining knowledge from them. Graph data mining, particularly with the integration of Graph Neural Networks (GNNs) [4], is inherently complex due to the vast volume of real-world graphs, which typically consist of an enormous number of nodes and edges. Various model-centric efforts have been made to tackle this challenge, such as developing efficient architectures [5] and improving the efficiency of graph data mining pipelines [6]. Recently, **G**raph **C**ondensation (**GC**) [7]–[9] has been emerging as a *data-centric* solution that focuses on reducing the volume of graph-structured data itself. The motivation of GC is to compress large-scale graphs into smaller yet informative condensed graphs, such that models trained on these condensed graphs achieve comparable performance to those trained on the original graphs for specific graph data mining tasks.

As illustrated in fig. 1, existing GC methods [10]–[17] employ a key information maintenance architecture, where a shared relay model [8] is used to extract key information from both graphs. Subsequently, the condensed graph is updated by matching the extracted key information between the two graphs. Recent advances in GC have extensively explored the design of the relay model, focusing on defining and extracting key information within a graph. Specifically, some methods align the spectral and spatial properties of the original and condensed graphs by matching features such as Laplacian energy distributions [13], eigenvalues and corresponding eigenvectors [18], node spectral embedding and distributions [19], [20], topological structures [21], and other statistical metrics; another group of methods directly preserves the performance of models trained on the original graph by aligning their parameters (trajectories) [14], [22], training gradients [10], [17], [23], intermediate representation and output logits [19], [24], etc., with the models trained on the condensed graph. However, in these GC methods, since the extracted key information from the original graph is not updated, this process can essentially be interpreted as one-way maintenance

* Corresponding author
Hongjia Xu, Sheng Zhou, Zhuonan Zheng, and Jiajun Bu are with Zhejiang Key Laboratory of Accessible Perception and Intelligent Systems, College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: xu_hj@zju.edu.cn; zhousheng_zju@zju.edu.cn; zhengzn@zju.edu.cn; bjj@zju.edu.cn).
Qiaoyu Tan is with the College of Computer Science, New York University (Shanghai), China (e-mail: qiaoyu.tan@nyu.edu).
Jiawei Chen is with the College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China (e-mail: sleepyhunt@zju.edu.cn).

of the extracted information in the condensed graph, with no mechanism to explicitly verify the condensed graph against the original. Thus we raise a question: *Can they ensure the sufficient maintenance of key information?*

To answer this question, we evaluate several state-of-the-art GC methods on benchmark datasets, specifically by training models on the condensed graph using the same protocol as the original. Experimental evidence, which aligns well with the result reported in recent benchmark studies [25]–[27], indicates that there exist performance gaps between the condensed graph and the original. For instance, the performance of the node classification model trained on the condensed graphs can be consistently inferior to that of models trained on the original graph, especially on large benchmark datasets like Ogbn-Arxiv. Meanwhile, take one of the SoTA methods SFGC [14] as an example, the key information it aims to maintain in the condensed graph, i.e. the training trajectories of a model, is theoretically sufficient to perfectly reproduce an identically parameterized model. Therefore, the observed performance gap suggests that current GC methods may fail to sufficiently preserve all the key information within the condensed graph, yet no mechanism was provided to effectively address the potential loss of key information within the condensed graph.

From the perspective of information compression, sufficient maintenance of information indicates a lossless compression [28], which inherently requires the ability to recover the original graph data from its condensed ones. Therefore, similar to the structure of Autoencoders, a decoder structure to recover the original information from the condensed graphs, is needed to ensure the sufficiency of key information in the condensed graphs. However, although the Autoencoder structures such as in [29] are straightforward to implement, they are not directly applicable for enhancing GC. Specifically, without proper constraints, the decoder can be complex enough to generate complete data even from random noise; Similarly, the intermediate representation between the encoder and decoder can also be complex enough to cover all the information required for data recovery. Both of these scenarios go against the original motivation of efficiently maintaining information within the small condensed graph. Moreover, the original graph may contain a substantial amount of noise and redundancy, while GC was designed to exclude them from the condensed graph. Lossless reconstruction objectives such as the L_2-norm inherently tend to **reintroduce these undesirable elements** into the condensed graph.

In information theory, the minimal sufficient compression of an object (which is also the optimal solution for GC) is described by the theory of **Kolmogorov Complexity** [30]. In this paper, by extending the definition of Kolmogorov Complexity to graphs, we derived several corollaries that offer a new theoretical basis for GC (which will be detailed in section IV). Moreover, we propose to add a way-back mechanism that enhances the one-way GC process by focusing on aligning key information during the reconstruction of the original graph, advancing from one-way maintenance to comprehensive alignment of the key information. This mechanism can serve as a plugin to enhance any GC method by bridging the gap between the original and the condensed graphs.

Our experiments demonstrate that adding this **W**ay-**b**ack mechanism improves the performance of existing **GC** methods, namely the **WbGC**. In conclusion, the contributions of this paper are summarized as follows:

- Our research highlights that current GC methods operate in a one-way manner, which carries the risk of insufficient maintenance of key information from the original graph. Based on that, we propose WbGC to enhance GC methods with an extra objective of key information alignment.
- By incorporating the theory of Kolmogorov Complexity, we offer new theoretical bases for understanding and conducting the GC process.
- We demonstrate that by incorporating WbGC, the quality of condensed graphs generated from several GC methods is enhanced, not only in the baseline task but also in terms of cross-architecture capabilities.

II. Related Work

A. Graph Condensation

Dataset Distillation [31] has attracted noteworthy attention and yielded significant success in various domains, including images [32], textual documents [33], and more recently, *graphs*. The success of GC relies on identifying and maintaining the key information from the original graph to the condensed graph. According to recent surveys [7], [9], the optimization objective of GC methods can be categorized into three types (1) preserving information directly from the spectral and spatial domain of the original graph (*Graph-oriented*), (2) maintaining the model (mainly GNNs) capabilities for specific downstream tasks (*Model-oriented*), and (3) simultaneously accomplishing both.

Specifically, *Graph-oriented* methods align the spectral or Spatial information between the original and the condensed graph. Spectral information of a graph is defined as Laplacian Energy Distribution in SGDD [13], the pseudoinverse of Laplacian matrix in ReduceG [34], few smallest eigenvalues and corresponding eigenvectors in GR [35] and SCAL [18], spectral embeddings of nodes in GDEM [19], etc. The spatial domain of a graph is essentially the original topology and node features. Leveraging relationships within the original dataset by identifying the k-nearest neighbors and aggregating them is a common strategy, as in [36]–[38]. The distinction among these methods lies in how 'neighbors' are defined: whether based on topological proximity, feature space similarity, or other predefined 'equivalence relationships' under specific metrics, as in the G-Skeleton approach [38]. CTRL [21] and HGCond [17] use the cluster centers on the original features as initialization of the condensed graphs. Moreover, graph statistics (such as graph density, average degree, and degree variance) as graph properties are utilized to align the condensed graph and the original graph in [39] and [40]. These methods ensure that the condensed and the original graphs exhibit similar properties in the spectral or spatial domain by defining and aligning graph information.

On the other hand, the core idea of *Model-oriented* methods is to align the parameters of the model trained on the original

graph and the condensed graph as the optimization objective, using backpropagation to directly update the condensed graph. By doing so, the condensed graph is optimized to reproduce the capability (parameters) of the model trained on the original graph, maximizing the preservation of the models' performances on specific tasks. Specifically, starting with GCond [10], leveraging the training gradients [10]–[12], [16], [17], [21], [23], [41]–[46] and training trajectories [14], [22] of model parameters as key information to preserve in the condensed graph has become a widely adopted practice. For those models trained on the original graph, the gradients of parameters between epochs (i.e., gradient) or the parameters themselves at each epoch (i.e., trajectory) essentially represent the knowledge the model has learned from the original graph for the given task; Despite information within the model parameter and its training dynamics, KiDD [47], GC-SNTK [48], SGDC [49] aligns the optimal solution of the downstream task, which is quantified by the solution of Kernal-Ridge-Regression problem, to align the performance of the model trained on the condensed graph, and update the condensed graph by backpropagation; The outputs of a trained network, i.e., embeddings and logits, are also utilized to align the original and the condensed graph. For example, DisCo [50] and GCDM [24] align class centroids, i.e., mean of node embeddings for each class; SimGC [51] and TCGU [52] align the distribution parameters of embeddings; class-wise prediction logits have been adopted as an alignment constraint in GDEM [19], etc.

Moreover, as the *Model-oriented* and *Graph-oriented* objectives are not mutually conflict with each other, there are methods such as [13], [15], [19], [39], [53] that optimize the scaled graph from both graph-guided and model-guided objectives. Either way, the **objective** describes the principle to follow when condensing a graph, and the **formulations** of the condensed graphs describe how to generate them. These formulations can also be divided into two categories: Synthetic and Modification: *Synthetic formulations* require directly optimizing the condensed graph from either randomly or crafted initialization, according to the aforementioned optimization objectives. This process may lack transparency, making it difficult to understand how the condensed graph maintains the properties of the original graph and why it performs well on certain tasks. In contrast, *Modification formulations* have well-defined rules for editing and aggregating the original graph. For example, in molecular graphs, recurrent functional groups are condensed into hypernodes; in social networks, tightly connected communities are aggregated into hypernodes, etc. These predefined rules enhance the interpretability of both the small graph and the generation process, making it easier to understand how the condensed graph maintains the original characteristics. However, the versatility of graph modification methods is limited to specific scenarios and human expertise.

In this paper, we identify that current GC methods with one-way maintenance of key information cannot ensure the condensed graphs are sufficiently informative. Therefore, we propose to add a reconstruction mechanism to ensure that the key information within the condensed graph is sufficient, thereby enhancing the performance of existing GC methods.

III. Preliminary

For any matrix, symbols $\top, -, +, \mp$ denote transpose, inverse, pseudoinverse, and transposed pseudoinverse operations, respectively. On top of anything, a hat symbol $\hat{}$ indicates its optimal version. $f(;\theta) : I \rightarrow O$ indicates a function parameterized by θ, I and O represents the corresponding input and output spaces.

A. Problem Definition

Given a graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}, \mathbf{A} \in \{0,1\}^{N\times N}, \mathbf{X} \in \mathbb{R}^{N\times d}\}$ and its labels $\mathbf{Y} \in \{1, \cdots, C\}^M$ that depends on the downstream task, N is the number of nodes and d is the feature dimension, the goal of **Graph Condensation** is to formulate a function $GC(;\theta) : \mathcal{G} \rightarrow \mathcal{G}_c$ with predefined $\mathbf{Y}'$ that:

$$\mathcal{G}_c = GC(\mathcal{G}; \hat{\theta}),\ \hat{\theta} = \arg\min_{\theta} \mathcal{O}(\psi(\mathcal{G}_c, \mathbf{Y}'), \psi(\mathcal{G}, \mathbf{Y})) \quad (1)$$

where the graph condensation objective $\mathcal{O}$ measures the loss of key information, which is quantified by a function $\psi()$. $GC(;\theta)$ describes how to formulate condensed graphs, and is parameterized by θ. A **Graph Reconstruction** function $CG(;\phi) : \mathcal{G}_c \rightarrow \mathcal{G}$ representing the inverse operation of condensation, which is supposed to reconstruct $\mathcal{G}$ from $\mathcal{G}_c$.

IV. Revisiting GC from Information Theory

From the perspective of information compression, the GC process is equal to compressing the original graph object into a condensed graph object. In information theory, the compressed object being sufficiently informative is defined as *being able to reconstruct the original object through a computer program*, and the ***minimal sufficient*** compression of an object is described by Kolmogorov Complexity [30]:

Definition 1: Kolmogorov Complexity. This complexity refers to the length of the shortest binary program that outputs the object, quantifying the algorithmic information content of the binary object s:

$$K(s) = \min_{\mathcal{P}(\xi)=s} |\mathcal{P}(\xi)| \quad (2)$$

where $s \in \{0,1\}^{|s|}$ and $\mathcal{P}(\xi)$ is a computer program with input ξ, $||$ measures the binary length of a given binary object.

More specifically, we can assert that a minimal condensed graph is informative if they can reconstruct the original graph. In this chapter, by extending the definition of Kolmogorov Complexity to graphs, we derived several corollaries that offer a new theoretical basis for GC. Specifically, we found that combining the condensed graph with a reconstruction model provides a sufficient description of the original graph object, with its minimal version aligning precisely with the definition of Kolmogorov Complexity in information theory.

A. New Theory Basis for Graph Condensation

Under the context of Graph Condensation, assume there is an optimal condensed graph with N' nodes $\hat{\mathcal{G}}_c = GC(\mathcal{G}; \hat{\theta})$ that can not be condensed further (a minimal informative compression); and assume there is a reconstruction function $CG(;\hat{\phi})$ with a minimal number of parameters that can recover $\mathcal{G}$ from $\hat{\mathcal{G}}_c$. In such cases, a minimal description of

$\mathcal{G}$ can be expressed as the condensed graph $\hat{\mathcal{G}}_c$ along with a reconstruction network $CG(;\hat{\phi})$. The problem is that a Graph $\mathcal{G}$ is not naturally a binary object. So we define the binary representation of a graph as follows:

Definition 2: Note the binary representation of a graph as $\mathcal{G}_b$. Since $\{\mathbf{A}, \mathbf{X}\}$ are sufficient to describe $\mathcal{G}$, for b be the bits for float storage, the length of $\mathcal{G}_b$ can be written as:

$$|\mathcal{G}_b| = N \times N + N \times d \times b \tag{3}$$

Suppose there exists $CG(\hat{\mathcal{G}}_c;\hat{\phi})$ that is sufficient to describe $\mathcal{G}$ and each component is assumed to be minimal, we write the Kolmogorov Complexity of the original graph as:

$$k(\mathcal{G}) = |\hat{\mathcal{G}}_c| + |CG(;\hat{\phi})| \tag{4}$$

Theorem 1: [54] The upper bound of the Kolmogorov Complexity of a binary string x is:

$$\frac{1}{n}K\left(s \mid n = |s|\right) \leq H_b\left(\frac{1}{n}\sum_i s_i\right) + \frac{\log n}{2n} + O(1/n) \tag{5}$$

where H_b is the binary entropy function $H_b(p) = -p \times \log_2 p - (1-p) \times \log_2(1-p)$, $p \in [0-1]$. We use $\frac{\sum \mathcal{G}_b}{|\mathcal{G}_b|}$ to represent the proportion of non-zero positions in the binarized graph object $\mathcal{G}_b$.

Replacing $n, k(\mathcal{G})$ in theorem 1 by eq. (3) and eq. (4):

$$\begin{aligned} |\hat{\mathcal{G}}_c| + |CG(;\hat{\phi})| \leq H_b(\frac{\sum \mathcal{G}_b}{|\mathcal{G}_b|}) \times N \times (N + bd) \\ + \frac{\log(N \times (N + bd))}{2} + O(1) \end{aligned} \tag{6}$$

From this inequation, we propose the following corollaries:

Corollary 1: Fixing the size of the condensed graph, the minimum parameter space of the reconstruction function $CG(;\hat{\phi})$ is upper bounded, i.e., not infinitely complex, and is thus computable.

Corollary 2: Fixing the size of the reconstruction function, the size of the minimal condensed graph $\hat{\mathcal{G}}_c$ is upper bounded by constant. Any $\mathcal{G}_c$ larger than this constant must have redundant information.

Corollary 3: Suppose $GC(;\theta)$ and $CG(;\phi)$ have a dual structure, i.e., $|\theta| = |\phi|$, for a downstream model $GNN(;\gamma)$ which needs to be repeat trained for k times, Graph Condensation is useful, i.e., time consumed by acquiring $\mathcal{G}_c$ not exceed time saved by repeated training on $\mathcal{G}_c$, is ensured if:

$$\frac{|\theta| \times (N^2 + N'^2)}{|\gamma| \times (N^2 - N'^2)} \leq k \tag{7}$$

N, N' be the number of nodes in $\mathcal{G}$, $\mathcal{G}_c$; $|\theta|$ and $|\gamma|$ be the parameter space of the model of GC and downstream model.

Proof of these corollaries and experimental validation through a synthetic dataset will be presented in the Appendix sections A to C. Recall that we assumed the existence of an algorithm, $CG(;\phi)$, which is expected to solve the graph reconstruction problem. However, many problems are unsolvable, e.g., there is no universal algorithm for solving the halting problem. As in corollary 1, the parameter space of the assumed graph reconstruction model has an upper bound, indicating that finding this graph reconstruction model is an NP problem but not an unsolvable one. The corollary 2 indicates that the size of the minimum condensed graph is upper-bounded, thus providing a theoretical basis for selecting an appropriate condensation ratio when condensing an unknown dataset, eliminating the need for blind guessing and validation.

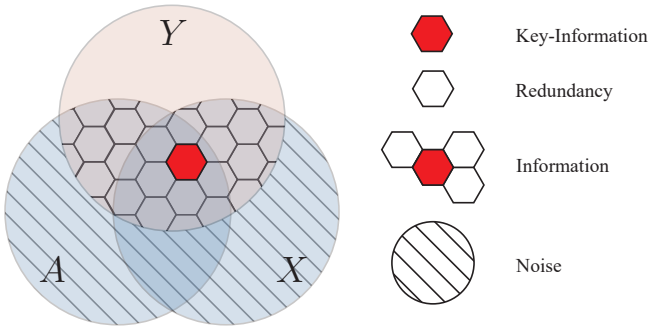


Fig. 2. Illustration of Information and Noise, Redundancy. A, X, Y represent the topology, node features, and label information of a graph, respectively.

Additionally, since GC is a one-time effort: the condensed graph can be reused indefinitely, thus the efficiency gains in fields such as Network Architecture Search, Continual Learning, Meta-learning, etc., can be linearly amplified with repeated downstream tasks on the condensed graph dataset. This renders the time consumed in the GC process itself negligible. However, this raises a question that has not yet been explored: *How many repetitions of downstream tasks are needed for the time saved by GC to surpass the time spent on the GC process itself?* The proposed corollary 3 directly answers the question mentioned above. **In summary**, the idea is that involving a comprehensive reconstruction mechanism ensures the condensed graph sufficiently retains all information from the original content, thereby enhancing the performance of GC methods. Meanwhile, to be considered minimally sufficient, both the reconstruction function and the intermediate representation (i.e., the condensed graph) should satisfy an upper bound on their binary length.

V. Enhancing GC with a Way-back Mechanism

Most existing GC methods $GC(;\theta) : \mathcal{G} \rightarrow \mathcal{G}_c$ are not reversible: they focus on condensing the graph without considering a mechanism to reconstruct and verify the condensed graph against the original, i.e., the $CG(;\phi) : \mathcal{G}_c \rightarrow \mathcal{G}$ in our proposed framework. To this end, we propose incorporating a feedback loop to reconstruct the original graph from the condensed version, thereby enhancing the retention of information, where the aforementioned Kolmogorov Complexity and corollaries provide the basis to formalize a minimal sufficiency scenario.

However, a comprehensive reconstruction will also bring the noise back to the compressed object. We present our understanding of the relation between useful Information, Noise, and Redundancy of a graph given task $\mathbf{Y}$ in fig. 2. By incorporating node features $\mathbf{X}$ and graph topology $\mathbf{A}$, models such as GNNs can effectively extract useful Information (the hexagonal grid coverage area in fig. 2) for task $\mathbf{Y}$, such as predicting node labels for classification, etc. In the meantime,

information that did not contribute to the current task is considered Noise (the parallel diagonal lines coverage area in fig. 2). The popular approaches of GC can be concluded as defining task-specific Key Information (the red hexagon in fig. 2) of a graph and aligning this key information between the condensed and the original graph. For example, the Key Information of a graph is defined as training gradients of GNNs in GCond [10], and the condensed graph is updated by optimizing an alignment objective. Directly incorporating such GC methods with a comprehensive reconstruction mechanism will reintroduce the Noise within the condensed graph.

Meanwhile, any information beyond the defined key information yet within the scope of useful Information can be considered Redundancy (the white hexagonal grid in fig. 2). While useful for training downstream tasks, they may not be strictly necessary. In graphs, such redundancy often manifests as the unique characteristics of graph data, where local patterns tend to repeat throughout the global structure. To enhance existing GC methods without introducing noise beyond the defined key information into the compressed graphs, the ideal situation is to recover only the redundant information within the reconstruction mechanism.

A. The Proposed Reconstructor

With a shared motivation of GC, the graph reduction [35], [55], [56] focuses on reducing node redundancy within the original graph and thereby maintaining key information in the reduced graph. Meanwhile, this process is reversible, i.e., redundancy can be recovered through the reverse operation of graph reduction. These methods employ a carefully designed projection matrix to aggregate similar nodes in the original graph, thereby reducing node redundancy. Specifically, $\mathbf{P} \in \mathbb{R}^{\mathbf{N}\times\mathbf{N}'}$ was defined as a projection matrix, indicating that nodes $\mathcal{V}_{(i)}$ in $\mathcal{G}$ were aggregated to a new node v_i' in $\mathcal{G}_c$:

$$\mathbf{P}_{i,j} = \begin{cases} 1 & \text{if } v_j' \in \mathcal{V}_{(i)} \\ 0 & \text{otherwise} \end{cases} \tag{8}$$

In the original definition [35], each row of $\mathbf{P}$ contains exactly one nonzero entry, indicating that the original node is aggregated only once. A more general formulation can be defined that, each row of the projection matrix $\mathbf{P}$ may contain an uncertain number of nonzero entries, ranging from 0 (the node is considered dropped, e.g., in [57]) to 1 and even multiple (aggregation clusters have overlapping issues, some node will be aggregated for multiple times). Further, the non-zero entries of $\mathbf{P}$ are not necessarily to be 1, the self-weighted attention mechanism can also be integrated. The graph reduction process using this projection matrix is written as follows:

$$\mathbf{A}' = \mathbf{P}^\top \mathbf{A}\mathbf{P},\ \mathbf{X}' = \mathbf{P}^+\mathbf{X} \tag{9}$$

From this formulation, the reverse operation of graph reduction can be easily derived and formulated as:

$$\mathbf{A_r} = \mathbf{P}^{\mp}\mathbf{A}'\mathbf{P}^+,\ \mathbf{X_r} = \mathbf{P}\mathbf{X}' \tag{10}$$

During the reverse operation, a graph with N' nodes can be lifted to the scale of N nodes, recovering the node redundancy. We propose optimizing the projection matrix to perform graph reconstruction as the inverse operation of graph reduction, thereby lifting the condensed graph back to the scale of the original graph. Here $\mathcal{G}_r = \{\mathbf{X_r}, \mathbf{A_r}, \mathbf{Y}\}$ is referred to as the reconstructed graph. An illustration of how $\mathbf{P}$ works with a toy example is provided in the Appendix section D.

The condensed graph, designed to retain only the key information from the original graph, is expected to be free of noise according to the optimization objective of GC eq. (11):

$$\hat{\mathcal{G}}_c = \arg\min_{\mathcal{G}_c} \ \mathcal{O}(\psi(\mathcal{G}), \psi(\mathcal{G}_c)) \tag{11}$$

According to our initial motivation, the reconstructed graph can be then used to align with the original graph. But how should the matrix $\mathbf{P}$ be optimized? If $\mathbf{P}$ indeed performs the function of redundancy elimination, then the amount of 'key information' contained in the reconstructed graph and the condensed graph should be equal. Given that the graph condensation process might not sufficiently maintain the key information, we propose an alignment approach to ensure that the key information in the reconstructed and original graphs is properly aligned, thereby getting the optimized $\hat{\mathbf{P}}$:

$$\hat{\mathbf{P}} = \arg\min_{\mathbf{P}} \ \mathcal{O}(\psi(\mathcal{G}), \psi(\mathcal{G}_\mathbf{r})) \tag{12}$$

B. Way-back Graph Condensation

Assuming that the objective in eq. (12) has converged, we have $\hat{\mathbf{P}}$ and thus $\hat{\mathcal{G}}_r$. Moreover, it is easy to observe that $\frac{\partial\mathcal{O}(\psi(\mathcal{G}),\psi(\mathcal{G}_r))}{\partial\mathcal{G}_c} = \frac{\partial\mathcal{O}(\psi(\mathcal{G}),\psi(\mathcal{G}_r))}{\partial\mathcal{G}_r}\frac{\partial\mathcal{G}_r}{\partial\mathcal{G}_c}$, and obviously $\frac{\partial\mathcal{G}_r}{\partial\mathcal{G}_c} = f(\mathbf{P})$. Thus, the knowledge from optimizing $\hat{\mathbf{P}}$ can be directly synthesized to the optimization of the condensed graph $\mathcal{G}_c$:

$$\begin{aligned}\hat{\mathcal{G}}_c = \arg\min_{\mathcal{G}_c,\mathbf{P}} \ & \mathcal{O}(\psi(\mathcal{G}), \psi(\mathcal{G}_c)) + \\ & \mathcal{O}(\psi(\mathcal{G}), \psi(\mathcal{G}_r))\end{aligned} \tag{13}$$

As illustrated in fig. 3, this optimization framework, namely the **WbGC**, can be applied directly to enhance existing GC methods as a plugin. The optimization of the additional objective not only learns to reconstruct but also enhances the alignment of key information of the base GC method.

C. The Proposed Reconstruction Constraint

The most widely used technique in GC is the gradient matching strategy [10]–[17]. Starting from GCond [10], this strategy involves matching the gradients generated by training the same graph neural network on both the original graph $\mathcal{G} = \{\mathbf{A}, \mathbf{X}, \mathbf{Y}\}$ and the condensed graph $\mathcal{G}_c = \{\mathbf{A}', \mathbf{X}', \mathbf{Y}'\}$ for the same task. This method ensures that the condensed graph $\mathcal{G}_c$ achieves similar performance to the original graph $\mathcal{G}$ on downstream tasks, thereby condensing **task-specific key information** from the original graph to the condensed graphs. The objective is written as eq. (14):

$$\begin{aligned}\min_{\mathbf{X}',\Phi} \mathrm{E}_{\boldsymbol{\theta}_0\sim P_{\boldsymbol{\theta}_0}} \Bigg[\sum_{t=0}^{T-1} D\Big(& \nabla_{\boldsymbol{\theta}}\mathcal{L}\left(\mathbf{GNN}_{\boldsymbol{\theta}_t}\left(g_\Phi\left(\mathbf{X}'\right), \mathbf{X}'\right), \mathbf{Y}'\right), \\ & \nabla_{\boldsymbol{\theta}}\mathcal{L}\left(\mathbf{GNN}_{\boldsymbol{\theta}_t}(\mathbf{A}, \mathbf{X}), \mathbf{Y}\right)\Big)\Bigg]\end{aligned} \tag{14}$$

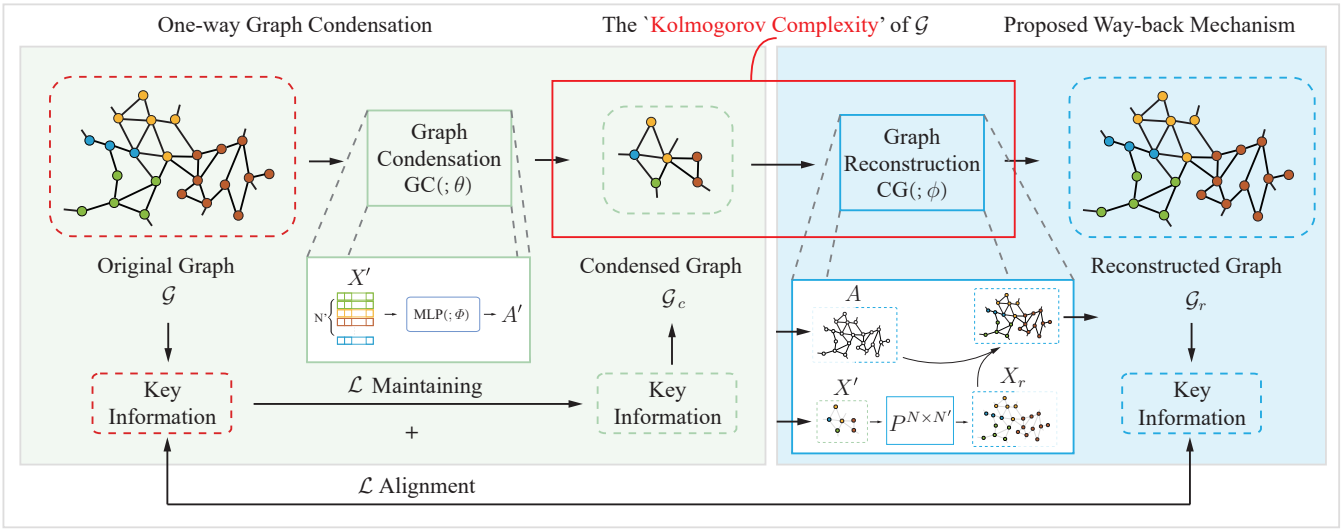


Fig. 3. Overall framework of the proposed method. Specifically, current GC methods involve optimizing the condensed graph by maintaining key information from the original graph. We propose that adding a graph reconstruction module and aligning the key information between the reconstructed graph and the original helps to enhance the process of GC.

with $\boldsymbol{\theta}_{t+1} = \text{opt}_{\boldsymbol{\theta}}\left(\mathcal{L}\left(\text{GNN}_{\boldsymbol{\theta}_t}\left(\mathbf{A}', \mathbf{X}'\right), \mathbf{Y}'\right)\right)$, $D(,)$ is the distance metric which sums the cosine distance of two gradient tensors layer by layer, t is the index of the training step, and g_Φ is a function modeling the condensed graph structure from the condensed node features, i.e., $\mathbf{A}' = g_\Phi\left(\mathbf{X}'\right)$, with

$$\mathbf{A}'_{ij} = \text{Sigmoid}\left(\frac{\text{MLP}_\Phi\left(\left[\mathbf{x}'_i; \mathbf{x}'_j\right]\right) + \text{MLP}_\Phi\left(\left[\mathbf{x}'_j; \mathbf{x}'_i\right]\right)}{2}\right) \tag{15}$$

As illustrated in eq. (13), we propose to simultaneously optimize the condensed graph $\mathcal{G}_c$ and a projection matrix $\mathbf{P}$ by incorporating an additional key information alignment objective into the existing GC objective. The proposed additional objective is written as eq. (16), with $CG(\mathcal{G}_c; \phi)$ be the proposed graph reconstructor (i.e., $\phi = \mathbf{P}$):

$$\min_{\mathbf{X}', \Phi, \phi} \text{E}_{\boldsymbol{\theta}_0 \sim P_{\boldsymbol{\theta}_0}}\left[\sum_{t=0}^{T-1} D\Big(\nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{GNN}_{\boldsymbol{\theta}_t}(CG(\mathcal{G}_c; \phi)), \mathbf{Y}), \nabla_{\boldsymbol{\theta}}\mathcal{L}(\mathbf{GNN}_{\boldsymbol{\theta}_t}(\mathbf{A}, \mathbf{X}), \mathbf{Y})\Big)\right] \tag{16}$$

D. Efficiency Concern

So far, the proposed method involves generating $\mathcal{G}_r$ by eq. (10), and then aligning the training gradients with the original graph. However, there is a process with an **unacceptable** time complexity: computing the pseudo-inverse of an $N \times N'$ matrix to perform $\mathbf{A_r} = \mathbf{P}^{\mp}\mathbf{A}'\mathbf{P}^{+}$. Although there are fast inversion methods like in [35], they are only applicable when the elements of the $\mathbf{P}$ matrix are all 0 or 1. Additionally, GNNs need to resample on $\mathbf{A_r}$ to perform message passing, which is also quite time-consuming. To this end, we propose replacing $\mathbf{A_r}$ in $\mathcal{G}_r$ with the original graphs's $\mathbf{A}$, i.e., $\mathcal{G}_r = \{\mathbf{X_r}, \mathbf{A}, \mathbf{Y}\}$. In this way, the efficiency of the proposed method is significantly improved, and the sampled neighborhood used for message passing on the original graph can be reused, eliminating the need for the GNN to resample subgraphs on the reconstructed graph.

In doing so, the structure noise within the original graph may be involved during the optimization of the final condensed graph. However, this is exactly why only the key information, extracted by GNNs' training between $\mathcal{G}_r$ and $\mathcal{G}$, should be aligned rather than sufficiently performing graph reconstruction. Besides, a sufficient graph reconstruction involves addressing critical tasks simultaneously such as feature completion [58], link prediction [59] and structure learning [37], etc., yet none of these subfields have yet offered a perfect solution. Note that the issue of label noise [60] and how to perform sufficient reconstruction is beyond this paper's scope.

Furthermore, it is not difficult to observe that $\mathbf{A_r}$ is entirely determined by the three components: $\mathbf{X}'$(the condensed graph features), the MLP g_Φ, and the $\mathbf{P}$ matrix. As WbGC already incorporates the information in $\mathbf{P}$ to the condensed graph, specifically the $\mathbf{X}'$, acquiring the complete information in $\mathbf{A_r}$ need only to acquire the knowledge from the MLP g_Φ. The ablation study of $\mathbf{A_r}$ being integrated into WbGC will be presented in table VII, and analyses from the view of information flow will be presented in Appendix section E.

E. The Initialization and Constraint on $\mathbf{P}$

Recall that the empirical definition of node redundancy is that they are similar, which implies that the aggregated node is more like a cluster center. Meanwhile, the node labels $\mathbf{Y}'$ of $\mathcal{G}_c$ are predefined through sampling at the start. Therefore, the training labels in the original graph $\mathbf{Y}_t \subset \mathbf{Y}$ can be integrated to initialize the $\mathbf{P}$ matrix. For nodes that were not in the training set split, we randomly select one of the neighboring labeled nodes as their pseudo-label $\mathbf{Y}_p$ in a recursive manner, and thus we have $\tilde{\mathcal{Y}} = \mathbf{Y}_t \cup \mathbf{Y}_p$. Further discussion and

ablation on pseudo-labeling is in the Appendix section F. Additionally, since connections in a graph may occur between different classes, we incorporate statistical information from the original graph. Considering that the original graph $\mathcal{G}$ has k independent classes of nodes, the tendency of connections between different classes can be calculated with a $k \times k$ matrix, namely the class relation matrix $\mathbf{C_r}$. Normalizing each row of $\mathbf{C_r}$ with a sum of 1, this matrix can statistically reflect the probability of connections between nodes with different class labels. Specifically, we initialize $\mathbf{P}^{\mathbf{N}\times\mathbf{N}'}$ as follows:

$$\mathbf{P}_{i,j} = \begin{cases} \alpha & \text{if } \tilde{\mathcal{Y}}_i = \mathbf{Y}'_j \text{ and rand}() \leq \beta \\ 1-\alpha & \text{if } \tilde{\mathcal{Y}}_i = \mathbf{Y}'_j \text{ and rand}() > \beta \\ (1-\alpha)\mathbf{C_r}[\tilde{\mathcal{Y}}_\mathbf{i}][\mathbf{Y}'_\mathbf{j}] & \text{otherwise} \end{cases} \tag{17}$$

where rand() generates a real number randomly within the interval $[0, 1]$, α is the global homophily ratio parameter, and β is another parameter controlling that nodes of the same class do not dominate excessively, i.e., not emphasizing the homophily assumption, with $\alpha, \beta \in \mathbb{R},\ 0 \leq \alpha, \beta \leq 1$. In this way, the initialized matrix $\mathbf{P}$ can aggregate information from nodes with different labels according to the connection preferences between different classes. To constrain the reconstructed node feature, an L_2-norm constraint on $\mathbf{P}$ is applied to make the sum of each row equal to 1:

$$\min_{\mathbf{P}} ||\mathbf{1_N} - \mathbf{P}\mathbf{1_{N'}}||_\mathbf{2} \tag{18}$$

In conclusion, $\mathcal{G}, \mathcal{G}_c, \mathcal{G}_r$ represent the original graph, the condensed graph, and the reconstructed graph respectively, the optimization of the proposed **WbGC** can be written as:

$$\begin{aligned} \mathcal{G}_c = \arg\min_{\mathcal{G}_c, \theta, \mathbf{P}} \ & \mathcal{O}(\psi(\mathcal{G}_c, \mathbf{Y}'), \psi(\mathcal{G}, \mathbf{Y})) + \\ & \mathcal{O}(\psi(\mathcal{G}_r, \mathbf{Y}), \psi(\mathcal{G}, \mathbf{Y})) + ||1_N - \mathbf{P}\mathbf{1_{N'}}||_\mathbf{2} \ , \\ \text{with } \mathcal{G}_c = & GC(\mathcal{G}; \theta),\ \mathcal{G}_r = CG(\mathcal{G}_c; \phi) \end{aligned} \tag{19}$$

where we follow GCond [10] that $\mathcal{O}$ is a matching function and ϕ quantifies the training gradient of a GNN model in the downstream task. The overall process of WbGC when the base method is GCond is presented in Algorithm 1. The algorithm can be summarized as three steps: (Step 1, line 1) Initialize the condensed graph and the projection matrix; (Step 2, lines 5-8) Perform graph reconstruction, and calculate the alignment loss $\mathcal{L}_{recons}$, i.e., the $\mathcal{O}(\psi(\mathcal{G}_r, \mathbf{Y}), \psi(\mathcal{G}, \mathbf{Y}))$ term in eq. (19); (Step 3, line 9-17) Calculate the alignment loss in the base GC method, and perform parameter optimization.

Algorithm 1: Our method: **WbGC**

Data: $\mathcal{G} = \{\mathbf{X}, \mathbf{A}\}$, $GNN(;\theta)$

1 Initialize $\mathcal{G}_c = \{\mathbf{X}', \mathbf{A}', \mathbf{Y}'\}$; Initialize $\mathbf{P}$;
2 **for** e *in epoches* **do**
3 $\mathbf{P_e}, \mathbf{X'_e}, \theta_\mathbf{e} \leftarrow$ Optimizers$\{\mathbf{P}, \mathbf{X}', \theta\}$;
4 Total_loss $= 0$;
5 $\mathbf{X}_r = \mathbf{P_e}\mathbf{X'_e}$; $\mathcal{G}_r = \{\mathbf{X}_r, \mathbf{A}\}$;
6 $\mathcal{L}^{\mathcal{O}} = \mathcal{L}\left(\text{GNN}\left(\mathbf{A}, \mathbf{X}; \theta_e\right), \mathbf{Y}\right)$;
7 $\mathcal{L}^{\mathcal{R}} = \mathcal{L}\left(\text{GNN}\left(\mathbf{A}, \mathbf{X}_r; \theta_e\right), \mathbf{Y}\right)$;
8 Total_loss $+= D\left(\nabla_{\theta_e}\mathcal{L}^{\mathcal{O}}, \nabla_{\theta_e}\mathcal{L}^{\mathcal{R}}\right)$;
9 **if** *base model* == *GCond* **then**
10 $\mathcal{L}^{\mathcal{C}} = \mathcal{L}\left(\text{GNN}\left(MLPs(\mathbf{X}'_e), \mathbf{X}'_e; \theta_e\right), \mathbf{Y}'\right)$;
11 Total_loss $+= D\left(\nabla_{\theta_e}\mathcal{L}^{\mathcal{O}}, \nabla_{\theta_e}\mathcal{L}^{\mathcal{C}}\right)$;
12 **else**
13 Calculate $\mathcal{L}_{base}$; # the loss of the base method
14 Total_loss $+= \mathcal{L}_{base}$;
15 **end**
16 Total_loss.$backward()$;
17 $\mathbf{P_{e+1}}, \mathbf{X'_{e+1}}, \theta_{\mathbf{e+1}} \leftarrow$ Optimizers$\{\mathbf{P}, \mathbf{X}', \theta\}.step()$;
18 **end**
19 return $\mathcal{G}_c$

VI. Experiment

A. Experiment Settings

Datasets. Following previously released benchmark papers [25]–[27], we evaluate GC methods on six commonly used datasets, i.e., Cora, Citeseer, Pubmed, Flickr, Ogbn-arxiv, and Reddit. We use the public splits for these datasets. The condense ratio, $r\% = \frac{N'}{N}$ is the fraction of condensed nodes to original nodes, and was fixed following the previous benchmarks. The labels of the condensed graph nodes $\mathbf{Y}'$ are predefined, specifically they were randomly sampled from the original labels to keep the class distribution the same as the original. Dataset statistics can be found in table I.

TABLE I
Dataset statistics

Dataset	Node	Edge	Class	Homophily	Feature
Cora	2,708	5,429	7	0.81	1,433
Citeseer	3,327	4,732	6	0.74	3,703
Flickr	89,250	899,756	7	0.24	500
Ogbn-Arxiv	169,343	1,166,243	40	0.65	128
Reddit	232,965	57,307,946	210	0.78	602
Pubmed	19,717	44,338	3	0.80	500

Baselines. The following methods are selected as baselines: (1) Traditional graph reduction methods: Random, Herding, K-Center, and (2) Optimization-based GC methods: GEOM [61], SFGC [14], GCond [10] and GDEM [19]. Among them, GEOM, SFGC, and GCond match the training trajectories of $\mathcal{G}$ and $\mathcal{G}_c$ as their main objective, while GDEM aims to match the eigenbasis, i.e., key information in spectral domain of the two graphs, as their main objective. We ran the official code with the provided hyperparameters for each baseline method and added our proposed graph reconstruction process as an additional loss term during the distillation stage of each baseline method in (2). Specifically, GCond and SGDD are end-to-end methods, while GEOM, SFGC, and GDEM have independent preprocessing stages. During the optimization stage of $\mathcal{G}_c$, we add an additional optimizer for the projection matrix $\mathbf{P}$, and add the alignment loss of reconstruction $\mathcal{L}_{recons} = \mathcal{O}(\phi(\mathcal{G}_r, \mathbf{Y}), \phi(\mathcal{G}, \mathbf{Y}))$ and the constraint on P to the original loss, namely, **GC + ours**. The hyperparameters used in our method are α and β, which were used to initialize the projection matrix $\mathbf{P}$. α was calculated according to dataset statistic, and β was set to be 0.5 for all experiments. The ablation of these two parameters was covered in table VI.

TABLE II
NODE CLASSIFICATION PERFORMANCES ON CORA, CITESEER, AND FLICKR, REPORTED AS MEAN ACCURACY (%) ± STANDARD DEVIATION. 'M', 'D', AND 'R' REPRESENT THE BASELINE AND OUR PROPOSED METHOD, DATASET, AND CONDENSATION RATIO (R), RESPECTIVELY

D/r / M	Cora				Citeseer				Flickr			
	1.3%	2.6%	5.2 %	↓↑	0.9 %	1.8 %	3.6 %	↓↑	0.1 %	0.5 %	1 %	↓↑
Full Graph		81.2 ± 0.2		- 00.0		71.7 ± 0.7		- 00.0		47.2 ± 0.1		- 00.0
Random	62.3 ± 0.8	72.6 ± 0.6	76.8 ± 0.5	↓ 10.6	45.3 ± 0.5	63.5 ± 0.4	69.3 ± 0.5	↓ 12.3	41.8 ± 0.6	43.1 ± 0.3	43.7 ± 0.2	↓ 4.3
Herding	68.2 ± 0.4	74.3 ± 0.4	76.5 ± 0.6	↓ 8.2	62.0 ± 0.9	67.9 ± 0.7	69.2 ± 0.5	↓ 5.3	40.9 ± 0.6	44.7 ± 0.3	44.1 ± 0.5	↓ 3.9
K-Center	65.6 ± 1.2	73.5 ± 1.0	76.6 ± 0.7	↓ 9.3	54.4 ± 0.9	62.7 ± 0.9	69.1 ± 0.5	↓ 9.6	42.4 ± 0.7	43.2 ± 0.5	43.7 ± 0.4	↓ 4.1
GEOM [61]	79.8 ± 0.6	79.8 ± 0.2	81.6 ± 1.4	↓ 0.8	69.0 ± 0.8	68.6 ± 0.2	70.8 ± 1.6	↓ 2.3	38.7 ± 0.1	40.2 ± 0.7	41.7 ± 0.3	↓ 7.0
GEOM + ours	80.2 ± 1.5	81.6 ± 0.4	82.2 ± 0.3	↑ 0.1	71.6 ± 1.2	70.0 ± 2.8	72.0 ± 1.2	↓ 0.5	46.2 ± 0.2	46.8 ± 0.3	48.6 ± 0.1	↓ 0.1
SFGC [14]	62.4 ± 0.2	64.8 ± 0.5	67.8 ± 1.1	↓ 16.2	55.8 ± 0.8	58.0 ± 0.0	59.4 ± 1.4	↓ 13.9	44.1 ± 0.7	45.8 ± 0.6	45.2 ± 1.5	↓ 2.1
SFGC + ours	61.0 ± 0.4	64.2 ± 0.4	68.2 ± 0.3	↓ 16.7	60.6 ± 1.4	60.8 ± 0.4	62.4 ± 0.2	↓ 10.4	45.9 ± 1.0	46.7 ± 0.4	45.8 ± 1.4	↓ 1.0
GCond [10]	79.2 ± 0.9	80.3 ± 0.9	81.0 ± 0.6	↓ 1.0	72.1 ± 1.4	72.2 ± 0.5	70.7 ± 0.9	↓ 0.1	46.1 ± 0.6	46.7 ± 0.3	46.7 ± 0.4	↓ 0.7
GCond + ours	81.4 ± 0.4	81.9 ± 0.6	81.4 ± 0.5	↑ 0.4	73.1 ± 0.5	73.2 ± 0.4	72.7 ± 0.3	↑ 1.3	47.0 ± 0.2	47.2 ± 0.1	47.3 ± 0.4	↓ 0.3
GDEM [19]	77.3 ± 1.3	81.1 ± 0.5	81.2 ± 0.6	↓ 1.3	72.8 ± 0.3	72.9 ± 0.5	OOM	↑ 1.1	49.4 ± 0.1	49.3 ± 0.2	49.5 ± 0.1	↑ 2.2
GDEM + ours	77.1 ± 1.1	81.2 ± 0.6	81.1 ± 0.2	↓ 1.4	73.3 ± 0.3	72.7 ± 0.4	OOM	↑ 1.3	50.3 ± 0.4	50.1 ± 0.2	50.1 ± 0.4	↑ 3.0

TABLE III
SECOND PART OF NODE CLASSIFICATION PERFORMANCES, ON OGBN-ARXIV, REDDIT, AND PUBMED.

D/r / M	Ogbn-arxiv				Reddit				Pubmed			
	0.05%	0.25%	0.5 %	↓↑	0.05 %	0.1 %	0.2 %	↓↑	0.08 %	0.15 %	0.3 %	↓↑
Full Graph		71.4 ± 0.1		- 00.0		93.9 ± 0.2		- 00.0		79.3 ± 0.2		- 00.0
Random	44.1 ± 1.5	57.0 ± 0.5	59.1 ± 0.6	↓ 18.0	47.3 ± 1.5	53.1 ± 1.9	75.9 ± 1.3	↓ 35.1	69.4 ± 0.2	73.3 ± 0.7	77.8 ± 0.3	↓ 5.8
Herding	49.7 ± 1.7	56.8 ± 0.5	58.5 ± 0.3	↓ 16.4	55.8 ± 2.8	67.3 ± 1.6	79.5 ± 1.5	↓ 26.3	73.3 ± 0.8	75.2 ± 0.6	78.0 ± 0.4	↓ 3.8
K-Center	48.5 ± 0.9	55.4 ± 0.8	59.7 ± 0.7	↓ 16.8	46.6 ± 1.8	67.3 ± 1.5	76.2 ± 1.2	↓ 30.5	68.9 ± 0.6	73.7 ± 0.4	77.8 ± 0.8	↓ 5.8
GEOM [61]	63.7 ± 2.1	66.3 ± 1.4	68.2 ± 1.4	↓ 5.3	81.4 ± 0.6	83.7 ± 0.5	85.8 ± 0.7	↓ 10.2	74.7 ± 1.7	77.5 ± 0.6	78.9 ± 0.7	↓ 2.2
GEOM + ours	64.2 ± 1.8	67.5 ± 0.6	69.1 ± 0.5	↓ 4.4	83.2 ± 0.3	84.3 ± 0.6	86.6 ± 1.0	↓ 9.2	76.4 ± 0.5	79.2 ± 0.6	78.6 ± 0.4	↓ 1.2
SFGC [14]	61.6 ± 0.7	66.3 ± 0.4	66.6 ± 0.5	↓ 6.5	83.1 ± 0.9	83.6 ± 0.4	84.1 ± 0.4	↓ 10.3	70.4 ± 0.4	72.2 ± 0.4	74.8 ± 0.6	↓ 6.8
SFGC + ours	63.5 ± 1.4	66.8 ± 0.4	67.2 ± 0.2	↓ 5.5	83.4 ± 0.5	82.4 ± 1.5	83.6 ± 1.6	↓ 10.7	71.2 ± 0.2	73.0 ± 0.6	74.9 ± 0.3	↓ 6.2
GCond [10]	56.2 ± 2.4	63.7 ± 0.5	63.9 ± 0.4	↓ 10.1	88.1 ± 1.1	88.4 ± 0.8	88.9 ± 1.2	↓ 5.4	77.6 ± 0.1	77.6 ± 0.5	77.8 ± 0.4	↓ 1.6
GCond + ours	56.5 ± 1.0	64.9 ± 0.4	65.7 ± 0.6	↓ 9.0	88.5 ± 0.3	89.8 ± 0.5	90.6 ± 0.4	↓ 4.2	78.1 ± 0.4	78.2 ± 0.3	78.2 ± 0.2	↓ 1.1
GDEM [19]	63.5 ± 0.9	63.9 ± 0.6	63.7 ± 0.5	↓ 7.7	92.6 ± 0.2	93.0 ± 0.1	93.2 ± 0.3	↓ 1.0	77.6 ± 0.6	78.7 ± 0.6	78.2 ± 1.3	↓ 1.1
GDEM + ours	64.1 ± 0.5	64.6 ± 0.4	64.2 ± 0.5	↓ 7.1	92.8 ± 0.3	93.2 ± 0.2	93.3 ± 0.2	↓ 0.8	78.3 ± 0.3	79.1 ± 0.5	78.8 ± 0.5	↓ 0.5

Evaluation. We evaluate the effectiveness of GC methods by evaluating the condensed graphs. The condensed graphs on six benchmark datasets are evaluated by the accuracy of training node classification models, where we train the GCN [62] on the condensed graphs from scratch and then evaluate their performance (ACC) on the test set of real graphs. Each experiment was repeated 5 times, and the mean test accuracy and standard deviation were reported. For each dataset, the average performances at different condensation rates are compared with the performances of models trained on the full dataset. These comparisons are presented in the fourth column of each dataset (↓↑), a ↓ indicates that the average performance decreased compared to the full dataset, while a ↑ indicates that it exceeded the performance of the full dataset.

B. Results and Analysis

The node classification performances on Cora, Citeseer, and Flickr are reported in table II, and the performances on Ogbn-arxiv, Reddit, and Pubmed are shown in table III. Firstly, **optimization-based GC methods consistently outperform the traditional methods**, including Random, Herding, and K-Center. This is because these optimization-based GC methods are task-oriented, which involves the powerful representation learning capabilities of graph models like GNNs, and the downstream task information was incorporated in the distillation process. Secondly, **as the condensation ratio increases, the performance of all methods improves**. In some baselines, e.g. the GDEM on Citeseer and Flickr, the models trained on the condensed graphs even outperform those trained on the full graphs. We believe this is because the total amount of information increases with the size of the training dataset, but so does the amount of noisy information. The superior performance of the condensed graphs may be due to the removal of such noisy information during GC.

Thirdly, each optimization-based GC method is modified as GC + ours, the results show that **our proposed additional reconstruction module enhances all performances of baselines on each dataset.** Not only does it allow these methods to be analyzed within our unified theoretical framework, but it also improves their performance, and particularly, **the perfor-**

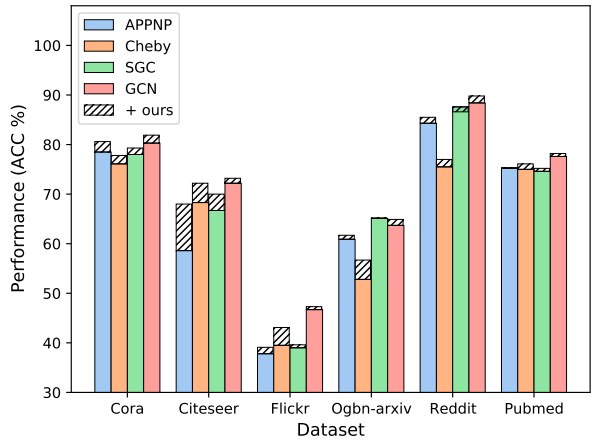


Fig. 4. Downstream GNN architecture ablation. Colors correspond to different architectures; The performance of the base method GCond is presented by the colored bar in the histogram, while the impact of incorporating our method is shown as the change in the bar, filled with diagonal lines.

mance of 'GCond + ours' on Cora and Citeseer exceeded that of the baseline trained on the full graph where the 'GCond' itself did not. However, in table III, neither the baselines nor our modifications exceeded the performance achieved when training on the original graph in Ogbn-arxiv, Reddit, and Pubmed, only GDEM and 'GDEM + ours' are close enough in Reddit and Pubmed. We believe that this might be due to the larger scale of these datasets, which contain more complex information and patterns, thus the objectives of key information alignment can be difficult to converge.

C. Ablation Study

This section focuses on the base GC method 'GCond' and performs ablation studies. Firstly, to test the generalizability of our proposed method on different downstream GNN architectures, an ablation study of four different GNN architectures is presented in fig. 4. Further, the ablation of cross-architecture on Pubmed and Cora are presented in table IV and table V, respectively (results on other datasets can be found in the Appendix section G). Note that in these tables, for the last two columns and last two rows, '↓ / ↑' reports the average, decrease/increase in the corresponding column or row between 'GCond' and 'GCond + ours'; 'STDEV' be the standard deviation of performances. The results show that with our enhancement, **'GCond + Ours' outperforms the baseline 'GCond' across all experiments**, and the **robustness of GCond over architecture ablation ('Stdev' over multiple experiments) is improved** over several datasets by adding our way-back mechanism, especially on Cireseer and Flickr.

Secondly, our proposed reconstruction model has three key components, namely the key information alignment loss, the initialization process, and the constraint on $\mathbf{P}$. The ablation of the three components is presented in table VI. The results indicate that **each of the three proposed modules contributes differently, with the best performance occurring when all modules are combined.**

Thirdly, the graph structure was not reconstructed, as the aim was to empirically seek the trade-off between effectiveness and efficiency. To ablate the effect of structure reconstruction, we conduct experiments with reconstructed $\mathbf{A}_r$. Given that GCond utilizes a learnable MLP() to predict $\mathbf{A}'$ from $\mathbf{X}'$, we can also use this MLP to predict $\mathbf{A}_r$ from $\mathbf{X}_r$, thereby introducing structural information (as illustrated in section V-D). Experiment results are presented in table VII.

TABLE IV
CROSS-ARCHITECTURE ABLATION ON PUBMED (R=0.15%). C, T MEANS THE CONDENSATION AND THE TEST GNN ARCHITECTURE.

Method	C\T	APPNP	Cheby	GCN	SGC	↓↑	STDEV
GCond	APPNP	77.3	74.1	76.6	67.4	-	4.51
+ ours		77.8	75.7	77.3	68.3	↑ 0.93	4.40
GCond	Cheby	63.4	66.0	41.5	49.4	-	11.6
+ ours		69.9	65.7	44.4	50.5	↑ 2.55	12.1
GCond	GCN	47.2	44.0	60.6	51.1	-	7.19
+ ours		58.5	59.4	63.3	56.8	↑ 8.78	2.75
GCond	SGC	75.3	75.0	77.6	74.6	-	1.34
+ ours		75.2	76.1	78.2	75.2	↑ 0.55	1.41
+ ours	↓↑	↑ 4.55	↑ 4.45	↑ 1.72	↑ 2.07	↑ 3.20	↓ 0.99
GCond	STDEV	13.8	14.4	16.9	12.4	-	13.11
+ ours		8.55	8.12	15.8	11.2	↓ 2.52	10.59

TABLE V
CROSS-ARCHITECTURE ABLATION ON CORA(R=2.6%). C, T MEANS THE CONDENSATION AND THE TEST GNN ARCHITECTURE.

Method	C\T	APPNP	Cheby	GCN	SGC	↑↓	STDEV
GCond	APPNP	77.9	64.4	76.5	68.3	-	6.48
+ ours		78.9	66.9	80.7	71.7	↑ 2.77	6.41
GCond	Cheby	52.0	46.4	24.9	30.7	-	12.8
+ ours		50.2	47.8	31.5	32.1	↑ 1.90	9.98
GCond	GCN	61.9	69.1	74.6	59.3	-	6.95
+ ours		65.3	67.3	75.4	59.9	↑ 0.75	6.42
GCond	SGC	78.5	76.1	80.3	78.0	-	1.72
+ ours		80.6	77.8	81.9	79.3	↑ 1.68	1.75
+ ours	↓↑	↑ 1.18	↑ 0.95	↑ 3.30	↑ 1.68	↑ 1.78	↓ 0.84
GCond	STDEV	12.9	12.7	26.2	20.3	-	17.20
+ ours		14.1	12.5	24.0	20.7	↓ 0.39	16.81

TABLE VI
MODULE ABLATION. THE ABLATION OF INITIALIZATION ALSO COVERS THE ABLATION OF HYPERPARAMETERS USED TO INITIALIZE $\mathbf{P}$.

Dataset	GCond	+①	+①+②	+①+③	+①+②+③
Cora-r2.6	80.3	81.3	81.4	81.6	81.9
Citeseer-r1.8	72.2	72.4	72.8	73.2	73.2
Flickr-r0.5	46.7	47.1	46.4	47.0	47.2
Arxiv-r0.25	63.7	64.0	64.5	65.1	64.9
Reddit-r0.1	88.4	89.0	89.6	89.4	89.8
Pubmed-r0.15	77.6	78.0	78.0	78.2	78.2

ps: ①=$\mathcal{L}_{recons}$, ②=Initialization of $\mathbf{P}$, ③=$\mathcal{L}$ Constraint on $\mathbf{P}$

This approach (simultaneously reconstructing the graph structure) **shows improvement on small dataset but does not on larger one**. This may be due to the complex topology of large datasets, where a simple MLP is insufficient for effective link prediction. Moreover, this process significantly increases process time, as the GNN needs to resample the neighborhood.

TABLE VII
ABLATION STUDY WITH RECONSTRUCTED GRAPH STRUCTURE

Method	Cora (r=2.6%)	Ogbn-arxiv (r=0.25%)
GCond	80.3±0.9	63.7±0.5
+ ours ($G_r = \{X_r, A\}$)	81.9±0.6	64.9±0.4
+ ours ($G_r = \{X_r, MLP(X_r)\}$)	82.2±0.7	62.9±0.8

D. Time Complexity Analysis

Let r be the number of sampled neighbors per node, d for embedding dimension and L be layers of GCN, e be epochs. Our additional operation is 1. Calculating $\mathbf{X_r}$; 2. Forward of $\mathcal{G}_r$; 3. Gradient and backward propagation on $\mathbf{P}$; So the additional time complexity is $e*(O(NN'd)+O(r^L Nd^2)+O(LNN'))$, the additional space complexity is $O(NN')$. Part of the actual consumed time is presented in table VIII:

TABLE VIII
PART OF THE ACTUAL RUNNING TIME ON CORA AND ARXIV

Cora		(r=1.3%)	(r=2.6%)	(r=5.2%)
Time (50 epoch)	GCond	39.1 (s)	41.5 (s)	43.5 (s)
	+ ours	50.2 (s)	52.3 (s)	56.0 (s)
Ogbn-arxiv		(r=0.05%)	(r=0.25%)	(r=0.5%)
Time (50 epoch)	GCond	816 (s)	1345 (s)	1455 (s)
	+ ours	1097 (s)	1414 (s)	1971 (s)

VII. DISCUSSION

A. Connection to Information Bottleneck Principle

From the Information Bottleneck perspective, specifically the supervised IB Lagrangian:

$$\max\{I(\mathbf{Y};\mathbf{Z_X}) - \beta \mathbf{I}(\mathbf{X};\mathbf{Z_X})\} \tag{20}$$

has been interpreted as representations $\mathbf{Z_X}$ should contain only the ***minimal sufficient*** information for predicting $\mathbf{Y}$. We found that this formulation is particularly similar to gradient-based GC, where the condensed graph is also expected to contain the ***minimal sufficient*** information for training the task only, and the noise in the original graph is expected to be minimized in the condensed graph. Take GCond as an example, they: 1) training on the original graph, get trajectory gt_1; 2) training on the condensed graph, get trajectory gt_2; 3) minimize the distance between gt_1 and gt_2; 4) Optimize the condensed graph through gradient backpropagation.

It is clear that the training gradients contain only information relevant to the current task, with no inclusion of task-independent noise. The final result can be summarized that the following objectives are being maximized simultaneously:

$$\begin{aligned}\hat{\mathcal{G}}_c = \arg\max_{\mathcal{G}_c,\theta}\ & a \times (I(\mathbf{Y};\psi(\mathcal{G})) - \beta_1 I(\mathcal{G};\psi(\mathcal{G}))) \\ & + b \times (I(\mathbf{Y}';\psi(\mathcal{G}_c)) - \beta_2 I(\mathcal{G}_c;\psi(\mathcal{G}_c))) \\ & + c \times (I(\psi(\mathcal{G});\psi(\mathcal{G}_c)))\end{aligned} \tag{21}$$

where $\psi(\mathcal{G}) = \nabla_\theta \mathcal{L}(GNN(\mathcal{G};\theta),\mathbf{Y})$ is the gradient of parameters θ under learning task $\mathcal{L}(GNN(\mathcal{G};\theta),\mathbf{Y})$, $\mathcal{L}$ could be the cross-entropy loss. Our method can be thereby interpreted as adding two more terms to optimize:

$$\begin{aligned}\hat{\mathcal{G}}_c = \arg\max_{\mathcal{G}_c,\theta,\phi}\ & a \times (\cdots) + b \times (\cdots) + c \times (\cdots) \\ & + d \times ((I(\mathbf{Y};\psi(\mathcal{G}_r)) - \beta_3 I(\mathcal{G}_r;\psi(\mathcal{G}_r)))) \\ & + e \times (I(\psi(\mathcal{G});\psi(\mathcal{G}_r)))\end{aligned} \tag{22}$$

where $\mathcal{G}_r = CG(\mathcal{G}_c;\phi)$. From these formulations, it can be observed that the denoise objective terms from the IB interpretation is effective in our method, i.e., our method will attempt to eliminate the noise within the reconstructed graph, while the key information alignment between $\mathcal{G}$ and $\mathcal{G}_c$ has been enhanced by further constraint.

Kolmogorov Complexity provides the most general description of data, independent of any specific task or goal. In contrast, the information bottleneck principle, as our proposed method, focuses on extracting information from data that is relevant to a particular task, optimizing for the task at hand. To some extent, Kolmogorov Complexity essentially describes the optimal and minimal solution of the information bottleneck principle, where it is not the original object but the extracted key information that is being compressed (i.e., $\{\nabla_{\theta_0}, \nabla_{\theta_1}, \ldots, \nabla_{\theta_e}\}$ is actually being compressed and reconstructed in our method). From partial to sufficient reconstruction, we provide a theoretical foundation for using the encoder-decoder architecture to verify the sufficiency of GC.

B. Limitations

The effectiveness of the proposed method is demonstrated in the previous sections, as it enhances the performance of all baseline GC methods as a plugin. Notably, many baselines exhibit a significant performance gap when trained on the condensed graph versus the full graph, especially in large-scale datasets like Ogbn-arxiv (169,343 nodes) and Reddit (232,965 nodes). Although integrating our method has enhanced the performance of all baselines and partially narrowed the performance gap, this gap still persists on large datasets. Moreover, although we have made progress on the robustness of model architecture ablation, the standard deviations are still not low enough to perform tasks like network architecture search. In such a scenario, the performances of the condensed graphs must be consistently high to be considered reliable substitutions for the original graphs.

VIII. CONCLUSION

In this study, we propose to address the challenge of insufficient graph data condensation via Way-back Graph Condensation (WbGC), a novel framework that integrates a reconstruction mechanism with constraints to enhance the key information alignment of current GC methods. Furthermore, the theoretical framework of GC is enriched by incorporating the theory of Kolmogorov Complexity, providing a deeper understanding of graph information compression. In the future, it will be highly valuable to explore the performance limits of GC, specifically identifying under what conditions the condensed graphs could outperform the original, and perform consistently high against the change of model architectures.

REFERENCES

[1] A. Fout, J. Byrd, B. Shariat, and A. Ben-Hur, "Protein interface prediction using graph convolutional networks," *Advances in neural information processing systems*, vol. 30, 2017.

[2] Y. Li, J. Fan, Y. Wang, and K.-L. Tan, "Influence maximization on social graphs: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 10, pp. 1852–1872, 2018.

[3] C. C. Aggarwal and C. C. Aggarwal, "An introduction to recommender systems," *Recommender systems: the textbook*, pp. 1–28, 2016.

[4] B. Zhang, C. Fan, S. Liu, K. Huang, X. Zhao, J. Huang, and Z. Liu, "The expressive power of graph neural networks: A survey," *IEEE Transactions on Knowledge and Data Engineering*, 2024.

[5] Z. Zhong, C.-T. Li, and J. Pang, "Hierarchical message-passing graph neural networks," *Data Mining and Knowledge Discovery*, vol. 37, no. 1, pp. 381–408, 2023.

[6] W. Hamilton, Z. Ying, and J. Leskovec, "Inductive representation learning on large graphs," *Advances in neural information processing systems*, vol. 30, 2017.

[7] M. Hashemi, S. Gong, J. Ni, W. Fan, B. A. Prakash, and W. Jin, "A comprehensive survey on graph reduction: Sparsification, coarsening, and condensation," in *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*. ijcai.org, 2024, pp. 8058–8066.

[8] X. Gao, J. Yu, T. Chen, G. Ye, W. Zhang, and H. Yin, "Graph condensation: A survey," *IEEE Transactions on Knowledge and Data Engineering*, vol. 37, no. 4, pp. 1819–1837, 2025.

[9] H. Xu, L. Zhang, Y. Ma, S. Zhou, Z. Zheng, and B. Jiajun, "A survey on graph condensation," *arXiv preprint arXiv:2402.02000*, 2024.

[10] W. Jin, L. Zhao, S. Zhang, Y. Liu, J. Tang, and N. Shah, "Graph condensation for graph neural networks," in *International Conference on Learning Representations*, 2021.

[11] W. Jin, X. Tang, H. Jiang, Z. Li, D. Zhang, J. Tang, and B. Yin, "Condensing graphs via one-step gradient matching," in *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022, pp. 720–730.

[12] X. Li, K. Wang, H. Deng, Y. Liang, and D. Wu, "Attend who is weak: Enhancing graph condensation via cross-free adversarial training," *arXiv preprint arXiv:2311.15772*, 2023.

[13] B. Yang, K. Wang, Q. Sun, C. Ji, X. Fu, H. Tang, Y. You, and J. Li, "Does graph distillation see like vision dataset counterpart?" in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[14] X. Zheng, M. Zhang, C. Chen, Q. V. H. Nguyen, X. Zhu, and S. Pan, "Structure-free graph condensation: From large-scale graphs to condensed graph-free data," *Advances in Neural Information Processing Systems*, vol. 36, 2024.

[15] X. Gao, T. Chen, Y. Zang, W. Zhang, Q. V. H. Nguyen, K. Zheng, and H. Yin, "Graph condensation for inductive node representation learning," *arXiv preprint arXiv:2307.15967*, 2023.

[16] J. Gao and J. Wu, "Multiple sparse graphs condensation," *Knowledge-Based Systems*, vol. 278, p. 110904, 2023.

[17] J. Gao, J. Wu, and J. Ding, "Heterogeneous graph condensation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 7, pp. 3126–3138, 2024.

[18] Z. Huang, S. Zhang, C. Xi, T. Liu, and M. Zhou, "Scaling up graph neural networks via graph coarsening," in *Proceedings of the 27th ACM SIGKDD conference on knowledge discovery & data mining*, 2021, pp. 675–684.

[19] Y. Liu, D. Bo, and C. Shi, "Graph distillation with eigenbasis matching," in *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net, 2024.

[20] Y. Liu, R. Qiu, Y. Tang, H. Yin, and Z. Huang, "Puma: Efficient continual graph learning for node classification with graph condensation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 37, no. 1, pp. 449–461, 2025.

[21] T. Zhang, Y. Zhang, K. Wang, K. Wang, B. Yang, K. Zhang, W. Shao, P. Liu, J. T. Zhou, and Y. You, "Two trades is not baffled: Condense graph via crafting rational gradient matching," *arXiv preprint arXiv:2402.04924*, 2024.

[22] Y. Zhang, T. Zhang, K. Wang, Z. Guo, Y. Liang, X. Bresson, W. Jin, and Y. You, "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, 2024.

[23] X. Gao, H. Yin, T. Chen, G. Ye, W. Zhang, and B. Cui, "Robgc: Towards robust graph condensation," *arXiv preprint arXiv:2406.13200*, 2024.

[24] M. Liu, S. Li, X. Chen, and L. S, "Graph condensation via receptive field distribution matching," *arXiv preprint arXiv:2206.13697*, 2022.

[25] Q. Sun, Z. Chen, B. Yang, C. Ji, X. Fu, S. Zhou, H. Peng, J. Li, and S. Y. Philip, "Gc-bench: An open and unified benchmark for graph condensation," in *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

[26] Y. Liu, R. Qiu, and Z. Huang, "Gcondenser: Benchmarking graph condensation," *arXiv preprint arXiv:2405.14246*, 2024.

[27] S. Gong, J. Ni, N. Sachdeva, C. Yang, and W. Jin, "Gc4nc: A benchmark framework for graph condensation on node classification with new insights," *arXiv preprint arXiv:2406.16715*, 2024.

[28] L. Zhou, K. S. Candan, and J. Zou, "Deepmapping: Learned data mapping for lossless compression and efficient lookup," in *2024 IEEE 40th International Conference on Data Engineering (ICDE)*. IEEE, 2024, pp. 1–14.

[29] H. Chen, H. Wang, H. Chen, Y. Zhang, W. Zhang, and X. Lin, "Denoising variational graph of graphs auto-encoder for predicting structured entity interactions," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 3, pp. 1016–1029, 2023.

[30] A. N. Kolmogorov, "On tables of random numbers," *Sankhyā: The Indian Journal of Statistics, Series A*, pp. 369–376, 1963.

[31] R. Yu, S. Liu, and X. Wang, "Dataset distillation: A comprehensive review," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2023.

[32] G. Cazenavette, T. Wang, A. Torralba, A. A. Efros, and J.-Y. Zhu, "Dataset distillation by matching training trajectories," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 4750–4759.

[33] J.-g. Yao, X. Wan, and J. Xiao, "Recent advances in document summarization," *Knowledge and Information Systems*, vol. 53, pp. 297–336, 2017.

[34] G. Bravo Hermsdorff and L. Gunderson, "A unifying framework for spectrum-preserving graph sparsification and coarsening," *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[35] A. Loukas, "Graph reduction with spectral and cut guarantees." *J. Mach. Learn. Res.*, vol. 20, no. 116, pp. 1–42, 2019.

[36] C. Deng, Z. Zhao, Y. Wang, Z. Zhang, and Z. Feng, "Graphzoom: A multi-level spectral approach for accurate and scalable graph embedding," *arXiv preprint arXiv:1910.02370*, 2019.

[37] Y. Jin, A. Loukas, and J. JaJa, "Graph coarsening with preserved spectral properties," in *International Conference on Artificial Intelligence and Statistics*. PMLR, 2020, pp. 4452–4462.

[38] L. Cao, H. Deng, Y. Yang, C. Wang, and L. Chen, "Graph-skeleton:˜ 1% nodes are sufficient to represent billion-scale graph," in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 570–581.

[39] J. Xu, R. Huang, X. Jiang, Y. Cao, C. Yang, C. Wang, and Y. Yang, "Better with less: A data-active perspective on pre-training graph neural networks," *Advances in Neural Information Processing Systems*, vol. 36, pp. 56 946–56 978, 2023.

[40] M. Kumar, A. Sharma, S. Saxena, and S. Kumar, "Featured graph coarsening with similarity guarantees," in *International Conference on Machine Learning*. PMLR, 2023, pp. 17 953–17 975.

[41] J. Wu, N. Lu, Z. Dai, W. Fan, S. Liu, Q. Li, and K. Tang, "Backdoor graph condensation," *arXiv preprint arXiv:2407.11025*, 2024.

[42] M. Ding, X. Liu, T. Rabbani, and F. Huang, "Faster hyperparameter search on graphs via calibrated dataset condensation," in *NeurIPS 2022 Workshop: New Frontiers in Graph Learning*, 2022.

[43] J. Fang, X. Li, Y. Sui, Y. Gao, G. Zhang, K. Wang, X. Wang, and X. He, "EXGC: bridging efficiency and explainability in graph condensation," in *Proceedings of the ACM on Web Conference 2024, WWW 2024, Singapore, May 13-17, 2024*. ACM, 2024, pp. 721–732.

[44] R. Mao, W. Fan, and Q. Li, "Gcare: Mitigating subgroup unfairness in graph condensation through adversarial regularization," *Applied Sciences*, vol. 13, no. 16, p. 9166, 2023.

[45] Y. Liu and Y. Shen, "Tinygraph: joint feature and node condensation for graph neural networks," *arXiv preprint arXiv:2407.08064*, 2024.

[46] B. Yan, "Federated graph condensation with information bottleneck principles," *arXiv preprint arXiv:2405.03911*, 2024.

[47] Z. Xu, Y. Chen, M. Pan, H. Chen, M. Das, H. Yang, and H. Tong, "Kernel ridge regression-based graph dataset distillation," in *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023, pp. 2850–2861.

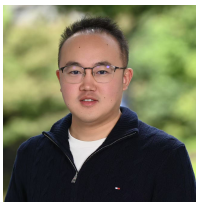
[48] L. Wang, W. Fan, J. Li, Y. Ma, and Q. Li, "Fast graph condensation with structure-based neural tangent kernel," in *Proceedings of the ACM on Web Conference 2024*, 2024, pp. 4439–4448.

[49] Y. Wang, X. Yan, S. Jin, H. Huang, Q. Xu, Q. Zhang, B. Du, and J. Jiang, "Self-supervised learning for graph dataset condensation," in *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2024, pp. 3289–3298.
[50] Z. Xiao, S. Liu, Y. Wang, T. Zheng, and M. Song, "Disentangled condensation for large-scale graphs," *arXiv preprint arXiv:2401.12231*, 2024.
[51] Z. Xiao, Y. Wang, S. Liu, H. Wang, M. Song, and T. Zheng, "Simple graph condensation," in *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*. Springer, 2024, pp. 53–71.
[52] F. Li, X. Wang, D. Cheng, W. Zhang, Y. Zhang, and X. Lin, "Tcgu: Data-centric graph unlearning based on transferable condensation," *arXiv preprint arXiv:2410.06480*, 2024.
[53] Q. Feng, Z. Jiang, R. Li, Y. Wang, N. Zou, J. Bian, and X. Hu, "Fair graph distillation," in *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.
[54] T. M. Cover, *Elements of information theory*. John Wiley & Sons, 1999.
[55] J. Dietrich, L. Chang, L. Qian, L. M. Henry, C. McCartin, and B. Scholz, "Efficient sink-reachability analysis via graph reduction," *IEEE Transactions on Knowledge and Data Engineering*, vol. 34, no. 11, pp. 5321–5335, 2021.
[56] Z. Liu, C. Wang, H. Feng, and Z. Chen, "Efficient unsupervised graph embedding with attributed graph reduction and dual-level loss," *IEEE Transactions on Knowledge and Data Engineering*, vol. 36, no. 12, pp. 8120–8134, 2024.
[57] D. Luo, W. Cheng, W. Yu, B. Zong, J. Ni, H. Chen, and X. Zhang, "Learning to drop: Robust graph neural network via topological denoising," in *Proceedings of the 14th ACM international conference on web search and data mining*, 2021, pp. 779–787.
[58] H. Taguchi, X. Liu, and T. Murata, "Graph convolutional networks for graphs containing missing features," *Future Generation Computer Systems*, vol. 117, pp. 155–168, 2021.
[59] A. Kumar, S. S. Singh, K. Singh, and B. Biswas, "Link prediction techniques, applications, and performance: A survey," *Physica A: Statistical Mechanics and its Applications*, vol. 553, p. 124289, 2020.
[60] B. Frénay and M. Verleysen, "Classification in the presence of label noise: a survey," *IEEE transactions on neural networks and learning systems*, vol. 25, no. 5, pp. 845–869, 2013.
[61] Y. Zhang, T. Zhang, K. Wang, Z. Guo, Y. Liang, X. Bresson, W. Jin, and Y. You, "Navigating complexity: Toward lossless graph condensation via expanding window matching," in *Forty-first International Conference on Machine Learning*.
[62] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Hongjia Xu received the degree of B.E. in Computer Science and Technology from Zhejiang University in 2020. He is currently working toward the PhD degree with the College of Computer Science and Technology, Zhejiang University, China. His current research interests include Data mining, Graph Neural Networks, and Dataset Distillation.

Sheng Zhou received the PhD degree from the College of Computer Science and Technology, Zhejiang University, Hangzhou, Zhejiang, China. He is currently working as an assistant professor with the College of Software and Engineering, Zhejiang University. He is working with the Zhejiang Provincial Key Laboratory of Service Robot, College of Computer Science, Zhejiang University. His current research interests include Data mining and Multimodal Large-Language-Models (MLLMs).

Zhuonan Zheng received the BS degree in software engineering from Zhejiang University of Technology, China, in 2021. He is currently working towards the PhD degree with the College of Computer Science and Technology, Zhejiang University. His research interests include graph neural networks, graph heterophily, and data mining.



Qiaoyu Tan is currently an assistant professor in the Computer Science Department at New York University (Shanghai). He earned his PhD in Computer Science and Engineering from Texas A&M University in 2023. His research interests focus on machine learning and data mining, with particular emphasis on graph machine learning, foundation models, multimodal learning, and their applications in bioinformatics and healthcare. His innovative work has been featured at top-tier conferences such as KDD, WWW, WSDM, SIGIR, NeurIPS, ACL, NAACL, and TKDE. Prior to joining NYU Shanghai, he worked as a research intern at Alibaba and Samsung Research America.

Jiawei Chen is a Research Fellow in School of Computer Science and Technology, Zhejiang University. His research interests include Information Retrieval, Graph Learning, and Large Language Model. He received Ph.D. in Computer Science from Zhejiang University in 2020. He has published over 60 academic papers on top-tier conferences or journals such as WWW, SIGIR, AAAI, KDD, TOIS and TKDE, and won the best paper awards on WSDM'25 and the best paper honorable mentions on SIGIR'23.

Jiajun Bu received the BS and Ph.D. degrees in computer science from Zhejiang University, China, in 1995 and 2000, respectively. He is currently a distinguished professor with the College of Computer Science, Zhejiang University. He is the Associate Dean of the Graduate School of Zhejiang University and a fellow of the China Computer Federation (CCF). He is a senior member of both IEEE and ACM. His research interests include intelligent media computing, data mining, and information accessibility, with over 150 publications in NeurIPS, KDD, ICCV, AAAI, etc. He also serves as the editorial board of Neurocomputing.