An Efficient and Scalable Graph Condensation with Structure-Preserving

Yulin Hu
Southwest University
Chongqing, China
hylinserein@email.swu.edu.cn

Fuyan Ou
Southwest University
Chongqing, China
omaksimovbe@gmail.com

Ye Yuan
Southwest University
Chongqing, China
yuanyekl@swu.edu.cn

***Abstract*—Graph condensation (GC) is pivotal for enabling Graph Neural Networks (GNNs) deployment in resource-constrained scenarios by compressing large-scale graphs into compact synthetic counterparts. Existing GC methods commonly suffer from computational inefficiency due to coupled optimization as well as encountering poor generalization across GNN architectures. To address these challenges, this study proposes an Efficient and Scalable Graph Condensation with Structure-Preserving (SP-ESGC), which possesses a decoupled design that separates node condensation from graph structure generation. Specifically, it first employs heat kernel feature propagation to generate node representation via spectral graph theory-inspired diffusion. Further, a novel hybrid clustering strategy is designed to extracts discriminative intra-class centroids from the node representation. Finally, a pre-trained edge predictor infers transferable structural patterns from the original graph, ensuring accurate synthetic graph generation. Extensive experiments on real-world graph datasets demonstrate that the proposed SP-ESGC implementes a precise GC with significantly high computational efficiency. Moreover, SP-ESGC also generalizes well across diverse GNN architectures.**

***Index Terms*—Graph Condensation, Graph Neural Networks, Graph Representation Learning, Efficiency**

I. INTRODUCTION

Graph-structured data is fundamental to many applications, including social networks [1], molecular modeling [2], and recommender systems [3]. Graph neural networks (GNNs) [4] can effectively learn from such data, but training GNNs on large graphs is increasingly expensive, limiting their use in resource-constrained scenarios, such as neural architecture search and continual learning. Graph condensation has emerged as a promising solution by condensing large graphs into small synthetic ones while preserving downstream performance. Inspired by dataset condensation in the image domain [5], early methods typically retain information through gradient matching, distribution matching, or trajectory matching.

Despite improved condensation quality, these methods still have several problems. A major issue lies in their optimization design. Most methods tightly couple the optimization of node features, graph structures, and GNN parameters. It often requires bi-level or even tri-level optimization, making the condensation process slow and difficult to scale, especially for large-scale graphs. Another issue is the dependence on specific GNN architectures. In many methods, the condensation objective is closely tied to the relay model. This tight coupling not only weakens generalization across different architectures but also leads to unstable performance when applied in practical settings. Finally, preserving graph structural information remains a challenge. Although some methods [6] attempt to introduce structural constraints, they often do so at the cost of increased computational overhead or reduced flexibility.

Recent work has explored more efficient condensation schemes through simplified optimization objectives. However, several challenges remain unresolved, including how to obtain stable node representations with lightweight optimization, how to preserve class-level discriminability under large node reduction, and how to efficiently generate the topology of the synthetic graph. To address these challenges, we propose SP-ESGC, an efficient graph condensation framework with a decoupled design. We separate node feature condensation from graph structure generation to improve efficiency and stability. Specifically, we employ heat kernel feature propagation to obtain node representations, followed by a clustering-based condensation strategy that integrates low-rank approximation and kernel feature expansion to extract representative class centroids. Finally, we generate the synthetic graph structure using an edge predictor pre-trained on the original graph.

Our main contributions can be summarized as follows:

- We propose a graph condensation framework that avoids costly bi-level optimization and repeated GNN training.
- We design a feature-based graph structure generation mechanism that enables efficient and scalable topology construction.
- Our method significantly reduces the computational cost without compromising the performance of downstream tasks, making it suitable for large-scale graph scenarios.

II. RELATED WORK

Gradient Matching. Gradient matching is an early and influential paradigm in graph condensation, formulating condensation as a bi-level optimization problem by aligning GNN parameter gradients between original and condensed graphs. GCond [7] enforces gradient consistency to approximate original training dynamics and achieves high condensation ratios. DosCond [8] simplifies this framework via one-step gradient matching, eliminating relay updates and hyperparameter tuning. SGDD [6] further incorporates structural information to mitigate spectral mismatches. Although these methods are

effective, gradient matching is still computationally expensive and sensitive to GNN architectures, which limits its scalability.

Trajectory Matching. Trajectory matching extends gradient matching by aligning full training dynamics rather than gradients. SFGC [10] argues that gradient matching cannot capture complex GNN learning behavior, it matches multi-step training trajectories to encode richer dynamics from the original graph. GEOM [11] further observes that limited supervision in trajectory matching leads to performance gaps. Despite improved condensation quality, trajectory matching incurs substantial computational and storage overhead due to teacher model pretraining and trajectory storage, limiting scalability to large graphs and resource-constrained settings.

Kernel Ridge Regression. Kernel-based methods reformulate graph condensation to avoid expensive GNN training. GC-SNTK [12] models GNN behavior using graph neural tangent kernels and casts condensation as a kernel ridge regression problem, eliminating bi-level optimization and repeated GNN training. KiDD [13] extends this formulation to graph classification by removing nonlinear activations in the kernel, simplifying matrix multiplications during condensation and achieving higher accuracy than gradient matching. While these methods improve efficiency and stability, their performance depends on kernel expressiveness and may be limited in capturing complex nonlinear patterns in real-world graphs.

Distribution Matching. Distribution matching aims to align statistical distributions between the original and condensed graphs. GCDM [9] models node receptive fields and represents the original graph as a collection of receptive field distributions. Consistency between the original and synthetic graphs is enforced using Maximum Mean Discrepancy (MMD), which reduces dependence on specific GNN architectures and improves generalization across models. SimGC [14] further incorporates pre-trained GNNs to align node representation variations and enhance performance. However, as these methods rely on static distribution alignment, they often struggle to preserve structural information and dynamic training behavior, particularly under large condensation ratios.

III. METHODOLOGY

The objective of graph condensation is to construct a synthetic graph from a large-scale original graph, while retaining the information that is essential for downstream tasks. By reducing the size of the training data, graph condensation aims to significantly lower computational and storage costs without sacrificing model performance. In this section, we will introduce the graph condensation framework SP-ESGC that we proposed, as illustrated in Fig.1. Consider that we have a graph set $\mathcal{T} = \{A, X, Y\}$, where $A \in \mathbb{R}^{N\times N}$ is the adjacency matrix, N is the number of nodes, $X \in \mathbb{R}^{N\times d}$ is the d-dimensional node feature matrix and $Y \in \{0, \ldots, C-1\}^N$ represents the node labels.Graph condensation will generate a small synthetic graph $\mathcal{S} = \{A', X', Y'\}$, where $A' \in \mathbb{R}^{n\times n}$, $X' \in \mathbb{R}^{n\times d}$, $Y' \in \{0, \ldots, C-1\}^n$, such that the GNN trained on $\mathcal{S}$ can achieve comparable performance to the GNN trained on $\mathcal{T}$.

A. Heat Kernel Feature Propagation

In graph-structured data, node features are often affected by noise and inconsistencies in graph structure or limited local neighborhoods can make it difficult for learning models to capture global information. To further enhance the stability of feature representations and improve structural coherence, we introduces heat kernel feature propagation, a mechanism inspired by spectral graph theory that provides a smooth, stable and controllable way to diffuse node features across the graph.

The theoretical foundation of heat kernel feature propagation originates from the classical heat equation defined on Euclidean spaces [16]. In particular, the heat equation describes how heat diffuses through a medium over time:

$$\frac{\partial u}{\partial t} = \Delta u, \tag{1}$$

where Δ is the continuous Laplace operator. However, graph is not continuous space, its nodes do not have continuous coordinates but only discrete edge relationships, so the continuous heat equation cannot be applied directly.

To adapt the heat diffusion model to graph, we convert the continuous heat equation into a discrete form on graphs [17]. We first construct the symmetric normalized adjacency matrix $S = D^{-\frac{1}{2}}AD^{-\frac{1}{2}}$, where A is the adjacency matrix and D is the degree matrix. Based on this, the symmetric normalized graph Laplacian is defined as $L = I - S$. By replacing the continuous Laplace operator Δ with the graph Laplacian L, the heat equation on graphs can be written as:

$$\frac{\partial X(t)}{\partial t} = -LX(t), \quad with\ initial \quad X(0) = X, \tag{2}$$

where $X(t)$ denotes the node features evolving over time t. This equation admits an analytical solution:

$$X(t) = e^{-tL}X, \tag{3}$$

where e^{-tL} is known as the heat kernel on graphs. Directly computing e^{-tL} on large-scale graphs is computationally expensive. To maintain scalability, we adopt a truncated Taylor series approximation:

$$e^{-tL} \approx \sum_{k=0}^{K} \frac{(-t)^k}{k!} L^k, \tag{4}$$

where K is the truncation coefficient. By retaining the first K terms of the infinite series, we obtain an approximation that balances accuracy and computational cost [18]. Therefore, we ultimately obtain the node representation matrix after heat kernel feature propagation:

$$H = \mathbf{X}(t) \approx \sum_{k=0}^{K} \frac{(-tL)^k}{k!} X. \tag{5}$$

H not only has a smoother representation, but also effectively integrates the local and global information of the graph, providing more expressive and stable input features for the subsequent graph condensation module.

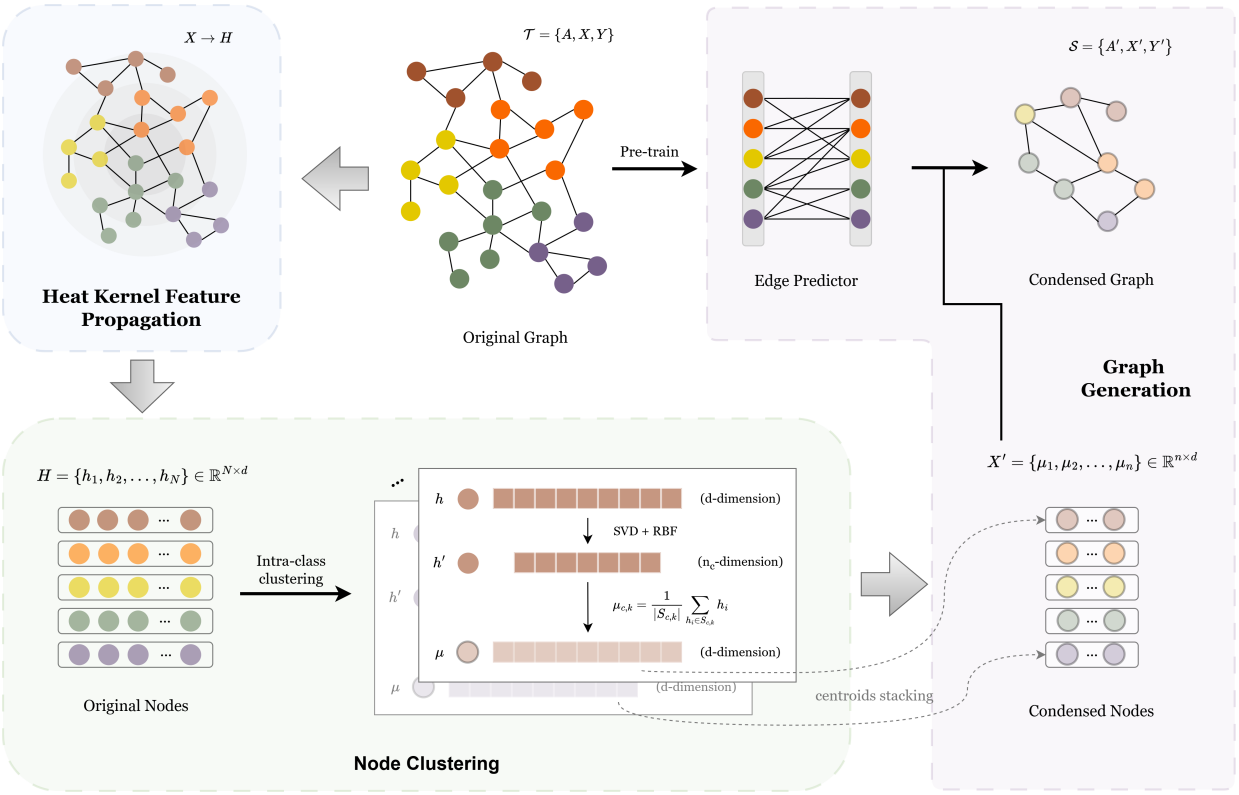

Fig. 1. A schematic diagram of the proposed SP-ESGC framework.

B. Node Clustering

In graph condensation tasks, we aim to condense the set of nodes belonging to each class into a small number of representative intra-class centroids while preserving discriminative information among different classes. These centroids serve as compact representations of intra-class information for subsequent tasks, and their quality directly affects the overall performance of downstream models trained on the condensed graph [15]. To achieve this, we propose a hybrid clustering method for condensing node representations, which integrates low-rank approximation, randomized kernel feature expansion and spectral space clustering to obtain a robust and discriminative set of intra-class centroids.

Given node representation matrix $H = \{h_1, h_2, \ldots, h_N\} \in \mathbb{R}^{N \times d}$ and the corresponding class labels $y \in \{0, \ldots, C-1\}^N$, the objective is to generate n_c representative node centroids within each class c from the set of node representations $H_c = \{h_i \mid y_i = c\} \in \mathbb{R}^{N_c \times d}$. These centroids form a class representation matrix

$$H_c' = \{\mu_1, \mu_2, \ldots, \mu_{n_c}\}, \tag{6}$$

where each centroid $\mu_j \in \mathbb{R}^d$ approximates the representation of samples within the same cluster and serves as a representative of the corresponding class.

For each class c, we extract the corresponding feature matrix H_c. To suppress mean shift, we first apply mean-centering to the features. We then perform truncated singular value decomposition (SVD) [19] to extract a low-rank information with rank is aligned with the target number of intra-class clusters. This step captures the directions of greatest variance within the class. By discarding components associated with smaller singular values, it provides a low-noise and low-rank basis space U_c for kernel expansion. Since node representations often lie on complex nonlinear structures, relying solely on linearly reduced features fails to achieve effective separation. To enhance the nonlinear discriminability of the features, we apply Random Fourier Features(RFF) to U_c, approximating the mapping of the Radial Basis Function(RBF) kernel:

$$Z_c = \phi_{\text{RBF}}(U_c) \in \mathbb{R}^{N_c \times m}, \tag{7}$$

where m is the random feature dimension used to approximate the RBF kernel, which represents the length of the feature vector generated for each sample through RFF [20]. According to Bochner's theorem, any continuous and positive-definite function can be expressed as the Fourier transform of some probability density function. Therefore:

$$\phi(x) = \sqrt{\frac{2}{m}} \cos(\omega^\top x + b). \tag{8}$$

This process embeds node representations into a more discriminative feature space without explicitly constructing the kernel matrix. The resulting Z_c is then further condensed in feature dimension to extract the most stable cluster information, ensuring that the retained features better capture the variation directions of local clusters. This yields the spectral

embedding space $Q_c \in \mathbb{R}^{N_c \times n_c}$. Clustering is then performed within each class in Q_c. The intra-class centroids are computed in the original feature space by using the cluster labels $\{S_{c,k}\}$:

$$\mu_{c,k} = \frac{1}{|S_{c,k}|} \sum_{h_i \in S_{c,k}} h_i. \tag{9}$$

Finally, by sequentially stacking the centroids of all classes, we obtain the node representations of the condensed graph:

$$X' = \bigcup_{c=0}^{C-1} \bigcup_{k=1}^{n_c} \mu_{c,k} \in \mathbb{R}^{n \times d}. \tag{10}$$

C. Graph Generation

We construct the condensed graph structure using a feature-based edge predictor learned from the original graph. This approach consists of two stages: (i) pretraining a parameterized edge predictor on the original graph, (ii) performing all-pairs inference with the pre-trained model on the synthetic node features to construct the final condensed graph structure.

Our goal is given a set of synthetic node features X' to generate a condensed graph $\mathcal{S} = \{A', X', Y'\}$ whose structure reflects the patterns of edge connections in the original graph, while preserving consistent behavior in downstream tasks. To achieve this, we first learn a mapping f_θ on original graph.The input to this mapping is the concatenated feature vector of a node pair $z_{uv} = [x_u; x_v] \in \mathbb{R}^{2d}$ and the output is the probability of the corresponding edge $p_{uv} = f_\theta(z_{uv})$. We then apply this mapping to the synthetic nodes to obtain the topology of the condensed graph [21]. By training on original graph, the edge predictor learns which combinations of node features tend to form edges, effectively capturing a transferable relationship between node and graph topology.

After obtaining the pre-trained edge predictor f_θ, we perform all-pairs inference on the set of synthetic node features X' to construct the condensed graph structure. For each node pair (i, j), we compute

$$A'_{ij} = f_\theta([x'_i; x'_j]) \tag{11}$$

to obtain a probabilistic adjacency matrix $A' \in \mathbb{R}^{n \times n}$, where each element represents the confidence of edge existing between the corresponding node pair. Since the all-pairs matrix produces a dense graph, we further sparsify A' by selecting a high-quantile threshold τ based on the distribution of all edge probabilities. Only connections with probabilities significantly above random levels are retained, while entries below the threshold are set to 0:

$$A'_{ij} = \begin{cases} A'_{ij}, & A'_{ij} \geq \tau, \\ 0, & \text{otherwise.} \end{cases} \tag{12}$$

Finally, we extract the nonzero entries of the sparsified matrix A' as the edge set and edge weights of the condensed graph, thereby obtaining the condensed graph structure. By leveraging the pre-trained edge predictor, we treat structure generation as a learnable functional mapping, eliminating the need for the original graph's adjacency structure during node synthesis. As long as the synthetic features preserve similarity to original graph features, the edge predictor can reconstruct a reasonable topology in the new feature space.

IV. EXPERIMENTS

In this section, we conduct a series of experiments to evaluate the effectiveness of the proposed framework SP-ESGC. We first examine the expressive capacity of the condensed graphs by comparing the performance of node classification models trained on them. We then analyze the efficiency of SP-ESGC against other condensation methods, together with its generalization ability across models and datasets. In addition, we perform ablation studies to better understand the contribution of each component in our framework. All experiments are conducted on an NVIDIA GeForce RTX 3050 GPU.

TABLE I
THE STATISTICS OF FIVE DATASETS.

Dataset	#Nodes	#Edges	#Classes	#Features
Cora	2,708	5,429	7	1,433
Citeseer	3,327	4,732	6	3,703
Ogbn-arxiv	169,343	1,166,243	40	128
Flickr	89,250	899,756	7	500
Reddit	232,965	57,307,946	210	602

A. Experimental Settings

We evaluate the condensation performance of SP-ESGC on five datasets, among which three are transductive datasets like Cora, Citeseer and Ogbn-arxiv, two are inductive datasets like Flickr and Reddit. For all datasets, we adopt the publicly available splits and experimental settings to ensure fair comparison. Detailed statistics of the datasets are reported in Table I. For each dataset, we consider three condensation ratios (r). Specifically, r is defined as the ratio between the number of condensed nodes n and the number of original nodes N. For transductive datasets, N denotes the total number of nodes in the entire graph, whereas for inductive datasets, N refers to the number of nodes in the training subgraph of the graph.We compared the proposed method with several baselines: (1) three coreset methods including Random, Herding and K-Center; (2) five graph condensation methods including GCOND [7], SGDD [6], SimGC [14], GC-SNTK [12] and SFGC [10].

B. Prediction Accuracy

We train node classification models on the condensed graphs and predict labels on the test sets. Classification accuracy is used as the evaluation metric to measure the representational capacity of the condensed graphs. We evaluate the representation quality of the condensed graphs under three different condensation ratios and report the test accuracy of all methods on different datasets in Table II. From the results, we observe that coreset methods perform worse than approaches specifically designed for graph condensation. This gap becomes more evident as the condensation ratio decreases. These observations further underline the importance of condensation

TABLE II

NODE CLASSIFICATION PERFORMANCE UNDER DIFFERENT GRAPH CONDENSATION RATIOS ON DIFFERENT GRAPH CONDENSATION METHODS.PERFORMANCE IS REPORTED AS TEST ACCURACY (%), WITH "±" THE STANDARD DEVIATION ACROSS THREE TRIALS.BEST RESULTS ARE IN BOLD AND THE SECOND-BESTS ARE UNDERLINED.OOM MEANS OUT-OF-MEMORY.

Dataset	Ratio (r)	Random	Herding	K-Center	Gcond	SGDD	SimGC	GC-SNTK	SFGC	SP-ESGC	Whole Dataset
Cora	1.30%	63.6±3.7	67.0±1.3	64.0±2.3	79.8±1.3	80.1±0.7	80.8±2.3	81.3±0.2	80.1±0.4	**82.6±0.6**	81.2±0.2
	2.60%	72.8±1.1	73.4±1.0	73.2±1.2	80.1±0.6	80.6±0.8	80.9±2.6	81.5±0.7	81.7±0.5	**82.7±0.1**	
	5.20%	76.8±0.1	76.8±0.1	76.7±0.1	79.3±0.3	80.4±1.6	82.1±1.3	81.3±0.2	81.6±0.8	**82.5±0.4**	
Citeseer	0.90%	54.4±4.4	57.1±1.5	52.4±2.8	70.5±1.2	69.5±0.4	**73.8±2.5**	66.4±1.0	71.4±0.5	73.5±0.1	71.7±0.1
	1.80%	64.2±1.7	66.7±1.0	64.3±1.0	70.6±0.9	70.2±0.8	72.2±0.5	68.4±1.1	72.4±0.4	**72.8±0.2**	
	3.60%	69.1±0.1	69.0±0.1	69.1±0.1	69.8±1.4	70.3±1.7	71.1±2.8	69.8±0.8	70.6±0.7	**72.0±0.4**	
Ogbn-arxiv	0.05%	47.1±3.9	52.4±1.8	47.2±3.0	59.2±1.1	60.8±1.3	63.6±0.8	64.4±0.2	**65.5±0.7**	64.3±0.4	71.4±0.1
	0.25%	57.3±1.1	58.6±1.2	56.8±0.8	63.2±0.3	65.8±1.2	66.4±0.3	65.1±0.8	66.1±0.4	**66.7±0.1**	
	0.50%	60.0±0.9	60.4±0.8	60.3±0.4	64.0±0.4	66.3±0.7	**66.8±0.4**	65.4±0.5	**66.8±0.4**	**66.8±0.1**	
Flickr	0.10%	41.8±2.0	42.5±1.8	42.0±0.7	46.5±0.4	46.5±0.1	45.3±0.7	46.7±0.1	46.6±0.2	**47.2±0.2**	47.2±0.1
	0.50%	44.0±0.4	43.9±0.9	43.2±0.1	45.2±0.3	46.4±0.2	45.6±0.4	46.8±0.1	47.0±0.1	**47.2±0.3**	
	1.00%	44.6±0.2	44.4±0.6	44.1±0.4	**47.1±0.1**	46.3±0.1	43.8±1.5	46.5±0.2	**47.1±0.1**	**47.1±0.3**	
Reddit	0.05%	46.1±4.4	53.1±2.5	46.6±2.3	88.0±1.8	90.3±0.1	89.6±0.6	OOM	89.7±0.2	**90.7±0.0**	93.9±0.0
	0.10%	58.0±2.2	62.7±1.0	53.0±3.3	89.6±0.7	90.7±0.1	90.6±0.3	OOM	90.0±0.3	**91.6±0.0**	
	0.20%	66.3±1.9	71.0±1.6	58.5±2.1	90.1±0.5	91.3±0.2	91.4±0.2	OOM	89.9±0.4	**92.3±0.0**	

methods to graph data. SP-ESGC achieves the best or second-best performance in the majority of cases.Specifically, under low-ratio settings such as Cora (r=1.30%), Flickr (r=0.10%), and Reddit (r=0.05%), SP-ESGC consistently achieves the best results, indicating that it can effectively preserve both structural and semantic information even at low condensation ratio. In contrast, GC-SNTK encounters out-of-memory (OOM) issues on Reddit, revealing scalability limitations on large-scale graph datasets, whereas SP-ESGC maintains high efficiency.

C. Condensation Time

In this section, we compare the condensation time of SP-ESGC with that of five graph condensation baselines. To ensure a fair evaluation, we separately record the time spent on graph condensation, edge predictor model pretraining and condensed graph performance evaluation for SP-ESGC. The total runtime is added these time together and then compared against other methods. As shown in Table III, SP-ESGC is shown to be significantly faster than existing graph condensation methods. Notably, on the Reddit dataset, the condensation time of SP-ESGC is reduced to approximately one-sixteenth of that required by the second fastest baseline. This result highlights the efficiency advantage of SP-ESGC, especially in large-scale graph.

TABLE III

THE CONDENSATION TIME (SECONDS) OF OUR PROPOSED SP-ESGC AND BASELINES ON FIVE DATASETS. THE R OF THE FIVE DATASETS ARE RESPECTIVELY SET TO 2.60%, 1.80%, 0.25%, 0.50% AND 0.10%.

	Cora	Citeseer	Ogbn-arxiv	Flickr	Reddit
GCond	653.9	940.3	13521.7	1455.6	20528.8
SGDD	4848.6	3091.2	35179.7	26767.4	378220.9
SimGC	289.3	495.7	476.6	744.5	2655.9
GC-SNTK	92.1	69.7	28897.8	889.8	OOM
SFGC	5895.8	3807.2	156975.2	46706.7	370089.0
SP-ESGC	**12.4**	**26.4**	**143.4**	**22.6**	**162.5**

D. Generalization Ability

In this section, we evaluate the generalization ability of different graph condensation methods by training various GNN architectures on the condensed graphs. Specifically, we consider GCN, SGC, GAT, GraphSAGE, APPNP, and compare SP-ESGC with two representative graph condensation methods, GCond and SFGC. As shown in Fig.2, SP-ESGC demonstrates consistently strong performance across diverse GNN architectures and datasets. The performance of GCond exhibits noticeable degradation on certain architectures (e.g. GAT), while SP-ESGC maintains relatively stable accuracy with smaller fluctuations across GNNs. The results suggest that SP-ESGC provides better generalization and stability.

E. Ablation Study

To analyze the contribution of each component in SP-ESGC, we conduct ablation studies on multiple datasets with three variants. In "w/o HKP", we remove the heat kernel feature propagation module and directly use the original node features for node condensation. In "w/o EP", the pre-trained edge predictor is removed, edges between condensed nodes are constructed based on cosine similarity of node features. In "w K-means", we replace the proposed class condensation strategy with K-means clustering. Results are reported in Table IV. We observe that the full model consistently achieves the best performance across all datasets, indicating that each component contributes to the effectiveness of SP-ESGC. The ablation results clearly show that: (1) Heat kernel feature propagation introduces global information prior to condensation, providing a more stable node representation for subsequent clustering and structure generation. (2) The method based on feature similarity are insufficient for generating meaningful graph structures. The edge predictor is able to capture connection rules for edges from the original graph and

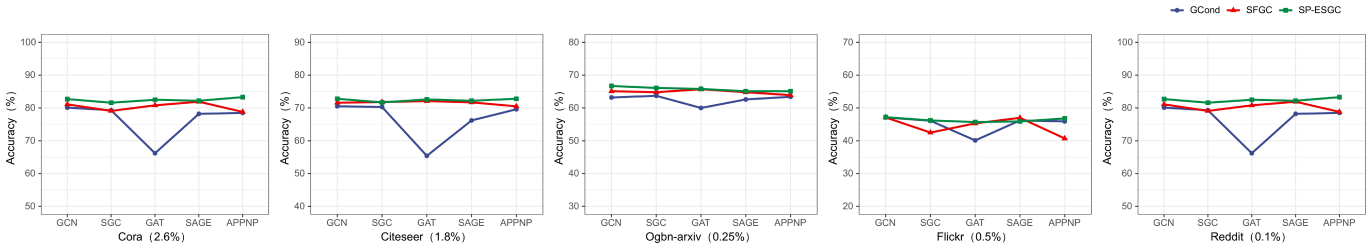

Fig. 2. The generalization capabilities of SP-ESGC on five datasets.

transfer them to the condensed nodes. (3) Simple Euclidean-space clustering fails to capture the intrinsic structure of node representations. In contrast, intra-class spectral space clustering more accurately models the nonlinear distributions, producing more representative node centroids.

TABLE IV
THE ABLATION STUDY OF SP-ESGC.

Method	Cora	Citeseer	Ogbn-arxiv	Flickr	Reddit
SP-ESGC w/o HKP	$78.6_{\pm 0.3}$	$70.4_{\pm 0.2}$	$65.4_{\pm 0.3}$	$46.7_{\pm 0.4}$	$90.5_{\pm 0.1}$
SP-ESGC w/o EP	$81.9_{\pm 0.2}$	$72.5_{\pm 0.2}$	$66.0_{\pm 0.2}$	$45.8_{\pm 0.1}$	$91.1_{\pm 0.1}$
SP-ESGC w K-means	$81.4_{\pm 0.2}$	$72.0_{\pm 0.2}$	$66.6_{\pm 0.0}$	$46.7_{\pm 0.2}$	$91.4_{\pm 0.0}$
SP-ESGC	$\mathbf{82.7}_{\pm \mathbf{0.1}}$	$\mathbf{72.8}_{\pm \mathbf{0.2}}$	$\mathbf{66.7}_{\pm \mathbf{0.1}}$	$\mathbf{47.2}_{\pm \mathbf{0.3}}$	$\mathbf{91.6}_{\pm \mathbf{0.0}}$

V. CONCLUSION

In this work, we study graph condensation from an efficiency and structure-aware perspective and propose a simple and effective framework, SP-ESGC. By decoupling node condensation from graph structure generation, SP-ESGC avoids complex bi-level optimization and reduces computational overhead. The proposed heat kernel feature propagation introduces global information in a stable and controllable manner, while the clustering strategy preserves intra-class discriminability under condensation. Moreover, the edge predictor enables to generate flexible and scalable topology construction. Experiments on both transductive and inductive datasets show that SP-ESGC achieves strong performance across various condensation ratios and GNN architectures. Overall, SP-ESGC offers an efficient solution for graph condensation, making it well suited for large-scale and resource-constrained scenarios.

REFERENCES

[1] S. Huang, W. Lin, Z. Bao, and J. Sun, "Influence maximization in real-world closed social networks," *arXiv preprint arXiv:2209.10286*, 2022.

[2] C. Ying, T. Cai, S. Luo, S. Zheng, G. Ke, D. He, Y. Shen, and T.-Y. Liu, "Do transformers really perform badly for graph representation?" *Advances in neural information processing systems*, vol. 34, pp. 28 877–28 888, 2021.

[3] W. Fan, Y. Ma, Q. Li, Y. He, E. Zhao, J. Tang, and D. Yin, "Graph neural networks for social recommendation," in *The world wide web conference*, 2019, pp. 417–426.

[4] L. Waikhom and R. Patgiri, "Graph neural networks: Methods, applications, and opportunities," *arXiv preprint arXiv:2108.10733*, 2021.

[5] T. Wang, J.-Y. Zhu, A. Torralba, and A. A. Efros, "Dataset distillation," *arXiv preprint arXiv:1811.10959*, 2018.

[6] 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?" *Advances in Neural Information Processing Systems*, vol. 36, pp. 53 201–53 226, 2023.

[7] W. Jin, L. Zhao, S. Zhang, Y. Liu, J. Tang, and N. Shah, "Graph condensation for graph neural networks," *arXiv preprint arXiv:2110.07580*, 2021.

[8] 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.

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

[10] 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, pp. 6026–6047, 2023.

[11] 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," *arXiv preprint arXiv:2402.05011*, 2024.

[12] 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 Web Conference 2024*, 2024, pp. 4439–4448.

[13] 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.

[14] 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.

[15] X. Gao, G. Ye, T. Chen, W. Zhang, J. Yu, and H. Yin, "Rethinking and accelerating graph condensation: A training-free approach with class partition," in *Proceedings of the ACM on Web Conference 2025*, 2025, pp. 4359–4373.

[16] F. Zhang and E. R. Hancock, "Graph spectral image smoothing using the heat kernel," *Pattern recognition*, vol. 41, no. 11, pp. 3328–3342, 2008.

[17] J. Zhao, Y. Dong, M. Ding, E. Kharlamov, and J. Tang, "Adaptive diffusion in graph neural networks," *Advances in neural information processing systems*, vol. 34, pp. 23 321–23 333, 2021.

[18] C.-C. Tseng and S.-L. Lee, "Distributed implementation of heat kernel smoothing for graph signal denoising," in *2022 IEEE 65th International Midwest Symposium on Circuits and Systems (MWSCAS)*. IEEE, 2022, pp. 1–4.

[19] S. L. Shishkin, A. Shalaginov, and S. D. Bopardikar, "Fast approximate truncated svd," *Numerical Linear Algebra with Applications*, vol. 26, no. 4, p. e2246, 2019.

[20] A. Rahimi and B. Recht, "Random features for large-scale kernel machines," *Advances in neural information processing systems*, vol. 20, 2007.

[21] Z. Xiao, Y. Wang, S. Liu, B. Hu, H. Wang, M. Song, and T. Zheng, "Disentangled condensation for large-scale graphs," in *Proceedings of the ACM on Web Conference 2025*, 2025, pp. 4494–4506.