Appendix

This is the appendix of our work: **'Structure-free Graph Condensation: From Large-scale Graphs to Condensed Graph-free Data'**. In this appendix, we provide more details of the proposed SFGC in terms of related works, potential application scenarios, dataset statistics, method analysis, and experimental settings with some additional results.

A Related Works

Dataset Distillation (Condensation) aims to synthesize a small typical dataset that distills the most important knowledge from a given large target dataset, such that the synthesized small dataset could serve as an effective substitution of the large target dataset for various scenarios [30, 49], *e.g.*, model training and inference, architecture search, and continue learning. Typically, DD [59] and DC-KRR [39] adopted the meta-learning framework to solve bi-level distillation objectives through calculating meta-gradients. In contrast, DC [77], DM [76], and MTT [4] designed surrogate functions to avoid unrolled optimization through the gradient matching, feature distribution matching, and training trajectory matching, respectively, where the core idea is to effectively mimic the large target dataset in the synthesized small dataset. Except for the image data condensed by the above-mentioned works, GCOND [27] first extended the online gradient matching scheme in DC [77] to structural graph data, along with parameterized graph structure learning module to synthesize condensed edge connections. Furthermore, DosCond [26] proposed single-step gradient matching to synthesize graph nodes, with a probabilistic graph model to condense structures on the graph classification task. In this work, we eliminate the process of synthesizing graph structures and propose a novel structure-free graph condensation paradigm, to distill the large-scale graph to the small-scale graph-free data, leading to the easier optimization process of condensation. Meanwhile, the structure-free characteristic allows condensed data better generalization ability to different GNN architectures.

Graph Size Reduction aims to reduce the graph size to fewer nodes and edges for effective and efficient GNN training, including graph sampling [66, 6], graph coreset [47, 60], graph sparsification [1, 5], graph coarsening [3, 28], and recently rising graph condensation [27, 9, 26]. Concretely, graph sampling methods [66, 6] and graph coreset methods [47, 60] sample or select the subset of nodes and edges from the whole graph, such that the information of the derived sub-graph is constrained by the whole large-scale graph, which considerably limits the expressiveness of the size-reduced graph. Moreover, graph sparsification methods [1, 5] and graph coarsening methods [3, 28] reduce the number of edges and nodes by simplifying the edge connections and grouping node representations of the large-scale graph, respectively. The core idea of both sparsification and coarsening is to preserve specific large-scale graph properties (*e.g.*, spectrum and principle eigenvalues) in the sparse and coarsen small graph. The preserved graph properties in the small-scale graph, however, might not be suitable for downstream GNN tasks. In contrast, our work focuses on graph condensation to directly optimize and synthesize the small-scale condensed data, which breaks information constraints of the large-scale graph and encourages consistent GNN test performance.

B Potential Application Scenarios

We would like to highlight the significance of graph condensation task to various application scenarios within the research field of dataset distillation/condensation, while comprehensive overviews can be found in survey works [30, 59]. Specifically, we present several potential scenarios where our proposed structure-free graph condensation method could bring benefits:

Graph Neural Architecture Search. Graph neural architecture search (GraphNAS) aims to develop potential and expressive GNN architectures beyond existing human-designed GNNs. By automatically searching in a space containing various candidate GNN architecture components, GraphNAS could derive powerful and creative GNNs with superior performance on specific graph datasets for specific tasks [80, 81, 46, 13, 18]. Hence, GraphNAS needs to repeatedly train different potential GNN architectures on the specific graph dataset, and ultimately selects the optimal one. When in the large-scale graph, this would incur severe computation and memory costs. In this case, searching on our developed small-scale condensed graph-free data, a representative substitution of the large-scale graph, could significantly benefit for saving many computation costs and accelerating new GNN architecture development in GraphNAS research field.

Privacy Protection. Considering the outsourcing scenario of graph learning tasks, the original large-scale graph data is not allowed to release due to privacy, for example, patients expect to use GNNs for medical diagnosis without their personal medical profiles being leaked [52, 11]. In this case, as a compact and representative substitution, the synthesized small-scale condensed graph could be used to train GNN models, so that the private information of the original graph data can be protected. Besides, considering the scenario that over-parameterized GNNs might easily memorize training data, inferring the well-trained models could cause potential privacy leakage issue. In this case, we could release a GNN model trained by the synthesized small-scale condensed graph, so that the model avoids explicitly training on the original large-scale graph and consequently helps protect its data privacy.

Adversarial Robustness. In practical applications, GNNs might be attacked with disrupted performance, when attackers impose adversarial perturbations to the original graph data [68], for instance, poisoning attacks on graph data [53, 15, 84], where attackers attempt to alter the edges and nodes of training graphs of a target GNN. Training on poisoned graph data could significantly damage GNNs' performance. In this case, given a poisoned original training graph, graph condensation could synthesize a new condensed graph from it, which we use to train the target GNN would achieve comparable test performance with that trained by the original training graph before being poisoned. Hence, the new condensed graph could eliminate adversarial samples in the original poisoned graph data with great adversarial robustness, so that using it to train a GNN would not damage its performance for inferring test graphs.

Continual learning. Continual learning (CL) aims to progressively accumulates knowledge over a continuous data stream to support future learning while maintaining previously learned information [45, 12, 65]. One of key challenges of CL is catastrophic forgetting [31, 83], where knowledge extracted and learned from old data/tasks are easily forgotten when new information from new data/tasks are learned. Some works have studied that data distillation/condensation is an effective solution to alleviate catastrophic forgetting [8, 48, 50, 62], where the distilled and condensed data is taken as representative summary stored in a replay buffer that is continually updated to instruct the training of subsequent data/tasks.

To summarize, graph condensation task holds great promise and is expected to bring significant benefits to various graph learning tasks and applications. By producing compact, high-quality, small-scale condensed graph data, graph condensation has the potential to enhance the efficiency and effectiveness of future graph machine learning works.

C Dataset Details

We provide the details of the original dataset statistics in Table A1. Moreover, we also compare the statistics of our condensed graph-free data with GCOND [27] condensed graphs in Table A2. It can be observed that both GCOND [27] and our proposed `SFGC` significantly reduce the numbers of nodes and edges from large-scale graphs, as well as the data storage. Importantly, our proposed `SFGC` directly reduces the number of edges to 0 by eliminating graphs structures in the condensation process, but with superior node attribute contexts integrating topology structure information.

Table A1: Details of dataset statistics.

Datasets	#Nodes	#Edges	#Classes	#Features	Train/Val/Test
Cora	2,708	5,429	7	1,433	140/500/1000
Citeseer	3,327	4,732	6	3,703	120/500/1000
Ogbn-arxiv	169,343	1,166,243	40	128	90,941/29,799/48,603
Flickr	89,250	899,756	7	500	44,625/22312/22313
Reddit	232,965	57,307,946	41	602	15,3932/23,699/55,334

D More Analysis of Structure-free Paradigm

In this section, we theoretically analyze the rationality of the proposed structure-free paradigm from the views of statistical learning and information flow, respectively.

Table A2: The statistic comparison of condensed graphs by GCOND [27] and condensed graph-free data by our SFGC.

Dataset	Citeseer ($r = 1.8\%$)			Cora ($r = 2.6\%$)			Ogbn-arxiv ($r = 0.25\%$)			Flickr ($r = 0.5\%$)			Reddit ($r = 0.1\%$)		
Methods	Whole	GCOND [27]	SFGC (ours)	Whole	GCOND [27]	SFGC (ours)	Whole	GCOND [27]	SFGC (ours)	Whole	GCOND [27]	SFGC (ours)	Whole	GCOND [27]	SFGC (ours)
Accuracy	70.7	70.5	72.4	81.5	79.8	81.7	71.4	63.2	66.1	47.1	47.1	47.0	94.1	89.4	90.0
#Nodes	3,327	60	60	2,708	70	70	169,343	454	454	44,625	223	223	153,932	153	153
#Edges	4,732	1,454	0	5,429	2,128	0	1,166,243	3,354	0	218,140	3,788	0	10,753,238	301	0
Sparsity	0.09%	80.78%	-	0.15%	86.86%	-	0.01%	3.25%	-	0.02%	15.23%	-	0.09%	2.57%	-
Storage	47.1MB	0.9MB	0.9MB	14.9MB	0.4MB	0.4MB	100.4MB	0.3MB	0.2MB	86.8MB	0.5MB	0.4MB	435.5MB	0.4MB	0.4MB

The View of Statistical Learning. We start from the graph condensation optimization objective of synthesizing graphs structures in Eq. (1) of the main submission. Considering its inner loops $\boldsymbol{\theta}_{\mathcal{T}'} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}\left[\mathrm{GNN}_{\boldsymbol{\theta}}\left(\mathbf{X}', \mathbf{A}'\right), \mathbf{Y}'\right]$ with $\mathbf{A}' = \mathrm{GSL}_{\boldsymbol{\psi}}\left(\mathbf{X}'\right)$, it equals to learn the conditional probability $Q(\mathbf{Y}' \mid \mathcal{T}')$ given the condensed graph $\mathcal{T}' = (\mathbf{X}', \mathbf{A}', \mathbf{Y}')$ as

$$\begin{aligned} Q(\mathbf{Y}' \mid \mathcal{T}') &\approx \sum_{\mathbf{A}' \in \boldsymbol{\psi}(\mathbf{X}')} Q(\mathbf{Y}' \mid \mathbf{X}', \mathbf{A}') Q(\mathbf{A}' \mid \mathbf{X}') \\ &= \sum_{\mathbf{A}' \in \boldsymbol{\psi}(\mathbf{X}')} Q(\mathbf{X}', \mathbf{A}', \mathbf{Y}') / Q(\mathbf{X}', \mathbf{A}') \cdot Q(\mathbf{X}', \mathbf{A}') / Q(\mathbf{X}') \\ &= \sum_{\mathbf{A}' \in \boldsymbol{\psi}(\mathbf{X}')} Q(\mathbf{X}', \mathbf{A}', \mathbf{Y}') / Q(\mathbf{X}') \\ &= Q(\mathbf{X}', \mathbf{Y}') / Q(\mathbf{X}') = Q(\mathbf{Y}' \mid \mathbf{X}'), \end{aligned} \tag{8}$$

where we simplify the notation of graph structure learning module $\mathrm{GSL}_{\boldsymbol{\psi}}$ as parameterized $\boldsymbol{\psi}\left(\mathbf{X}'\right)$. As can be observed, when the condensed graph structures are learned from the condensed nodes as $\mathbf{A}' \in \boldsymbol{\psi}\left(\mathbf{X}'\right)$, the optimization objective of the conditional probability is not changed, while its goal is still to solve the posterior probability $Q(\mathbf{Y}' \mid \mathbf{X}')$. In this way, eliminating graph structures to conduct structure-free condensation is rational from the view of statistical learning. By directly synthesizing the graph-free data, the proposed SFGC could ease the optimization process and directly transfer all the informative knowledge of the large-scale graph to the condensed graph node set without structures. Hence, the proposed SFGC conducts more compact condensation to derive the small-scale graph-free data via Eq. (6) of the main manuscript, whose node attributes already integrate implicit topology structure information.

The View of Information Flow. For training on large-scale graphs to obtain offline parameter trajectories, we solve the node classification task on $\mathcal{T} = (\mathbf{X}, \mathbf{A}, \mathbf{Y})$ with a certain GNN model as

$$\boldsymbol{\theta}_{\mathcal{T}}^{*} = \arg\min_{\boldsymbol{\theta}} \mathcal{L}_{\text{cls}}\left[\mathrm{GNN}_{\boldsymbol{\theta}}(\mathbf{X}, \mathbf{A}), \mathbf{Y}\right], \tag{9}$$

where $*$ denotes the optimal training parameters that build the training trajectory distribution $P_{\boldsymbol{\Theta}_{\mathcal{T}}}$. The whole graph information, *i.e.*, node attributes $\mathbf{X}$ and topology structures $\mathbf{A}$ are both embedded in the latent space of GNN network parameters. Hence, the large-scale graph information flows to GNN parameters as $(\mathbf{X}, \mathbf{A}) \Rightarrow P_{\boldsymbol{\Theta}_{\mathcal{T}}}$. In this way, by meta-sampling in the trajectory distribution, Eq. (4) and Eq. (5) in the main manuscript explicitly transfer learning behaviors of the large-scale graph to the parameter space $\widetilde{\boldsymbol{\theta}}_{\mathcal{S}}$ of $\mathrm{GNN}_{\mathcal{S}}$ as $P_{\boldsymbol{\Theta}_{\mathcal{T}}} \Rightarrow \widetilde{\boldsymbol{\theta}}_{\mathcal{S}}$. As a result, the informative knowledge of the large-scale graphs, *i.e.*, node attributes and topology structure information $(\mathbf{X}, \mathbf{A})$, would be comprehensively transferred as $(\mathbf{X}, \mathbf{A}) \Rightarrow P_{\boldsymbol{\Theta}_{\mathcal{T}}} \Rightarrow \widetilde{\boldsymbol{\theta}}_{\mathcal{S}}$. In this way, we could identify the critical goal of graph condensation is to further transfer the knowledge in $\widetilde{\boldsymbol{\theta}}_{\mathcal{S}}$ to the output condensed graph data as:

$$\begin{aligned} &(\mathbf{X}, \mathbf{A}) \Rightarrow P_{\boldsymbol{\Theta}_{\mathcal{T}}} \Rightarrow \boldsymbol{\Theta}_{\mathcal{S}} \Rightarrow \mathcal{T}' = (\mathbf{X}', \mathbf{A}'), \quad \text{GC}. \\ &(\mathbf{X}, \mathbf{A}) \Rightarrow P_{\boldsymbol{\Theta}_{\mathcal{T}}} \Rightarrow \boldsymbol{\Theta}_{\mathcal{S}} \Rightarrow \mathcal{S} = (\widetilde{\mathbf{X}}), \quad \texttt{SFGC}. \end{aligned} \tag{10}$$

where GC and SFGC are corresponding to the existing graph condensation and the proposed structure-free graph condensation, respectively.

Hence, from the view of information flow, we could observe that condensing structures would not inherit more information from the large-scale graph. Compared with GC which formulates the condensed graph into nodes and structures, the proposed SFGC directly distills all the large-scale graph knowledge into the small-scale graph node set without structures. Consequently, the proposed SFGC conducts more compact condensation to derive the small-scale graph-free data, which implicitly encodes the topology structure information into the discriminative node attributes.

Table A3: Running time comparison (seconds) of the proposed SFGC and GCOND [27] for 50 epochs with a single GeForce RTX 3080 GPU.

Ogbn-arxiv	r=0.05%	r=0.25%	r=0.5%
GCOND[27]	296.34	442.58	885.58
SFGC **(ours)**	101.07	183.54	150.35

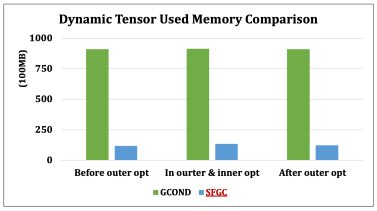

Figure A1: Comparison of the dynamic tensor used memory cost between online short-range gradient matching method GCOND [27] and our proposed SFGC.

E More Experimental Settings and Results

E.1 Time Complexity Analysis & Dynamic Memory Cost Comparison

We first analyze the time complexity of the proposed method and compare the running time between our proposed SFGC and GCOND [27].

Let the number of GCN layers be L, the large-scale graph node number be N, the small-scale condensed graph node number be N', the the feature dimension be d, the time complexity of calculating training trajectory meta-matching objective function is about $TK\mathcal{O}(LN'd^2 + LN'd)$, for each process of the forward, backward, and training trajectory meta-matching loss calculation, where T denotes the number of iterations and K denotes the meta-matching steps. Note that the offline expert training stage costs an extra $TK\mathcal{O}(LEd + LNd^2)$ on the large-scale graph, where E is the number of edges.

In contrast, for GCOND, it has at least $TK\mathcal{O}(LN'^2d + LN'd) + TK\mathcal{O}(N'^2d^2)$, and also additional $TK\mathcal{O}(LEd+LNd^2)$ on the large-scale graph, where K denotes the number of different initialization here. It can be observed that our proposed SFGC has a smaller time complexity compared to GCOND, which can be mainly attributed to our structure-free paradigm when the adjacency matrix related calculation in $\mathcal{O}(LN'^2d)$ can be avoided. The corresponding comparison of running time in the graph condensation process can be found in Table A3. As can be observed, both results on time complexity and running time could verify the superiority of the proposed SFGC.

Moreover, we present the comparison result of the dynamic tensor used memory cost between the online short-range gradient matching method GCOND [27] and our offline long-range meta-matching SFGC. As shown in Fig. A1, we consider three stages of optimizing the objective function, *i.e.*, before outer optimization, in the outer and inner optimization, and after outer optimization. It can be observed that the proposed SFGC could significantly alleviate heavy online memory and computation costs. This can be attributed to its offline parameter matching schema.

E.2 Effectiveness of Graph Neural Feature Score in SFGC

To verify the effectiveness of graph neural feature score γ_{gnf} in the proposed SFGC, we consider the following two aspects in dynamic evaluation: (1) node classification performance at different meta-matching steps in Table A4; (2) learning time comparison between iterative GNN training and our closed-form γ_{gnf} in Fig. A2.

As shown in Table A4, we select certain meta-matching step intervals, *i.e.*, 1000, 2000, and 3000, for testing their condensed data's performance, which is a commonly-used evaluation strategy for existing methods. Here, we set long-enough meta-matching steps empirically to ensure sufficient learning to expert training trajectory-built parameter distribution. And we compare these interval-step results with the performance of our condensed graph-free data, which is selected at certain steps of

Table A4: Performance of the condensed graph-free data between different meta-matching steps and γ_{gnf} dynamic evaluation selected steps in the proposed SFGC.

Datasets (Ratio)	Meta-matching Steps			γ_{gnf}	
	1000	2000	3000	ACC	Selected Steps
Citeseer ($r = 1.8\%$)	61.8±3.1	64.2±5.2	-	72.4±0.4	46
Cora ($r = 2.6\%$)	81.2±0.5	81.8±0.7	-	81.7±0.5	929
Ogbn-arxiv ($r = 0.25\%$)	64.5±0.8	65.8±0.3	-	66.1±0.4	90
Flickr ($r = 0.5\%$)	46.3±0.2	44.7±0.3	-	47.0±0.1	200
Reddit ($r = 0.1\%$)	86.9±0.5	89.8±0.3	89.9±0.5	90.0±0.3	2299

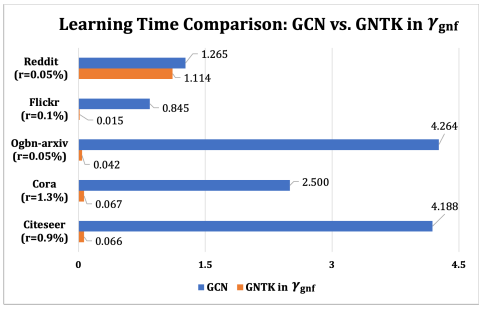

Figure A2: Learning time comparison (seconds) in dynamic evaluation between GNN iterative training and closed-form GNTK in γ_{gnf} of the proposed SFGC.

the meta-matching process according to the metric γ_{gnf}. Overall, γ_{gnf} could select optimal condensed graph-free data with superior effectiveness at best meta-matching steps.

For the learning time comparison between GNN iterative training *vs*. GNTK-based closed-form solutions of γ_{gnf} in Fig. A2, we consider the time of GNN iterative training that covers all training epochs under the best test performance for fair comparisons. This is due to the fact that the iterative training evaluation strategy mandates the complete training of a GNN model from scratch at each meta-matching step. For instance, in Flickr dataset ($r = 0.1\%$), we calculate 200 epochs running time, *i.e.*, 0.845s, which is the optimal parameter setting for training GNN under 0.1% condensation ratio. As can be generally observed, for all datasets, the proposed GNTK-based closed-form solutions of γ_{gnf} significantly save the learning time for evaluating the condensed graph-free data in meta-matching, illustrating our SFGC's high dynamic evaluation efficiency.

E.3 Analysis of Different Meta-matching Ranges

To explore the effects of different ranges of long-term meta-matching, we present the different step combinations of q steps (student) in $\text{GNN}_{\mathcal{S}}$ and p steps (expert) of $\text{GNN}_{\mathcal{T}}$ in Eq. (5) of the main manuscript on Ogbn-arxiv dataset under $r = 0.05\%$. The results are shown in Fig. A3. As can be observed, there exists the optimal step combination of q student steps (600) and expert p steps (1800). Under such a setting, the condensed small-scale graph-free data has the best node classification performance. Moreover, the quality and expressiveness of the condensed graph-free data moderately vary with different step combinations, but the variance is not too drastic.

More detailed settings of hyper-parameters of q steps (student) in $\text{GNN}_{\mathcal{S}}$ and p steps (expert) of $\text{GNN}_{\mathcal{T}}$ in the long-term meta-matching, as well as the meta-matching learning rate (LR) in the outer-level optimization and $\text{GNN}_{\mathcal{S}}$ learning rate (step size) ζ (Algorithm 1 of the main manuscript) in the inner-level optimization, are listed in Table A5.

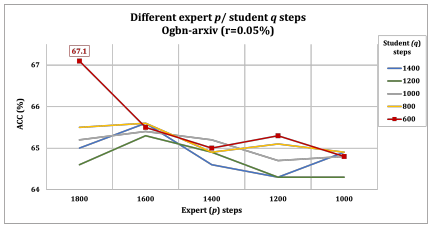

Figure A3: Performance with different step combinations of q student steps and expert p steps on Ogbn-arxiv ($r = 0.05\%$).

Table A5: Hyper-parameters of p expert steps and q student steps with meta-matching learning rate (LR) in the outer-level optimization and $\mathrm{GNN}_{\mathcal{S}}$ learning rate (step size) ζ in the inner-level optimization.

Datasets	Ratios (r)	p steps (expert)	q steps (student)	Meta-matching LR	ζ for $\mathrm{GNN}_{\mathcal{S}}$
Citeseer	0.9%	500	200	0.0005	1.0
	1.8%	500	200	0.001	1.0
	3.6%	400	300	0.001	1.0
Cora	1.3%	1500	400	0.0001	0.5
	2.6%	1200	500	0.0001	0.5
	5.2%	2000	500	0.0001	0.5
Ogbn-arxiv	0.05%	1800	600	0.2	0.2
	0.25%	1900	1200	0.1	0.1
	0.5%	1900	1000	0.1	0.1
Flickr	0.1%	700	600	0.1	0.3
	0.5%	900	600	0.01	0.2
	1%	900	900	0.02	0.2
Reddit	0.05%	900	900	0.02	0.5
	0.1%	900	900	0.05	0.5
	0.2%	900	900	0.2	0.2

E.4 Performance on Graph Node Clustering Task

Taking graph node clustering as the downstream task, we verified that, our condensed graph-free data, synthesized based on the node classification task, can be effectively utilized for other graph machine learning tasks, demonstrating the applicability of our condensed data. The experimental results are shown in Table A6 and Table A7 below.

Concretely, we use our condensed graph-free data, which is generated using GNN classification experts, to train a GCN model. Then, the trained GCN model conducts clustering on the original large-scale graph. The clustering results in percentage on Cora and Citeseer datasets are shown by four commonly-used metrics, including clustering accuracy (C-ACC), Normalized Mutual Information (NMI), F1-score (F1), and Adjusted Rand Index (ARI).

Table A6: Performance comparison on Cora in terms of graph node clustering. Best results are in bold and the second best are with underlines.

Clusterings on Cora	C-ACC	NMI	F1	ARI
K-means	50.0	31.7	37.6	23.9
VGAE [29]	59.2	40.8	45.6	34.7
ARGA [42]	64.0	44.9	61.9	35.2
MGAE [58]	68.1	48.9	53.1	**56.5**
AGC [75]	68.9	<u>53.7</u>	65.6	44.8
DAEGC [57]	70.4	52.8	68.2	49.6
SUBLIME [35]	<u>71.3</u>	**54.2**	63.5	<u>50.3</u>
SFGC **(ours)** ($r = 1.3\%$)	70.5	51.9	<u>71.0</u>	43.7
SFGC **(ours)** ($r = 2.6\%$)	69.4	51.3	70.1	42.2
SFGC **(ours)** ($r = 5.2\%$)	**71.8**	53.0	**73.1**	43.8

Table A7: Performance comparison on Citeseer in terms of graph node clustering. Best results are in bold and the second best are with underlines.

Clusterings on Citeseer	C-ACC	NMI	F1	ARI
K-means	54.4	31.2	41.3	28.5
VGAE [29]	39.2	16.3	27.8	10.1
ARGA [42]	57.3	35.0	54.6	34.1
MGAE [58]	66.9	<u>41.6</u>	52.6	<u>42.5</u>
AGC [75]	67.0	41.1	62.5	41.5
DAEGC [57]	<u>67.2</u>	39.7	<u>63.6</u>	41.0
SUBLIME [35]	**68.5**	**44.1**	63.2	**43.9**
SFGC **(ours)** ($r = 0.9\%$)	64.9	38.1	<u>63.6</u>	37.3
SFGC **(ours)** ($r = 1.8\%$)	66.5	39.4	**64.9**	39.7
SFGC **(ours)** ($r = 3.6\%$)	65.3	37.6	63.4	38.0

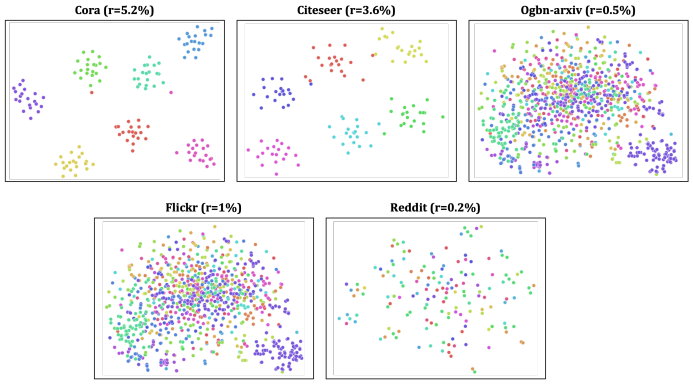

Figure A4: Visualization of t-SNE on condensed graph-free data by SFGC.

As can be observed, our condensed graph enables the GNN model to achieve comparable results with many graph node clustering baseline methods, even though we do not customize the optimization objective targeting node clustering task in the condensation process. These results could justify that: (1) the condensed graph-free data that is synthesized based on GNN classification experts, could also work well in other tasks, even without task-specific customization in the condensation process; (2) the condensed graph-free data contains adequate information about the original large-scale graph, which can be taken as the representative and informative substitution of the original large-scale graph, reflecting the good performance of our proposed method in graph condensation.

E.5 Visualization of Our Condensed Graph-free Data

we present t-SNE [55] plots of the condensed graph-free data of our proposed SFGC under the minimum condensation ratios over all datasets in Fig. A4. The condensed graph-free data shows a well-clustered pattern over Cora and Citeseer. In contrast, on larger-scale datasets with larger condensation ratios, we can also observe some implicit clusters within the same class. These results show that the small-scale graph-free data synthesized by our method has discriminative and representative node attributes that capture comprehensive information from large-scale graphs.