What Matters in Graph Class Incremental Learning? An Information Preservation Perspective

Jialu Li[1,2,3,*] jialuli@tju.edu.cn
Yu Wang[1,2,3,*] wang.yu@tju.edu.cn
Pengfei Zhu[1,2,3,†] zhupengfei@tju.edu.cn
Wanyu Lin[4] wan-yu.lin@polyu.edu.hk
Qinghua Hu[1,2,3] huqinghua@tju.edu.cn

[1]College of Intelligence and Computing, Tianjin University, Tianjin, China
[2]Engineering Research Center of City Intelligence and Digital Governance, Ministry of Education of the People's Republic of China, Tianjin, China
[3]Haihe Lab of ITAI, Tianjin, China
[4]Department of Computing, The Hong Kong Polytechnic University, Hong Kong, China

Abstract

Graph class incremental learning (GCIL) requires the model to classify emerging nodes of new classes while remembering old classes. Existing methods are designed to preserve effective information of old models or graph data to alleviate forgetting, but there is no clear theoretical understanding of what matters in information preservation. In this paper, we consider that present practice suffers from high semantic and structural shifts assessed by two devised shift metrics. We provide insights into information preservation in GCIL and find that maintaining graph information can preserve information of old models in theory to calibrate node semantic and graph structure shifts. We correspond graph information into low-frequency local-global information and high-frequency information in spatial domain. Based on the analysis, we propose a framework, Graph Spatial Information Preservation (GSIP). Specifically, for low-frequency information preservation, the old node representations obtained by inputting replayed nodes into the old model are aligned with the outputs of the node and its neighbors in the new model, and then old and new outputs are globally matched after pooling. For high-frequency information preservation, the new node representations are encouraged to imitate the near-neighbor pair similarity of old node representations. GSIP achieves a 10% increase in terms of the forgetting metric compared to prior methods on large-scale datasets. Our framework can also seamlessly integrate existing replay designs. The code is available through https://github.com/Jillian555/GSIP.

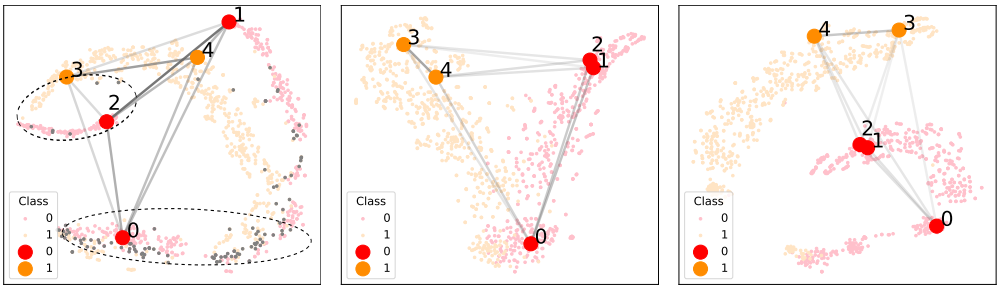
1 Introduction

In real-world applications, graph data is continuously generated. For instance, in citation networks, new types of papers and their citations may constantly emerge, an ideal literature classifier needs to continuously distinguish literature in emerging research areas [1, 2]. Therefore, it is critical for a graph model to incrementally integrate new classes on an extended graph, which is referred to as Graph class incremental learning (GCIL). However, this poses a major challenge known as catastrophic forgetting, where the model needs to preserve previous information while continuously acquiring new information [3, 4].

[*]Equal contribution.
[†]Corresponding author.

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

Many approaches attempt to preserve information from previous models or graph data to prevent catastrophic forgetting in GCIL, which can be divided into four groups. The parameter isolation methods entirely or partially preserve parameters of different tasks to protect model performance, such as dynamically incrementing feature extractors and prototypes [2]. Regularization methods, on the one hand, preserve important parameters, such as assessing parameter importance by considering loss and topology [1], and maintaining orthogonality with parameters from previous tasks [5], on the other hand, preserve the absolute position of nodes in feature space or output space, such as aligning the outputs of samples on old and new models [6]. The replay methods preserve a few nodes or subgraphs to retrain the model to prevent forgetting, such as saving representative nodes [7], selecting subgraphs according to node degree [8], and compressing training graphs [9]. The hybrid methods combine different learning paradigms (*i.e.* the combination of replay and regularization methods), such as feature distillation after identifying critical nodes [10] and minimizing distribution disparity of selected nodes across new and prior models [11]. These hybrid methods have demonstrated considerable potential and achieved state-of-the-art results. Despite their effectiveness, the information preservation mechanism by existing methods remains unclear, making it challenging to develop effective solutions for GCIL. This motivates us to explore a fresh perspective: ***What matters in information preservation when learning from the old model to the new model for GCIL?***



(a) Baseline (b) Ours (c) Target

Figure 1: Visualizations of semantic shift and structure shift.

We investigate the unique characteristics of catastrophic forgetting on graphs and find **node semantic and graph structure shifts** in GCIL. The visualization of node embeddings for new model of baseline (ERGNN [7]) and our method on old classes of CoraFull dataset are exhibited in Figure 1. The structure learned by old model is selected as target instead of original topology due to noisy real structure [12]. Five nodes belonging to two old classes are randomly selected and connected (darker edges indicate more similar features). We detect distortion in the features of baseline (Figure 1(a)) relative to those of the target (Figure 1(c)), especially nodes located within the black dotted box. The two categories can be well separated in the feature distribution of target but not in the baseline model and lead to false predictions (grey nodes in Figure 1(a)). The topological correlation is also significantly changed, node #2 is in proximity to node #1 but is distant from node #3. Surprisingly, within the representation space of baseline, node #2 appears to be moving closer to node #3 while simultaneously becoming more distant from node #1, which exacerbates catastrophic forgetting. Our method (Figure 1(b)) designs graph information preservation modules to mitigate shifts successfully.

In this paper, we inspect GCIL from the perspective of information preservation and theoretically find a key factor in reducing catastrophic forgetting risk with hybrid methods is preserving old graph information. We correspond graph information into low-frequency local-global information and high-frequency information in spatial domain. Subsequently, a Graph Spatial Information Preservation (GSIP) framework is proposed for calibrating semantic and structural shifts. In detail, the old and new representations of nodes are obtained after replay graph data is input to old and new models. The old representations of a node are locally aligned with new representations of a node and its neighbors. Further, old and new representations are globally matched after mean pooling. Finally, new representations of nodes are encouraged to mimic neighbor distance similarities that appear in old representations.

The proposed GSIP can outperform existing replay designs by up to 10% in terms of the forgetting metric on large-scale datasets. It is easy to implement and can be easily adapted to information-preserving approaches to boost their performance. Experiments show that GSIP greatly improves

over current information-preserving methods under different experimental settings and calibrates node semantic and graph structure shifts. Our main contributions can be summarized as follows:

- We provide theoretical insights into GCIL and find that preserving old graph information corresponding to low-frequency local-global and high-frequency information in spatial domain can calibrate semantic and structural shifts and reduce catastrophic forgetting risk.
- We propose a simple yet effective method that utilizes node representations on old and new models to preserve node features, graph representations, and neighbor distances.
- By combining with graph replay-based methods, our framework consistently achieves performance improvements across several benchmark datasets and shows the effectiveness of all the proposed components.

2 Related Work

2.1 Incremental Learning

Incremental learning requires the model to retain the capability of predicting old tasks while acquiring information about new ones [3, 13, 14, 15, 16, 17, 18]. Class incremental learning is not assigned a task ID and has greater training difficulty than task incremental learning [19, 20]. Existing methods can be categorized into three groups. Parameter isolation methods dynamically adapt the model without restricting its structure and capacity, providing distinct parameters for each task [21, 22, 23, 24, 25]. Replay-based methods replay a subset of examples stored in previous tasks or generated using generative models to mitigate forgetting [26, 27, 28, 29]. Regularization-based methods introduce an additional regularization term in the loss function to prevent modifications to crucial parameters related to previous tasks [6, 30, 31, 32, 33].

Traditional incremental learning methods for images or text lack topology learning, making it challenging to achieve effective topology mining and information preservation. By contrast, we analyze the basics of preventing catastrophic forgetting in GCIL from information preservation and solve them in the spatial domain.

2.2 Graph Incremental Learning

Graph incremental learning focuses on handling streaming graph data, and numerous methods have been developed explicitly for graph data [34, 35, 36, 37, 38, 39, 40, 41]. Topology-aware Weight Preserving (TWP) preserves key parameters and topology of previous tasks through regularization terms [1]. Experience Replay Graph Neural Network (ERGNN) framework incorporates memory replay by storing representative nodes [7]. Sparsified Subgraph Memory (SSM) stores sampled sparse subgraphs in a memory repository to preserve structural information [8]. Su *et al.* introduced regularization terms to mitigate catastrophic forgetting from structural drift [11]. Zhang *et al.* redesigned the architecture into a three-layer prototype that adaptively selects different parameter combinations for different tasks [2]. The Condense and Train (CaT) [9] framework compresses the graph into a small but informative synthetic replay graph. Furthermore, two graph incremental learning benchmarks have recently been developed [42, 43].

In comparison, GSIP combines graph information preservation to avoid catastrophic forgetting through low-frequency local-global and high-frequency information preservation.

3 Problem Analysis

Graph Class Incremental Learning (GCIL). GCIL addresses the problem of supervised node classification within the context of an expanding graph. Specifically, each G^t denotes a newly emerging subgraph within the overarching graph. A G^t consists of a node set V^t and an edge set $\mathcal{E}^t$ with its connectivity captured by adjacency matrix $A^t \in \mathbb{R}^{n\times n}$, where n is the number of nodes. Each vertex v is associated with node features X_v and a target label $Y_v \in \{0,1\}^c$, where c represents the total number of classes. At time t, the GCIL problem denoted as $\mathbb{P}_{GCIL}$ is provided with a subgraph $G^t = \{X^t, A^t\}$. The $\mathbb{P}_{GCIL}$ problem is formally defined with the following signature:

$$\mathbb{P}^t_{GCIL} : \left\langle f^{t-1}, \left(G^t, Y^t\right), \mathcal{M}^{t-1}\right\rangle \rightarrow \left\langle f^t, \mathcal{M}^t\right\rangle, \tag{1}$$

where f is graph neural networks and $\mathcal{M}$ signifies an external memory capable of storing a subset of training nodes or other useful graph data.

In the scenario of graph incremental learning, disparate tasks are mapped into distinct partitions of the graph. Once the learning for a specific task is completed, access to the corresponding data is restricted. Our objective is to learn a shared graph neural network model that distinguishes all classes from existing ones. Formally, we aim to minimize the loss caused by previously seen nodes at time step τ in $\mathbb{P}_{GCIL}$, the statistical risk of catastrophic forgetting is defined as:

$$\min_{\theta^\tau} \sum_{t=1}^{\tau} \mathbb{E}_{(G^t,Y^t)} \left[\mathcal{H} \left(Y^t, \sigma \left(f \left(G^t; \theta^\tau \right) \right) \right) \right], \tag{2}$$

where θ indicates parameters of the model, $\mathcal{H}$ represents cross-entropy loss, and σ denotes softmax activation function.

Replay-Based GCIL. Replay-based methods store replayed nodes or subgraphs in memory $\mathcal{M}$ by sampling. Catastrophic forgetting is solved by maintaining the historical distribution. These methods train a new model by minimizing loss of old task nodes on new model concerning the true labels. Given node representations on new model $Z^{new}(Z = f(\mathcal{M}; \theta))$, the replay loss is calculated as:

$$\mathcal{L}_{replay} = \frac{1}{|\mathcal{M}|} \sum_{i \in \mathcal{M}} \mathcal{H} \left(Y_i, \sigma \left(Z_i^{new} \right) \right). \tag{3}$$

Semantic Shift and Structural Shift. Due to memory and privacy limitations, a large amount of old graph data cannot be accessed in graph incremental learning, which leads to material information of old models being gradually forgotten and seriously damages new model performance on old classes. We design two novel shift metrics measuring semantic and structural forgetting degrees when trained on novel classes to show that model divergence manifests in node-level semantics and graph-level structure aspects. On CoraFull dataset, we conduct shift tests using model representations Z^{old} and Z^{new} generated by classical replay method ERGNN [7]. Specifically, central kernel alignment [44] scheme is leveraged to compute Semantic Shift Score (SSS_X):

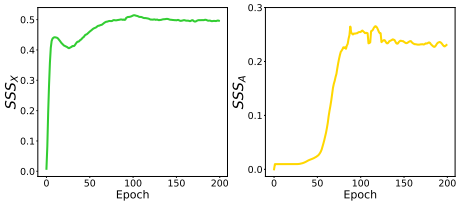


Figure 2: Semantic shift (left) and structural shift (right) between old and new models.

$$SSS_X(Z^{old}, Z^{new}) = 1 - \frac{HS(Z^{old}, Z^{new})}{\sqrt{HS(Z^{old}, Z^{old}) HS(Z^{new}, Z^{new})}}, HS(Z^{old}, Z^{new}) = \frac{tr(Z^{old} C Z^{new} C)}{(n-1)^2}, \tag{4}$$

where C is centering matrix $C_n = I_n - 1/n \mathbf{1}\mathbf{1}^\top$. In particular, Structural Shift Score (SSS_A) is derived by performing structure $\widehat{A}$ inference using feature cosine similarity, then computing differences between graph representations obtained by Anonymous Walk Embedding (AWE) [45]:

$$SSS_A(Z^{old}, Z^{new}) = 1 - COS(AWE(\widehat{A}^{old}), AWE(\widehat{A}^{new})), \widehat{A}_{ij} = \mathbb{1}\left[COS(Z_i, Z_j) > \delta \right], \tag{5}$$

where cosine similarity function $COS(a, b) = a^\top b / (\|a\| \, \|b\|)$ is used to calculate feature similarity degree, and δ is similarity threshold. Each task is trained for 200 epochs, and shift scores range from 0 (no shift) to 1 (completely different). We observe that SSS_X and SSS_A in Figure 2 gradually rise with the increase of epochs. Serious shifts are found in both node semantic and graph structure between old and new models as new classes are trained.

4 Graph Spatial Information Preservation

4.1 Graph Information Preservation

Model information preservation for GCIL can be defined as the mutual information of graph information across old and new models when considering the corresponding model parameters:

$$\mathcal{P}_{\theta^{old} \to \theta^{new}} = \mathcal{I} \left(\mathscr{Z}^{old}; \mathscr{Z}^{new} \right), \tag{6}$$

here, $\mathscr{Z}^{old}$ and $\mathscr{Z}^{new}$ are graph information on old and new models. We directly maximize mutual information between $\mathscr{Z}^{old}$ and $\mathscr{Z}^{new}$, which inherits powerful encoding capability of θ^{old} to θ^{new}.

Proposition 1. *The upper bound on graph information preservation can be estimated as:*

$$-\mathcal{I}\left(\mathscr{Z}^{old};\mathscr{Z}^{new}\right) \leq \|\mathscr{Z}^{old}-\mathscr{Z}^{new}\|_2^2 = \|\Delta\mathscr{Z}\|_2^2, \tag{7}$$

we expect to maximize mutual information $\mathcal{I}\left(\mathscr{Z}^{old};\mathscr{Z}^{new}\right)$, thus minimizing $-\mathcal{I}\left(\mathscr{Z}^{old};\mathscr{Z}^{new}\right)$ in estimation is needed.

Proposition 1 is proved in Appendix A.1, which suggests that graph information preservation is bounded with the square of Euclidean norm between old graph information $\mathscr{Z}^{old}$ and new graph information $\mathscr{Z}^{new}$.

4.2 Spatial Property

Based on the spatial properties of graphs, we analyze the maintenance of graph information in spatial domain to capture complex spatial relationships between nodes and edges in graphs.

Lemma 1. *(Graph spatial information factorization [46]) The graph convolution between convolution kernel $\mathcal{F}$ and the signal $\mathbf{x}$ to obtain graph information is formulated as follows:*

$$\mathcal{F}*\mathbf{x} = \frac{1}{2}\left(\mathcal{F}^l+\mathcal{F}^h\right)*\mathbf{x} = \frac{1}{2}\left(\mathcal{F}^l*\mathbf{x}+\mathcal{F}^h*\mathbf{x}\right) = \mathbf{x}, \tag{8}$$

where $\mathcal{F}^l*\mathbf{x}$ and $\mathcal{F}^h*\mathbf{x}$ are low-/high- frequency graph information, $\mathcal{F}^l = I_n+\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}$, $\mathcal{F}^h = I_n-\widetilde{D}^{-\frac{1}{2}}\widetilde{A}\widetilde{D}^{-\frac{1}{2}}$, $\widetilde{D}$ is diagonal degree matrix with $\widetilde{D}_{i,i}=\sum_j\widetilde{A}_{i,j}$, and $\widetilde{A}=A+I_n$ represents adjacency matrix with self-loop. Two pieces of information in spatial domain can be derived as follows:

$$\mathcal{F}^l*\mathbf{x}_i \rightarrow \mathbf{x}_i^l = x_i+\sum_{j\in\mathcal{N}_i}\frac{x_j}{\sqrt{|\mathcal{N}_i|\,|\mathcal{N}_j|}}, \mathcal{F}^h*\mathbf{x}_i \rightarrow \mathbf{x}_i^h = x_i-\sum_{j\in\mathcal{N}_i}\frac{x_j}{\sqrt{|\mathcal{N}_i|\,|\mathcal{N}_j|}}, \tag{9}$$

where $\mathcal{N}$ represents node neighbors.

According to Lemma 1, there is an identity map that filters out graph information $\mathscr{Z}$ with graph convolution, which provides effective solutions to correspond $\mathscr{Z}^{old}$ and $\mathscr{Z}^{new}$ to spatial domain. For each component i, low-frequency information preserving $\left\|\Delta\mathscr{Z}_i^l\right\|_2^2$ is defined as:

$$\left\|\Delta\mathscr{Z}_i^l\right\|_2^2 = \left\|\left(Z_i^{old}+\sum_{j\in\mathcal{N}_i}\frac{Z_j^{old}}{\sqrt{|\mathcal{N}_i|\,|\mathcal{N}_j|}}\right)-\left(Z_i^{new}+\sum_{j\in\mathcal{N}_i}\frac{Z_j^{new}}{\sqrt{|\mathcal{N}_i|\,|\mathcal{N}_j|}}\right)\right\|_2^2, \tag{10}$$

where Z^{old} and Z^{new} denote obtained representations on old and new models. This implies that the low-frequency information preserving is a semantic gap between the sum of node features and their neighbor features of replay data on old and new models.

Sustained global connectivity is crucial to avert the erasure of global semantic information inherited from the preceding model. As displayed in Figure 3, global semantic shift does exist. We extend the concept of first-hop neighbor nodes in the previous equation to include the entire replay graph (*i.e.* multi-hop neighbors), which is denoted as:

$$\left\|\Delta\mathscr{Z}_i^{\widehat{l}}\right\|_2^2 = \left\|\left(Z_i^{old}+\sum_{j\in\mathcal{M}}\frac{Z_j^{old}}{\sqrt{|\mathcal{M}|\,|\mathcal{M}|}}\right)-\left(Z_i^{new}+\sum_{j\in\mathcal{M}}\frac{Z_j^{new}}{\sqrt{|\mathcal{M}|\,|\mathcal{M}|}}\right)\right\|_2^2. \tag{11}$$

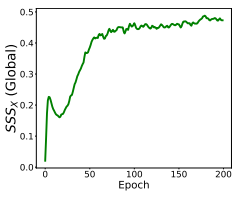


Figure 3: Global semantic shift on old and new models.

Similarly, the generalized low-frequency information preserving is the gap between the sum of node features and all replay data features on old and new models. It is worth noting that Eq. (10) provides a semantic comparison from a local perspective, whereas Eq. (11) compares from a global perspective.

For each component i, high-frequency information preserving $\left\|\Delta\mathscr{Z}_i^h\right\|_2^2$ is defined as:

$$\left\|\Delta\mathscr{Z}_i^h\right\|_2^2 = \left\|\left(Z_i^{old}-\sum_{j\in\mathcal{N}_i}\frac{Z_j^{old}}{\sqrt{|\mathcal{N}_i|\,|\mathcal{N}_j|}}\right)-\left(Z_i^{new}-\sum_{j\in\mathcal{N}_i}\frac{Z_j^{new}}{\sqrt{|\mathcal{N}_i|\,|\mathcal{N}_j|}}\right)\right\|_2^2. \tag{12}$$

High-frequency information preserving captures the gap between the difference in node features and neighbor features on old and new models from topological space.

Motivated by the above concepts, we introduce the following definition:

Definition 1. *(Graph spatial information preservation) A graph spatial information preservation model mainly consists of three kinds of information preservation* $\|\Delta\mathscr{Z}\|_2^2 \approx \|\Delta\mathscr{Z}^l\|_2^2 \cup \left\|\Delta\mathscr{Z}^{\widehat{l}}\right\|_2^2 \cup \|\Delta\mathscr{Z}^h\|_2^2$ *(defined in Eq.* (10)*,* (11)*, and* (12)*).*

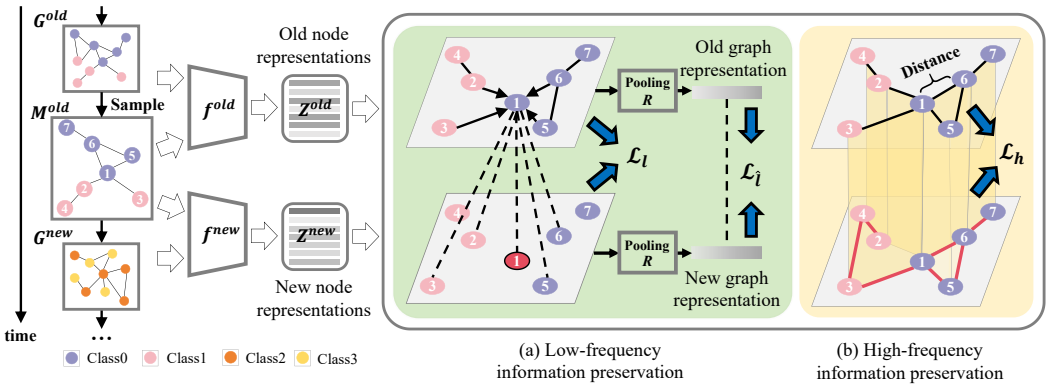


Figure 4: A high-level overview of GSIP framework. It consists of low-/high- frequency modules to preserve old information. The old and new node representations are used to calculate information preserving loss of node representations, graph representations, and neighbor distances.

4.3 Instantiations for Graph Spatial Information Preservation

The above analysis yields two crucial insights: (1) old model information preservation can be solved by preserving the learned graph information; (2) graph information preserving can correspond to low-frequency local-global information and high-frequency information from spatial domain to calibrate node semantic and graph structure shifts. Inspired by these two insights, we propose low-/high-frequency information preservation to adequately capture the old model's information. A high-level overview of GSIP framework is shown in Figure 4. The pseudo-code can be found in Appendix A.4.

Low-Frequency Information Preservation. The node representations within the previous model are derived via iterative feature integration and neighborhood communication, so it contains low-frequency graph information. The information of old model is aligned into the neighborhood of new model to better utilize low-frequency information, which can be represented as:

$$\mathcal{L}_l = \sum_{i\in\mathcal{M}} \sum_{j\in\mathcal{N}_i\cup i} \left\|Z_i^{old}, Z_j^{new}\right\|_2^2, \tag{13}$$

where Z is the output given by model. Low-frequency local information preserving loss uses Mean Squared Error (MSE) loss to locally match representations of nodes on old model Z^{old} with representations of nodes and their neighbors on new model Z^{new}. For replay methods that do not explicitly save neighbors, neighbor selection can be found in Appendix A.2. It is worth noting that since inputs become sparse when converted to probabilities, the softmax followed by Kullback Leibler (KL) divergence loss is not applied [47].

Preserving global information about low-frequency components aligns old model information as a whole and prevents catastrophic forgetting. Low-frequency global information preserving loss is introduced to minimize difference between global representations of old and new models, which is defined as:

$$\mathcal{L}_{\widehat{l}} = \left\|R(Z^{old}), R(Z^{new})\right\|_2^2, \tag{14}$$

where R represents pooling method, which is computed by mean pooling $R(Z) = 1/|\mathcal{M}| \sum_{i\in\mathcal{M}} Z_i$. Similarly, MSE loss is used to calculate global representation gaps.

High-Frequency Information Preservation. The high-frequency part of spatial domain represents the difference between the features of nodes and neighbors. The updated model preserves old topology by incorporating prior local contextual information and then mitigates heterogeneous information propagation blockage caused by smoothness assumption. Specifically, for node v_i, $\mathcal{N}_i$ denotes neighborhood node set and defines $S^{old}(v_i, \mathcal{N}_i)$ as the similarity of selected node vector with adjacent nodes computed by old model:

$$S^{old}\left(v_i, \mathcal{N}_i\right) = \left[S_1^{old}, \ldots, S_{|\mathcal{N}_i|}^{old}\right], S_j^{old} = \frac{\exp\left(\mathcal{K}\left(Z_i^{old}, Z_j^{old}\right)\right)}{\sum_{j' \in \mathcal{N}_i} \exp\left(\mathcal{K}\left(Z_i^{old}, Z_{j'}^{old}\right)\right)}, \tag{15}$$

where $\mathcal{K}(\cdot, \cdot)$ represents kernel function that measures pairwise distances between each node and its neighbors in the latent feature space, and element-wise absolute values $\mathcal{K}\left(Z_i, Z_j\right) = |Z_i - Z_j|$ is used. Then, we measure similarity distribution from new model $S^{new}(v_i, \mathcal{N}_i)$, which is formed by:

$$S^{new}\left(v_i, \mathcal{N}_i\right) = \left[S_1^{new}, \ldots, S_{|\mathcal{N}_i|}^{new}\right], S_j^{new} = \frac{\exp\left(\mathcal{K}\left(Z_i^{new}, Z_j^{new}\right)\right)}{\sum_{j' \in \mathcal{N}_i} \exp\left(\mathcal{K}\left(Z_i^{new}, Z_{j'}^{new}\right)\right)}. \tag{16}$$

High-frequency information preserving is proposed to map neighborhood pairwise differences between old and new models in topological space, information loss from old structure to new structure is more easily recognized with the help of KL divergence, which is denoted as follows:

$$\mathcal{L}_h = \sum_{i \in \mathcal{M}} S^{old}\left(v_i, \mathcal{N}_i\right) \log \frac{S^{old}\left(v_i, \mathcal{N}_i\right)}{S^{new}\left(v_i, \mathcal{N}_i\right)}. \tag{17}$$

Model Learning. To combine different preserving losses, the final graph information preservation loss function is defined as:

$$\mathcal{L}_{gip} = \mathcal{L}_l + \beta \mathcal{L}_{\widehat{l}} + \gamma \mathcal{L}_h, \tag{18}$$

where β and γ are loss weights.

Node classification loss is obtained by $\mathcal{L}_{nc} = 1/|G| \sum_{i \in G} \mathcal{H}\left(Y_i, \sigma\left(f\left(G; \theta^{new}\right)\right)\right)$. Therefore, the overall model learning objective is the weighted sum of current node classification loss, replay loss, and graph information preserving loss:

$$\mathcal{L} = \mathcal{L}_{nc} + \alpha_{replay} \mathcal{L}_{replay} + \alpha_{gip} \mathcal{L}_{gip}, \tag{19}$$

where α_{replay} and α_{gip} are loss weights, and the value of α_{replay} is relevant to the design of replayed method. More analysis about the preservation of other graph frequency information (*i.e.* mid-frequency information and high-frequency global information) is given in Appendix A.3.

5 Experiments

5.1 Datasets and Setups

Datasets and Settings. We utilize five public datasets to evaluate the effectiveness of the proposed method in GCIL, the statistics of datasets are reported in Appendix B.1. Three ways of dividing classes are used: one is divided unequally, and the other two are divided equally, with equal classes per task. The first dataset is CoraFull [48], which has 70 classes, 30 classes are used as base classes for dividing unequally, then 20 classes are used as an increment, and we divide classes equally into 10 or 2 classes per task. Arxiv [49] and Reddit [50], both containing 40 classes, dividing unequally using 10 classes as base classes, then in increments of 5 classes, and dividing equally with 10 or 2 classes per task. Each dataset has 3 tasks with 2 classes per task on Cora [51] and Citeseer [51]. The latest benchmark [42] is employed to implement ERGNN, along with CaT [9] is used to implement SSM and CaT, we follow their settings in graph class incremental learning. Our implementation and detailed settings are available in Appendix B.4 and B.5.

Baselines and Metrics. We compare our method with the following baselines, including Finetuning, Joint, EWC [30], GEM [31], MAS [32], LwF [6], TWP [1], SSRM [11], and three replay-based methods (*i.e.* ERGNN [7], SSM [8], and CaT [9]), where three graph replay methods apply our framework. Finetuning is the lower bound baseline updating the model only with newly emerging graph data. Joint is the ideal upper bound and inputs contain all previous graph data. We choose two widely used metrics to evaluate the performance of the compared methods, including Average Performance (AP) and Average Forgetting (AF) [31].

Table 1: Performance comparison on CoraFull, Arxiv, and Reddit for GCIL setting. Results are averaged among three trials. The best performing results (excluding Joint) are highlighted in **bold**.

Method	CoraFull						Arxiv						Reddit					
	Unequally		Equally (10)		Equally (2)		Unequally		Equally (10)		Equally (2)		Unequally		Equally (10)		Equally (2)	
	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑
Finetuning	23.95	-76.59	11.06	-85.77	2.70	-95.48	11.64	-70.41	5.41	-50.00	4.91	-87.61	14.66	-91.80	22.90	-94.42	5.83	-94.23
EWC	24.09	-75.78	11.15	-86.08	5.13	-93.08	11.93	-68.97	14.83	-57.33	4.91	-87.58	13.79	-95.35	22.30	-95.27	9.66	-93.85
GEM	23.95	-76.05	11.23	-85.78	7.97	-90.00	11.61	-60.27	8.27	-44.42	4.92	-86.66	18.51	-89.79	22.58	-93.93	35.11	-65.67
MAS	24.20	-75.97	10.94	-82.37	4.43	-89.22	11.09	-66.76	12.32	-57.99	5.29	-81.64	15.45	-0.50	25.54	0.01	5.98	-14.17
LwF	23.99	-76.14	11.14	-85.67	2.72	-95.08	11.93	-70.66	14.69	-58.93	4.91	-88.14	16.13	-90.31	24.39	-93.29	7.59	-88.98
TWP	23.86	-75.74	11.01	-85.43	3.56	-94.66	11.93	-69.26	14.41	-56.56	4.90	-87.75	13.95	-96.17	21.22	-96.41	9.34	-94.24
SSRM	63.62	-16.24	31.39	-60.61	3.22	-89.29	31.51	-45.12	26.61	-46.22	26.16	-61.24	78.40	-20.92	76.78	-23.16	83.96	-15.41
ERGNN	60.91	-19.47	24.39	-69.31	3.01	-94.34	31.18	-45.45	24.47	-49.11	24.70	-62.26	76.60	-23.22	75.22	-25.26	83.16	-16.21
+GSIP	67.22	-10.91	71.15	-11.37	44.79	-44.60	34.09	-32.59	33.88	-27.97	40.21	-28.96	90.82	-6.05	89.59	-2.03	93.03	-5.50
Improve ↑	6.31	8.56	46.76	57.94	41.78	49.74	2.91	12.86	9.41	21.14	15.51	33.30	14.22	17.17	14.37	23.23	9.87	10.71
SSM	50.51	-10.56	62.90	-6.02	79.02	-4.24	63.48	-12.41	60.57	-10.09	63.91	-12.48	90.10	-5.83	86.91	-3.24	96.24	-1.64
+GSIP	55.32	-2.50	63.86	**0.08**	79.31	0.70	63.36	-7.27	61.34	-6.34	64.16	-8.87	90.74	-3.97	87.41	0.13	96.25	-0.65
Improve ↑	4.81	8.06	0.96	6.10	0.29	4.94	-0.12	5.14	0.77	3.75	0.25	3.61	0.64	1.86	0.50	3.37	0.01	0.99
CaT	70.55	-5.26	76.35	-5.44	80.64	-4.31	**71.66**	-8.33	70.16	-7.25	66.21	-12.73	**96.39**	-0.77	93.97	-1.31	**97.64**	-0.49
+GSIP	**71.06**	**-0.28**	**78.29**	-1.25	**81.10**	**2.68**	71.52	**-4.76**	**70.57**	**-3.97**	**68.80**	**3.49**	96.15	**-0.23**	**94.23**	**0.21**	97.55	**1.04**
Improve ↑	0.51	4.98	1.94	4.19	0.46	6.99	-0.14	3.57	0.41	3.28	2.59	16.22	-0.24	0.54	0.26	1.52	-0.09	1.53
Joint	85.3	-	85.3	-	85.3	-	63.5	-	63.5	-	63.5	-	98.2	-	98.2	-	98.2	-

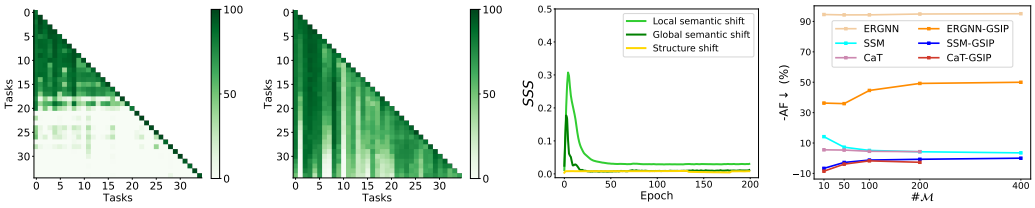
5.2 Performance Comparison

GSIP can improve the performance of existing replay-based information preservation methods. The effect of GCIL on five datasets is presented in Table 1 and Table 2, and the results with standard deviation are presented in Appendix C.1. Joint does not provide AF due to its non-compliance with the incremental learning setting. The existing regularization term relies on the correlation between old and new classes leading to catastrophic forgetting, and some of them do not take topology into account. The hybrid method SSRM absorbs partial old information, resulting in extremely limited performance gains. GSIP consistently demonstrates significant improvements in AP and AF when combined with existing replay methods. For example, ERGNN-GSIP improves both AP and AF by about 10% on Reddit. On CoraFull, SSM-GSIP under unequal partitioning situation improves AP and AF by 4.81% and 8.06%, respectively. CaT-GSIP performs remarkably well on Arxiv, even surpassing the performance of Joint, which has a 16.22% increase in AF on a setting with an increment of 2. CaT experiences a slight decrease in AP in some settings after using GSIP. This can be attributed to GSIP enabling better preservation of graph spatial information from old model, resulting in a lower forgetting rate. In addition, for ERGNN-GSIP, the AP and AF increase by 5.81% and 9.14% on Cora and by 13.64% and 21.74% on Citeseer. On Cora and Citeseer datasets, the AF of SSM-GSIP improves by more than 5%, and CaT-GSIP achieves the highest performance in most cases, with the AP approaching the value of Joint.

Table 2: Performance comparison on Cora and Citeseer for GCIL setting. Results are averaged among three trials. The best performing results (excluding Joint) are highlighted in **bold**.

Method	Cora		Citeseer	
	Equally (2)		Equally (2)	
	AP↑	AF↑	AP↑	AF↑
Finetuning	32.58	-96.83	31.46	-77.86
EWC	32.58	-97.16	31.26	-78.22
GEM	32.70	-97.12	31.39	-77.70
MAS	31.84	-97.17	31.25	-76.67
LwF	32.58	-97.57	31.44	-78.29
TWP	32.58	-97.32	31.22	-78.14
SSRM	35.48	-70.01	51.91	-67.66
ERGNN	65.48	-46.09	47.65	-51.12
+GSIP	71.29	-36.95	61.29	-29.38
Improve ↑	5.81	9.14	13.64	21.74
SSM	67.64	-19.78	60.99	-13.60
+GSIP	69.92	-11.82	61.86	**-8.39**
Improve ↑	2.28	7.96	0.87	5.21
CaT	88.22	-4.40	75.08	-10.93
+GSIP	**89.60**	**1.84**	**77.02**	-9.95
Improve ↑	1.38	6.24	1.94	0.98
Joint	93.09	-	78.27	-

GSIP can consistently achieve excellent performance on old classes in different task IDs. The performance matrices of ERGNN on CoraFull before and after incorporating GSIP are shown in Figure 5(a) and Figure 5(b). It is difficult to remember the information of old classes before employing graph spatial information preservation. After implementing the GSIP scheme, the performance matrix demonstrates a deceleration in the forgetting process (*i.e.*, the color of each column does not change much and the color deepens), which indicates that the catastrophic forgetting problem is mitigated due to the preserving of old graph information.



(a) ERGNN (b) ERGNN-GSIP (c) Shift calibration (d) The analysis of $\#\mathcal{M}$

Figure 5: (a)-(b) Performance matrices on CoraFull dataset. (c) Semantic and structural shift calibration of old and new models during increments. (d) Performance changes affected by $\#\mathcal{M}$ on CoraFull dataset.

GSIP can offer information preservation capability to calibrate semantic shift and structural shift. Figure 5(c) exhibits the curves of shift scores during the incremental process for ERGNN on CoraFull. It can be noted that shift scores start at a relatively high value, gradually decrease, and smooth out after graph spatial information is maintained, demonstrating that the low-frequency local-global information and high-frequency information of old model are well captured, and semantic and structural shifts are nicely calibrated.

5.3 Ablation Study

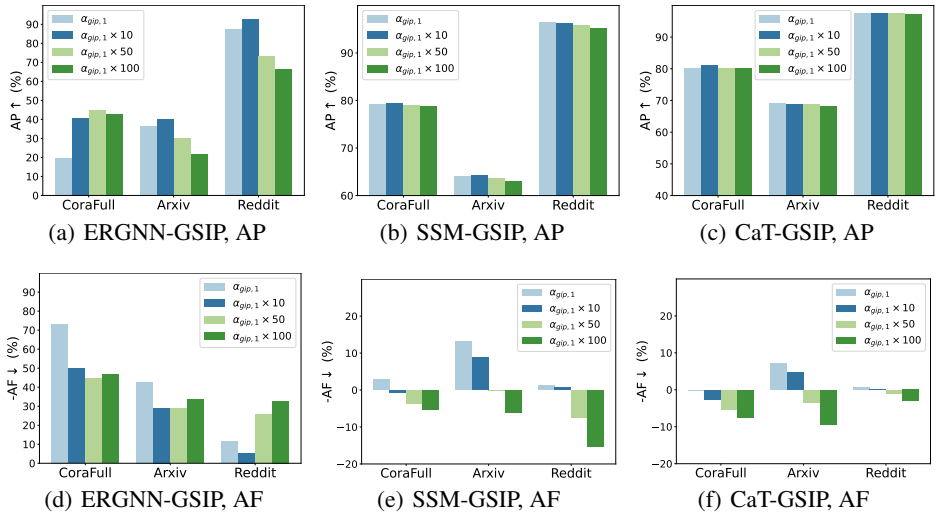
We investigate the effectiveness of low-frequency local modules (LL), low-frequency global modules (LG), and high-frequency modules (H), the experimental results use ERGNN as baseline (B). The results of ERGNN, SSM, and CaT with standard deviation are summarized in Appendix C.2. The above components are added one by one to baseline for performance comparison. From Table 3 we observe that: (1) When LL is utilized, the model can easily learn the aggregation rules of nodes and neighbors from old model locally. AP (AF) improves by about 2% (6%) to 45% (55%) over the baseline demonstrating the superiority of LL. (2) Combining LG significantly improves performance, especially on Reddit. The reason may be that larger datasets have greater overall shifts during increments. (3) H also brings significant improvements, with AP (AF) improving by about 1.6% (0.9%) to 2.4% (7.4%) over B+LL+LG on Reddit. This indicates that the H module can extract more topology information for better performance.

Table 3: Ablation comparisons of graph spatial information preservation.

Method	CoraFull						Arxiv						Reddit					
	Unequally		Equally (10)		Equally (2)		Unequally		Equally (10)		Equally (2)		Unequally		Equally (10)		Equally (2)	
	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑
B	60.91	-19.47	24.39	-69.31	3.01	-94.34	31.18	-45.45	24.47	-49.11	24.70	-62.26	76.60	-23.22	75.22	-25.26	83.16	-16.21
B+LL	65.79	-13.20	69.02	-14.15	41.37	-47.39	33.27	-34.60	27.10	-40.95	38.09	-35.04	84.63	-12.09	84.26	-12.37	87.52	-10.97
B+LL+LG	66.22	-12.78	69.77	-13.13	41.84	-46.73	34.00	-33.61	32.80	-34.40	39.89	**-28.85**	89.21	-6.97	87.17	-9.45	91.34	-7.12
B+LL+LG+H	**67.22**	**-10.91**	**71.15**	**-11.37**	**44.79**	**-44.60**	**34.09**	**-32.59**	**33.88**	**-27.97**	**40.21**	-28.96	**90.82**	**-6.05**	**89.59**	**-2.03**	**93.03**	**-5.50**

5.4 Further Analysis

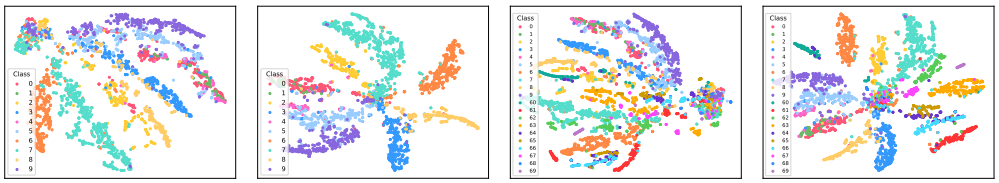
Hyper-Parameter Analysis. We analyze the impact of the number of storage nodes for each task $\#\mathcal{M}$ on performance. As depicted in Figure 5(d), it can be observed that the proposed method consistently outperforms the original method in terms of the -AF metric (the lower, the better), regardless of the value of $\#\mathcal{M}$. Interestingly, even with less memory, the proposed method still achieves better performance. CaT cannot be trained on 400 nodes due to Cuda memory constraints. We analyze the impact of loss weight α_{gip} on ERGNN, SSM, and CaT across CoraFull, Arxiv, and Reddit datasets with increments of 2 in Figure 6. For ERGNN, SSM, and CaT, $\alpha_{gip,1}$ is set to $[1, 1, 0.1]$, $[0.01, 0.01, 0.01]$, and $[0.1, 0.01, 0.01]$ for three datasets. It can be observed that the performance change is not as significant with the variation of α_{gip} on SSM and CaT. However, different α_{gip} has a greater impact on performance with ERGNN-GSIP. The possible reason is that ERGNN selects representative nodes for replay, which may cause class imbalance and topology discarding. For ERGNN, SSM, and CaT, the optimal hyper-parameters α_{gip} on three datasets are $[50, 10, 1]$, $[0.1, 0.1, 0.1]$, and $[1, 0.5, 0.5]$. Because of space limitation, we provide more curves about $\#\mathcal{M}$ in Appendix C.3 and hyper-parameter analysis of loss weights β and γ in Appendix C.4.



(a) ERGNN-GSIP, AP (b) SSM-GSIP, AP (c) CaT-GSIP, AP

(d) ERGNN-GSIP, AF (e) SSM-GSIP, AF (f) CaT-GSIP, AF

Figure 6: The analysis of α_{gip} in ERGNN, SSM, and CaT on CoraFull, Arxiv, and Reddit datasets.



(a) ERGNN, Task 1 (b) ERGNN-GSIP, Task 1 (c) ERGNN, Task 7 (d) ERGNN-GSIP, Task 7

Figure 7: The visualization of node embeddings in Task 1 and Task 7 on CoraFull dataset.

Visulization. To qualitatively demonstrate the effectiveness of our representations, we adopt t-SNE [52] to visualize the learned node embeddings. After learning the last task, Figure 7(a) and Figure 7(b) show the results of the learned node embeddings in Task 1 on CoraFull, while Figure 7(c) and Figure 7(d) demonstrate the results of the last task. We can clearly observe that GSIP possesses better representation ability by considering representations and classifying old and new classes well.

6 Conclusion

We contribute to the literature of GCIL by addressing the issue of information preservation from old model when adapting to new classes. The key insight is that preserving graph information from spatial domain plays a vital role in preserving information about old model, and subsequently calibrates semantic and structural shifts and reduces catastrophic forgetting risk. To accomplish this objective, we introduce a framework, GSIP, which utilizes the outputs of nodes in old model to diffuse the outputs of new model and its neighbors, then aligns the outputs of new model with old model after pooling. Finally, GSIP maintains kernel distance of neighbor pairs on both old and new models. The graph information is remembered by low-frequency local-global information preserving and high-frequency information preserving in feature and topological space. Evaluations over benchmark datasets show the superiority of GSIP in handling different dataset splitting cases. In the future, we will investigate comprehensive analysis for the preservation of complicated graph signals.

Acknowledgement

This work was supported in part by the National Science and Technology Major Project under Grant 2022ZD0116500, in part by the National Natural Science Foundation of China under Grants 62436002, 62476195, 61925602, U23B2049, and 62222608, in part by Tianjin Natural Science Funds for Distinguished Young Scholar under Grant 23JCJQJC00270, and in part by Zhejiang Provincial Natural Science Foundation of China under Grant LD24F020004.

References

[1] Huihui Liu, Yiding Yang, and Xinchao Wang. Overcoming catastrophic forgetting in graph neural networks. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 8653–8661, 2021.

[2] Xikun Zhang, Dongjin Song, and Dacheng Tao. Hierarchical prototype networks for continual graph representation learning. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(4):4622–4636, 2023.

[3] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory G. Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(7):3366–3385, 2022.

[4] Xinting Hu, Kaihua Tang, Chunyan Miao, Xian-Sheng Hua, and Hanwang Zhang. Distilling causal effect of data in class-incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 3957–3966, 2021.

[5] Jie Cai, Xin Wang, Chaoyu Guan, Yateng Tang, Jin Xu, Bin Zhong, and Wenwu Zhu. Multimodal continual graph learning with neural architecture search. In *The ACM Web Conference*, pages 1292–1300, 2022.

[6] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2935–2947, 2018.

[7] Fan Zhou and Chengtai Cao. Overcoming catastrophic forgetting in graph neural networks with experience replay. In *Thirty-Fifth AAAI Conference on Artificial Intelligence*, pages 4714–4722, 2021.

[8] Xikun Zhang, Dongjin Song, and Dacheng Tao. Sparsified subgraph memory for continual graph representation learning. In *IEEE International Conference on Data Mining*, pages 1335–1340, 2022.

[9] Yilun Liu, Ruihong Qiu, and Zi Huang. Cat: Balanced continual graph learning with graph condensation. In *IEEE International Conference on Data Mining*, pages 1157–1162, 2023.

[10] Seoyoon Kim, Seongjun Yun, and Jaewoo Kang. Dygrain: An incremental learning framework for dynamic graphs. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence*, pages 3157–3163, 2022.

[11] Junwei Su, Difan Zou, Zijun Zhang, and Chuan Wu. Towards robust graph incremental learning on evolving graphs. In *International Conference on Machine Learning*, volume 202, pages 32728–32748, 2023.

[12] Seungyoon Choi, Wonjoong Kim, Sungwon Kim, Yeonjun In, Sein Kim, and Chanyoung Park. DSLR: diversity enhancement and structure learning for rehearsal-based graph continual learning. In *The ACM Web Conference*, pages 733–744, 2024.

[13] Dipam Goswami, Yuyang Liu, Bartlomiej Twardowski, and Joost van de Weijer. Fecam: Exploiting the heterogeneity of class distributions in exemplar-free continual learning. In *Advances in Neural Information Processing Systems*, pages 1–14, 2023.

[14] Yaoyao Liu, Bernt Schiele, and Qianru Sun. RMM: reinforced memory management for class-incremental learning. In *Advances in Neural Information Processing Systems*, pages 3478–3490, 2021.

[15] Qi-Wei Wang, Da-Wei Zhou, Yi-Kai Zhang, De-Chuan Zhan, and Han-Jia Ye. Few-shot class-incremental learning via training-free prototype calibration. In *Advances in Neural Information Processing Systems*, pages 1–17, 2023.

[16] Depeng Li, Tianqi Wang, Junwei Chen, Qining Ren, Kenji Kawaguchi, and Zhigang Zeng. Towards continual learning desiderata via hsic-bottleneck orthogonalization and equiangular embedding. In *Thirty-Eighth AAAI Conference on Artificial Intelligence*, pages 13464–13473, 2024.

[17] Liyuan Wang, Jingyi Xie, Xingxing Zhang, Mingyi Huang, Hang Su, and Jun Zhu. Hierarchical decomposition of prompt-based continual learning: Rethinking obscured sub-optimality. In *Advances in Neural Information Processing Systems*, pages 1–23, 2023.

[18] Qi Chen, Changjian Shui, Ligong Han, and Mario Marchand. On the stability-plasticity dilemma in continual meta-learning: Theory and algorithm. In *Advances in Neural Information Processing Systems*, pages 1–15, 2023.

[19] Marc Masana, Xialei Liu, Bartlomiej Twardowski, Mikel Menta, Andrew D. Bagdanov, and Joost van de Weijer. Class-incremental learning: Survey and performance evaluation on image classification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 45(5):5513–5533, 2023.

[20] Songsong Tian, Lusi Li, Weijun Li, Hang Ran, Xin Ning, and Prayag Tiwari. A survey on few-shot class-incremental learning. *Neural Networks*, 169:307–324, 2024.

[21] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *CoRR*, abs/1606.04671, 2016.

[22] Jaehong Yoon, Eunho Yang, Jeongtae Lee, and Sung Ju Hwang. Lifelong learning with dynamically expandable networks. In *Sixth International Conference on Learning Representations*, pages 1–11, 2018.

[23] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 374–382, 2019.

[24] Mitchell Wortsman, Vivek Ramanujan, Rosanne Liu, Aniruddha Kembhavi, Mohammad Rastegari, Jason Yosinski, and Ali Farhadi. Supermasks in superposition. In *Advances in Neural Information Processing Systems*, pages 15173–15184, 2020.

[25] Johannes von Oswald, Christian Henning, João Sacramento, and Benjamin F. Grewe. Continual learning with hypernetworks. In *Eighth International Conference on Learning Representations*, pages 1–12, 2020.

[26] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems*, pages 2990–2999, 2017.

[27] Jeremias Knoblauch, Hisham Husain, and Tom Diethe. Optimal continual learning has perfect memory and is np-hard. In *Proceedings of the Thirty-Seventh International Conference on Machine Learning*, volume 119, pages 5327–5337, 2020.

[28] Lucas Caccia, Eugene Belilovsky, Massimo Caccia, and Joelle Pineau. Online learned continual compression with adaptive quantization modules. In *Proceedings of the Thirty-Seventh International Conference on Machine Learning*, volume 119, pages 1240–1250, 2020.

[29] Liyuan Wang, Xingxing Zhang, Kuo Yang, Longhui Yu, Chongxuan Li, Lanqing Hong, Shifeng Zhang, Zhenguo Li, Yi Zhong, and Jun Zhu. Memory replay with data compression for continual learning. In *The Tenth International Conference on Learning Representations*, pages 1–13, 2022.

[30] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

[31] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017.

[32] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *European Conference on Computer Vision*, volume 11207, pages 144–161, 2018.

[33] Yilin Lyu, Liyuan Wang, Xingxing Zhang, Zicheng Sun, Hang Su, Jun Zhu, and Liping Jing. Overcoming recency bias of normalization statistics in continual learning: Balance and adaptation. In *Advances in Neural Information Processing Systems*, pages 1–12, 2023.

[34] Qiao Yuan, Sheng-Uei Guan, Pin Ni, Tianlun Luo, Ka Lok Man, Prudence W. H. Wong, and Victor Chang. Continual graph learning: A survey. *CoRR*, abs/2301.12230, 2023.

[35] Falih Gozi Febrinanto, Feng Xia, Kristen Moore, Chandra Thapa, and Charu Aggarwal. Graph lifelong learning: A survey. *IEEE Comput. Intell. Mag.*, 18(1):32–51, 2023.

[36] Zonggui Tian, Du Zhang, and Hong-Ning Dai. Continual learning on graphs: A survey. *CoRR*, abs/2402.06330, 2024.

[37] Dongjie Wang, Zhengzhang Chen, Yanjie Fu, Yanchi Liu, and Haifeng Chen. Incremental causal graph learning for online root cause analysis. In *Proceedings of the Twenty-Ninth ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 2269–2278, 2023.

[38] Chen Wang, Yuheng Qiu, Dasong Gao, and Sebastian A. Scherer. Lifelong graph learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13709–13718, 2022.

[39] Bin Lu, Xiaoying Gan, Lina Yang, Weinan Zhang, Luoyi Fu, and Xinbing Wang. Geometer: Graph few-shot class-incremental learning via prototype representation. In *The Twenty-Eighth ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 1152–1161, 2022.

[40] Thanh Duc Hoang, Do Viet Tung, Duy-Hung Nguyen, Bao-Sinh Nguyen, Huy Hoang Nguyen, and Hung Le. Universal graph continual learning. *Trans. Mach. Learn. Res.*, 1(1):1–15, 2023.

[41] Jiajun Liu, Wenjun Ke, Peng Wang, Ziyu Shang, Jinhua Gao, Guozheng Li, Ke Ji, and Yanhe Liu. Towards continual knowledge graph embedding via incremental distillation. In *Thirty-Eighth AAAI Conference on Artificial Intelligence*, pages 8759–8768, 2024.

[42] Xikun Zhang, Dongjin Song, and Dacheng Tao. CGLB: benchmark tasks for continual graph learning. In *Advances in Neural Information Processing Systems*, pages 13006–13021, 2022.

[43] Jihoon Ko, Shinhwan Kang, and Kijung Shin. Begin: Extensive benchmark scenarios and an easy-to-use framework for graph continual learning. *CoRR*, abs/2211.14568, 2022.

[44] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey E. Hinton. Similarity of neural network representations revisited. In *Proceedings of the Thirty-Sixth International Conference on Machine Learning*, volume 97, pages 3519–3529, 2019.

[45] Eunjeong Jeong, Seungeun Oh, Hyesung Kim, Jihong Park, Mehdi Bennis, and Seong-Lyun Kim. Communication-efficient on-device machine learning: Federated distillation and augmentation under non-iid private data. *CoRR*, abs/1811.11479, 2018.

[46] Lirong Wu, Haitao Lin, Yufei Huang, Tianyu Fan, and Stan Z. Li. Extracting low-/high-frequency knowledge from graph neural networks and injecting it into mlps: An effective gnn-to-mlp distillation framework. In *Thirty-Seventh AAAI Conference on Artificial Intelligence*, pages 10351–10360, 2023.

[47] Chaitanya K. Joshi, Fayao Liu, Xu Xun, Jie Lin, and Chuan Sheng Foo. On representation knowledge distillation for graph neural networks. *IEEE Trans. Neural Networks Learn. Syst.*, 35(4):4656–4667, 2024.

[48] Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *Sixth International Conference on Learning Representations*, pages 1–11, 2018.

[49] Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. In *Advances in Neural Information Processing Systems*, pages 22118–22133, 2020.

[50] William L. Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in Neural Information Processing Systems*, pages 1024–1034, 2017.

[51] Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. Collective classification in network data. *AI Mag.*, 29(3):93–106, 2008.

[52] Laurens van der Maaten. Accelerating t-sne using tree-based algorithms. *J. Mach. Learn. Res.*, 15(1):3221–3245, 2014.

[53] Sungsoo Ahn, Shell Xu Hu, Andreas C. Damianou, Neil D. Lawrence, and Zhenwen Dai. Variational information distillation for knowledge transfer. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 9163–9171, 2019.

[54] Jincheng Huang, Lun Du, Xu Chen, Qiang Fu, Shi Han, and Dongmei Zhang. Robust mid-pass filtering graph convolutional networks. In *the ACM Web Conference*, pages 328–338, 2023.

[55] Haitong Luo, Xuying Meng, Suhang Wang, Hanyun Cao, Weiyao Zhang, Yequan Wang, and Yujun Zhang. Spectral-based graph neural networks for complementary item recommendation. In *Thirty-Eighth AAAI Conference on Artificial Intelligence*, pages 8868–8876, 2024.

A Method

A.1 The Proof of Proposition 1

Proposition 1. *The upper bound on graph information preservation can be estimated as:*

$$-\mathcal{I}\left(\mathscr{Z}^{old};\mathscr{Z}^{new}\right) \leq \|\mathscr{Z}^{old}-\mathscr{Z}^{new}\|_2^2 = \|\Delta\mathscr{Z}\|_2^2, \tag{20}$$

we expect to maximize mutual information $\mathcal{I}\left(\mathscr{Z}^{old};\mathscr{Z}^{new}\right)$, thus minimizing $-\mathcal{I}\left(\mathscr{Z}^{old};\mathscr{Z}^{new}\right)$ in estimation is needed.

Proof. The mutual information of graph information on old model and new model [53] can be equivalent to:

$$\begin{aligned}
\mathcal{I}\left(\mathscr{Z}^{old};\mathscr{Z}^{new}\right) &= \int d\mathscr{Z}^{new}d\mathscr{Z}^{old}p(\mathscr{Z}^{new},\mathscr{Z}^{old})\log\left(\frac{p(\mathscr{Z}^{new},\mathscr{Z}^{old})}{p(\mathscr{Z}^{new})p(\mathscr{Z}^{old})}\right)\\
&= \sum_{\mathscr{Z}^{new},\mathscr{Z}^{old}} p(\mathscr{Z}^{new},\mathscr{Z}^{old})\log\left(\frac{p(\mathscr{Z}^{new},\mathscr{Z}^{old})}{p(\mathscr{Z}^{new})p(\mathscr{Z}^{old})}\right)\\
&= \sum_{\mathscr{Z}^{new},\mathscr{Z}^{old}} p(\mathscr{Z}^{new},\mathscr{Z}^{old})\log\left(\frac{p(\mathscr{Z}^{new},\mathscr{Z}^{old})}{p(\mathscr{Z}^{new})}\right)\\
&- \sum_{\mathscr{Z}^{new},\mathscr{Z}^{old}} p(\mathscr{Z}^{new},\mathscr{Z}^{old})\log p(\mathscr{Z}^{old})\\
&= \sum_{\mathscr{Z}^{new},\mathscr{Z}^{old}} p(\mathscr{Z}^{new})p(\mathscr{Z}^{old}\mid\mathscr{Z}^{new})\log(p(\mathscr{Z}^{old}\mid\mathscr{Z}^{new}))\\
&- \sum_{\mathscr{Z}^{new},\mathscr{Z}^{old}} p(\mathscr{Z}^{new},\mathscr{Z}^{old})\log p(\mathscr{Z}^{old})\\
&= -\sum_{\mathscr{Z}^{new}} p(\mathscr{Z}^{new})\mathcal{H}(\mathscr{Z}^{old}\mid\mathscr{Z}^{new}=\mathscr{Z}^{new}) - \sum_{\mathscr{Z}^{old}}\log p(\mathscr{Z}^{old})p(\mathscr{Z}^{old})\\
&= \mathcal{H}\left(\mathscr{Z}^{old}\right) - \mathcal{H}\left(\mathscr{Z}^{old}\mid\mathscr{Z}^{new}\right)\\
&= \mathcal{H}(\mathscr{Z}^{old}) + \mathbb{E}_{\mathscr{Z}^{old},\mathscr{Z}^{new}}[\log p(\mathscr{Z}^{old}\mid\mathscr{Z}^{new})]\\
&= \mathcal{H}(\mathscr{Z}^{old}) + \mathbb{E}_{\mathscr{Z}^{old},\mathscr{Z}^{new}}[\log q(\mathscr{Z}^{old}\mid\mathscr{Z}^{new})]\\
&+ \mathbb{E}_{\mathscr{Z}^{new}}\left[\mathcal{D}_{KL}(p(\mathscr{Z}^{old}\mid\mathscr{Z}^{new})\|q(\mathscr{Z}^{old}\mid\mathscr{Z}^{new}))\right]\\
&\geq \mathbb{E}_{\mathscr{Z}^{old},\mathscr{Z}^{new}}[\log q(\mathscr{Z}^{old}\mid\mathscr{Z}^{new})],
\end{aligned} \tag{21}$$

where expectations are over distribution $p(\mathscr{Z}^{old},\mathscr{Z}^{new})$, $\mathcal{H}(\mathscr{Z}^{old})$ has been removed since it is constant with respect to optimized parameters. The reason for the inequality is the non-negativity of entropy $\mathcal{H}$ and Kullback-Leiber divergence $\mathcal{D}_{KL}$.

We expect to maximize mutual information $\mathcal{I}\left(\mathscr{Z}^{old};\mathscr{Z}^{new}\right)$, so we need to minimize $-\mathcal{I}\left(\mathscr{Z}^{old};\mathscr{Z}^{new}\right)$ in the loss function, hence we can obtain:

$$-\mathcal{I}\left(\mathscr{Z}^{old};\mathscr{Z}^{new}\right) \leq -\mathbb{E}_{\mathscr{Z}^{old},\mathscr{Z}^{new}}[\log q(\mathscr{Z}^{old}\mid\mathscr{Z}^{new})]. \tag{22}$$

The conditional likelihood is maximized to fit the information of old model, and the new model receives compressed information needed to recover old model. A Gaussian distribution with mean $\boldsymbol{\mu}$ and variance $\boldsymbol{\sigma}$ is employed to estimate variational distribution $q(\mathscr{Z}^{old}\mid\mathscr{Z}^{new})$, which is expressed as follows:

$$\begin{aligned}
-\mathcal{I}\left(\mathscr{Z}^{old};\mathscr{Z}^{new}\right) &\leq -\sum_{i=1}^{|\mathcal{M}|}\log q\left(\mathscr{Z}_i^{old}\mid\mathscr{Z}^{new}\right)\\
&= \sum_{i=1}^{|\mathcal{M}|}\log\boldsymbol{\sigma}_i + \frac{\left(\mathscr{Z}_i^{old}-\boldsymbol{\mu}_i(\mathscr{Z}^{new})\right)^2}{2\boldsymbol{\sigma}_i^2} + \mathcal{C},
\end{aligned} \tag{23}$$

where $\mathcal{C}$ is constant. Mean squared error matching can be seen as a specific instance of the above formula when $\boldsymbol{\mu}(\mathscr{Z}^{new}) = \mathscr{Z}^{new}$ and $\boldsymbol{\sigma} = 1$, thus $-\mathcal{I}\left(\mathscr{Z}^{old};\mathscr{Z}^{new}\right) \leq \|\mathscr{Z}^{old}-\mathscr{Z}^{new}\|_2^2$. □

A.2 Neighbor Selection

Some replay-based methods do not explicitly save topology, so we treat nodes with high representation similarity and label correlation on old model as neighbors for each node when computing low-frequency local information preserving loss. Nodes whose feature similarity is greater than the given threshold and whose labels are the same can be neighbors, formulated as follows:

$$\mathcal{N}_i = \{j \mid (COS(Z_i^{old}, Z_j^{old}) > \widehat{\delta}) \cap (Y_i = Y_j), \forall j \in \mathcal{M}\}, i \in \mathcal{M}, \tag{24}$$

where the cosine similarity function $COS(Z_i, Z_j) = Z_i^\top Z_j / (\|Z_i\| \, \|Z_j\|)$ is used to calculate the degree of feature similarity, and $\widehat{\delta}$ is similarity threshold.

We only use node feature similarity when calculating the loss of high-frequency information preservation for neighbor selection. The absence of label correlation captures the distance between nodes on different classes, which is conducive to the model to distinguish different classes. The neighborhood selection process is expressed as follows:

$$\mathcal{N}_i = \{j \mid COS(Z_i^{old}, Z_j^{old}) > \delta, \forall j \in \mathcal{M}\}, i \in \mathcal{M}, \tag{25}$$

where δ is the similarity threshold.

A.3 The Analysis of Other Graph Frequency Information Preservation

We perform analyses and experiments to assess the preservation of various aspects of graph frequency information, including mid-frequency information and high-frequency global information.

Mid-Frequency Information Preservation. According to the definition of mid-frequency filtering graph convolutional networks [54, 55], the mid-frequency convolution can be expressed as:

$$\mathcal{F}^m = (I_n - \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}})(I_n + \widetilde{D}^{-\frac{1}{2}} \widetilde{A} \widetilde{D}^{-\frac{1}{2}}). \tag{26}$$

Through a similar analysis as mentioned above, mid-frequency information preservation is defined as:

$$\|\Delta \mathscr{Z}_i^m\|_2^2 = \left\| \left(Z_i^{old} - \sum_{j \in \mathcal{N}_i^2} \frac{Z_j^{old}}{\sqrt{|\mathcal{N}_i^2| \, |\mathcal{N}_j^2|}} \right) - \left(Z_i^{new} - \sum_{j \in \mathcal{N}_i^2} \frac{Z_j^{new}}{\sqrt{|\mathcal{N}_i^2| \, |\mathcal{N}_j^2|}} \right) \right\|_2^2, \tag{27}$$

where $\mathcal{N}^2$ is the second-hop neighbors of nodes. The distinction between mid-/high- frequency information preservation lies in that mid-frequency signals calculate the differences between the node and its second-hop neighbors.

High-Frequency Global Information Preservation. The general formula for high-frequency global information preservation is represented as follows:

$$\left\| \Delta \mathscr{Z}_i^{\widehat{h}} \right\|_2^2 = \left\| \left(Z_i^{old} - \sum_{j \in \mathcal{M}} \frac{Z_j^{old}}{\sqrt{|\mathcal{M}| \, |\mathcal{M}|}} \right) - \left(Z_i^{new} - \sum_{j \in \mathcal{M}} \frac{Z_j^{new}}{\sqrt{|\mathcal{M}| \, |\mathcal{M}|}} \right) \right\|_2^2. \tag{28}$$

It implies that every node in replayed graph needs to calculate disparity with other nodes. Since graph neural network assumes that neighboring nodes have similar representations, the penalizing distance between nodes and their neighbors at multiple hops away is redundant and does not contribute to structural preservation. Moreover, the inclusion of this term imposes an optimization burden and exhibits high time complexity ($O(|\mathcal{M}|^2 \cdot \mathbb{k})$, where $\mathbb{k}$ denotes the dimensions of the hidden spaces).

We add mid-frequency information preservation (**M**) and high-frequency global information preservation (**HG**) to GSIP in Table 4, which can yield a slight performance improvement in some cases. However, it does not lead to better performance enhancements or outstanding results. The possible reason is that the preservation of first-hop neighbors is sufficient to calibrate structural shift.

A.4 Algorithm

The proposed method is summarized in Algorithm 1.

Table 4: Performance comparison before and after adding other graph frequency information preservation on CoraFull dataset.

Method	Unequally				Equally (10)				Equally (2)			
	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑
GSIP	**55.32** ±0.75	**-2.50** ±1.13	**67.22** ±0.44	-10.91 ±0.62	63.86 ±0.85	0.08 ±0.76	**71.15** ±0.98	**-11.37** ±0.74	**79.31** ±0.50	0.70 ±0.25	**44.79** ±1.77	**-44.60** ±1.67
+M / HG	55.28 ±0.65	-2.61 ±1.04	66.63 ±0.49	**-10.07** ±1.75	**63.89** ±0.86	**0.20** ±0.68	69.96 ±0.85	-12.85 ±0.94	79.29 ±0.67	**0.75** ±0.52	24.66 ±1.47	-68.48 ±1.35

Algorithm 1 Framework of GSIP

Input: At time step $t > 1$: New input G, Memory $\mathcal{M}$, Graph neural networks f, Labels Y, Learned parameter θ^{old}, Max epochs U, Loss weights α_{replay}, α_{gip}

Output: Parameter θ^{new} which can mitigate catastrophic forgetting of preceding classes

Initialize θ^{new} at random
for $u = 1$ to U **do**
    $\mathcal{L}_{nc} = \ell_{ce}(Y^G, f(G; \theta^{new}))$
    $\mathcal{L}_{replay} = \ell_{ce}(Y^{\mathcal{M}}, f(\mathcal{M}; \theta^{new}))$
    $\mathcal{L}_{gip} = \ell_{reg}(f(\mathcal{M}; \theta^{old}), f(\mathcal{M}; \theta^{new}))$
    $\theta^{new} \leftarrow \arg\min \mathcal{L} = \mathcal{L}_{nc} + \alpha_{replay}\mathcal{L}_{replay} + \alpha_{gip}\mathcal{L}_{gip}$
end for
Add selected nodes to memory $\mathcal{M}$
return Parameter θ^{new}

B Implementation Details

B.1 Datasets

As illustrated in Table 5, we utilize five public datasets to evaluate the effectiveness of our proposed method in graph class incremental learning. Three different ways for partitioning datasets are employed: one involves an unequal division, where more classes are designated as base classes to enhance model robustness, while the remaining classes are treated as novel classes; the other two ways involve an equal division, with an equal number of classes allocated per task. The first dataset is CoraFull [48], which encompasses 70 classes. For the unequal division, 30 classes are used as base classes, and an additional 20 classes are selected as increments. Additionally, the classes are divided equally into either 10 or 2 classes per task. Arxiv [49] and Reddit [50], both of which consist of 40 classes. In the unequal division, 10 classes are designated as base classes, with increments of 5 classes. Similarly, the classes are evenly divided into either 10 or 2 classes per task. Each dataset has 3 tasks with 2 classes per task on Cora [51] and Citeseer [51].

Table 5: Statistics of datasets.

Datasets	CoraFull	Arxiv	Reddit	Cora	Citeseer
# nodes	19,793	169,343	227,853	2,708	3,327
# edges	130,622	1,166,243	114,615,892	5,429	4,732
# class	70	40	40	7	6
# task	3 / 7 / 35	7 / 4 / 20	7 / 4 / 20	3	3
# base class	30 / 10 / 2	10 / 10 / 2	10 / 10 / 2	2	2
# novel class	20 / 10 / 2	5 / 10 / 2	5 / 10 / 2	2	2

B.2 Baselines

In this subsection, we introduce the baselines in the main paper. These baselines are as follows:

- **Finetuning** is the lower bound baseline updating the model only with newly emerging graph data.
- **Joint** is the ideal upper bound and inputs contain all previous graph data.

- Elastic Weight Consolidation **(EWC)** [30] quadratically penalizes model weights according to their importance to previous tasks.
- Gradient Episodic Memory **(GEM)** [31] modifies gradients of the current task using gradients computed from stored graph data.
- Memory Aware Synapses **(MAS)** [32] utilizes analysis of parameter prediction as the importance of parameters when adding regularization terms.
- Learning without Forgetting **(LwF)** [6] utilizes information distillation to reduce the discrepancy between old and new models.
- Topology-aware Weight Preserving **(TWP)** [1] preserves the key parameters and topology of the previous task through regularization terms.
- Structural Shift Risk Mitigation **(SSRM)** [11] introduces regularization terms to mitigate catastrophic forgetting from structural drift.
- Experience Replay Graph Neural Network **(ERGNN)** [7] framework incorporates memory replay by storing representative nodes.
- Sparsified Subgraph Memory **(SSM)** [8] stores sampled sparse subgraphs in memory repository to preserve structural information.
- The Condense and Train **(CaT)** [9] framework compresses the graph into a small but informative synthetic replay graph.

B.3 Metrics

We choose two widely used metrics to evaluate the performance of the compared methods, including Average Performance (AP) and Average Forgetting (AF) [31]. When the model learns the latest task, all previous tasks are evaluated and a lower triangular performance matrix $W = \{w_{tt'}\} \in w^{\tau \times \tau}$ is formed, where $w_{tt'}$ is node classification accuracy on task t after learning task t' ($t \leq t'$) and τ is the total number of tasks. Average performance $\mathrm{AP} = 1/\tau \sum_{t=1}^{\tau} w_{\tau,t}$ evaluates the average performance of model on previous task after learning from new task τ. Average Forgetting $\mathrm{AF} = 1/(\tau-1) \sum_{t=1}^{\tau-1} (w_{\tau,t} - w_{t,t})$ represents the average performance degradation on previous tasks after learning from task τ.

B.4 Reproducibility

Our method is trained using a fixed random seed to ensure the consistency and verifiability of results. We are committed to open-sourcing and sharing our code to promote academic collaboration and knowledge sharing, enabling other researchers to reproduce and validate our experimental results.

Table 6: Incremental learning settings.

ERGNN-GSIP	d: 0.5, sampler: CM
SSM-GSIP	subgraph_sampler: random
CaT-GSIP	n_encoders: 500, feat_init: randomChoice, feat_lr: 0.001, hid_dim: 512, hop: 1

B.5 Detailed Settings

Our model is deployed in PyTorch with an NVIDIA RTX 3090 GPU and trained with 200 epochs for every task. We use Adam with weight decay for optimization, and the learning rate is set to 0.005. We use a two-layer GCN with a hidden dimension 256 as the backbone. All results are reported in means and standard deviations over 3 trials. The train-validation-test splitting ratios are 60%, 20%, and 20% for all datasets. The train-validation-test split is achieved through random sampling, resulting in variations in performance across different rounds of random sampling. $\widehat{\delta}$ is set to 0.99 and the search space of δ is $\{0.5, 0.6, 0.7, 0.8, 0.9\}$. Table 6 is the hyper-parameters we adopt from [42] and [9].

C Experimental Results

C.1 Performance Comparison

Table 7: Performance comparison on CoraFull, Arxiv, and Reddit for GCIL setting. Results are averaged among three trials. The best performing results (excluding Joint) are highlighted in **bold**, and the standard deviations are shown in gray.

Method	CoraFull						Arxiv						Reddit					
	Unequally		Equally (10)		Equally (2)		Unequally		Equally (10)		Equally (2)		Unequally		Equally (10)		Equally (2)	
	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑
Finetuning	23.95 ±0.18	-76.59 ±0.80	11.06 ±0.14	-85.77 ±0.03	2.70 ±0.28	-95.48 ±0.04	11.64 ±0.20	-70.41 ±0.76	5.41 ±2.03	-50.00 ±3.93	4.91 ±0.01	-87.61 ±0.41	14.66 ±1.68	-91.80 ±3.51	22.90 ±1.89	-94.42 ±1.74	5.83 ±0.78	-94.23 ±1.32
EWC	24.09 ±0.48	-75.78 ±0.61	11.15 ±0.25	-86.08 ±0.32	5.13 ±1.99	-93.08 ±1.94	11.93 ±0.15	-68.97 ±1.45	14.83 ±0.31	-57.33 ±1.16	4.91 ±0.01	-87.58 ±0.30	13.79 ±0.27	-95.35 ±1.54	22.30 ±0.49	-95.27 ±1.30	9.66 ±2.03	-93.85 ±2.07
GEM	23.95 ±0.37	-76.05 ±0.63	11.23 ±0.29	-85.78 ±0.07	7.97 ±0.67	-90.00 ±0.67	11.61 ±0.38	-60.27 ±5.38	8.27 ±1.53	-44.42 ±1.49	4.92 ±0.08	-86.66 ±0.61	18.51 ±4.23	-89.79 ±6.06	22.58 ±2.58	-93.93 ±1.08	35.11 ±4.31	-65.67 ±4.57
MAS	24.20 ±0.59	-75.97 ±0.97	10.94 ±0.48	-82.37 ±0.71	4.43 ±0.64	-89.22 ±1.43	11.09 ±0.23	-66.76 ±0.59	12.32 ±0.69	-57.99 ±1.82	5.29 ±0.83	-81.64 ±1.34	15.45 ±3.38	-0.50 ±0.76	25.54 ±1.41	0.01 ±0.03	5.98 ±1.82	-14.17 ±1.17
LwF	23.99 ±0.62	-76.14 ±0.55	11.14 ±0.14	-85.67 ±0.63	2.72 ±0.25	-95.08 ±0.05	11.93 ±0.20	-70.66 ±0.74	14.69 ±1.48	-58.93 ±1.25	4.91 ±0.01	-88.14 ±0.09	16.13 ±3.17	-90.31 ±3.99	24.39 ±0.69	-93.29 ±0.60	7.59 ±0.51	-88.98 ±0.74
TWP	23.86 ±0.17	-75.74 ±0.70	11.01 ±0.21	-85.43 ±0.18	3.56 ±1.02	-94.66 ±0.84	11.93 ±0.08	-69.26 ±0.24	14.41 ±0.54	-56.56 ±3.36	4.90 ±0.01	-87.75 ±0.32	13.95 ±0.90	-96.17 ±1.11	21.22 ±1.58	-96.41 ±0.11	9.34 ±3.46	-94.24 ±3.78
SSRM	63.62 ±0.56	-16.24 ±1.22	31.39 ±2.25	-60.61 ±3.11	3.22 ±0.29	-89.29 ±0.48	31.51 ±0.44	-45.12 ±0.55	26.61 ±0.11	-46.22 ±1.09	26.16 ±0.85	-61.24 ±0.84	78.40 ±7.42	-20.92 ±8.64	76.78 ±4.35	-23.16 ±5.62	83.96 ±3.04	-15.41 ±3.18
ERGNN	60.91 ±1.12	-19.47 ±1.43	24.39 ±0.39	-69.31 ±0.73	3.01 ±0.20	-94.34 ±0.51	31.18 ±0.83	-45.45 ±1.75	24.47 ±2.67	-49.11 ±3.03	24.70 ±1.00	-62.26 ±0.75	76.60 ±5.77	-23.22 ±6.47	75.22 ±11.67	-25.26 ±14.13	83.16 ±1.92	-16.21 ±2.06
ERGNN-GSIP (Ours)	67.22 ±0.44	-10.91 ±0.62	71.15 ±0.98	-11.37 ±0.74	44.79 ±1.77	-44.60 ±1.67	34.09 ±0.77	-32.59 ±1.37	33.88 ±0.87	-27.97 ±1.63	40.21 ±0.92	-28.96 ±0.94	90.82 ±0.70	-6.05 ±0.68	89.59 ±2.04	-2.03 ±4.62	93.03 ±3.87	-5.50 ±4.11
SSM	50.51 ±1.03	-10.56 ±0.94	62.90 ±0.50	-6.02 ±0.37	79.02 ±0.50	-4.24 ±0.23	63.48 ±0.78	-12.41 ±0.01	60.57 ±0.80	-10.09 ±1.19	63.91 ±0.35	-12.48 ±0.58	90.10 ±1.56	-5.83 ±1.14	86.91 ±1.77	-3.24 ±0.56	96.24 ±0.24	-1.64 ±0.31
SSM-GSIP (Ours)	55.32 ±0.75	-2.50 ±1.13	63.86 **0.08** ±0.85	±0.76	79.31 ±0.50	0.70 ±0.25	63.36 ±1.13	-7.27 ±0.82	61.34 ±0.77	-6.34 ±0.70	64.16 ±0.37	-8.87 ±0.58	90.74 ±0.44	-3.97 ±0.40	87.41 ±1.60	0.13 ±0.91	96.25 ±0.37	-0.65 ±0.64
CaT	70.55 ±0.67	-5.26 ±0.40	76.35 ±0.41	-5.44 ±0.58	80.64 ±0.30	-4.31 ±0.43	**71.66** ±0.73	-8.33 ±0.26	70.16 ±0.14	-7.25 ±0.79	66.21 ±0.12	-12.73 ±0.09	**96.39** ±0.17	-0.77 ±0.40	93.97 ±0.30	-1.31 ±0.08	**97.64** ±0.09	-0.49 ±0.04
CaT-GSIP (Ours)	**71.06** ±0.54	**-0.28** ±0.07	**78.29** ±0.11	-1.25 ±0.25	**81.10** ±0.18	**2.68** ±0.16	71.52 ±0.64	**-4.76** ±0.08	**70.57** ±0.14	**-3.97** ±0.51	**68.80** ±0.24	**3.49** ±0.39	96.15 ±0.30	**-0.23** ±0.31	**94.23** ±0.28	**0.21** ±0.64	97.55 ±0.05	**1.04** ±0.30
Joint	85.3 ±0.1	-	85.3 ±0.1	-	85.3 ±0.1	-	63.5 ±0.3	-	63.5 ±0.3	-	63.5 ±0.3	-	98.2 ±0.0	-	98.2 ±0.0	-	98.2 ±0.0	-

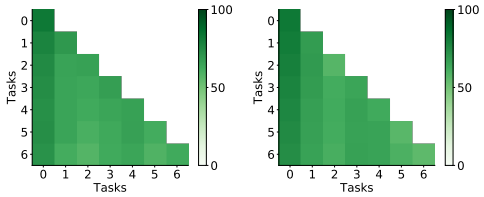
The effect of GCIL on five datasets with standard deviation is presented in Table 7 and 8. GSIP improves existing information preservation methods under different dataset splitting scenarios. The performance matrices of SSM and CaT on CoraFull before and after incorporating GSIP are shown in Figure 8 and Figure 9. Remembering information from previous classes becomes challenging without the implementation of the proposed GSIP. The performance matrix illustrates a deceleration in the forgetting process after adopting the GSIP scheme. This is evident by the minimal changes and deepening in color of each column, indicating mitigation of the catastrophic forgetting problem. This is achieved through the preservation of information from previous model.

Table 8: Performance comparison on Cora and Citeseer for GCIL setting. Results are averaged among three trials. The best performing results (excluding Joint) are highlighted in **bold**, and the standard deviations are shown in gray.

Method	Cora		Citeseer	
	Equally (2)		Equally (2)	
	AP↑	AF↑	AP↑	AF↑
Finetuning	32.58 ±0.00	-96.83 ±0.39	31.46 ±0.27	-77.86 ±0.67
EWC	32.58 ±0.00	-97.16 ±0.55	31.26 ±0.15	-78.22 ±0.61
GEM	32.70 ±0.19	-97.12 ±0.16	31.39 ±0.09	-77.70 ±1.06
MAS	31.84 ±0.10	-97.17 ±0.22	31.25 ±0.55	-76.67 ±1.18
LwF	32.58 ±0.00	-97.57 ±0.27	31.44 ±0.00	-78.29 ±0.36
TWP	32.58 ±0.00	-97.32 ±0.17	31.22 ±0.23	-78.14 ±0.22
SSRM	35.48 ±0.49	-70.01 ±1.12	51.91 ±4.59	-67.66 ±6.67
ERGNN	65.48 ±0.76	-46.09 ±1.15	47.65 ±0.57	-51.12 ±1.47
ERGNN-GSIP (Ours)	71.29 ±0.91	-36.95 ±1.22	61.29 ±0.96	-29.38 ±2.28
SSM	67.64 ±3.14	-19.78 ±8.10	60.99 ±1.43	-13.60 ±3.61
SSM-GSIP (Ours)	69.92 ±1.46	-11.82 ±6.74	61.86 ±2.23	**-8.39** ±3.82
CaT	88.22 ±0.49	-4.40 ±1.04	75.08 ±0.35	-10.93 ±1.07
CaT-GSIP (Ours)	**89.60** ±0.62	**1.84** ±3.89	**77.02** ±0.70	-9.95 ±1.15
Joint	93.09 ±0.85	-	78.27 ±0.10	-

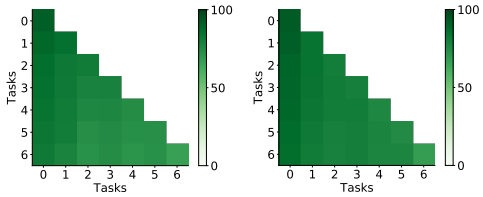
C.2 Ablation Study

Efficiency. As shown in Table 9, Table 10, and Table 11, the consequences of ablation experiments with standard deviation for ERGNN, SSM, and CaT as baselines (B) are presented. We investigate the effectiveness of low-frequency local modules (LL), low-frequency global modules (LG), and high-frequency modules (H). The above components are added one by one to baselines for performance comparison. B is the baseline. B+LL indicates that it only uses low-frequency local preservation. B+LL+LG represents that it uses low-frequency local-global preservation. B+LL+LG+H is the full model, which uses low-frequency and high-frequency preservation. All the components are effective and can bring great benefits to the model in performance improvement.



(a) CoraFull, SSM (b) CoraFull, SSM-GSIP

Figure 8: Performance matrices in SSM.



(a) CoraFull, CaT (b) CoraFull, CaT-GSIP

Figure 9: Performance matrices in CaT.

Table 9: Ablation comparisons of graph spatial information preserving strategy for ERGNN.

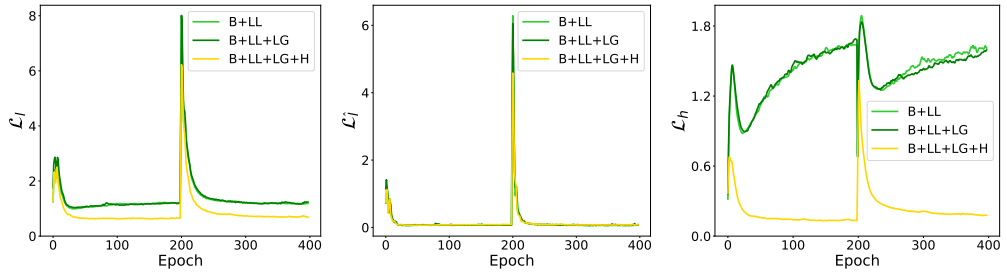
Method	CoraFull						Arxiv						Reddit					
	Unequally		Equally (10)		Equally (2)		Unequally		Equally (10)		Equally (2)		Unequally		Equally (10)		Equally (2)	
	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑
B	60.91	-19.47	24.39	-69.31	3.01	-94.34	31.18	-45.45	24.47	-49.11	24.70	-62.26	76.60	-23.22	75.22	-25.26	83.16	-16.21
	±1.12	±1.43	±0.39	±0.73	±0.20	±0.51	±0.83	±1.75	±2.67	±3.03	±1.00	±0.75	±5.77	±6.47	±11.67	±14.13	±1.92	±2.06
B+LL	65.79	-13.20	69.02	-14.15	41.37	-47.39	33.27	-34.60	27.10	-40.95	38.09	-35.04	84.63	-12.09	84.26	-12.37	87.52	-10.97
	±0.87	±1.26	±0.11	±0.05	±1.90	±2.19	±1.48	±1.29	±3.47	±4.17	±1.02	±1.67	±4.58	±6.36	±1.12	±1.96	±2.86	±2.98
B+LL+LG	66.22	-12.78	69.77	-13.13	41.84	-46.73	34.00	-33.61	32.80	-34.40	39.89	**-28.85**	89.21	-6.97	87.17	-9.45	91.34	-7.12
	±0.56	±1.12	±0.37	±0.29	±1.25	±1.16	±0.37	±0.58	±2.71	±3.17	±0.16	±0.49	±2.43	±0.94	±2.70	±3.86	±3.58	±3.91
B+LL+LG+H	**67.22**	**-10.91**	**71.15**	**-11.37**	**44.79**	**-44.60**	**34.09**	**-32.59**	**33.88**	**-27.97**	**40.21**	-28.96	**90.82**	**-6.05**	**89.59**	**-2.03**	**93.03**	**-5.50**
	±0.44	±0.62	±0.98	±0.74	±1.77	±1.67	±0.77	±1.37	±0.87	±1.63	±0.92	±0.94	±0.70	±0.68	±2.04	±4.62	±3.87	±4.11

Table 10: Ablation comparisons of graph spatial information preserving strategy for SSM.

Method	CoraFull						Arxiv						Reddit					
	Unequally		Equally (10)		Equally (2)		Unequally		Equally (10)		Equally (2)		Unequally		Equally (10)		Equally (2)	
	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑
B	50.51	-10.56	62.90	-6.02	79.02	-4.24	**63.48**	-12.41	60.57	-10.09	63.91	-12.48	90.10	-5.83	86.91	-3.24	96.24	-1.64
	±1.03	±0.94	±0.50	±0.37	±0.50	±0.23	±0.78	±0.01	±0.80	±1.19	±0.35	±0.58	±1.56	±1.14	±1.77	±0.56	±0.24	±0.31
B+LL	54.75	-5.45	63.68	-1.60	79.30	-0.49	62.99	-9.70	60.13	-7.04	63.93	-9.89	90.55	-4.06	87.34	0.06	95.98	-0.84
	±0.76	±1.06	±0.58	±0.60	±0.65	±0.36	±0.91	±0.18	±0.48	±0.98	±0.34	±0.58	±0.48	±0.46	±1.56	±0.93	±0.19	±0.30
B+LL+LG	55.16	-2.62	63.67	-0.02	**79.31**	0.61	63.40	-7.60	61.17	-6.57	64.01	-9.22	90.70	-4.05	87.38	**0.14**	96.16	-0.75
	±0.56	±1.21	±0.65	±0.59	±0.65	±0.60	±1.16	±0.96	±1.02	±1.32	±0.60	±1.07	±0.35	±0.33	±1.69	±0.93	±0.35	±0.57
B+LL+LG+H	**55.32**	**-2.50**	**63.86**	**0.08**	**79.31**	**0.70**	63.36	**-7.27**	**61.34**	**-6.34**	**64.16**	**-8.87**	**90.74**	**-3.97**	**87.41**	0.13	**96.25**	**-0.65**
	±0.75	±1.13	±0.85	±0.76	±0.50	±0.25	±1.13	±0.82	±0.77	±0.70	±0.37	±0.58	±0.44	±0.40	±1.60	±0.91	±0.37	±0.64

Table 11: Ablation comparisons of graph spatial information preserving strategy for CaT.

Method	CoraFull						Arxiv						Reddit					
	Unequally		Equally (10)		Equally (2)		Unequally		Equally (10)		Equally (2)		Unequally		Equally (10)		Equally (2)	
	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑	AP↑	AF↑
B	70.55	-5.26	76.35	-5.44	80.64	-4.31	**71.66**	-8.33	70.16	-7.25	66.21	-12.73	**96.39**	-0.77	93.97	-1.31	**97.64**	-0.49
	±0.67	±0.40	±0.41	±0.58	±0.30	±0.43	±0.73	±0.26	±0.14	±0.79	±0.12	±0.09	±0.17	±0.40	±0.30	±0.08	±0.09	±0.04
B+LL	70.74	-1.32	**78.31**	-2.03	80.91	-0.89	71.42	-6.02	70.42	-4.80	67.23	-2.85	96.12	-0.29	93.23	-0.12	97.26	0.35
	±0.57	±0.69	±0.16	±0.13	±0.14	±0.52	±0.77	±0.20	±0.37	±0.75	±0.24	±0.44	±0.28	±0.31	±0.39	±0.78	±0.28	±0.36
B+LL+LG	70.89	-0.39	77.91	-1.41	**81.12**	-0.10	71.41	-4.85	**70.59**	-3.99	68.61	3.10	96.14	-0.26	93.91	0.04	97.27	0.48
	±0.42	±0.07	±0.27	±0.33	±0.19	±0.17	±0.92	±0.04	±0.15	±0.59	±0.10	±0.34	±0.24	±0.24	±0.47	±1.09	±0.33	±0.36
B+LL+LG+H	**71.06**	**-0.28**	78.29	**-1.25**	81.10	**2.68**	71.52	**-4.76**	70.57	**-3.97**	**68.80**	**3.49**	96.15	**-0.23**	**94.23**	**0.21**	97.55	**1.04**
	±0.54	±0.07	±0.11	±0.25	±0.18	±0.16	±0.64	±0.08	±0.14	±0.51	±0.24	±0.39	±0.30	±0.31	±0.28	±0.64	±0.05	±0.30



(a) The change of loss $\mathcal{L}_l$ (b) The change of loss $\mathcal{L}_{\widehat{l}}$ (c) The change of loss $\mathcal{L}_h$

Figure 10: Graph information preservation training losses of different variants with epochs on CoraFull dataset.

Information Pattern. Training losses of ERGNN-GSIP during incremental processes in the dataset inequality partition setting are illustrated. Specifically, we focus on the last two tasks and examine how the loss of different variants changes with increasing epochs. We make three observations: (1) Low-frequency local graph information is well preserved. Figure 10(a) measures the degree of feature similarity. It can be observed that losses of the first two variants start to rise around the 50th epoch. However, the GSIP loss remains much lower than the first two variants and converges quickly. (2) The learning of low-frequency information is ensured after global graph embedding similarity correction. It can be seen that the loss of GSIP is slightly lower than the other two variants in Figure 10(b). (3) GSIP can preserve high-frequency information and reduce the forgetting of topology. In Figure 10(c), we can see that the losses of the first two variants decrease and then increase at each increment. This demonstrates that node difference information is almost completely discarded in these variants.

C.3 Hyper-Parameter Analysis of Memory Size

We analyze the impact of the number of storage nodes for each task $\#\mathcal{M}$ on performance of each task that has 2 classes. As depicted in Figure 11-Figure 15, it can be observed that the proposed method consistently outperforms the original method in terms of the AP (the higher, the better) and -AF metric (the lower, the better), regardless of the value of $\#\mathcal{M}$. CaT cannot be trained with 400 nodes on CoraFull due to Cuda memory constraints. Interestingly, despite having less memory, the proposed method demonstrates superior performance across three datasets. Furthermore, the performance remains relatively consistent when storing 200 or 400 nodes, indicating that our method does not incur higher storage costs. Notably, storing only 10 nodes per task yields performance comparable to storing 400 nodes on Reddit, highlighting the superiority of our approach.

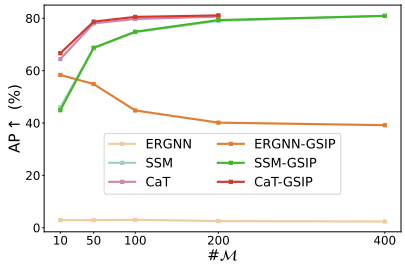


Figure 11: The change of AP affected by $\#\mathcal{M}$ on CoraFull dataset.

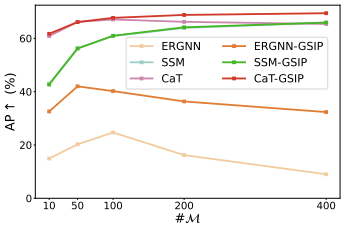


Figure 12: The change of AP affected by $\#\mathcal{M}$ on Arxiv dataset.

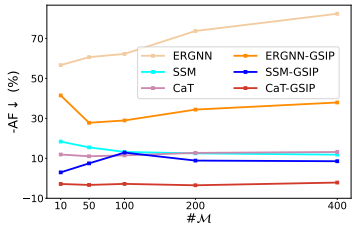


Figure 13: The change of AF affected by $\#\mathcal{M}$ on Arxiv dataset.

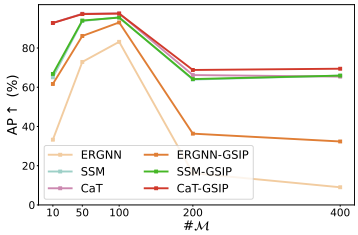


Figure 14: The change of AP affected by $\#\mathcal{M}$ on Reddit dataset.

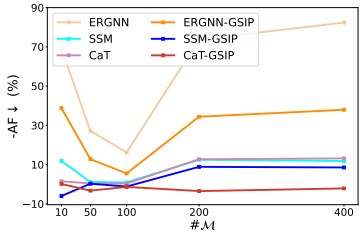
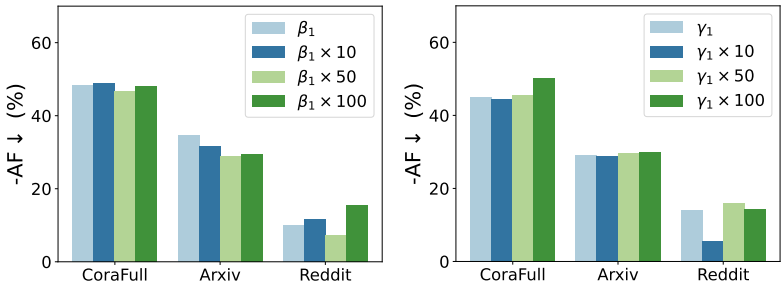


Figure 15: The change of AF affected by $\#\mathcal{M}$ on Reddit dataset.

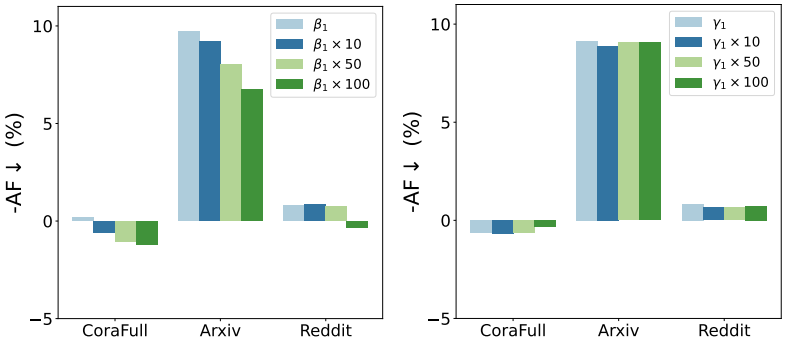
C.4 Hyper-Parameter Analysis of Loss Weights

As shown in Figure 16, Figure 17, and Figure 18, the analysis for two hyper-parameters, loss weights β and γ on ERGNN, SSM, and CaT in terms of AF is conducted, and the results are in an experimental setting of increment 2. For ERGNN, β_1 is set to $[2e-5, 1e-1, 1e-5]$ and γ_1 is set to $[2e-2, 1e-8, 1e-4]$ for different datasets. For SSM, β_1 is set to [1, 1, 10] and γ_1 is set to $[1e-7, 1e-8, 1e-3]$ for three datasets. For CaT, β_1 is set to $[1e-1, 2, 2e-4]$ and γ_1 is set to $[1e-1, 1e-6, 2e-1]$ for three datasets. We notice that the model performance remains unaffected by changes in β when γ is set to 0 and by changes in γ when β reaches its optimal value.



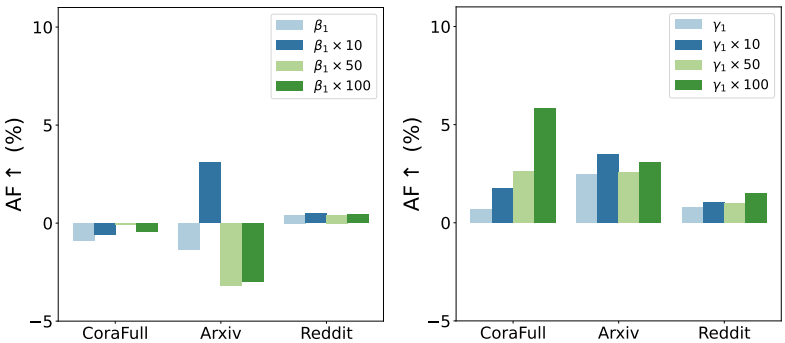
(a) Varying β with $\gamma = 0$ (b) Varying γ with $\beta = \beta^*$

Figure 16: The analysis of β and γ in ERGNN-GSIP on CoraFull, Arxiv, and Reddit datasets.



(a) Varying β with $\gamma = 0$ (b) Varying γ with $\beta = \beta^*$

Figure 17: The analysis of β and γ in SSM-GSIP on CoraFull, Arxiv, and Reddit datasets.

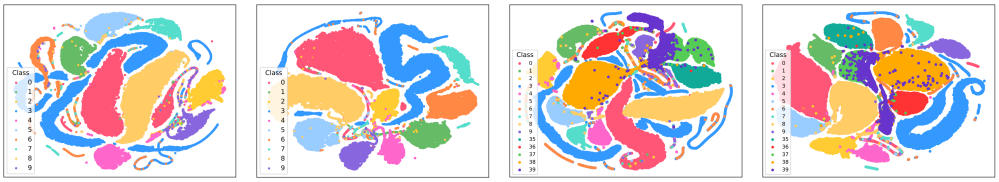


(a) Varying β with $\gamma = 0$ (b) Varying γ with $\beta = \beta^*$

Figure 18: The analysis of β and γ in CaT-GSIP on CoraFull, Arxiv, and Reddit datasets.

C.5 Visualization

To qualitatively demonstrate the effectiveness of representations, we utilize t-SNE [52] to visualize the node embeddings of ERGNN and ERGNN-GSIP. After learning the last task, Figure 19(a) and Figure 19(b) display the results of the learned node embeddings in Task 1 on Reddit, while Figure 19(c) and Figure 19(d) illustrate the results of the last task. GSIP exhibits superior representation ability, effectively considering representations and accurately classifying old and new classes.



(a) ERGNN, Task 1 (b) ERGNN-GSIP, Task 1 (c) ERGNN, Task 7 (d) ERGNN-GSIP, Task 7

Figure 19: The visualization of node embeddings from ERGNN and ERGNN-GSIP in Task 1 and Task 7 of Reddit dataset.

C.6 Time Complexity Analysis

The time complexity of framework mainly comes from three aspects: (1) node classification $O(|V|\mathbb{k}^G\mathbb{k}+|\mathcal{E}|\mathbb{k}+|V|\mathbb{k})$, (2) replay scheme $O(|\mathcal{M}|\mathbb{k}^G\mathbb{k}+|\mathcal{E}^{\mathcal{M}}|\mathbb{k}+|\mathcal{M}|\mathbb{k})$, and (3) graph spatial information preservation $O(|\mathcal{E}^{\mathcal{M}}|\mathbb{k}+\mathbb{k})$. The total time complexity is $O(|V|\mathbb{k}^G\mathbb{k}+|\mathcal{M}|\mathbb{k}^G\mathbb{k}+|V|\mathbb{k}+|\mathcal{M}|\mathbb{k}+|\mathcal{E}|\mathbb{k}+|\mathcal{E}^{\mathcal{M}}|\mathbb{k}+\mathbb{k})$, where V and $\mathcal{M}$ are subgraph vertices and memory, $\mathcal{E}$ and $\mathcal{E}^{\mathcal{M}}$ are edge sets of subgraph vertices and memory, then $\mathbb{k}^G$ and $\mathbb{k}$ denote the dimensions of inputs and hidden spaces. The time complexity increases linearly compared with baselines. The running time on CoraFull is presented in Table 12. For ERGNN, $|V| \gg |\mathcal{M}|$ leads to phenomenon that the higher the number of tasks, the shorter training time. On the contrary, more tasks result in longer training time due to the properties of compressed graphs in SSM and CaT.

Table 12: Running time (s) of each epoch under three dataset partitioning cases on CoraFull dataset.

Method	Unequally	Equally (10)	Equally (2)
ERGNN	0.8073	0.3694	0.1660
+GSIP	1.1913	0.4672	0.3792
SSM	0.0110	0.0279	0.0410
+GSIP	0.0113	0.0312	0.0494
CaT	0.0106	0.0189	0.0488
+GSIP	0.0147	0.0235	0.0524

D Discussion

D.1 Limitation

The primary limitation of GSIP lies in its focus on replayed designs and the lack of connection to other methods. Also, the paper does not examine other GCIL settings on the graph, such as the lack of a clear task boundary, which would be an interesting direction to explore in the future.

D.2 Broader Impact

Considering the broader implications of our work, we posit that the proposed framework for information preservation of the old graph model will support the development of systems in an open environment based on machine learning. However, accessing old data may raise privacy concerns, and the dynamic updating of systems could inadvertently marginalize under-represented groups, potentially have a negative impact on outcomes and interfere with fair decision-making.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: See abstract and introduction.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: The limitations are discussed in Appendix D.1.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

Justification: See Section 4 for assumptions and the detailed proof is provided in Appendix A.1.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: All the information needed to reproduce the main experimental results is added in Appendix B.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: They are enclosed in the supplemental material.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: The experiment settings are provided in Appendix B.5.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: See that Table 7-Table 11 report the standard deviation of experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The detailed description is provided in Appendix B.5 and C.6.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms with the NeurIPS Code of Ethics.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: Broader impacts are discussed in Appendix D.2.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: The paper poses no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [NA]

    Justification: The paper does not use existing assets.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.