Towards Robust Graph Incremental Learning on Evolving Graphs

Junwei Su [1] **Difan Zou** [1] **Zijun Zhang** [2] **Chuan Wu** [1]

Abstract

Incremental learning is a machine learning approach that involves training a model on a sequence of tasks, rather than all tasks at once. This ability to learn incrementally from a stream of tasks is crucial for many real-world applications. However, incremental learning is a challenging problem on graph-structured data, as many graph-related problems involve prediction tasks for each individual node, known as Node-wise Graph Incremental Learning (NGIL). This introduces non-independent and non-identically distributed characteristics in the sample data generation process, making it difficult to maintain the performance of the model as new tasks are added. In this paper, we focus on the inductive NGIL problem, which accounts for the evolution of graph structure (structural shift) induced by emerging tasks. We provide a formal formulation and analysis of the problem, and propose a novel regularization-based technique called Structural-Shift-Risk-Mitigation (SSRM) to mitigate the impact of the structural shift on catastrophic forgetting of the inductive NGIL problem. We show that the structural shift can lead to a shift in the input distribution for the existing tasks, and further lead to an increased risk of catastrophic forgetting. Through comprehensive empirical studies with several benchmark datasets, we demonstrate that our proposed method, Structural-Shift-Risk-Mitigation (SSRM), is flexible and easy to adapt to improve the performance of state-of-the-art GNN incremental learning frameworks in the inductive setting.

[1]Department of Computer Science, University of Hong Kong [2]Department of Computer Science, University of Wu Han. Correspondence to: Junwei Su <junweisu@connect.hku.hk>.

Proceedings of the 40th International Conference on Machine Learning, Honolulu, Hawaii, USA. PMLR 202, 2023. Copyright 2023 by the author(s).

Implementation available at: https://github.com/littleTown93/NGIL_Evolve

1. Introduction

Humans are capable of acquiring new information continuously while retaining previously obtained knowledge. This seemingly natural capability, however, is difficult but important for deep neural networks (DNNs) to acquire (Wu et al., 2021). Incremental learning, also known as continual learning or life-long learning, studies machine learning approaches that allow a model to continuously acquire new knowledge while retaining previously obtained knowledge. This is important because it allows the model to adapt to new information without forgetting past knowledge. In the general formulation of incremental learning, a stream of tasks arrive sequentially and the model goes through rounds of training sessions to accumulate knowledge for a particular objective (such as classification). The goal is to find a learning algorithm that can incrementally update the model's parameters based on the new task without suffering from *catastrophic forgetting* (Delange et al., 2021), which refers to the inability to retain previously learned information when learning new tasks. Incremental learning is crucial for the practicality of machine learning systems as it allows the model to adapt to new information without the need for frequent retraining, which can be costly.

While incremental learning has been extensively studied for Euclidean data (such as images and text) (Biesialska et al., 2020; Pfülb & Gepperth, 2019), there has been relatively little research on incremental learning for graph-structured data (Zhang et al., 2022). Graphs are often generated continuously in real-life scenarios, making them an ideal application for incremental learning. For example, in a citation network, new papers and their associated citations may emerge, and a document classifier needs to continuously adapt its parameters to distinguish the documents of newly emerged research fields (Zhou & Cao, 2021). In social networks, the distributions of users' friendships and activities depend on when and where the networks are collected (Gama et al., 2014). In financial networks, the payment flows between transactions and the appearance of illicit transactions have strong correlations with external contextual factors such as time and market (Zliobaite, 2010). Therefore, it is important to develop Graph Incremental Learning (GIL) methods that can handle new tasks over newly emerged graph data while maintaining model performance, particularly for highly dynamic systems such as citation networks, online

social networks, and financial networks. Graph neural networks (GNNs) are popular and effective tools for modelling graph and relational data structures (Wu et al., 2020), while the ability to incrementally learn from data streams is needed for building practical GNN-driven systems.

One of the major challenges in incremental learning on graph-structured data lies in the prediction of individual nodes, known as Node-wise GIL (NGIL). This type of problem introduces non-independent and non-identically distributed characteristics in the sample data generation process, making it difficult to maintain the performance of the model as new tasks are added. Existing research on NGIL has primarily focused on a *transductive* setting, where the graph structures among tasks are assumed to be independent (Zhou & Cao, 2021; Liu et al., 2021). However, in many real-life scenarios, such as the examples above, it is inevitable that the emerging graph associated with the new task would expand on the existing graph, thereby altering the structural information of the existing vertices, i.e., structural shift. As the structural information of the graph is a crucial input for GNNs, an evolving graph structure can greatly affect the model's prediction and generalization abilities. This is referred to as the *inductive* NGIL problem. Fig. 1 provides a graphical illustration of the difference between transductive and inductive NGIL. To advance the practicality of the NGIL framework, it is important to study this inductive setting. However, currently, there is a lack of a clear problem formulation and theoretical understanding of the problem, making it challenging to develop effective solutions for inductive NGIL.

In this paper, we delve into the complex inductive NGIL problem. We carefully formulate and rigorously analyze this problem, and propose a novel regularization-based technique to effectively mitigate the impact of the structural shift on catastrophic forgetting (as measured by retention of model performance in previous tasks) of the inductive NGIL problem. Our contributions are summarized as follows.

- We present a mathematical formulation of the inductive NGIL problem that quantifies the effect of changes in the graph structure on the model's ability to generalize to previously seen tasks/vertices. This formulation provides a clear and detailed understanding of the problem and is essential for designing and evaluating effective solutions.

- We conduct a formal analysis of the impact of the structural shift on catastrophic forgetting of the inductive NGIL problem. We show that structural shift can lead to a shift in the input distribution for existing tasks (Proposition 4.1), and we derive a bound on catastrophic forgetting that indicates that the risk of forgetting is positively related to the extent of structural shift present in the inductive setting (Theorem 4.3). This sheds insight into developing algorithms for inductive NGIL.

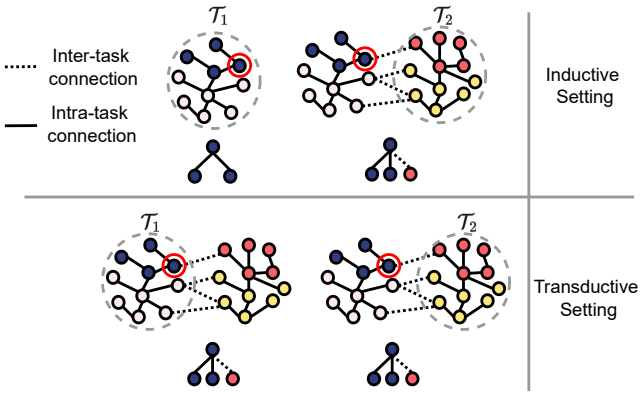


Figure 1. An illustration of the difference between transductive and inductive NGIL. $\mathcal{T}_1$ and $\mathcal{T}_2$ are two consecutive tasks. In the transductive setting, the 1-hop ego graph of the vertex with a red circle would remain the same, while in the inductive setting, the graph may change as new tasks are introduced and the overall graph structure evolves.

- Based on the analysis, we propose a novel regularization method utilizing the divergence minimization principle, which encourages the model to converge to a latent space where the difference of the distribution induced by the structural shift is minimised. We show that such an approach can reduce the risk of catastrophic forgetting (Thereom 5.1), and we term our proposed method as Structural-Shift-Risk-Mitigation (SSRM). SSRM is easy to implement and can be easily adapted to existing GNN incremental learning frameworks to boost their performance in the inductive setting.

- We validate the predicted effect of structural shift on catastrophic forgetting and the effectiveness of our proposed SSRM method through comprehensive empirical studies. Our results show that SSRM can consistently improve the performance of state-of-the-art incremental learning frameworks on the inductive NGIL problem.

2. Related Work

2.1. Incremental Learning

Incremental learning, also known as continual or lifelong learning, has gained increasing attention in recent years and has been extensively explored on Euclidean data. We refer readers to the surveys (Delange et al., 2021; Parisi et al., 2019; Biesialska et al., 2020) for a more comprehensive review of these works. The primary challenge of incremental learning is to address the catastrophic forgetting problem, which refers to the drastic degradation in a model's performance on previous tasks after being trained on new tasks.

Existing approaches for addressing this problem can be broadly categorized into three types: regularization-based methods, experience-replay-based methods, and parameter-isolation-based methods. Regularization-based methods aim to maintain the model's performance on previous tasks by penalizing large changes in the model parameters (Jung et al., 2016; Li & Hoiem, 2017; Kirkpatrick et al., 2017; Farajtabar et al., 2020; Saha et al., 2021). Parameter-isolation-based methods prevent drastic changes to the parameters that are important for previous tasks by continually introducing new parameters for new tasks (Rusu et al., 2016; Yoon et al., 2017; 2019; Wortsman et al., 2020; Wu et al., 2019). Experience-replay-based methods select a set of representative data from previous tasks, which are used to retrain the model with the new task data to prevent forgetting (Lopez-Paz & Ranzato, 2017; Shin et al., 2017; Aljundi et al., 2019; Caccia et al., 2020; Chrysakis & Moens, 2020; Knoblauch et al., 2020).

Our proposed method, SSRM, is a novel regularization-based technique that addresses the unique challenge of the structural shift in the inductive NGIL. Unlike existing regularization methods, which focus on minimizing the effect of updates from new tasks, SSRM aims to minimize the impact of the structural shift on the generalization of the model on the existing tasks by finding a latent space where the impact of the structural shift is minimized. This is achieved by minimizing the distance between the representations of vertices in the previous and current graph structures through the inclusion of a structural shift mitigation term in the learning objective. It is important to note that SSRM should not be used as a standalone method to overcome catastrophic forgetting but should be used as an additional regularizer to improve performance when there is a structural shift.

2.2. Graph Incremental Learning

Recently, there has been a surge of interest in GIL due to its practical significance in various applications (Wang et al., 2022; Xu et al., 2020; Daruna et al., 2021; Kou et al., 2020; Ahrabian et al., 2021; Cai et al., 2022; Wang et al., 2020a; Liu et al., 2021; Zhang et al., 2021; Zhou & Cao, 2021; Carta et al., 2021; Zhang et al., 2022; Kim et al., 2022; Tan et al., 2022). However, most existing works in NGIL focus on a transductive setting, where the entire graph structure is known before performing the task or where the interconnection between different tasks is ignored. In contrast, the inductive NGIL problem, where the graph structure evolves as new tasks are introduced, is less explored and lacks a clear problem formulation and theoretical understanding. There is currently a gap in understanding the inductive NGIL problem, which our work aims to address. In this paper, we highlight the importance of addressing the structural shift and propose a novel regularization-based technique, SSRM, to mitigate the impact of the structural shift on the inductive NGIL problem. Our work lays down a solid foundation for future research in this area.

Finally, it is important to note that there is another area of research known as dynamic graph learning (Galke et al., 2021; Wang et al., 2020b; Han et al., 2020; Yu et al., 2018; Nguyen et al., 2018; Ma et al., 2020; Feng et al., 2020; Bielak et al., 2022), which focuses on enabling GNNs to capture the changing graph structures. The goal of dynamic graph learning is to capture the temporal dynamics of the graph into the representation vectors, while having access to all previous information. In contrast, GIL addresses the problem of catastrophic forgetting, in which the model's performance on previous tasks degrades after learning new tasks. For evaluation, a dynamic graph learning algorithm is only tested on the latest data, while GIL models are also evaluated on past data. Therefore, dynamic graph learning and GIL are two independent research directions with different focuses and should be considered separately.

3. Preliminary and Problem Formulation

In this section, we present the preliminary and formulation of the inductive NGIL problem. We use bold letters to denote random variables, while the corresponding realizations are represented with thin letters.

We assume the existence of a stream of training tasks $\mathcal{T}_1, \mathcal{T}_2, ..., \mathcal{T}_m$, characterized by observed vertex batches $V_1, V_2, ..., V_m$ that are drawn from an unknown undirected graph $\mathcal{G}$. Each vertex v is associated with a node feature x_v and a target label y_v. The observed graph structure at training task $\mathcal{T}_i$ is induced by the accumulative vertices and given by $\mathcal{G}_{\mathcal{T}_i} = \mathcal{G}[\bigcup_{j=1}^{i} V_j]$. In this setting, the graph structure is evolving as the learning progresses through different training tasks. Fig. 2 provides a graphical illustration.

To accommodate the nature of node-level learning tasks, in which the information used for inference is aggregated within the k-hop neighborhood of a node, we adopt a local view in the learning problem formulation. We define $\mathrm{N}_k(v)$ as the k-hop neighborhood of vertex v, and the nodes in $\mathrm{N}_k(v)$ form an ego-graph G_v, which consists of a (local) node feature matrix $X_v = \{x_u | u \in \mathrm{N}_k(v)\}$ and a (local) adjacency matrix $A_v = \{a_{uw} | u, w \in \mathrm{N}_k(v)\}$. We use $\mathbf{G_v}$ as a random variable of the ego-graph for the target vertex v, whose realization is $G_v = (A_v, X_v)$. Let $G_V = \{G_v | v \in V\}$ denote the set of ego graphs associated with vertex set V. Under this formulation, the ego-graph G_v can be viewed as the Markov blanket (containing all necessary information for the prediction problem) for the root vertex v. Therefore, we can see the prediction problem associated with data $\{(G_v, y_v)\}_{v \in V_i}$ from training session $\mathcal{T}_i$ as drawn from an empirical joint distribution $P(\mathbf{y_v}, \mathbf{G}_v | V_i)$.

Let $\mathcal{H}$ denote the hypothesis space and $f \in \mathcal{H}$ be a classifier

with $\hat{y}_v = f(G_v)$ and $\mathcal{L}(.,.) \mapsto \mathbb{R}$ be a given loss function. We use $R^{\mathcal{L}}_{P(\mathbf{y}_v, \mathbf{G}_v|V)}(f)$ to denote the generalization risk of the classifier f with respect to $\mathcal{L}$ and $P(\mathbf{y}_v, \mathbf{G}_v|V)$, and it is defined as follows:

$$R^{\mathcal{L}}_{P(\mathbf{y}_v, \mathbf{G}_v|V)}(f) = \mathbb{E}_{P(\mathbf{y}_v, \mathbf{G}_v|V)}[\mathcal{L}(f(G_v), y_v)]. \quad (1)$$

Catastrophic Forgetting Risk. With the formulation above, the catastrophic forgetting risk (**CFR**) of a classifier f after being trained on $\mathcal{T}_m$ can be characterized by the retention of performance on previous vertices, given by:

$$\mathbf{CFR}(f) := R^{\mathcal{L}}_{P(\mathbf{y}_v, \mathbf{G}_v|V_1, ..., V_{m-1})}(f) \quad (2)$$

which translates to the retention of performance of the classifier f from $\mathcal{T}_m$ on the previous tasks $\mathcal{T}_1, ..., \mathcal{T}_{m-1}$.

4. Structural Shift and Catastrophic Forgetting

In this section, we present our main results on the connection between the structural shift of the evolving graph structure and the catastrophic forgetting risk of the NGIL problem. We start by showing how the evolving graph structure can lead to a distributional shift in the underlying learning problem. We then derive a bound for catastrophic forgetting risk with respect to structural shift.

4.1. Structural Shift

To illustrate the phenomenon of structural shift more concretely in NGIL, we consider a graph generation process with two communities (i.e., types of vertices): community 1 and community 2. The feature vectors x_u of vertices u in community 1 are drawn from a distribution N_1, and those of vertices in community 2 are drawn from another distribution N_2, where $\mathbb{E}_{N_1}[x_u] \neq \mathbb{E}_{N_2}[x_u]$. The connectivity probability for vertices of the same community is denoted as p_{in}, and the connectivity probability for vertices of different communities is denoted as p_{out}. Note the setting above is an instance of the commonly used graph generation model for node classification tasks, i.e., Contextual Stochastic Block Model (CSBM) (Deshpande et al., 2018).

Consider an incremental learning setting consisting of two training tasks $\mathcal{T}_1, \mathcal{T}_2$ and associated observed vertex batches V_1, V_2. We define $C_1(V) \mapsto \mathbb{N}$ to be a function that counts the number of vertices in community 1 for a given vertex batch, and function $C_2(V) \mapsto \mathbb{N}$ is defined similarly. Consider a mean aggregation function that averages the node features of the 1-hop neighbors, i.e., mean-agg$(v) = \frac{1}{|\mathrm{N}_1(v)|}\sum_{u \in \mathrm{N}_1(v)} x_u$.

Proposition 4.1 (Imbalanced Observation)**.** *If* $\frac{C_1(V_1)}{C_2(V_1)} \neq \frac{C_1(V_2)}{C_2(V_2)}$, *then we have that* $\mathbb{E}[\textit{mean-agg}(v)|\mathcal{G}_{\mathcal{T}_1}] \neq \mathbb{E}[\textit{mean-agg}(v)|\mathcal{G}_{\mathcal{T}_2}], \forall v \in V_1$.

The proof of Proposition 4.1 can be found in Appendix A. Proposition 4.1 indicates that if the ratio between the number of vertices from the two communities is different, then the appearance of $\mathcal{T}_2$ would alter the expected input of vertices from $\mathcal{T}_1$ in the new graph. Proposition 4.1 highlights the fact that while the underlying mechanism for generating features and connectivity may remain the same across tasks, imbalanced observations can still cause a shift in the input distribution of ego graphs. Furthermore, as the graph evolves, changes in observed properties such as node features and connectivity can also alter the properties of ego graphs of existing vertices, resulting in distributional differences between tasks. This dependency between the input distribution (ego graphs) of vertices V_i and the observed graph structure $\mathcal{G}_{\mathcal{T}_j}$ of the training session $\mathcal{T}_j$, i.e. $P(\mathbf{G}_v|V_i, \mathcal{G}_{\mathcal{T}_j})$, poses a unique risk of catastrophic forgetting for the model. Not only must the model retain information for existing data, but it must also continually adapt to changing input distributions of existing data as the graph evolves.

In this paper, we focus on the effect of structural shift on catastrophic forgetting and assume that the labelling rule is the same for vertices in different sessions, i.e., $P(\mathbf{y}|\mathbf{G}_v, \mathcal{T}_i) = P(\mathbf{y}|\mathbf{G}_v, \mathcal{T}_j), \forall i, j$. In addition, we gauge our analysis toward the case of two training tasks, referred to as the NGIL-2 problem. The NGIL-2 problem is characterized by three distributions: $P(\mathbf{y}, \mathbf{G}_v|V_1, \mathcal{G}_{\mathcal{T}_1})$, $P(\mathbf{y}, \mathbf{G}_v|V_1, \mathcal{G}_{\mathcal{T}_2})$, and $P(\mathbf{y}, \mathbf{G}_v|V_2, \mathcal{G}_{\mathcal{T}_2})$, which are the distributions of V_1 in graphs $\mathcal{G}_{\mathcal{T}_1}$ and $\mathcal{G}_{\mathcal{T}_2}$, and the empirical distribution of V_2 in graph $\mathcal{G}_{\mathcal{T}_2}$. The analysis can be easily extended to multiple training tasks by recursively applying the analysis and treating consecutive tasks as a joint task.

4.2. Structural Shift on Catastrophic Forgetting Risk

Definition 4.2 (Maximum Mean Discrepancy)**.** Let $\mathcal{F}$ be a reproducing kernel Hilbert space (RKHS) with kernel $\mathcal{K}$ and norm $\|.\|_{\mathcal{F}}$. Then the Maximum Mean Discrepancy (MMD) between distribution P_1 and P_2 is defined as:

$$d_{\mathrm{MMD}}(P_1, P_2) := \sup_{f \in \mathcal{F}: \|f\|_{\mathcal{F}} \leq 1} \mathbb{E}_{P_1}[f(x)] - \mathbb{E}_{P_2}[f(x)]$$

MMD is a distance metric between distributions that leverages kernel embedding and has been commonly used to quantify the difference between two distributions (Gretton et al., 2012). MMD admits efficient estimation, as well as fast convergence properties, which are of chief importance in our analysis and the proposed method.

In this paper, we use MMD distance to characterize the effect of the emerging vertex batch on the input distribution of the previous vertex batch. As discussed in subsection 4.1, if the structural shift between the vertex batches is large, the appearance of the new vertex batch will significantly change the input distribution of the previous vertex batch,

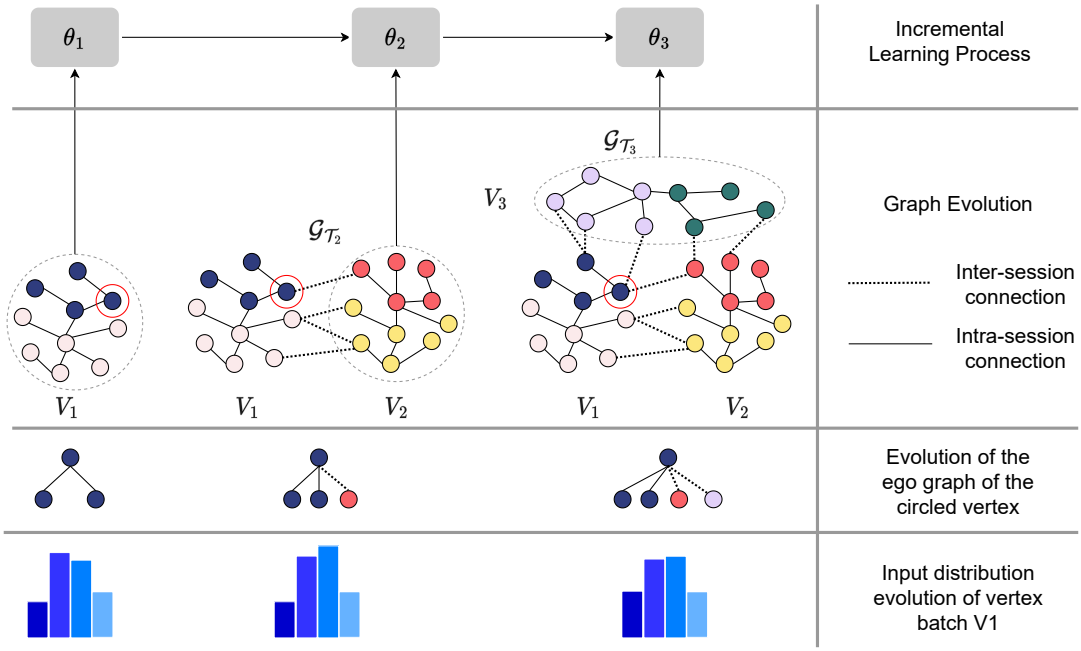


Figure 2. Illustration of the Progression in Inductive NGIL. Each task i results in an update to the model's parameters from θ_{i-1} to θ_i using data from the new task. As new vertex batches associated with each task are introduced, the graph structure changes, potentially altering the input distribution of existing vertices through changes in their ego graphs.

leading to a larger MMD distance. Next, we use MMD to formalize the relation between catastrophic forgetting risk and structural shift.

Theorem 4.3 (CFR Bound)**.** *Let $\mathcal{H} = \{f \in \mathcal{F}_\mathcal{K} : \|f\|_\mathcal{F} \leq 1\}$ where $\mathcal{F}_\mathcal{K}$ is a RKHS with its associated kernel $\mathcal{K}$. Let $P(\mathbf{y}, \mathbf{G}_v|V_1, \mathcal{G}_{\mathcal{T}_1})$, $P(\mathbf{y}, \mathbf{G}_v|V_1, \mathcal{G}_{\mathcal{T}_2})$ and $P(\mathbf{y}, \mathbf{G}_v|V_2, \mathcal{G}_{\mathcal{T}_2})$ be the three distributions that characterize a NGIL-2 problem. Then for a given loss function $\mathcal{L}^q(a, b)$ of the form $|a - b|^q$, for every $h \in \mathcal{H}$, we have*

$$\begin{aligned}\boldsymbol{CFR}(h) \leq\ & R^{\mathcal{L}^q}_{P(\mathbf{y},\mathbf{G}_v|V_2,\mathcal{G}_{\mathcal{T}_2})}(h) \\ & + \underbrace{2 * d_{\mathrm{MMD}}(P(\mathbf{G}_v|V_1, \mathcal{G}_{\mathcal{T}_1}), P(\mathbf{G}_v|V_1, \mathcal{G}_{\mathcal{T}_2}))}_{\textit{Structural Shift}} \\ & + \underbrace{d_{\mathrm{MMD}}(P(\mathbf{G}_v|V_1, \mathcal{G}_{\mathcal{T}_1}), P(\mathbf{G}_v|V_2, \mathcal{G}_{\mathcal{T}_2}))}_{\textit{Structural Shift}} + \lambda,\end{aligned}$$

where

$$\lambda = \min_{h \in \mathcal{H}} R^{\mathcal{L}^q}_{P(\mathbf{y},\mathbf{G}_v|V_1,\mathcal{G}_{\mathcal{T}_2})}(h) + R^{\mathcal{L}^q}_{P(\mathbf{y},\mathbf{G}_v|V_2,\mathcal{G}_{\mathcal{T}_2})}(h).$$

The proof of Theorem 4.3 can be found in Appendix B. Theorem 4.3 provides a formal bound on the catastrophic forgetting risk (as measured by the retention of performance on the previous task) in inductive NGIL. The first term $R^{\mathcal{L}^q}_{P(\mathbf{y},\mathbf{G}_v|V_2,\mathcal{G}_{\mathcal{T}_2})}(h)$ represents the model's performance on the newest task. The second term, $d_{\mathrm{MMD}}(P(\mathbf{G}_v|V_1, \mathcal{G}_{\mathcal{T}_1}), P(\mathbf{G}_v|V_1, \mathcal{G}_{\mathcal{T}_2}))$, captures the structural shift induced by the emergence of the new task, while the third term, $d_{\mathrm{MMD}}(P(\mathbf{G}_v|V_1, \mathcal{G}_{\mathcal{T}_1}), P(\mathbf{G}_v|V_2, \mathcal{G}_{\mathcal{T}_2}))$, captures the structural shift induced by the difference between the new and old tasks. This theorem formalizes the effect of structural shift on the catastrophic forgetting risk in NGIL and can be used to analyze and develop NGIL frameworks.

5. Structural Shift Risk Mitigation

As shown in the previous section, structural shift plays a significant role in causing catastrophic forgetting risks in NGIL problems. In order to mitigate this issue, we propose a method based on divergence minimization. This involves encouraging the model to converge to a latent space that is invariant to the distributional shift caused by the evolving graph structure. To achieve this, we modify the learning objective to minimize the distance between the representation of the vertices in the previous graph structure and the new graph structure. This is done through the inclusion of terms that capture the distance between the representations of vertices in different sessions and regularize the model.

Here, we consider a GNN as the backbone for incremental learning, where the GNN serves as an embedding function that combines the graph structure and node features to learn a representation vector for each node. This representation is then passed to a task-specific decoder function for prediction. Formally, let $\mathcal{Z}$ be the embedding space for the learned representation of vertices, and $\mathcal{H}_g$ represent the hypothesis space of the embedding functions defined by the given GNN model with varying parameters. Let $g : G_v \mapsto \mathcal{Z}$ be

a specific instance of a GNN model within the hypothesis space, that maps a vertex $v \in \mathcal{V}$ to a representation vector z_v in the embedding space $\mathcal{Z}$. Similarly, let $\mathcal{H}_f$ denote the hypothesis space of prediction functions, and $f : \mathcal{Z} \mapsto y_v$ be a specific instance of a prediction function that maps an embedding vector z_v to a label y_v in the label set. Then, an NGIL framework with GNN as the backbone and the aforementioned principle is equivalent to the following learning objective:

$$\begin{aligned} &\min_{g \in \mathcal{H}_g, f \in \mathcal{H}_f} \underbrace{\mathcal{L}(f(g(G_{V_i})), y_{V_i})}_{\text{Training Loss}} + \\ &\underbrace{\alpha \cdot \sum_{j=1}^{i-1} d_{\text{MMD}}(P(g(\mathbf{G}_{V_j})|\mathcal{G}_{\mathcal{T}_{i-1}}), P(g(\mathbf{G}_{V_j})|\mathcal{G}_{\mathcal{T}_i}))}_{\text{Structural Shift Mitigation}} \\ &+ \underbrace{\beta \cdot d_{\text{MMD}}(P(g(\mathbf{G}_{V_j})|\mathcal{G}_{\mathcal{T}_{i-1}}), P(g(\mathbf{G}_{V_i})|\mathcal{G}_{\mathcal{T}_i}))}_{\text{Structural Shift Mitigation}} \end{aligned} \tag{3}$$

where $\mathcal{L}(f(g(G_{V_i})), y_{V_i})$ is the training loss for measuring the prediction performance of the given model on the newest task, and α, β are the hyper-parameter that control the regularization effect. The second and third term captures the distance between the representations of vertices in the previous and current graph structures and serves to encourage the model to converge to a latent space that is invariant to the distributional shift caused by the evolving graph structure. We term equation 3 as structural shift risk mitigation, SSRM.

Theorem 5.1 (Induced CFR Bound)**.** *Let g be a given GNN model that maps vertices to the embedding space. Then for a given loss function $\mathcal{L}^q(a, b)$ of the form $|a - b|^q$ and every prediction function $f \in \mathcal{H}_f$, we have that*

$$\begin{aligned} \boldsymbol{CFR}(f|g) \leq\ & R^{\mathcal{L}}_{P(\mathbf{y}, g(\mathbf{G}_v)|V_2, \mathcal{G}_{\mathcal{T}_2})}(f|g) \\ &+ 2 * d_{\text{MMD}}(P(g(\mathbf{G}_v)|V_1, \mathcal{G}_{\mathcal{T}_1}), P(g(\mathbf{G}_v)|V_1, \mathcal{G}_{\mathcal{T}_2})) \\ &+ d_{\text{MMD}}(P(g(\mathbf{G}_v)|V_1, \mathcal{G}_{\mathcal{T}_1}), P(g(\mathbf{G}_v)|V_2, \mathcal{G}_{\mathcal{T}_2})) + \lambda', \end{aligned}$$

where $\boldsymbol{CFR}(f|g)$ is the catastrophic forgetting bound given a fixed GNN g and

$$\lambda' = \min_{f \in \mathcal{H}_f} R^{\mathcal{L}^q}_{P(\mathbf{y}, g(\mathbf{G}_v)|V_1, \mathcal{G}_{\mathcal{T}_2})}(h) + R^{\mathcal{L}^q}_{P(\mathbf{y}, g(\mathbf{G}_v)|V_2, \mathcal{G}_{\mathcal{T}_2})}(f)$$

The proof of Theorem 5.1 can be found in Appendix C. Theorem 5.1 can be interpreted in a similar way as Theorem 4.3. However, instead of operating on the input space, Theorem 5.1 operates on the latent space learned by the GNN model. The theorem suggests that by encouraging the GNN model to converge to a latent space that reduces the distance induced by structural shift, we can decrease the risk of catastrophic forgetting of the prediction function. This validates our proposed method as given in equation 3.

In practice, we can estimate MMD by comparing the square distance between the empirical kernel mean embedding, as shown in equation 4.

$$\begin{aligned} \widehat{d}^2_{\text{MMD}}(X, Y) &= \frac{1}{n_1^2} \sum_i^{n_1} \sum_j^{n_1} \mathcal{K}(x_i, x_j) \\ &+ \frac{1}{n_2^2} \sum_i^{n_2} \sum_j^{n_2} \mathcal{K}(y_i, y_j) - \frac{2}{n_1 n_2} \sum_i^{n_2} \sum_j^{n_1} \mathcal{K}(x_j, y_i), \end{aligned} \tag{4}$$

where n_1, n_2 represent the number of samples from the two distributions X, Y, respectively, and $\mathcal{K}$ is the chosen kernel function. In our experiment, we use the Gaussian kernel (Dziugaite et al., 2015) with $\mathcal{K}(x, y) = \sum_{\alpha_i} e^{-\alpha_i ||x-y||_2}$ ($\alpha_i = 1, 0.1, 0.01$). The overall procedure and graphical illustration of our proposed method can be found in Appendix D.

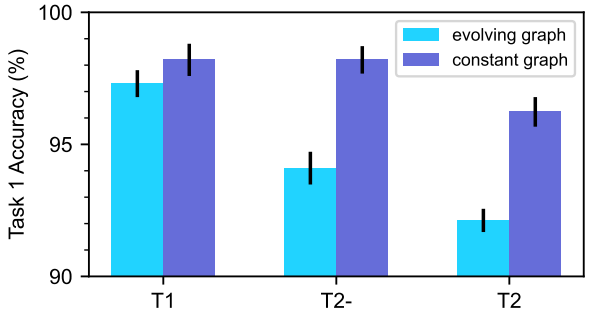
6. Experiment

In this section, we present the experimental results of our proposed method, SSRM, on several NGIL benchmark datasets. We first describe the datasets and experimental set-up, followed by the results and analysis of our proposed method when being applied to the state-of-the-art incremental learning methods in inductive NGIL. Due to space limitations, we provide a more comprehensive description of the datasets, experiment set-up and additional results in Appendix E and F.

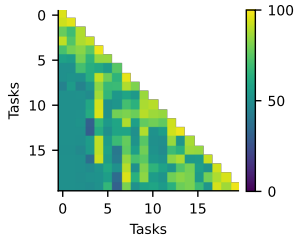
Table 1. Incremental learning settings for each dataset.

Datasets	OGB-Arxiv	Reddit	CoraFull
# vertices	169,343	227,853	19,793
# edges	1,166,243	114,615,892	130,622
# class	40	40	70
# tasks	20	20	35
# vertices / # task	8,467	11,393	660
# edges / # task	58,312	5,730,794	4,354

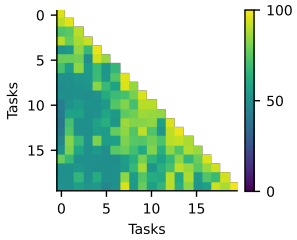
Datasets and Experimental Set-up. We evaluate our proposed method, SSRM, on OGB-Arxiv (Hu et al., 2020), Reddit (Hamilton et al., 2017), and CoraFull (Bojchevski & Günnemann, 2017). The experimental set-up follows the widely adopted task-incremental-learning (task-IL) (Zhang et al., 2022), where a k-class classification task is extracted from the dataset for each training session. For example, in OGB-Arxiv dataset, which has 40 classes, we divide them into 20 tasks: Task 1 is a 2-class classification task between classes 0 and 1, task 2 is between classes 2 and 3, and so on. In each task, the system only has access to the graph induced by the vertices at the current and earlier learning stages, following the formulation in Sec. 3. A brief description of the datasets, and how they are divided into different node classification tasks is given in Table 1. We adopt the implementation from the GIL recent benchmark (Zhang et al.,



(a) Evolution of Task 1 Performance

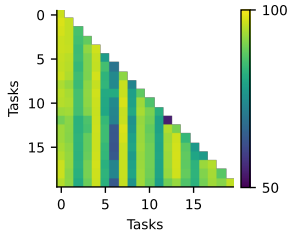


(b) Arxiv, Inductive

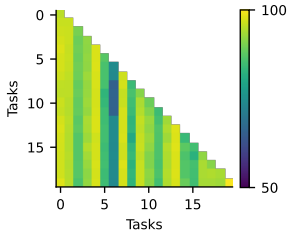


(c) Arxiv, Transductive

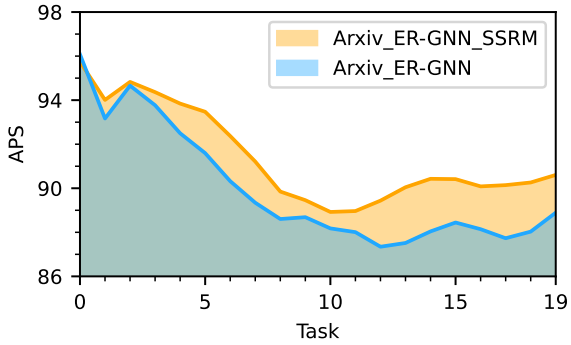
Figure 3. Learning Dynamics of Bare Model on Arxiv in Transductive and Inductive Settings. (a) captures the change of model performance of task 1 when transitioning into task 2 in the inductive and transductive settings. (b) and (c) are the complete performance matrix (x, y-axis are the i, j in $r_{i,j}$ correspondingly) of Bare model in the inductive and transductive settings.



(a) Arxiv, ER-GNN



(b) Arxiv, ER-GNN-SSRM



(c) Arxiv, Learning Curve

Figure 4. Learning Dynamic of ER-GNN on Arxiv w/w.o. SSRM. (a) and (b) are the complete performance matrix (x, y-axis are i, j in $r_{i,j}$ correspondingly) of ER-GNN on Arxiv w./w.o. SSRM. (c) is the learning curve of the two settings illustrating that SSRM leads to a higher APS for each task.

2022) for creating the task-IL setting and closely follow their set-up (such as the train/valid/test set split).

Incremental Learning Frameworks. In our experiments, we evaluate the effectiveness of our proposed method on two state-of-art NGIL frameworks, Experience Replay GNN (ER-GNN) (Liu et al., 2021) and Topology-ware Weight Preserving (TWP) (Zhou & Cao, 2021). ER-GNN stores a small set of representative nodes and replays them with new tasks, while TWP penalizes the update of model parameters to preserve the topological information of previous graphs. Additionally, we also include Gradient Episodic Memory (GEM) as a state-of-the-art incremental learning framework from the DNN field. GEM operates in a similar fashion as ER-GNN, but with a different strategy for selecting representative data. In addition to these incremental learning frameworks, we also include two natural baselines for NGIL: the Bare model and Joint Training. The Bare model denotes the backbone GNN without any continual learning techniques, serving as the lower bound on continual learning performance. On the other hand, Joint Training trains the backbone GNN on all tasks simultaneously, resulting in no forgetting problems and serving as the upper bound for continual learning performance.

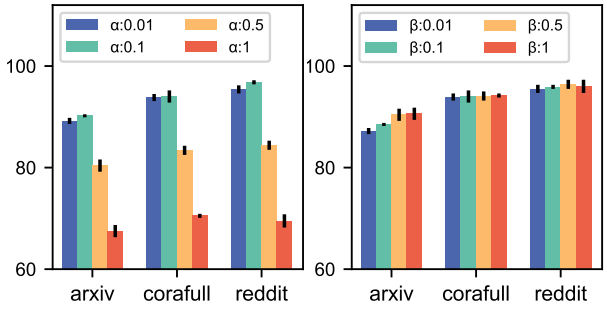
Evaluation Metric. Let $r_{i,j}$ denote the performance (e.g., accuracy) on task j after the model has been trained over a sequence of tasks from 1 to i. Then, the forgetting of task j after being trained over a sequence of tasks from 1 to i is measured by $r_{i,j} - r_{j,j}$. To better understand the dynamics of the overall performance while learning about the task sequence, we are interested in the average performance sequence (APS) $:= \{\frac{\sum_j^i r_{i,j}}{i} | i = 1, ..., m\}$ and the average forgetting sequence (AFS) $:= \{\frac{\sum_j^i r_{i,j} - r_{j,j}}{i} | i = 1, ..., m-1\}$. We use the final average performance (FAP) $:= \frac{\sum_j^m r_{m,j}}{m}$ and the final average forgetting (FAF) $:= \frac{\sum_j^m r_{m,j} - r_{j,j}}{m}$ to measure the overall effectiveness of an NGIL framework.

6.1. Results

Difference in Transductive and Inductive. We first show the difference in task-IL between a transductive setting and an inductive setting. We do so by comparing the learning dynamic (change of $r_{i,j}$ for different i, j) of the Bare model. The result of the Arxiv dataset is illustrated in Fig. 3. In Fig. 3(a), T_1 and T_2 on the x-axis denote the first and the second tasks (training sessions), respectively, and T_2- represents the state when Task 2 has arrived but the model has

Table 2. Performance comparison of existing NGIL frameworks w/w.o. SSRM (↑ higher means better). Results are averaged among five trials. We use $\alpha = 0.1, \beta = 0.5$ for SSRM. **Bold letter** with * indicates that the entry admits an improvement with SSRM.

Dataset	Arixv-CL		CoraFull-CL		Reddit-CL	
Performance Metric	FAP (%) ↑	FAF (%) ↑	FAP (%) ↑	FAF (%) ↑	FAP(%) ↑	FAF (%) ↑
Bare model	55.9 ± 1.2	-33.4 ± 2.3	58.2 ± 3.6	-33.7 ± 3.3	68.6 ± 4.8	-23.9 ± 5.7
Joint Training	92.5 ± 0.6	N.A.	94.4 ± 0.4	N.A.	98.3 ± 1.2	N.A.
GEM	76.6 ± 1.3	-4.1 ± 1.4	88.6 ± 1.1	-3.8 ± 0.7	78.8 ± 7.5	-17.7 ± 5.6
GEM-SSRM	**80.4*** ± 1.7	**-3.1*** ± 1.1	**91.8*** ± 1.2	-3.8 ± 0.8	**81.5*** ± 1.5	**-15.2*** ± 1.4
ER-GNN	86.5 ± 0.5	-10.7 ± 0.6	93.7 ± 1.0	-3.8 ± 0.4	95.1 ± 3.3	-1.9. ± 0.3
ER-GNN-SSRM	**91.2*** ± 0.7	**-8.7*** ± 0.7	**94.3*** ± 0.8	-4.0 ± 1.9	**97.5*** ± 0.4	**-1.8*** ± 0.2
TWP	86.6 ± 0.9	-5.6 ± 0.8	88.1 ± 0.9	-4.2 ± 0.9	89.3 ± 1.2	-8.2 ± 1.3
TWP-SSRM	**88.5*** ± 0.8	-7.2 ±0.9	**90.2*** ± 0.7	**-3.5*** ± 0.5	**92.5*** ± 1.7	**-7.1*** ± 1.6



(a) Varying α with $\beta = 0$ (b) Varying β with $\alpha = 0$

Figure 5. Parameter sensitivity of SSRM. x-axis are the different datasets and the y-axis is FAP. The results are average of five trials. (a) captures the model performance (FAP) when varying α with $\beta = 0$. (b) captures the model performance (FAP) when varying β with $\alpha = 0$. α and β are the two hyperparameters used in equation 3 to control the regularization effect.

not been trained on Task 2 data yet. As shown in the figure, in the case of an evolving graph (inductive), the model performance on Task 1 experiences two drops when entering stage 2: when vertices of Task 2 are added to the graph (structural shift) and when the model is trained on Task 2's data. The bottom half of Fig. 3 is the complete performance matrix (with $r_{i,j}$ as entries) for the two settings. The colour density difference of the two performance matrices illustrates that the existing tasks suffer more catastrophic forgetting in the inductive setting as more tasks arrive.

Effectiveness of SSRM. In Table 2, we show the FAP and FAF of each method after learning the entire task sequence. On average, the Bare model without any continual learning technique performs the worst, and Joint training performs the best. FAF is inapplicable to joint-trained models because they do not follow the continual learning setting and are simultaneously trained on all tasks. In terms of FAP, methods with SSRM consistently admit an improvement with a significant margin. On the other hand, some methods do experience a slight FAF drop after applying SSRM. The reason is that SSRM simultaneously allows for a better forward transfer (transferring information from existing tasks to new tasks) and hence leads to higher performance for each new task (i.e., larger $r_{i,i}$ for each task i), as illustrated in the performance matrices and learning curve in Fig. 4.

6.2. Ablation Study

In SSRM, there are two main hyperparameters: α and β (see equation 3 for more details). In Fig. 5, the performance of SSRM with different α and β in $[0, 1]$ is reported. The results show that the model performance is blunt to the change of β when $\alpha = 0$. On the other hand, if α is too large, the representations learnt by the GNN lose expressive power over different tasks and hence the performance would decrease.

7. Conclusion and Discussion

This paper presents a mathematical formulation of the inductive NGIL problem and provides a comprehensive analysis of the catastrophic forgetting problem in this setting. Based on the analysis, we propose a novel method, SSRM, that addresses the problem of the structural shift in inductive NGIL by minimizing the divergence between the representations of vertices in different tasks and graphs. Our method is simple to implement and can be integrated with existing incremental learning frameworks. Empirical evaluations on benchmark datasets for NGIL show that our method significantly improves the performance of state-of-the-art NGIL frameworks in the inductive setting.

This research opens up new opportunities for addressing the catastrophic forgetting problem in NGIL and we hope it lays down a solid foundation for further research in the area. The current formulation assumes a clear task boundary. In future work, one working direction is to extend our analysis and apply SSRM to other GIL settings such as link prediction in dynamic graphs where there is no clear task boundary.

Acknowledgement

We would like to thank the anonymous reviewers and area chairs for their helpful comments. This work was supported in part by the following grants: 1) Hong Kong RGC under the contracts HKU 17207621 and 17203522, 2) National Natural Science Foundation of China (62206203) and 3) Fundamental Research Funds for the Central Universities (2042022kf1034).

References

Ahrabian, K., Xu, Y., Zhang, Y., Wu, J., Wang, Y., and Coates, M. Structure aware experience replay for incremental learning in graph-based recommender systems. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, pp. 2832–2836, 2021.

Aljundi, R., Lin, M., Goujaud, B., and Bengio, Y. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019.

Bielak, P., Tagowski, K., Falkiewicz, M., Kajdanowicz, T., and Chawla, N. V. Fildne: a framework for incremental learning of dynamic networks embeddings. *Knowledge-Based Systems*, 236:107453, 2022.

Biesialska, M., Biesialska, K., and Costa-jussà, M. R. Continual lifelong learning in natural language processing: A survey. *arXiv preprint arXiv:2012.09823*, 2020.

Bojchevski, A. and Günnemann, S. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. *arXiv preprint arXiv:1707.03815*, 2017.

Caccia, L., Belilovsky, E., Caccia, M., and Pineau, J. Online learned continual compression with adaptive quantization modules. In *International conference on machine learning*, pp. 1240–1250. PMLR, 2020.

Cai, J., Wang, X., Guan, C., Tang, Y., Xu, J., Zhong, B., and Zhu, W. Multimodal continual graph learning with neural architecture search. In *Proceedings of the ACM Web Conference 2022*, pp. 1292–1300, 2022.

Carta, A., Cossu, A., Errica, F., and Bacciu, D. Catastrophic forgetting in deep graph networks: an introductory benchmark for graph classification. *arXiv preprint arXiv:2103.11750*, 2021.

Chrysakis, A. and Moens, M.-F. Online continual learning from imbalanced data. In *International Conference on Machine Learning*, pp. 1952–1961. PMLR, 2020.

Daruna, A., Gupta, M., Sridharan, M., and Chernova, S. Continual learning of knowledge graph embeddings. *IEEE Robotics and Automation Letters*, 6(2):1128–1135, 2021.

Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Deshpande, Y., Sen, S., Montanari, A., and Mossel, E. Contextual stochastic block models. *Advances in Neural Information Processing Systems*, 31, 2018.

Dziugaite, G. K., Roy, D. M., and Ghahramani, Z. Training generative neural networks via maximum mean discrepancy optimization. *arXiv preprint arXiv:1505.03906*, 2015.

Farajtabar, M., Azizan, N., Mott, A., and Li, A. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pp. 3762–3773. PMLR, 2020.

Feng, Y., Jiang, J., and Gao, Y. Incremental learning on growing graphs. *openReview preprint https://openreview.net/forum?id=nySHNUlKTVw*, 2020.

Galke, L., Franke, B., Zielke, T., and Scherp, A. Lifelong learning of graph neural networks for open-world node classification. In *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8. IEEE, 2021.

Gama, J., Žliobaitė, I., Bifet, A., Pechenizkiy, M., and Bouchachia, A. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Hamilton, W. L., Ying, R., and Leskovec, J. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, pp. 1025–1035, 2017.

Han, Y., Karunasekera, S., and Leckie, C. Graph neural networks with continual learning for fake news detection from social media. *arXiv preprint arXiv:2007.03316*, 2020.

Hu, W., Fey, M., Zitnik, M., Dong, Y., Ren, H., Liu, B., Catasta, M., and Leskovec, J. Open graph benchmark: Datasets for machine learning on graphs. *arXiv preprint arXiv:2005.00687*, 2020.

Jung, H., Ju, J., Jung, M., and Kim, J. Less-forgetting learning in deep neural networks. *arXiv preprint arXiv:1607.00122*, 2016.

Kim, S., Yun, S., and Kang, J. Dygrain: An incremental learning framework for dynamic graphs. In *31st International Joint Conference on Artificial Intelligence, IJCAI 2022*, pp. 3157–3163. International Joint Conferences on Artificial Intelligence, 2022.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Knoblauch, J., Husain, H., and Diethe, T. Optimal continual learning has perfect memory and is np-hard. In *International Conference on Machine Learning*, pp. 5327–5337. PMLR, 2020.

Kou, X., Lin, Y., Liu, S., Li, P., Zhou, J., and Zhang, Y. Disentangle-based continual graph representation learning. *arXiv preprint arXiv:2010.02565*, 2020.

Li, Z. and Hoiem, D. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Liu, H., Yang, Y., and Wang, X. Overcoming catastrophic forgetting in graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8653–8661, 2021.

Lopez-Paz, D. and Ranzato, M. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30, 2017.

Ma, Y., Guo, Z., Ren, Z., Tang, J., and Yin, D. Streaming graph neural networks. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 719–728, 2020.

Nguyen, G. H., Lee, J. B., Rossi, R. A., Ahmed, N. K., Koh, E., and Kim, S. Continuous-time dynamic network embeddings. In *Companion proceedings of the the web conference 2018*, pp. 969–976, 2018.

Parisi, G. I., Kemker, R., Part, J. L., Kanan, C., and Wermter, S. Continual lifelong learning with neural networks: A review. *Neural Networks*, 113:54–71, 2019.

Pfülb, B. and Gepperth, A. A comprehensive, application-oriented study of catastrophic forgetting in dnns. *arXiv preprint arXiv:1905.08101*, 2019.

Redko, I., Morvant, E., Habrard, A., Sebban, M., and Bennani, Y. A survey on domain adaptation theory: learning bounds and theoretical guarantees. *arXiv preprint arXiv:2004.11829*, 2020.

Rusu, A. A., Rabinowitz, N. C., Desjardins, G., Soyer, H., Kirkpatrick, J., Kavukcuoglu, K., Pascanu, R., and Hadsell, R. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016.

Saha, G., Garg, I., and Roy, K. Gradient projection memory for continual learning. *arXiv preprint arXiv:2103.09762*, 2021.

Shin, H., Lee, J. K., Kim, J., and Kim, J. Continual learning with deep generative replay. *Advances in neural information processing systems*, 30, 2017.

Su, J. and Wu, C. Towards robust inductive graph incremental learning via experience replay, 2023.

Tan, Z., Ding, K., Guo, R., and Liu, H. Graph few-shot class-incremental learning. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, pp. 987–996, 2022.

Wang, C., Qiu, Y., and Scherer, S. Bridging graph network to lifelong learning with feature interaction. 2020a.

Wang, C., Qiu, Y., Gao, D., and Scherer, S. Lifelong graph learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13719–13728, 2022.

Wang, J., Song, G., Wu, Y., and Wang, L. Streaming graph neural networks via continual learning. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1515–1524, 2020b.

Wortsman, M., Ramanujan, V., Liu, R., Kembhavi, A., Rastegari, M., Yosinski, J., and Farhadi, A. Supermasks in superposition. *Advances in Neural Information Processing Systems*, 33:15173–15184, 2020.

Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., and Fu, Y. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 374–382, 2019.

Wu, Z., Pan, S., Chen, F., Long, G., Zhang, C., and Philip, S. Y. A comprehensive survey on graph neural networks. *IEEE transactions on neural networks and learning systems*, 32(1):4–24, 2020.

Wu, Z., Baek, C., You, C., and Ma, Y. Incremental learning via rate reduction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1125–1133, 2021.

Xu, Y., Zhang, Y., Guo, W., Guo, H., Tang, R., and Coates, M. Graphsail: Graph structure aware incremental learning for recommender systems. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 2861–2868, 2020.

Yoon, J., Yang, E., Lee, J., and Hwang, S. J. Lifelong learning with dynamically expandable networks. *arXiv preprint arXiv:1708.01547*, 2017.

Yoon, J., Kim, S., Yang, E., and Hwang, S. J. Scalable and order-robust continual learning with additive parameter decomposition. *arXiv preprint arXiv:1902.09432*, 2019.

Yu, W., Cheng, W., Aggarwal, C. C., Zhang, K., Chen, H., and Wang, W. Netwalk: A flexible deep embedding approach for anomaly detection in dynamic networks. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining*, pp. 2672–2681, 2018.

Zhang, X., Song, D., and Tao, D. Hierarchical prototype networks for continual graph representation learning. *arXiv preprint arXiv:2111.15422*, 2021.

Zhang, X., Song, D., and Tao, D. Cglb: Benchmark tasks for continual graph learning. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022.

Zhou, F. and Cao, C. Overcoming catastrophic forgetting in graph neural networks with experience replay. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 4714–4722, 2021.

Zliobaite, I. Learning under concept drift: an overview. *arXiv preprint arXiv:1010.4784*, 41, 2010.

A. Appendix: Proof for Proposition 4.1

Proof. First recall that we are considering an incremental learning setting consisting of two training tasks $\mathcal{T}_1, \mathcal{T}_2$ and associated observed vertex batches V_1, V_2. We define $C_1(V) \mapsto \mathbb{N}$ to be a function that counts the number of vertices in community 1 for a given vertex batch, and similarly defined for $C_2(V) \mapsto \mathbb{N}$. Consider a mean aggregation function that averages the node features of the 1-hop neighbors, i.e., mean-agg$(v) = \frac{1}{|\mathrm{N}_1(v)|}\sum_{u\in\mathrm{N}_1(v)} x_u$.

Under this setup, we have the expected input of vertex of community 1 in task 1 as follows.

$$\mathbb{E}[\text{mean-agg}(v)|\mathcal{G}_{\mathcal{T}_1}] = \frac{C_1(V_1)p_{in}}{C_1(V_1)p_{in} + p_{out}C_2(V_1)}\mu_1 + \frac{C_2(V_1)p_{out}}{C_1(V_1)p_{in} + p_{out}C_2(V_1)}\mu_2 \tag{5}$$

Similarly, we have the expected input of the vertex of community 1 in task 2 as follows.

$$\mathbb{E}[\text{mean-agg}(v)|\mathcal{G}_{\mathcal{T}_2}] = \frac{C_1(V)p_{in}}{C_1(V)p_{in} + p_{out}C_2(V)}\mu_1 + \frac{C_2(V)p_{out}}{C_1(V)p_{in} + p_{out}C_2(V)}\mu_2 \tag{6}$$

where $V = V_1 + V_2$. For simplicity, let's denote $a = C_1(V_1)p_{in}, b = p_{out}C_2(V), c = C_1(V_2)p_{in}, d = C_2(V_2)p_{out}$.

$$\begin{aligned}
&\mathbb{E}[\text{mean-agg}(v)|\mathcal{G}_{\mathcal{T}_1}] \neq \mathbb{E}[\text{mean-agg}(v)|\mathcal{G}_{\mathcal{T}_2}] \\
&\Leftrightarrow \frac{a}{a+b} \neq \frac{a+c}{a+b+c+d} \\
&\Leftrightarrow \frac{a}{b} \neq \frac{c}{d}
\end{aligned} \tag{7}$$

Substitute back $a = C_1(V_1)p_{in}, b = p_{out}C_2(V), c = C_1(V_2)p_{in}, d = C_2(V_2)p_{out}$, we obtain the result of the proposition □

B. Appendix: Proof for Theorem 4.3

In this section, we provide proof for Theorem 4.3.

We use the following Lemma 35 from (Redko et al., 2020).

Lemma B.1. *Let $\mathcal{H} = \{f \in \mathcal{H}_k : \|f\|_{\mathcal{F}} \leq 1\}$ where $\mathcal{H}_k$ is a RKHS with its associated kernel k. Let P_1, P_2 be two arbitrary distributions that share the same input and output space as $\mathcal{H}_k$. Then for a given loss function $\mathcal{L}^q(a,b)$ of the form $|a-b|^q$, for every $h, h' \in \mathcal{H}$, we have*

$$R^{\mathcal{L}^q}_{P_1}(h,h') \leq R^{\mathcal{L}^q}_{P_2}(h,h') + d_{\mathrm{MMD}}(P_{X1}, P_{X2}),$$

where P_{X1}, P_{X2} are the input distribution of P_1, P_2 and $R^{\mathcal{L}^q}_{P}(h,h') = \mathbb{E}_{x\sim P_X}[\mathcal{L}^q(h(x), h'(x))]$.

Proof. Let $\mathcal{H} = \{f \in \mathcal{H}_k : \|f\|_{\mathcal{F}} \leq 1\}$ where $\mathcal{H}_k$ is a RKHS with its associated kernel k. Let $P(\mathbf{y}, \mathbf{G}_v|V_1, \mathcal{G}_{\mathcal{T}_1})$, $P(\mathbf{y}, \mathbf{G}_v|V_1, \mathcal{G}_{\mathcal{T}_2})$ and $P(\mathbf{y}, \mathbf{G}_v|V_2, \mathcal{G}_{\mathcal{T}_2})$ be the three distributions that characterize a NGIL-2 problem. Let $\mathcal{L}^q(a,b)$ be a loss function as given in the premise of the theorem. For simplicity, we denote $P_1 = P(\mathbf{y}, \mathbf{G}_v|V_1, \mathcal{G}_{\mathcal{T}_1}), P_2 = P(\mathbf{y}, \mathbf{G}_v|V_1, \mathcal{G}_{\mathcal{T}_2}), P = P(\mathbf{y}, \mathbf{G}_v|V_2, \mathcal{G}_{\mathcal{T}_2})$.

Let's denote the optimal function in the hypothesis space as,

$$h_2^* = \arg\min_{h\in\mathcal{H}} R^{\mathcal{L}^q}_{P_2}(h)$$

$$\lambda = \min_{h\in\mathcal{H}} R^{\mathcal{L}^q}_{P_3}(h) + R^{\mathcal{L}^q}_{P_2}(h)$$

Then, we have that $\forall h \in \mathcal{H}$,

$$\begin{aligned}
R^{\mathcal{L}^q}_{P_2}(h) &\leq R^{\mathcal{L}^q}_{P_2}(h_2^*) + R^{\mathcal{L}^q}_{P_2}(h, h_2^*) \\
&= R^{\mathcal{L}^q}_{P_2}(h_2^*) + R^{\mathcal{L}^q}_{P_2}(h, h_2^*) + R^{\mathcal{L}^q}_{P_1}(h, h_2^*) - R^{\mathcal{L}^q}_{P_1}(h, h_2^*) + R^{\mathcal{L}^q}_{P_3}(h, h_2^*) - R^{\mathcal{L}^q}_{P_3}(h, h_2^*) + R^{\mathcal{L}^q}_{P_2}(h, h_2^*) - R^{\mathcal{L}^q}_{P_2}(h, h_2^*) \\
&= R^{\mathcal{L}^q}_{P_2}(h_2^*) + [R^{\mathcal{L}^q}_{P_2}(h, h_2^*) - R^{\mathcal{L}^q}_{P_1}(h, h_2^*)] + [R^{\mathcal{L}^q}_{P_2}(h, h_2^*) - R^{\mathcal{L}^q}_{P_3}(h, h_2^*)] + R^{\mathcal{L}^q}_{P_1}(h, h_2^*) + R^{\mathcal{L}^q}_{P_3}(h, h_2^*) - R^{\mathcal{L}^q}_{P_2}(h, h_2^*)
\end{aligned} \tag{8}$$

Applying Lemma B.1 on P_1, P_2 we have that

$$\begin{aligned} &R^{\mathcal{L}^q}_{P_2}(h,h') \leq R^{\mathcal{L}^q}_{P_1}(h,h') + d_{\mathrm{MMD}}(P_{X1}, P_{X2}) \\ &\Leftrightarrow R^{\mathcal{L}^q}_{P_2}(h,h') - R^{\mathcal{L}^q}_{P_1}(h,h') \leq d_{\mathrm{MMD}}(P_{X1}, P_{X2}) \end{aligned} \tag{9}$$

Similarly, applying Lemma B.1 on P_2, P_3 we have that

$$R^{\mathcal{L}^q}_{P_2}(h,h') - R^{\mathcal{L}^q}_{P_3}(h,h') \leq d_{\mathrm{MMD}}(P_{X2}, P_{X3}) \tag{10}$$

Substitute back to the inequality above, we have that

$$\begin{aligned} R^{\mathcal{L}^q}_{P_2}(h) &\leq R^{\mathcal{L}^q}_{P_2}(h_2^*) + R^{\mathcal{L}^q}_{P_2}(h, h_2^*) \\ &\leq R^{\mathcal{L}^q}_{P_2}(h_2^*) + [R^{\mathcal{L}^q}_{P_2}(h, h_2^*) - R^{\mathcal{L}^q}_{P_1}(h, h_2^*)] + [R^{\mathcal{L}^q}_{P_2}(h, h_2^*) - R^{\mathcal{L}^q}_{P_3}(h, h_2^*)] + R^{\mathcal{L}^q}_{P_1}(h, h_2^*) + R^{\mathcal{L}^q}_{P_3}(h, h_2^*) - R^{\mathcal{L}^q}_{P_2}(h, h_2^*) \\ &\leq R^{\mathcal{L}^q}_{P_2}(h_2^*) + 2d_{\mathrm{MMD}}(P_{X1}, P_{X2}) + d_{\mathrm{MMD}}(P_{X2}, P_{X3}) + R^{\mathcal{L}^q}_{P_3}(h, h_2^*) \\ &\leq R^{\mathcal{L}^q}_{P_2}(h_2^*) + 2d_{\mathrm{MMD}}(P_{X1}, P_{X2}) + d_{\mathrm{MMD}}(P_{X2}, P_{X3}) + R^{\mathcal{L}^q}_{P_3}(h) + R^{\mathcal{L}^q}_{P_3}(h_2^*) \\ &= R^{\mathcal{L}^q}_{P_3}(h) + 2d_{\mathrm{MMD}}(P_{X1}, P_{X2}) + d_{\mathrm{MMD}}(P_{X2}, P_{X3}) + R^{\mathcal{L}^q}_{P_3}(h) + R^{\mathcal{L}^q}_{P_3}(h_2^*) + R^{\mathcal{L}^q}_{P_2}(h_2^*) \\ &\leq R^{\mathcal{L}^q}_{P_3}(h) + 2d_{\mathrm{MMD}}(P_{X1}, P_{X2}) + d_{\mathrm{MMD}}(P_{X2}, P_{X3}) + R^{\mathcal{L}^q}_{P_3}(h) + R^{\mathcal{L}^q}_{P_3}(h_2^*) + \lambda \end{aligned} \tag{11}$$

Substitute $P_1 = P(\mathbf{y}, \mathbf{G}_v | V_1, \mathcal{G}_{\mathcal{T}_1}), P_2 = P(\mathbf{y}, \mathbf{G}_v | V_1, \mathcal{G}_{\mathcal{T}_2}), P = P(\mathbf{y}, \mathbf{G}_v | V_2, \mathcal{G}_{\mathcal{T}_2})$ back to the equation above, we got

$$\mathbf{CFR}(h) \leq R^{\mathcal{L}^q}_{P(\mathbf{y}, \mathbf{G}_v | V_2, \mathcal{G}_{\mathcal{T}_2})}(h) + 2d_{\mathrm{MMD}}(P(\mathbf{G}_v | V_1, \mathcal{G}_{\mathcal{T}_1}), P(\mathbf{G}_v | V_1, \mathcal{G}_{\mathcal{T}_2})) + d_{\mathrm{MMD}}(P(\mathbf{G}_v | V_1, \mathcal{G}_{\mathcal{T}_1}), P(\mathbf{G}_v | V_2, \mathcal{G}_{\mathcal{T}_2})) + \lambda. \tag{12}$$

□

C. Proof of Theorem 5.1

In this section, we provide the proof for Theorem 5.1.

Proof. Let $\mathcal{H}_f$ be the hypothesis space for the prediction head and g be a given GNN that maps a vertex into an embedding space $\mathcal{Z}$. Then, a given g induces a distribution in the latent space $\mathcal{Z}$ for the prediction function $f \in \mathcal{H}_f$. For simplicity, let's denote the induced distribution as, $P_1 = R_{P(\mathbf{y}, g(\mathbf{G}_v) | V_1, \mathcal{G}_{\mathcal{T}_1})}, P_2 = P(\mathbf{y}, g(\mathbf{G}_v) | V_1, \mathcal{G}_{\mathcal{T}_2}), P_3 = P(\mathbf{y}, g(\mathbf{G}_v) | V_2, \mathcal{G}_{\mathcal{T}_2})$.

Let's denote the optimal function in the hypothesis space as,

$$f_2^* = \arg\min_{h \in \mathcal{H}_f} R^{\mathcal{L}^q}_{P_2}(f)$$

$$\lambda' = \min_{f \in \mathcal{H}_f} R^{\mathcal{L}^q}_{P_3}(f) + R^{\mathcal{L}^q}_{P_2}(f)$$

Then, we have that $\forall f \in \mathcal{H}_f$,

$$\begin{aligned} R^{\mathcal{L}^q}_{P_2}(f) &\leq R^{\mathcal{L}^q}_{P_2}(f_2^*) + R^{\mathcal{L}^q}_{P_2}(f, f_2^*) \\ &= R^{\mathcal{L}^q}_{P_2}(f_2^*) + R^{\mathcal{L}^q}_{P_2}(f, f_2^*) + R^{\mathcal{L}^q}_{P_1}(f, f_2^*) - R^{\mathcal{L}^q}_{P_1}(f, f_2^*) + R^{\mathcal{L}^q}_{P_3}(f, f_2^*) - R^{\mathcal{L}^q}_{P_3}(f, f_2^*) + R^{\mathcal{L}^q}_{P_2}(f, f_2^*) - R^{\mathcal{L}^q}_{P_2}(f, f_2^*) \\ &= R^{\mathcal{L}^q}_{P_2}(h_2^*) + [R^{\mathcal{L}^q}_{P_2}(f, f_2^*) - R^{\mathcal{L}^q}_{P_1}(f, f_2^*)] + [R^{\mathcal{L}^q}_{P_2}(f, f_2^*) - R^{\mathcal{L}^q}_{P_3}(f, f_2^*)] + R^{\mathcal{L}^q}_{P_1}(f, f_2^*) + R^{\mathcal{L}^q}_{P_3}(f, f_2^*) - R^{\mathcal{L}^q}_{P_2}(f, f_2^*) \end{aligned} \tag{13}$$

Then, following the same procedure in the proof of Theorem 4.3, we can get

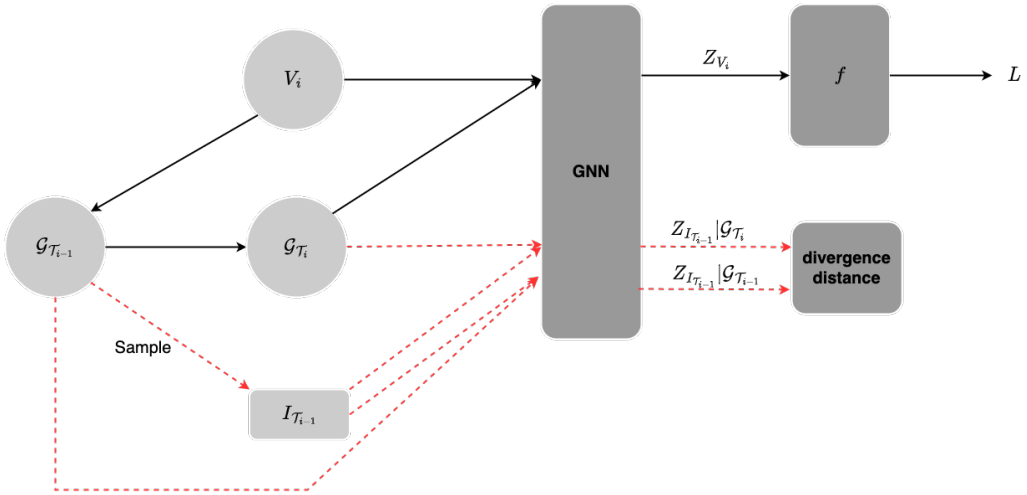


Figure 6. Illustration of the structural shift risk mitigation

$$\begin{aligned}\mathbf{CFR}(f|g) \leq & R^{\mathcal{L}}_{P(\mathbf{y},g(\mathbf{G}_v)|V_2,\mathcal{G}_{\mathcal{T}_2})}(f|g) + 2 * d_{\text{MMD}}(P(g(\mathbf{G}_v)|V_1,\mathcal{G}_{\mathcal{T}_1}), P(g(\mathbf{G}_v)|V_1,\mathcal{G}_{\mathcal{T}_2})) \\ & + d_{\text{MMD}}(P(g(\mathbf{G}_v)|V_1,\mathcal{G}_{\mathcal{T}_1}), P(g(\mathbf{G}_v)|V_2,\mathcal{G}_{\mathcal{T}_2})) + \lambda',\end{aligned} \tag{14}$$

□

D. Appendix:Overall Procedure

The overall learning procedure in each stage is summarized in Algorithm 1 and Fig. 6 provide a graphical illustration of the procedure. It involves training the GNN model on the current task while also minimizing the distance between the representations of vertices in the previous and current graph structures through the inclusion of the structural shift mitigation term. This results in a model that is able to maintain good performance on previous tasks while also learning the current task effectively, leading to improved generalization performance.

Algorithm 1 Learning Procedure for $\mathcal{T}_i$

Input: V_i //new coming vertex set for $\mathcal{T}_i$
Require: f, g //current model parameter
Require: $I_{\mathcal{T}_{i-1}}$ //a set of vertices sampled form $V_1, ..., V_{i-1}$
Require: β, α //hyper parameter for controling the regularization effect
Repeat until convergence:
Compute the representation for I before and after V_i

$$Z_{\text{bef}} = g(I_{\mathcal{T}_{i-1}}, \theta_g|\mathcal{G}_{\mathcal{T}_{i-1}}), \quad \mathcal{Z}_{\text{aft}} = g(I_{\mathcal{T}_{i-1}}, \theta_g|\mathcal{G}_{\mathcal{T}_i})$$

$$\mathcal{Z}_{V_i} = g(V_i, \theta_g|\mathcal{G}_{\mathcal{T}_i})$$

Update f, g through training procedure (e.g. Back Propogation) with loss function:

$$\mathcal{L}(f(g(G_{V_i}, \theta_g), \theta_f), y_{V_i}) + \text{Reg}$$

where

$$\text{Reg} = \beta \hat{d_{\text{MMD}^2}}(Z_{\text{bef}}, \mathcal{Z}_{\text{aft}}) + \alpha \hat{d_{\text{MMD}^2}}(Z_{\text{bef}}, \mathcal{Z}_{V_i}) \tag{15}$$

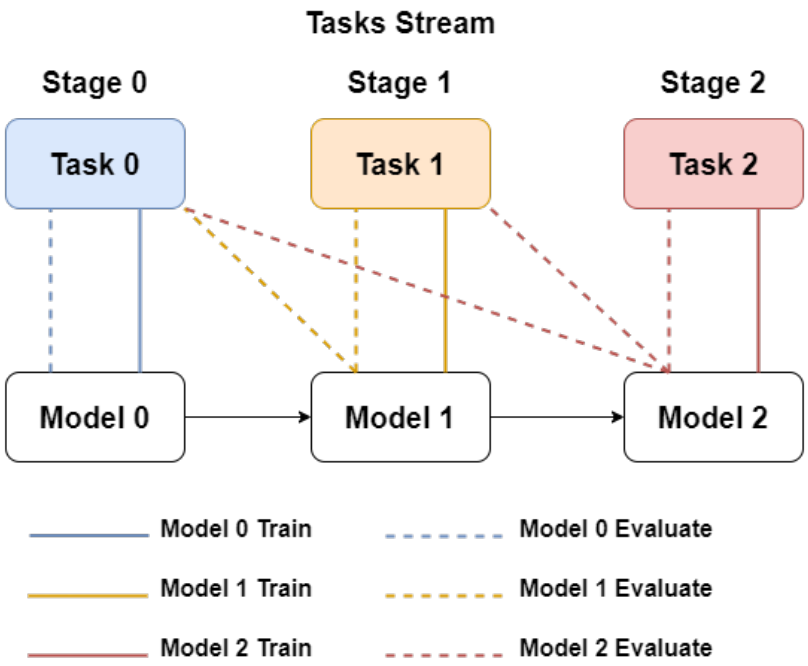


Figure 7. Incremental Training in Development Phase. The figure above illustrates what task data the model used for training and evaluation.

D.1. Space Complexity

To compute the maximum mean discrepancy (MMD) given in Eq. equation 15 in the SSRM framework, we need the representation of $Z_{\text{bef}}, \mathcal{Z}_{V_i}$ and $\mathcal{Z}_{\text{aft}}$. We can obtain $\mathcal{Z}_{\text{aft}}$ and $\mathcal{Z}_{V_i}$ from the updated graph at each task. The only extra storage cost incurred by the proposed method comes from Z_{bef}, and the number of data required for Z_{bef} is determined by the sampling efficiency of MMD, that has been empirically shown to be efficient. Furthermore, as SSRM is not a stand-alone incremental learning framework but serves as a regularization to mitigate the effect of structural shift and boost the performance of existing incremental learning frameworks. As most existing incremental learning frameworks require storage or access to the previous data, SSRM could utilize this fact and make use of the data that have already been used by the incremental learning framework. In other words, for most incremental learning frameworks, such as ER-GNN, SSRM does not incur extra storage costs.

D.2. Computation Complexity

First, recall that the equation we used for estimating the maximum mean discrepancy (MMD) is given as follows.

$$\widehat{d}^2_{\text{MMD}}(X,Y) = \frac{1}{n_1^2}\sum_i^{n_1}\sum_j^{n_1}\mathcal{K}(x_i,x_j) + \frac{1}{n_2^2}\sum_i^{n_2}\sum_j^{n_2}\mathcal{K}(y_i,y_j) - \frac{2}{n_1 n_2}\sum_i^{n_2}\sum_j^{n_1}\mathcal{K}(x_j,y_i), \tag{16}$$

Let $N = \max\{n_1, n_2\}$ where n_1, n_2 are the constants in Eq. 16. It is obvious that the computation complexity of Eq. 16 above is $\mathcal{O}(N^2)$. Note that subsampling is commonly employed for estimating MMD, i.e., n_1 and n_2 (hence N) would be small and therefore, the cost of SSRM is small compared with the standard training time. Our empirical study finds that the extra computation cost from MMD is about 5% of the total computation time. There exists other techniques such as kernel approximation to make MMD computation even more scalable.

E. Appendix: Additional Experiment Details

E.1. Hardware and Software

All the experiments of this paper are conducted on the following machine

CPU: two Intel Xeon Gold 6230 2.1G, 20C/40T, 10.4GT/s, 27.5M Cache, Turbo, HT (125W) DDR4-2933

GPU: four NVIDIA Tesla V100 SXM2 32G GPU Accelerator for NV Link

Memory: 256GB (8 x 32GB) RDIMM, 3200MT/s, Dual Rank

OS: Ubuntu 18.04LTS

E.2. Dataset and Processing

E.3. Dataset Description

OGB-Arxiv. The OGB-Arxiv dataset (Hu et al., 2020) is a benchmark dataset for node classification. It is constructed from the arXiv e-print repository, a popular platform for researchers to share their preprints. The graph structure is constructed by connecting papers that cite each other. The node features include the text of the paper's abstract, title, and its authors' names. Each node is assigned one of 40 classes, which corresponds to the paper's main subject area.

Cora-Full. The Cora-Full (Bojchevski & Günnemann, 2017) is a benchmark dataset for node classification. Similarly to OGB-Arxiv, it is a citation network consisting of 70 classes.

Reddit. The Reddit dataset (Hamilton et al., 2017) is a benchmark dataset for node classification that consists of posts and comments from the website Reddit.com. Each node represents a post or comment and each edge represents a reply relationship between posts or comments.

Table 3. Incremental learning settings for each dataset.

Datasets	OGB-Arxiv	Reddit	CoraFull
# vertices	169,343	227,853	19,793
# edges	1,166,243	114,615,892	130,622
# class	40	40	70
# tasks	20	20	35
# vertices / # task	8,467	11,393	660
# edges / # task	58,312	5,730,794	4,354

E.3.1. License

The datasets used in this paper are curated from existing public data sources and follow their licenses. OGB-Arxiv is licensed under Open Data Commons Attribution License (ODC-BY). Cora-Full dataset and the Reddit dataset are two datasets built from publicly available sources (public papers and Reddit posts) without a license attached by the authors.

E.3.2. Data Processing

For the datasets, we remove the 41-th class of Reddit-CL, following closely in (Zhang et al., 2022). This aims to ensure an even number of classes for each dataset to be divided into a sequence of 2-class tasks. For all the datasets, the train-validation-test splitting ratios are 60%, 20%, and 20%. The train-validation-test splitting is obtained by random sampling, therefore the performance may be slightly different with splittings from different rounds of random sampling.

E.4. Hyperparameter of Incremental Learning Framework

Table 4 is the hyperparameter research space we adopt from (Zhang et al., 2022).

Table 4. Incremental learning settings for each dataset.

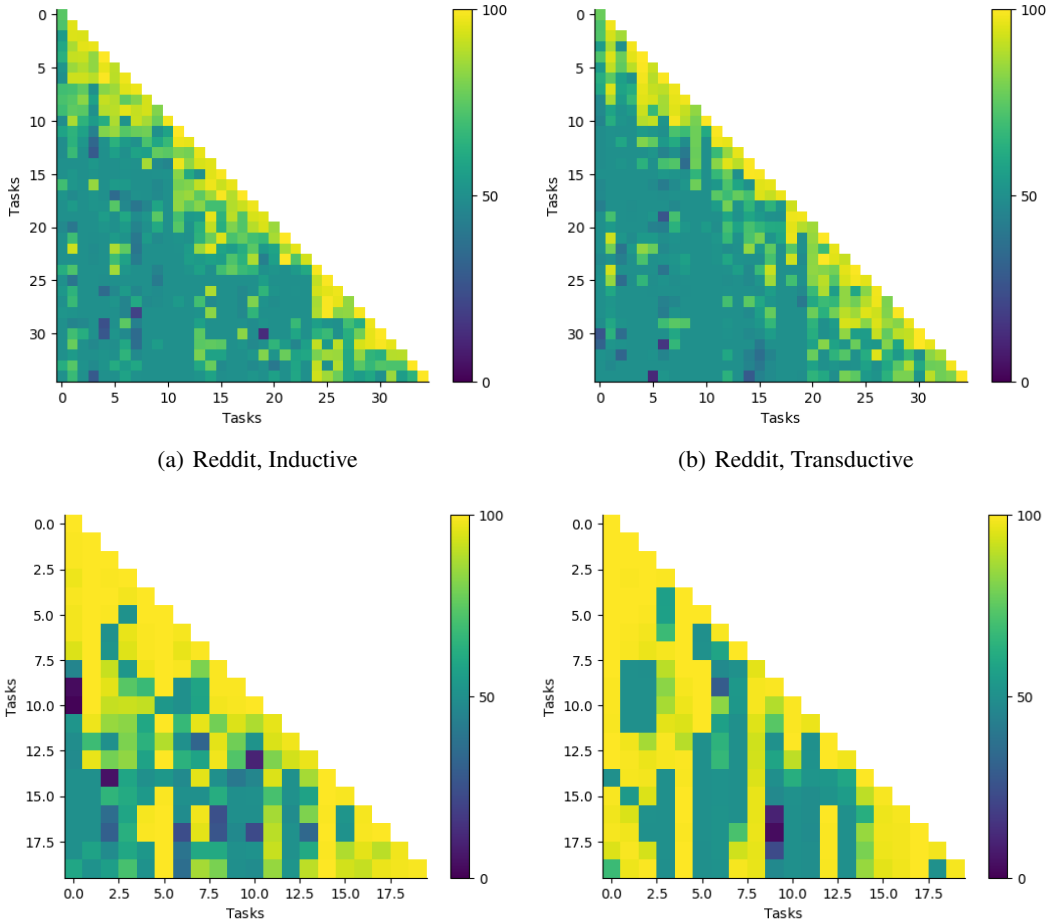
GEM	memory_strength:[0.05,0.5,5]; n_memories:[10,100,1000]
TWP	lambda_1:[100,10000]; lambda_t:[100,10000]; beta:[0.01,0.1]
ER-GNN	budget:[10,100]; d:[0.05,0.5,5.0]; sampler:[CM]

F. Appendix:Additional Experiment Result

In this section, we present additional experimental results.

F.1. Difference between Inductive and Transductive

In this subsection, we provide the additional results on the remaining dataset for the difference between inductive and transductive settings. The results are reported in Fig. 8 and Fig. 9.



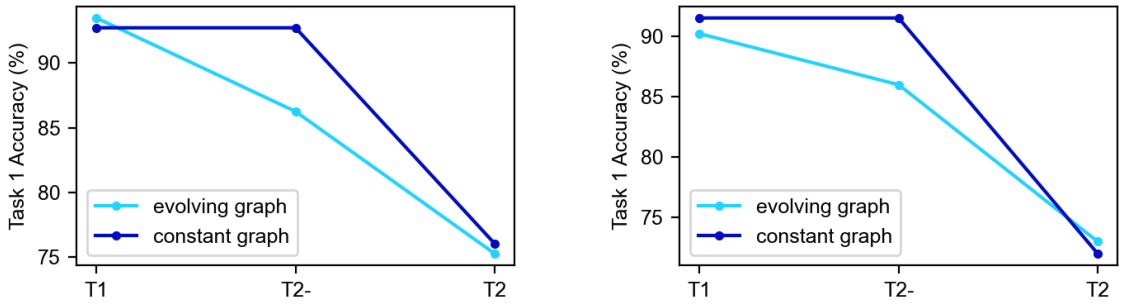
(a) Reddit, Inductive (b) Reddit, Transductive

(c) CoraFull, Inductive (d) CoraFull, Transductive

Figure 8. Performance Matrix of Bare Model on Transductive and Inductive Setting on CoraFull and Reddit Datasets.

F.2. Incremental Learning w/w.o. SSRM

In this subsection, we provide the additional results on the remaining dataset for the effect of SSRM on the complete learning dynamic. The results are reported in Fig. 11.



(a) CoraFull, Inductive

(b) Reddit, Transductive

Figure 9. Evolution of Task 1 performance.

G. Additional Related Work

G.1. Incremental Learning

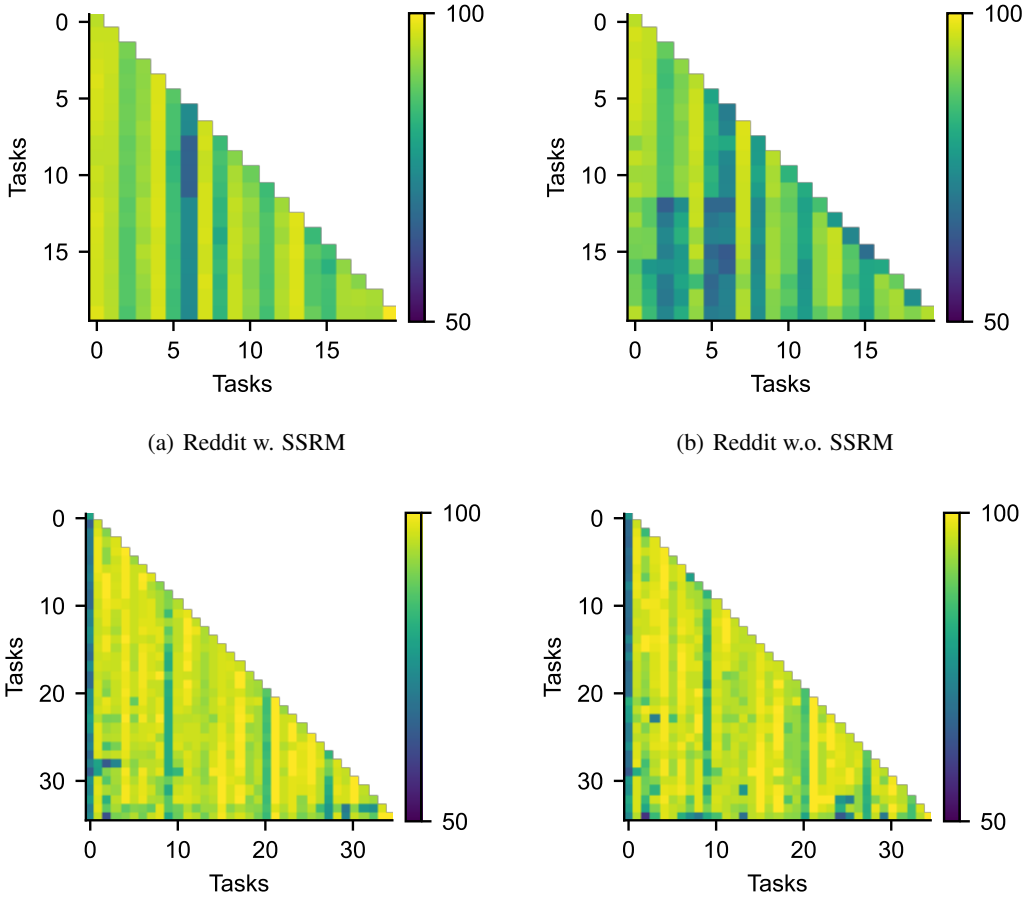
Incremental learning, also known as continual or lifelong learning, has gained increasing attention in recent years and has been extensively explored on Euclidean data. We refer readers to the surveys (Delange et al., 2021; Parisi et al., 2019; Biesialska et al., 2020) for a more comprehensive review of these works. The primary challenge of incremental learning is to address the catastrophic forgetting problem, which refers to the drastic degradation in a model's performance on previous tasks after being trained on new tasks.

Existing approaches for addressing this problem can be broadly categorized into three types: regularization-based methods, experience-replay-based methods, and parameter-isolation-based methods. Regularization-based methods aim to maintain the model's performance on previous tasks by penalizing large changes in the model parameters (Jung et al., 2016; Li & Hoiem, 2017; Kirkpatrick et al., 2017; Farajtabar et al., 2020; Saha et al., 2021). Parameter-isolation-based methods prevent drastic changes to the parameters that are important for previous tasks by continually introducing new parameters for new tasks (Rusu et al., 2016; Yoon et al., 2017; 2019; Wortsman et al., 2020; Wu et al., 2019). Experience-replay-based methods select a set of representative data from previous tasks, which are used to retrain the model with the new task data to prevent forgetting (Lopez-Paz & Ranzato, 2017; Shin et al., 2017; Aljundi et al., 2019; Caccia et al., 2020; Chrysakis & Moens, 2020; Knoblauch et al., 2020).

Our proposed method, SSRM, is a novel regularization-based technique that addresses the unique challenge of the structural shift in the inductive NGIL. Unlike existing regularization methods, which focus on minimizing the effect of updates from new tasks, SSRM aims to minimize the impact of the structural shift on the generalization of the model on the existing tasks by finding a latent space where the impact of the structural shift is minimized. This is achieved by minimizing the distance between the representations of vertices in the previous and current graph structures through the inclusion of a structural shift mitigation term in the learning objective. It is important to note that SSRM should not be used as a standalone method to overcome catastrophic forgetting but should be used as an additional regularizer to improve performance when there is a structural shift.

G.2. Graph Incremental Learning

Recently, there has been a surge of interest in GIL due to its practical significance in various applications (Wang et al., 2022; Xu et al., 2020; Daruna et al., 2021; Kou et al., 2020; Ahrabian et al., 2021; Cai et al., 2022; Wang et al., 2020a; Liu et al., 2021; Zhang et al., 2021; Zhou & Cao, 2021; Carta et al., 2021; Zhang et al., 2022; Kim et al., 2022; Tan et al., 2022; Su & Wu, 2023). However, most existing works in NGIL focus on a transductive setting, where the entire graph structure is known before performing the task or where the inter-connection between different tasks is ignored. In contrast, the inductive NGIL problem, where the graph structure evolves as new tasks are introduced, is less explored and lacks a clear problem formulation and theoretical understanding. There is currently a gap in understanding the inductive NGIL problem, which our work aims to address. In this paper, we highlight the importance of addressing the structural shift and propose a novel regularization-based technique, SSRM, to mitigate the impact of the structural shift on the inductive NGIL problem. Our work lays down a solid foundation for future research in this area.



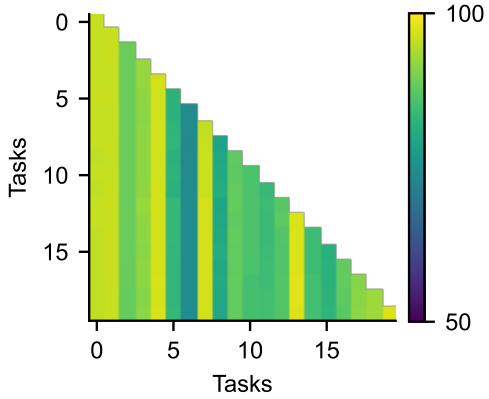
(a) Reddit w. SSRM

(b) Reddit w.o. SSRM

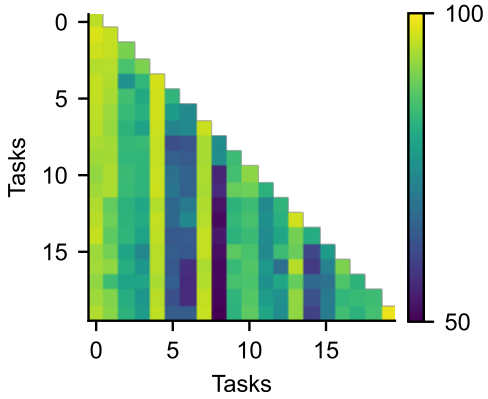
(c) CoraFull w. SSRM

(d) CoraFull, w.o. SSRM

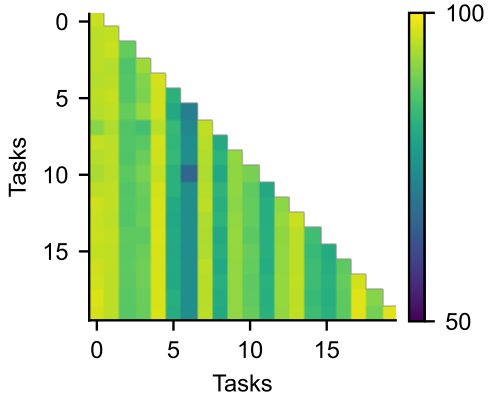
Figure 10. Performance Matrix of ER-GNN with SSRM and without SSRM on CoraFull and Reddit Datasets.



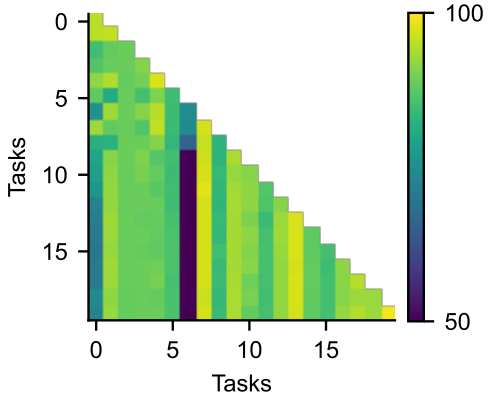
(a) Arxiv w. SSRM



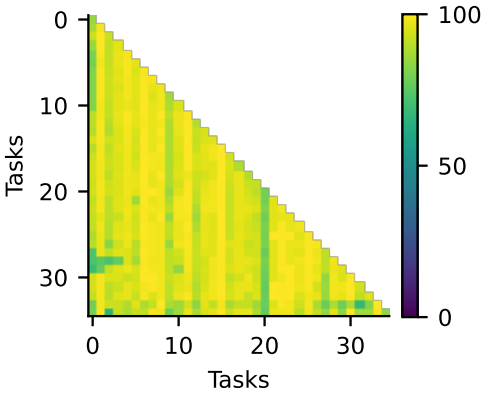
(b) Arxiv w.o. SSRM



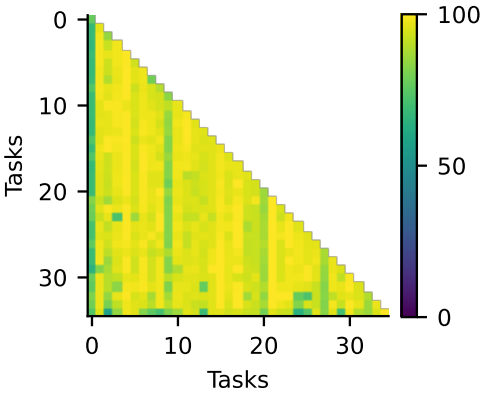
(c) Reddit w. SSRM



(d) Reddit w.o. SSRM



(e) CoraFull w. SSRM



(f) CoraFull, w.o. SSRM

Figure 11. Performance Matrix of TWP with SSRM and without SSRM on Arxiv, CoraFull and Reddit Datasets.

Finally, it is important to note that there is another area of research known as dynamic graph learning (Galke et al., 2021; Wang et al., 2020b; Han et al., 2020; Yu et al., 2018; Nguyen et al., 2018; Ma et al., 2020; Feng et al., 2020; Bielak et al., 2022), which focuses on enabling GNNs to capture the changing graph structures. The goal of dynamic graph learning is to capture the temporal dynamics of the graph into the representation vectors, while having access to all previous information. In contrast, GIL addresses the problem of catastrophic forgetting, in which the model's performance on previous tasks degrades after learning new tasks. For evaluation, a dynamic graph learning algorithm is only tested on the latest data, while GIL models are also evaluated on past data. Therefore, dynamic graph learning and GIL are two independent research directions with different focuses and should be considered separately.