

Exploring Rationale Learning for Continual Graph Learning

Lei Song¹, Jiaxing Li¹, Qinghua Si¹, Shihan Guan¹, Youyong Kong^{1, 2*}

¹Jiangsu Provincial Joint International Research Laboratory of Medical Information Processing,
School of Computer Science and Engineering, Southeast University

²Key Laboratory of New Generation Artificial Intelligence Technology and Its Interdisciplinary Applications (Southeast University), Ministry of Education, China

{230238577, jiaxing_li, 220225771, 230228507, kongyouyong}@seu.edu.cn

Abstract

Catastrophic forgetting poses a significant challenge for graph neural networks in continuously updating their knowledge base with data streams. To address this issue, much of the research has focused on node-level continual learning using parameter regularization or rehearsal-based strategies, while little attention given to graph-level tasks. Furthermore, current paradigms for continual graph learning may inadvertently capture spurious correlations for specific tasks through shortcuts, thereby exacerbating the forgetting of previous knowledge when new tasks are introduced. To tackle these challenges, we propose a novel paradigm, **Rationale Learning GNN (RL-GNN)**, for graph-level continual graph learning. Specifically, we harness the invariant learning principle to incorporate environmental interventions into both the current and historical distributions, aiming to uncover rationales by minimizing empirical risk across all environments. The rationale serves as the sole factor guiding the learning process. Therefore, continual graph learning is redefined as capturing these invariant rationales within task sequences, alleviating catastrophic forgetting caused by spurious features. Extensive experiments on real-world datasets with varying task lengths demonstrate the effectiveness of our RL-GNN in continuous knowledge assimilation and reduction of catastrophic forgetting.

Introduction

Modern graph neural network (GNN) algorithms have demonstrated exceptional performance in specialized tasks such as recommendation systems (Wu et al. 2022b), social network analysis (Li et al. 2023), and drug discovery (Hu et al. 2023), etc. However, they are criticized for their difficulty in retaining previously acquired knowledge when adapting to a new task, a phenomenon referred to as catastrophic forgetting (CF). One critical issue arises from the distinct data distributions across different tasks. Without targeted interventions, model parameters are inclined to adapt predominantly to the latest distribution to minimize empirical risk, complicating their compatibility with previous distributions. Recent studies have explored strategies such as parameter regularization (Liu, Yang, and Wang 2021), memory replay (also known as rehearsal-based methods) (Zhou

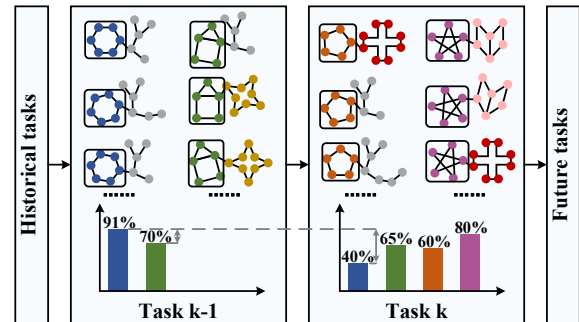


Figure 1: An example of catastrophic forgetting induced by spurious correlations between labels and environments. Each graph consists of a rationale (marked with a black box) and an environment (the remaining portion). From left to right, it represents a sequence of tasks generated over time. Our objective is to distinguish all previously encountered classes, with each graph’s label determined solely by the corresponding rationale. The histogram at the bottom depicts the prediction rationale accuracy for each class.

and Cao 2021; Zhang, Song, and Tao 2022c), and parameter isolation (Zhang, Song, and Tao 2022b) to mitigate CF. Among them, rehearsal-based methods demonstrate superior performance, which involve sampling a limited subset of historical data and feeding it into a fixed-size memory buffer. Upon encountering a new task, these samples are incorporated into a new collection to strike a trade-off between plasticity and stability.

Despite the commendable results achieved by these methodologies, they predominantly concentrate on node-level tasks, neglecting graph-level tasks. In reality, much of real-world data is inherently organized as independent graphs. For instance, drug discovery models require ongoing updates to the systemic knowledge to pinpoint target molecules, while medical agents must constantly integrate new information on disease or virus structures to ensure precise and reliable diagnostics (Tian, Zhang, and Dai 2024). To bridge this gap, our focus lies in addressing the challenges posed by graph-level continual graph learning (CGL). Rather than segregating past experiences and new tasks to safeguard their independence, this work aims to

*Corresponding Author.

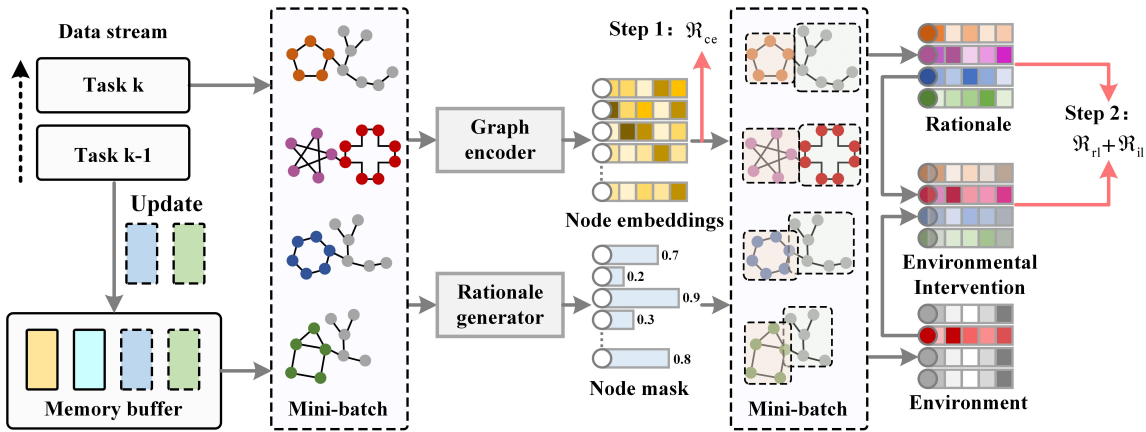


Figure 2: An overview of the RL-GNN framework. The memory buffer \mathcal{M} stores historical data from tasks $\mathcal{T}_{1:k-1}$. Upon the arrival of a new task \mathcal{T}_k , mini-batch samples are drawn from both the current task and \mathcal{M} , then processed through the graph encoder $\Phi(\cdot)$ and the rationale generator $g(\cdot)$. Best viewed in color.

delve deeper into the following pivotal question:

Is all information within the graph pertinent to our needs, and how can we extract the maximally informative subset from the graph structure at hand?

To answer this question, let’s delve into an example from Figure 1. If blue hexagons (the label’s maximal unique solution, also referred to as rationale) frequently co-occur with gray subtrees (environmental noise) in task \mathcal{T}_{k-1} , a classification model minimizing cross-entropy loss will erroneously associate the label with the environment through shortcuts (Wei et al. 2023). Within the same task, green houses linked to gray subtrees are susceptible to misjudgment, resulting in lower accuracy in predicting the category. Upon the arrival of task \mathcal{T}_k , if some orange pentagons are also linked to gray subtrees, previously acquired spurious correlations will become ineffective. This necessitates the model to restructure existing knowledge to prioritize learning more discriminative features, exacerbating CF and significantly diminishing performance on previously encountered categories. To put it another way, if the model can acquire rationales (maximally informative subsets) for diverse categories in each task, it will minimize perturbations to existing knowledge when adapting to new tasks. Furthermore, existing empirical knowledge can enhance its capacity to assimilate new information.

To tackle these challenges, inspired by recent research on out-of-distribution (OOD) generalization (Wu et al. 2022c; Li et al. 2022; Yu, Liang, and He 2023), we incorporate the invariant learning principle into CGL. Our goal is to discern rationales from graphs intertwined with environmental factors, thereby mitigating the forgetting of previously acquired knowledge through rationale learning. However, applying invariant learning to CGL is non-trivial for two reasons: (1) Existing invariant learning frameworks are often arduous to train, posing significant challenges for CGL; (2) Using cross-entropy loss as the optimization objective could induce models to exhibit recency effect (Zhao et al. 2020). Hence, we propose a novel graph-level CGL paradigm grounded in invariant learning, as illustrated

in Figure 2. The graph encoder $\Phi(\cdot)$ derives node representations for each graph, offering dependable features to the rationale generator $g(\cdot)$. The rationale generator $g(\cdot)$ assigns a scalar mask score (ranging from 0 to 1) to each node, distinguishing rationales from environmental factors. During training, the graph encoder $\Phi(\cdot)$ and the entire framework $f(\cdot)$ update parameters through alternating backpropagation with distinct optimization objectives. During inference, class prototypes computed from the memory buffer \mathcal{M} guide classification.

Our contributions manifest in three folds:

- We propose a novel graph-level CGL paradigm, **Rationale Learning GNN (RL-GNN)**, which incorporates the invariant learning principle into class-incremental learning, redefining CGL as capturing invariant rationales within task sequences. To the best of our knowledge, this is the pioneering effort to apply invariant learning on CGL.
- To tackle issues (1) and (2), we propose an alternating training regimen alongside a supervised contrastive invariant learning objective. The aim is to better identify and learn rationales to alleviate CF.
- Extensive experiments on three real-world datasets with varying task lengths validate that our RL-GNN exhibits stronger resilience against CF. Visualization results underscore the efficacy of invariant learning for CGL.

Related Work

Rehearsal-based Continual Graph Learning

Continual graph learning has recently been widely studied due to its capability in counteracting CF. Prevailing methodologies encompass constraining parameter updates via parameter estimation or knowledge distillation (parameter regularization), replaying historical distributions using a fixed-size memory buffer (memory replay), and dynamically extending the model architecture as required (parameter isolation). Among these, memory replay, or rehearsal-based methods, exhibit the most robust performance. A seminal

work in (Zhou and Cao 2021) introduced three experience selection strategies based on complementary learning systems theory. Subsequently, a series of CGL algorithms expanded upon it, such as sparsified subgraph memory (Zhang, Song, and Tao 2022c), condensed graph memory (Liu, Qiu, and Huang 2023), and topology-aware embedding memory (Zhang et al. 2024). Unfortunately, these efforts consistently prioritized node-level CGL, overlooking the crucial role of graph-level CGL in practical applications. To bridge this gap, our work falls within the family of rehearsal-based methods and proposes a new paradigm for graph-level CGL aimed at combating the well-known CF.

Invariant Learning on Graphs

Invariant learning has been extensively harnessed for OOD generalization, with the intent of extending the efficacy of GNNs trained in specific environments to perform robustly in unseen ones. A common practice involves exploring the causal mechanism (Pearl, Glymour, and Jewell 2016) underlying data generation and employing risk extrapolation (Krueger et al. 2021; Wu et al. 2022a) to ensure stability across varying environments. However, acquiring environmental labels for graphs is non-trivial due to the intertwined nature of environments with rationales, rendering them unobservable. Furthermore, the restricted variability across environments poses challenges for invariant learning. To this end, existing approaches (Wu et al. 2022c; Li et al. 2022; Yang et al. 2022; Yu, Liang, and He 2023) employed distribution interventions to generate multiple virtual environments, mimicking distributional shifts happening in real-world scenarios. Nevertheless, they were all geared towards static models, implying that more enduring knowledge could not be incrementally acquired over time. Indeed, data evolves progressively, evidenced by the continual exploration of new molecular structures and the emergence of nascent topic communities. Our work is tailored for task streams, aimed at leveraging invariant learning to sustainably update existing knowledge repositories, ameliorating CF.

Preliminaries

Before delving into the details of our RL-GNN, we initiate with the definition of class-incremental CGL, followed by the formalization of invariant learning principle on CGL.

Problem Formulation

Continual graph learning assumes receiving a sequence of tasks $\mathcal{T} = \{\mathcal{T}_1, \mathcal{T}_2, \dots, \mathcal{T}_K\}$ ($|\mathcal{T}| = K$) that are successively fed into the framework $f(\cdot)$ for learning. In this paper, each task $\mathcal{T}_k = \{\mathcal{D}_k^{tr}, \mathcal{D}_k^{te}\} \in \mathcal{T}$ is defined as a graph-level classification task, with \mathcal{D}_k^{tr} for training and \mathcal{D}_k^{te} for testing. Each graph $G_k^j \in \mathcal{G}_k$ corresponds to a class label $y^l \in \mathcal{Y}_k$, where $\mathcal{Y}_k = \{y^1, y^2, \dots, y^{c_k}\}$ is the label set and c_k denotes the number of classes in task \mathcal{T}_k . In the class-incremental learning scenario, $\mathcal{Y}_m \cap \mathcal{Y}_n = \emptyset$ if $m \neq n$. Due to memory constraints and privacy concerns, historical data is often concealed when learning a new task. Alternatively, a fixed-size memory buffer \mathcal{M} is utilized to preserve a few histor-

ical samples to replay their distributions. Upon completing training on task \mathcal{T}_k , the GNN is mandated to differentiate between all classes encountered in previous tasks $\mathcal{T}_{1:k}$. Below, we omit superscripts for brevity. From a probabilistic view, it can be interpreted with Bayes' theorem (Kirkpatrick et al. 2017) as:

$$\log p(\Theta|\mathcal{D}_{1:k}) = \log p(\mathcal{D}_k|\Theta) + \log p(\Theta|\mathcal{D}_{1:k-1}) - \log p(\mathcal{D}_k), \quad (1)$$

where $\log p(\mathcal{D}_k|\Theta)$ is the log likelihood for task \mathcal{T}_k , $\log p(\Theta|\mathcal{D}_{1:k-1})$ represents the posterior probability of Θ (parameterized representation of $f(\cdot)$), and $\log p(\mathcal{D}_k)$ denotes the evidence. To ensure the efficacy of CGL, it is imperative that all knowledge from tasks $\mathcal{T}_{1:k-1}$ is assimilated into the posterior distribution. Rehearsal-based methods leverage memory replay to keep its reliability. Consequently, the optimization objective can be formulated as:

$$\min \mathfrak{R}_{\text{MR}} = \mathfrak{R}(f(\mathcal{G}_k), \mathcal{Y}_k) + \lambda \mathfrak{R}(f(\mathcal{G}_{1:k-1}|\mathcal{M}), \mathcal{Y}_{1:k-1}), \quad (2)$$

where the two terms on the right-hand side compute the risk associated with the current task and the memory buffer, respectively. λ balances between the two.

Invariant Learning Principle on CGL

Let's begin with the OOD generalization problem, which is pervasive in contexts where discrepancies exist between training and testing distributions. Recent researches ascribe it to the data generation process, highlighting the biased distributions influenced by an underlying environment variable e . Invariant learning principle involves finding an optimal mapping $f^*(\cdot) : \mathbb{G} \rightarrow \mathbb{Y}$ that minimizes the worst-case risk, formulated as:

$$f^*(\cdot) = \arg \min_f \max_{e \in \mathcal{E}} \mathbb{E}_{(x,y) \sim p(x,y|e=e)} [l(f(x), y)|e], \quad (3)$$

where \mathcal{E} represents the support set of environments, $l(\cdot, \cdot)$ denotes the loss function. $\mathbb{E}_{(x,y) \sim p(x,y|e=e)} [l(f(x), y)|e]$ is defined as the risk function $\mathfrak{R}(f(\mathcal{X}), \mathcal{Y}|e)$ according to (Yang et al. 2022; Yu, Liang, and He 2023). Instead of focusing on the divergences between training and testing distributions within static data, our work prioritizes the distribution shifts both within and across tasks in CGL, which means that invariant learning pervades the entire sequence of tasks, aiming to discern stable rationales across diverse environments for mitigating CF. Hence, Eq. 3 can be reformulated as:

$$f^*(\cdot) = \arg \min_f \max_{e \in \mathcal{E}} \mathfrak{R}(f(\mathcal{G}_k), \mathcal{Y}_k|e) + \mathfrak{R}(f(\mathcal{G}_{1:k-1}|\mathcal{M}), \mathcal{Y}_{1:k-1}|e). \quad (4)$$

Methodology

In this section, we delve into the internal workings of the RL-GNN framework, comprising a graph encoder and a rationale generator, geared towards decoupling rationales and environments in each task. Moreover, we propose a supervised contrastive invariant learning objective grounded in the constructed virtual environments, alternately trained with cross-entropy loss. During inference, class prototypes are employed for prediction to avoid the recency effect.

Rationale-Environment Decoupling

As outlined in the Introduction, *not all information within the graph is conducive to CGL*. Our focus lies in grasping rationales, rather than inadvertently fostering spurious correlations with environments. To achieve this goal, we propose an invariant learning framework that disentangles rationales from environments, leveraging a graph encoder $\Phi(\cdot)$ and a rationale generator $g(\cdot)$. Given a graph $G_k^j = \{\mathbf{A}_k^j, \mathbf{X}_k^j\} \in \mathcal{G}_k$, where $\mathbf{A}_k^j \in \mathbb{R}^{N \times N}$ is the binary adjacency matrix and $\mathbf{X}_k^j \in \mathbb{R}^{N \times C}$ is the node feature matrix, the graph encoder $\Phi(\cdot)$ derives the node embedding $\mathbf{Z}_k^j \in \mathbb{R}^{N \times d}$ via iterative message passing as:

$$\begin{aligned} \mathbf{Z}_k^j &= \text{GNN}_c^m(\dots \text{GNN}_c^{(2)}(\text{GNN}_c^{(1)}(\mathbf{X}_k^j, \mathbf{A}_k^j), \mathbf{A}_k^j), \dots, \mathbf{A}_k^j) \\ &= \text{GNN}_c^m(G_k^j), \end{aligned} \quad (5)$$

where m is the number of layers in the model. Following (Li et al. 2022), we suppose that graph G_k^j comprises an invariant rationale $G_k^{j,(r)}$ and an environment $G_k^{j,(e)}$, where $G_k^{j,(r)} \cap G_k^{j,(e)} = \emptyset$ while $G_k^{j,(r)} \cup G_k^{j,(e)} = G_k^j$. The rationale generator $g(\cdot)$ is crafted to extract $G_k^{j,(r)}$ from G_k^j , mitigating the risk of the GNN spuriously correlating $G_k^{j,(e)}$ with its label in a specific task environment. More specifically, $g(\cdot)$ computes a scalar mask score $\mathbf{S}_k^{j,(r)}[i]$ (ranging from 0 to 1) for each node, indicating whether it should be filtered out. If $\mathbf{S}_k^{j,(r)}[i]$ is below 0.5, it is deemed an environment node; otherwise, as a rationale node. Ideally, scores for environment nodes approach 0 and for rationale nodes tend towards 1. The formal formulation is as follows:

$$\mathbf{S}_k^{j,(r)} = \text{Sigmoid}(\text{MLP}_g(\text{GNN}_g^m(G_k^j))), \quad (6)$$

$$\mathbf{S}_k^{j,(e)} = \mathbf{1}_N - \mathbf{S}_k^{j,(r)}, \quad (7)$$

where $\text{GNN}_g^m(\cdot)$ shares its structure with $\text{GNN}_c^m(\cdot)$ but differs in parameters, $\text{MLP}_g(\cdot)$ stands for a multilayer perceptron, and $\mathbf{1}_N \in \mathbb{R}^N$ is filled with ones. Then, $\mathbf{S}_k^{j,(r)}$ and $\mathbf{S}_k^{j,(e)}$ are utilized for computing the representations of the rationale and the environment as:

$$\mathbf{H}_k^{j,(r)} = \text{READOUT}(\mathbf{Z}_k^j \odot \mathbf{S}_k^{j,(r)}), \quad (8)$$

$$\mathbf{H}_k^{j,(e)} = \text{READOUT}(\mathbf{Z}_k^j \odot \mathbf{S}_k^{j,(e)}), \quad (9)$$

where \odot is the element-wise product, and $\text{READOUT}(\cdot)$ aggregates rationale and environment nodes separately using a permutation-invariant operation to derive graph-level embeddings. Subsequently, we can harness the rationales and environments for invariant learning.

Virtual Environment Construction

As demonstrated in Eq. 4, the goal of invariant learning on CGL is to solve a bi-level optimization problem, which entails acquiring stable embeddings (also called rationales) that exhibit robust performance across diverse tasks. As acquiring environment labels for graphs poses challenges, the

crux for this problem lies in how to effectively build virtual environments to simulate distribution shifts observed in the real world. Following (Liu et al. 2022), we leverage environment replacement to infer distinct environments within the latent space, ensuring sufficient potential environments for each sample within a mini-batch. Assuming a memory bank $\mathbf{H}_k^{(\mathcal{E})} = \{\mathbf{H}_{1:k-1|\mathcal{M}}^{1,(e)}, \dots, \mathbf{H}_{1:k-1|\mathcal{M}}^{p,(e)}, \mathbf{H}_k^{1,(e)}, \dots, \mathbf{H}_k^{q,(e)}\}$ storing all the potential environments, where p and q denote the number of samples from the memory buffer \mathcal{M} and the current task \mathcal{T}_k within a mini-batch, respectively. We combine $\mathbf{H}_k^{(\mathcal{R})}$ (the complement of $\mathbf{H}_k^{(\mathcal{E})}$) with any $\mathbf{H}_k^{(\mathcal{E})}[i]$ ($i \in [1, p+q]$) in pairs. Taking sample $\mathbf{H}_k^{1,(r)} \in \mathbf{H}_k^{(\mathcal{R})}$ as an illustration:

$$\mathbf{H}_k^1 = \{\mathbf{H}_k^{1,(r)} + \mathbf{H}_{1:k-1|\mathcal{M}}^{1,(e)}, \dots, \mathbf{H}_k^{1,(r)} + \mathbf{H}_k^{q,(e)}\}. \quad (10)$$

Each rationale is contextualized across $p+q$ distinct environments, thereby yielding $(p+q)^2$ latent embeddings within a mini-batch. It facilitates invariant learning over diverse tasks without imposing substantial computational overhead.

Contrastive Invariant Learning

Existing literatures (Wu et al. 2022c; Liu et al. 2022; Li et al. 2022; Yu, Liang, and He 2023) on OOD generalization commonly employ cross-entropy loss to solve Eq. 3 within static datasets. However, cross-entropy can potentially induce pronounced recency bias during inference in the context of CGL, given that the sample volume from new tasks often far exceeds that of old-class samples in the memory buffer \mathcal{M} . This suggests that historical samples are prone to being misclassified as new classes, exacerbating the phenomenon of CF. To address this issue, we propose a novel contrastive invariant learning objective to solve Eq. 4, comprising an invariant learning term \mathfrak{R}_{il} and a relational learning term \mathfrak{R}_{rl} . The primary challenge lies in how to adequately train the GNN to achieve a profound perception of the rationales after moving beyond the constraint of cross-entropy loss. Having acquired multiple rationales across diverse environments, we strive for robust consistency in the predictions of these samples according to Eq. 4. Hence, we compute as follows:

$$\begin{aligned} \mathfrak{R}_{\text{il}} &= \mathbb{E}_{G \in \{\mathcal{G}_{1:k-1|\mathcal{M}} \cup \mathcal{G}_k\}} \\ &\left[\frac{1}{p+q} \sum_{i=1}^{p+q} (\text{sg}[\mathbf{H}^{(r)}] - \text{MLP}_1(\mathbf{H}[i]))^2 \right], \end{aligned} \quad (11)$$

where the index of the task is omitted for simplicity, G is an instance from the training set of task \mathcal{T}_k or memory buffer \mathcal{M} , $\mathbf{H}^{(r)}$ signifies the rationale, while \mathbf{H} denotes the constructed virtual environments. $\text{sg}[\cdot]$ stops gradient backpropagation to avert model collapse (Grill et al. 2020). By minimizing the empirical risk \mathfrak{R}_{il} , our RL-GNN adeptly acquires stable knowledge in the novel task, concurrently reinforcing previous experiences for alleviating CF. However, since Eq. 11 does not account for graph category information, the learned knowledge may not necessarily represent the maximally informative subset for prediction. For instance, when infants learn to recognize animals, parents usually instruct

them on labels while simultaneously imparting essential attributes. Inspired by this insight, the relational learning term \mathfrak{R}_{rl} employs supervised contrastive learning to capture both intra-task and inter-task label dependencies as:

$$\mathfrak{R}_{\text{rl}} = \mathbb{E}_{G \in \{\mathcal{G}_{1:k-1|\mathcal{M}} \cup \mathcal{G}_k\}} \left[\frac{-1}{|\mathcal{P}_i|} \sum_{j \in \mathcal{P}_i} \log \left(\frac{\exp(\text{MLP}_2(\mathbf{H}^i) \cdot \text{MLP}_2(\mathbf{H}^j) / \tau)}{\sum_{u \neq i} \exp(\text{MLP}_2(\mathbf{H}^i) \cdot \text{MLP}_2(\mathbf{H}^u) / \tau)} \right) \right], \quad (12)$$

where $i \in \{1, 2, \dots, 2(p+q)\}$, as each instance in the mini-batch corresponds to a rationale and a stochastically selected virtual environment, $\mathcal{P}_i = \{j \in \{1, 2, \dots, 2(p+q)\} | y_j = y_i, j \neq i\}$, and τ is the temperature coefficient. Of particular note, our RL-GNN does not necessitate predefined data augmentation to achieve diverse views, leveraging the rationales and constructed virtual environments as paired positive samples. Moreover, to mitigate the risk of excessively sparse or dense scores in the generated node masks, a regularization term is introduced to penalize the rationale generator $g(\cdot)$ as follows:

$$\mathfrak{R}_{\text{reg}} = \mathbb{E}_{G \in \{\mathcal{G}_{1:k-1|\mathcal{M}} \cup \mathcal{G}_k\}} \left[\left| \sum_{i=1}^N \mathbf{S}^{(r)}[i] / N - \gamma \right| \right], \quad (13)$$

where γ functions as a hyperparameter that delimits the scope of rationales, a smaller value indicates a preference for a narrower range.

Risk Function Optimization for Training

Thus far, the risk function for training is delineated as:

$$\mathfrak{R}_{\text{RL-GNN}} = \mathfrak{R}_{\text{rl}} + \alpha \cdot \mathfrak{R}_{\text{il}} + \beta \cdot \mathfrak{R}_{\text{reg}}, \quad (14)$$

where α and β are hyperparameters calibrated to equilibrate the contributions of the invariant learning term \mathfrak{R}_{il} and the regularization term $\mathfrak{R}_{\text{reg}}$. However, as (Chang et al. 2020) suggests, the node mask scores generated by $g(\cdot)$ align with $\Phi(\cdot)$'s node embeddings to discern rationales and environments, forming an intrinsic cooperative relationship. Hence, we propose a novel alternating training paradigm with two steps as follows:

- **Step 1:** Ensuring the fidelity of node embeddings derived from the graph encoder $\Phi(\cdot)$ is imperative for rationale identification. We endeavor for $\Phi(\cdot)$ to acquire reasonable initialization parameters attuned to categorical nuances, thus advancing further learning of the rationale generator $g(\cdot)$. In implementation, we focus solely on minimizing the cross-entropy risk \mathfrak{R}_{ce} to update $\Phi(\cdot)$ as:

$$\mathfrak{R}_{\text{ce}} = \mathbb{E}_{G \in \{\mathcal{G}_{1:k-1|\mathcal{M}} \cup \mathcal{G}_k\}} [-y \cdot \log(\omega(\text{READOUT}(\mathbf{Z})))] \quad (15)$$

where $\omega(\cdot)$ is an extra auxiliary classifier deployed exclusively for this step.

- **Step 2:** Minimizing $\mathfrak{R}_{\text{RL-GNN}}$ to update the entire RL-GNN, including the graph encoder $\Phi(\cdot)$.

Step 1 and Step 2 alternate throughout the training process for CGL. Algorithm 1 delineates the complete training process of our RL-GNN.

Algorithm 1: Rationale Learning for Continual Graph Learning

Input: Task stream $\{\{\mathbf{A}_k^j, \mathbf{X}_k^j\}_{j=1}^K\}$, disjoint class sets $\{\mathcal{Y}_k\}_{k=1}^K$, Memory buffer \mathcal{M} , replay size p , batch size q , number of epochs E , learning rate η .

Initialize: The graph encoder $\Phi(\cdot)$, the rationale generator $g(\cdot)$, the auxiliary classifier $\omega(\cdot)$, the mappings $\text{MLP}_1(\cdot)$ and $\text{MLP}_2(\cdot)$.

```

1: for  $k = 1, 2, \dots, K$  do
2:   Retrieve the training data  $\{\mathcal{D}_k^{tr} \cup \mathcal{M}\}$  for the current task  $\mathcal{T}_k$ 
3:   for  $epoch = 1, 2, \dots, E$  do
4:     Draw a mini-batch  $\{\{\mathbf{A}^j, \mathbf{X}^j\}_{j=1}^{p+q}\}$  with  $q$  examples from  $\mathcal{D}_k^{tr}$  and  $p$  from  $\mathcal{M}$ 
5:     for all  $j \in \{1, 2, \dots, p+q\}$  do
6:        $\mathbf{Z}^j \leftarrow \Phi(\mathbf{A}^j, \mathbf{X}^j)$ 
7:        $\mathbf{S}^{j,(r)}, \mathbf{S}^{j,(e)} \leftarrow g(\mathbf{A}^j, \mathbf{X}^j), 1_N - g(\mathbf{A}^j, \mathbf{X}^j)$ 
8:       Decouple  $\mathbf{H}^{j,(r)}$  and  $\mathbf{H}^{j,(e)}$ 
9:       Construct virtual environments  $\mathbf{H}^j$ 
10:    end for
11:    if  $epoch \% 2 == 1$  then
12:      Compute  $\mathfrak{R}_{\text{ce}}$  according to Eq. 15
13:      Step 1: Update  $\theta \leftarrow \{\Phi, \omega\}$  by  $\theta \leftarrow \theta - \eta \nabla_{\theta} \mathfrak{R}_{\text{ce}}$ 
14:    else
15:      Compute  $\mathfrak{R}_{\text{RL-GNN}}$  according to Eq. 14
16:      Step 2: Update  $f \leftarrow \{\Phi, g, \text{MLP}_1, \text{MLP}_2\}$  by  $f \leftarrow f - \eta \nabla_f \mathfrak{R}_{\text{RL-GNN}}$ 
17:    end if
18:  end for
19:  Draw a fixed number of instances from each class in  $\mathcal{D}_k^{tr}$  and deposit them into  $\mathcal{M}$ .
20: end for

```

Class Prototypes for Inference

Given that RL-GNN updates via the contrastive learning objective rather than cross-entropy, direct inference of test instances across diverse tasks is unfeasible. Inspired by nearest class mean (NCM) classifier (Mai et al. 2021), we compute class prototypes from the memory buffer \mathcal{M} to classify all previously encountered tasks as follows:

$$\mu_c = \frac{1}{n_c} \sum_{j=1}^{n_c} \mathbf{H}_{1:k|\mathcal{M}}^{j,(r)} \cdot \mathbb{1}\{y_j = c\}, \quad (16)$$

$$\tilde{y} = \underset{c \in \mathcal{Y}_{1:k}}{\text{argmax}} \text{sim}(\mathbf{H}^{(r)}, \mu_c), \quad (17)$$

where $\mathbb{1}\{\cdot\}$ is an indicator that returns 1 if the condition is met, otherwise 0, $\text{sim}(\cdot, \cdot)$ denotes a similarity measure or a negative distance function.

Experiments

Next, we empirically investigate the following questions:

- **Q1:** Does RL-GNN demonstrate stronger resistance to CF over existing CGL approaches?
- **Q2:** Is RL-GNN sensitive to hyperparameters such as α , β and γ ?

Method	Aromaticity-CL (T=15)		REDDIT-MULTI-12K-CL (T=5)		ENZYMES-CL (T=3)	
	AP/%	AF/%	AP/%	AF/%	AP/%	AF/%
Joint-train	83.18±2.18	-	45.42±2.72	-	64.00±3.18	-
Fine-tune	5.31±1.25	-96.15±2.81	14.18±1.22	-77.54±2.86	23.33±2.24	-51.50±7.43
GEM	7.28±2.08	-69.54±5.17	15.75±1.49	-77.04±3.28	24.17±3.35	-44.25±10.00
EWC	11.16±0.50	-59.83±4.25	14.78±2.15	-76.28±2.87	23.83±3.17	-44.50±9.14
LWF	7.45±1.69	-95.91±1.68	15.15±1.85	-77.79±4.56	24.17±1.86	-53.50±7.00
MAS	8.37±1.87	-82.49±2.85	13.50±1.56	-71.71±2.92	24.33±2.38	-47.25±7.02
TWP	5.89±2.00	-67.91±5.70	13.13±0.95	-65.84±4.34	21.50±3.29	-53.75±9.10
ER-GNN	37.54±4.49	14.00±4.39	22.60±4.66	-32.06±4.36	23.17±2.29	-31.50±7.26
ER-GS-LS*	45.40±1.50	17.60±1.20	-	-	25.50±1.70	-20.00±5.10
RL-GNN ($n_c=10$)	56.00±7.85	-9.31±6.40	23.04±3.79	-18.57±5.06	26.17±5.43	-28.50±8.15
RL-GNN ($n_c=100$)	68.24±5.68	-4.95±3.31	29.73±2.00	-10.57±3.90	30.50±5.77	-26.25±8.96

Table 1: Performance comparison across three different datasets, where T denotes task length, * indicates results cited from the publication, and n_c is the number of graphs allocated to the memory buffer \mathcal{M} for each class, with the best results highlighted in bold.

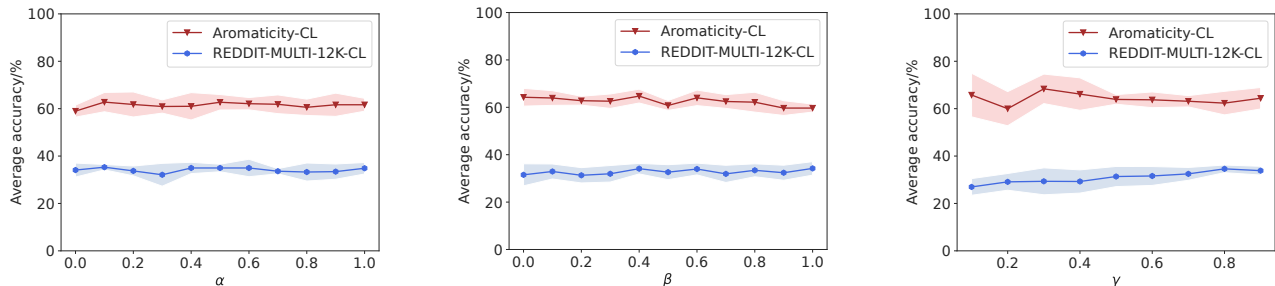


Figure 3: Sensitivity analysis of α , β , and γ on Aromaticity-CL and REDDIT-MULTI-12K-CL. Assessing a specific parameter necessitates maintaining the other hyperparameters constant.

- **Q3:** Can RL-GNN really differentiate between rationales and environments for CGL?

Experimental Setup

Datasets. To answer the aforementioned questions, we carry out experiments on three real-world datasets: Aromaticity (Xiong et al. 2019), REDDIT-MULTI-12K (Yanardag and Vishwanathan 2015), and ENZYMES (Borgwardt et al. 2005). Following (Zhang, Song, and Tao 2022a), we divide each dataset into 2-way graph classification tasks, with each category being stratified into training/validation/testing subsets in accordance with 8/1/1 ratio, yielding three task streams suffixed with "-CL" for CGL. More details regarding the datasets and task configurations can be found in Appendix A.1.

Baselines. For a rigorous comparison of performance across various methods, all experiments are conducted following Continual Graph Learning Benchmark (CGLB) (Zhang, Song, and Tao 2022a), where EWC (Kirkpatrick et al. 2017), MAS (Aljundi et al. 2018), GEM (Lopez-Paz and Ranzato 2017), TWP (Liu, Yang, and Wang 2021), and LWF (Li and Hoiem 2017) are regularization-based methods, ER-GNN (Zhou and Cao 2021) employs a rehearsal strategy, and Fine-tune and Joint-train denote the lower and upper bounds of CGL performance. Given that ER-GNN is tailored for node-level tasks, we re-implement it for appli-

cation on graph-level tasks. Moreover, ER-GS-LS (Hoang et al. 2023), serving as an ensemble method, is also included in our baselines.

Metrics. To gauge the performance of the baselines and our RL-GNN on CGL, we adopt average performance (AP) and average forgetting (AF) as metrics for the plasticity and stability, respectively. Formal definitions are provided in Appendix A.2. Conceptually, a robust method should exhibit superior performance across both metrics.

Environment and Hyperparameters. All experiments are implemented on the PyTorch 3.10 framework, with an NVIDIA 3090 GPU. Following (Zhang, Song, and Tao 2022a), we adhere to default settings for the baselines. As the source code for ER-GS-LS is not publicly available, the values in Table 1 are cited from the publication. Moreover, considering that CGL performance is intricately linked to certain hyperparameters such as batch size and training epochs, we maintain consistency across all experiments. Detailed settings are documented in Appendix A.3.

Comparative Study for Q1

Table 1 presents our comparison results on three datasets with varying task lengths. The visualization of the performance matrices and learning dynamics is available in Appendix B. It is evident that directly fine-tuning GNNs

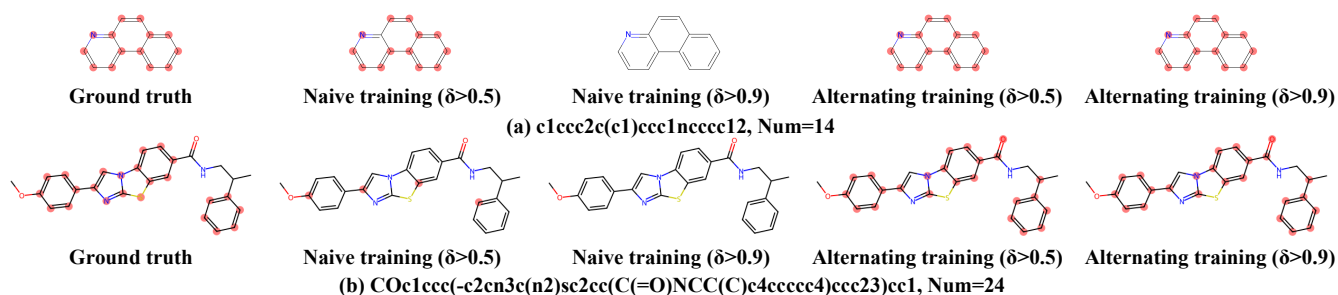


Figure 4: Visualization graphs of two molecules in Aromaticity-CL, where Num is the number of aromatic atoms in each molecule, and δ denotes a threshold used to filter atoms with lower mask scores. Two training strategies are compared.

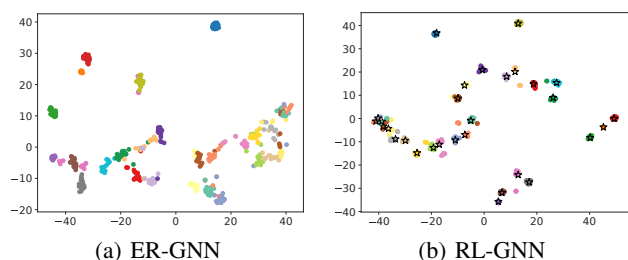


Figure 5: TSNE visualization on Aromaticity-CL reports ER-GNN and our RL-GNN, where different colors indicate different classes, and the star symbols in the right panel denote class prototypes.

results in pronounced degradation of retained knowledge. Regularization-based methods exhibit lower AF, underscoring the formidable challenge of mitigating CF in class-incremental learning scenario through existing parameter regularization strategies. Rehearsal-based methods (e.g., ER-GNN) demonstrate substantial superiority over regularization-based approaches regarding AP and AF, particularly on Aromaticity-CL and REDDIT-MULTI-12K-CL, which verifies the efficacy of memory replay in bolstering both plasticity and stability. ER-GS-LS, which combines memory replay and knowledge distillation, achieves elevated performance, indicating a synergistic relationship among different CGL techniques. Furthermore, while displaying higher AF values compared to RL-GNN, it shows lower AP, especially on Aromaticity-CL. This trade-off sacrifices plasticity for stability, resulting in inflated AF. Our RL-GNN excels in harmonizing both aspects and attains exceptional CGL performance with low replay costs. Increasing n_c leads to better results, suggesting that the CGL performance is correlated with the representativeness of samples in the memory buffer \mathcal{M} , which is not the focus of our work.

Sensitivity Study for Q2

To further explore the complexities of parameter tuning, we conduct sensitivity analysis on α , β , and γ without loss of generality. Figure 3 demonstrates that RL-GNN is insensitive to α and β , exhibiting consistent trends across various datasets. γ showcases divergent distributions on the two

datasets, implying the discrepancies in rationales underlying different types of graphs. A reasonable regularization term can aid in identifying rationales, thus improving the performance of CGL. Furthermore, the fluctuations on REDDIT-MULTI-12K-CL are less pronounced than on Aromaticity-CL, owing to the former having a larger number of instances for training. Our contrastive invariant learning necessitates more negative samples to some extent to achieve stability in performance.

Case Analysis for Q3

Ensuring that RL-GNN can really identify rationales is pivotal to our work. Hence, we casually select two cases from Aromaticity-CL to illustrate their node masks. As shown in Figure 4, we take a threshold δ to exclude atoms with lower mask scores, highlighting the aromatic atoms identified as rationales with red circles. We compare two training strategies, with naive training focusing solely on minimizing $\mathfrak{R}_{\text{RL-GNN}}$. Following the integration of initialization parameters with class information, our alternating training paradigm produces more accurate and more stable predictions across various thresholds. This suggests that RL-GNN can effectively discern the functional groups (rationales) within each molecule through the node mask scores. Moreover, Figure 5 visualizes the molecular graph embeddings, observing that our method generates a more compact embedding space across the task sequence compared to ER-GNN, with representations of diverse classes closely clustered around their corresponding class prototypes.

Conclusion

In this paper, we proposed a novel graph-level CGL paradigm, RL-GNN, which, to the best of our knowledge, was the first attempt to introduce the invariant learning principle into this domain. To be specific, we perturbed the instances in the current task and memory buffer through environment replacement, then achieved awareness of rationales by minimizing a supervised contrastive invariant learning risk, thus alleviating CF. Extensive experiments demonstrated that our RL-GNN exhibited strong resistance to CF. In the future, we will further investigate the applications of CGL grounded in invariant rationales across diverse domains, such as neurodegenerative disease diagnosis and financial risk assessment.

Acknowledgments

The work is supported in part by the National Natural Science Foundation of China under Grant No. 62471133, the Central University Basic Research Fund of China under Grant No. 2242024K40020, and the Big Data Computing Center of Southeast University.

References

- Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European conference on computer vision (ECCV)*, 139–154.
- Borgwardt, K. M.; Ong, C. S.; Schönauer, S.; Vishwanathan, S.; Smola, A. J.; and Kriegel, H.-P. 2005. Protein function prediction via graph kernels. *Bioinformatics*, 21(suppl-1): i47–i56.
- Chang, S.; Zhang, Y.; Yu, M.; and Jaakkola, T. 2020. Invariant rationalization. In *International Conference on Machine Learning*, 1448–1458. PMLR.
- Grill, J.-B.; Strub, F.; Altché, F.; Tallec, C.; Richemond, P.; Buchatskaya, E.; Doersch, C.; Avila Pires, B.; Guo, Z.; Gheshlaghi Azar, M.; et al. 2020. Bootstrap your own latent—a new approach to self-supervised learning. *Advances in neural information processing systems*, 33: 21271–21284.
- Hoang, T. D.; Tung, D. V.; Nguyen, D.-H.; Nguyen, B.-S.; Nguyen, H. H.; and Le, H. 2023. Universal graph continual learning. *arXiv preprint arXiv:2308.13982*.
- Hu, W.; Liu, Y.; Chen, X.; Chai, W.; Chen, H.; Wang, H.; and Wang, G. 2023. Deep learning methods for small molecule drug discovery: a survey. *IEEE Transactions on Artificial Intelligence*, 5(2): 459–479.
- Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13): 3521–3526.
- Krueger, D.; Caballero, E.; Jacobsen, J.-H.; Zhang, A.; Binias, J.; Zhang, D.; Le Priol, R.; and Courville, A. 2021. Out-of-distribution generalization via risk extrapolation (rex). In *International conference on machine learning*, 5815–5826. PMLR.
- Li, H.; Zhang, Z.; Wang, X.; and Zhu, W. 2022. Learning invariant graph representations for out-of-distribution generalization. *Advances in Neural Information Processing Systems*, 35: 11828–11841.
- Li, X.; Sun, L.; Ling, M.; and Peng, Y. 2023. A survey of graph neural network based recommendation in social networks. *Neurocomputing*, 549: 126441.
- Li, Z.; and Hoiem, D. 2017. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12): 2935–2947.
- Liu, G.; Zhao, T.; Xu, J.; Luo, T.; and Jiang, M. 2022. Graph rationalization with environment-based augmentations. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 1069–1078.
- Liu, H.; Yang, Y.; and Wang, X. 2021. Overcoming catastrophic forgetting in graph neural networks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, 8653–8661.
- Liu, Y.; Qiu, R.; and Huang, Z. 2023. Cat: Balanced continual graph learning with graph condensation. In *2023 IEEE International Conference on Data Mining (ICDM)*, 1157–1162. IEEE.
- Lopez-Paz, D.; and Ranzato, M. 2017. Gradient episodic memory for continual learning. *Advances in neural information processing systems*, 30.
- Mai, Z.; Li, R.; Kim, H.; and Sanner, S. 2021. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 3589–3599.
- Pearl, J.; Glymour, M.; and Jewell, N. P. 2016. *Causal inference in statistics: A primer*. John Wiley & Sons.
- Tian, Z.; Zhang, D.; and Dai, H.-N. 2024. Continual Learning on Graphs: A Survey. *arXiv preprint arXiv:2402.06330*.
- Wei, Y.; Ye, J.; Huang, Z.; Zhang, J.; and Shan, H. 2023. Online prototype learning for online continual learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 18764–18774.
- Wu, Q.; Zhang, H.; Yan, J.; and Wipf, D. 2022a. Handling distribution shifts on graphs: An invariance perspective. *arXiv preprint arXiv:2202.02466*.
- Wu, S.; Sun, F.; Zhang, W.; Xie, X.; and Cui, B. 2022b. Graph neural networks in recommender systems: a survey. *ACM Computing Surveys*, 55(5): 1–37.
- Wu, Y.-X.; Wang, X.; Zhang, A.; He, X.; and Chua, T.-S. 2022c. Discovering invariant rationales for graph neural networks. *arXiv preprint arXiv:2201.12872*.
- Xiong, Z.; Wang, D.; Liu, X.; Zhong, F.; Wan, X.; Li, X.; Li, Z.; Luo, X.; Chen, K.; Jiang, H.; et al. 2019. Pushing the boundaries of molecular representation for drug discovery with the graph attention mechanism. *Journal of medicinal chemistry*, 63(16): 8749–8760.
- Yanardag, P.; and Vishwanathan, S. 2015. Deep graph kernels. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, 1365–1374.
- Yang, N.; Zeng, K.; Wu, Q.; Jia, X.; and Yan, J. 2022. Learning substructure invariance for out-of-distribution molecular representations. *Advances in Neural Information Processing Systems*, 35: 12964–12978.
- Yu, J.; Liang, J.; and He, R. 2023. Mind the label shift of augmentation-based graph ood generalization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 11620–11630.
- Zhang, X.; Song, D.; Chen, Y.; and Tao, D. 2024. Topology-aware Embedding Memory for Learning on Expanding Graphs. *arXiv preprint arXiv:2401.13200*.
- Zhang, X.; Song, D.; and Tao, D. 2022a. Cglb: Benchmark tasks for continual graph learning. *Advances in Neural Information Processing Systems*, 35: 13006–13021.

Zhang, X.; Song, D.; and Tao, D. 2022b. Hierarchical prototype networks for continual graph representation learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(4): 4622–4636.

Zhang, X.; Song, D.; and Tao, D. 2022c. Sparsified subgraph memory for continual graph representation learning. In *2022 IEEE International Conference on Data Mining (ICDM)*, 1335–1340. IEEE.

Zhao, B.; Xiao, X.; Gan, G.; Zhang, B.; and Xia, S.-T. 2020. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 13208–13217.

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