GOTHAM: Graph Class Incremental Learning Framework under Weak Supervision

Anonymous authors
Paper under double-blind review

Abstract

Graphs are growing rapidly and so are the number of different categories associated with it. Applications like e-commerce, healthcare, recommendation systems, and various social media platforms are rapidly moving towards graph representation of data due to their ability to capture both structural and attribute information. One crucial task in graph analysis is node classification, where unlabeled nodes are categorized into predefined classes. In practice, novel classes appear incrementally sometimes with just a few labels (seen classes) or even without any labels (unseen classes), either because they are new or haven't been explored much. Traditional methods assume abundant labeled data for training, which isn't always feasible. We investigate a broader objective: *Graph Class Incremental Learning under Weak Supervision (GCL)*, addressing this challenge by meta-training on base classes with limited labeled instances. During the incremental streams, novel classes can have few-shot or zero-shot representation. Our proposed framework GOTHAM efficiently accommodates these unlabeled nodes by finding the closest prototype representation, serving as class representatives in the attribute space. For Text-Attributed Graphs (TAGs), our framework additionally incorporates semantic information to enhance the representation. By employing teacher-student knowledge distillation to mitigate forgetting, GOTHAM achieves promising results across various tasks. Experiments on datasets such as Cora-ML, Amazon, and OBGN-Arxiv showcase the effectiveness of our approach in handling evolving graph data under limited supervision. The code implementation is available here: `https://shorturl.at/2VCEc`

1 Introduction

Graph-structured data are ubiquitously used in many real-world applications, such as citation graphs (Cummings & Nassar, 2020; Tang et al., 2008), biomedical graphs (Subramanian et al., 2005; Zhai et al., 2023), circuit optimization (Shahane et al., 2023; Hakhamaneshi et al., 2023) and social networks (Qi et al., 2012). Recently, Graph Neural Networks (GNNs) have been proposed (Cao et al., 2016; Subramanian et al., 2005; Henaff et al., 2015; Xu et al., 2019; Kipf & Welling, 2017; Veličković et al., 2018; Hamilton et al., 2017) to model graph-structured data by leveraging the structural and attributed information along the graph. As a central task in graph machine learning, node classification (Wang et al., 2020b; Xhonneux et al., 2020; Wang et al., 2021b; Zhu et al., 2021) has achieved remarkable progress with the rise of GNNs. While these methods concentrate on static graphs to classify unlabeled nodes into predetermined classes, real-world graphs are dynamic. In practice, graphs grow (You et al., 2022; Lu et al., 2022; Tan et al., 2022) rapidly incorporating nodes and edges belonging to novel classes incrementally. For example, (1) think of a biomedical graph. Each node represents a rare disease category, and edges show how these diseases relate to each other. As new disease categories emerge, they are gradually added to this graph. Such methods significantly aid the drug discovery process. (2) For food delivery systems nodes correspond to various zip codes and the edges indicate the spatial distances between them. As the company expands its reach, new zip codes are systematically incorporated into this graph, facilitating efficient supply chain management. GNNs typically require a large amount of labeled data to learn effective node representations (Ding et al., 2020b; Zhou et al., 2019b). In practice, catching up with these newly emerging classes is tough, and obtaining extensive labeled data for each class is even harder. The annotation process can be extremely time-consuming and expensive (Ding et al., 2022; Guo et al., 2021; Wang et al., 2022c). Naturally, it becomes crucial to empower models to classify the nodes from: limited labeled classes and those unseen classes having no labeled instances, collectively referred to as *weakly supervised.* In this regard, we investigate the problem of *Graph Class Incremental Learning under Weak Supervision (GCL).*

Recent studies (Lu et al., 2022; Tan et al., 2022), have delved into a specific aspect of the broader problem, termed graph few-shot class incremental learning (GFSCIL). This approach operates under the assumption that the base classes possess abundant labeled instances, while novel classes introduced during streaming sessions always have representations in the form of k-shots. Additionally, there is a separate line of research (Wang et al., 2021b; 2023b; Hanouti & Borgne, 2022), focusing on zero-shot node classification. Furthermore, addressing the issue of limited labeled data availability during base training, which impacts the model's generalizability for better node representations during finetuning, is discussed in Wang et al.. Finally, studies like (Wang et al., 2023a; Wang et al.) address few-shot node classification. Graph data, existing in a non-Euclidean space with constantly changing network structures, poses unique challenges. Unlike the progress made in class incremental learning in computer vision, incremental learning in graphs remains relatively unexplored. Therefore, for developing a framework for *GCL* the key challenges include: *(1) Can the model learn good node representations with just k-shots for base training classes during finetuning? (2) Is there a universal framework to address both the GFSCIL problem (classes in novel streams represented by k-shots) and the GCL task (including classes with no training instances)? and finally, (3) How to prevent forgetting old knowledge while learning new information?*

Sometimes *"Less is Plenty"*, by heuristically sampling the neighborhood corresponding to the k-shot representations our approach extends the support set for each class. These prototypes serve a crucial role in steering the orientation of both base and novel classes within the graph during streaming sessions. We adopt the popular meta-learning strategy, called episodic learning (Finn et al., 2017), which has shown great promise in few-shot learning. We propose ***Graph Orientation Through Heuristics And Meta-learning (GOTHAM)***, an incremental learning framework that effectively addresses all the aforementioned issues. Finally, the teacher-student knowledge distillation in GOTHAM prevents catastrophic forgetting. The paper is structured into six sections, focusing on class orientation through prototypes, proposed approach, experimental analysis, and concluding remarks.

2 Related Work

Continual Graph Learning: The Continual Graph Learning Benchmark (CGLB) (ZHANG et al., 2022) categorizes tasks in evolving graph structures into Continual Graph Learning (CGL) (Wang et al., 2022a; Xu et al., 2020; Daruna et al., 2021; Ahrabian et al., 2021; Kou et al., 2020), Dynamic Graph Learning (DGL) (Galke et al., 2020; Wang et al., 2020a; Yu et al., 2018; Han et al., 2020), and Few-Shot Graph Learning (FSGL) (Zhou et al., 2019a; Guo et al., 2021; Yao et al., 2020). CGL focuses on mitigating catastrophic forgetting without relying on past data, DGL captures temporal dynamics with access to historical data, and FSGL enables rapid adaptation to new tasks using meta-learning. Our work lies at the intersection of CGL and FSGL. We examine related works in few-shot, zero-shot, and incremental learning for graph-based tasks, situating our contributions within this broader graph learning framework.

Few-Shot Node Classification: Despite several advancements in applying GNNs to node classification tasks (Kipf & Welling, 2017; Veličković et al., 2018; Hamilton et al., 2017; Wang et al., 2022b), more recently, many studies (Ding et al., 2020b; Wang et al., 2021a; Zhou et al., 2019b; Wang et al.) have shown that the performance of the GNNs is severely affected when number of labeled instances are limited. Consequently, there has been a surge in interest in the area of few-shot node classification. These works are broadly categorized into two main streams: (1) Optimization based approaches (Zhou et al., 2019b; Huang & Zitnik, 2020; Liu et al., 2021; Lan et al., 2020) and (2) Metric based approaches (Wang et al., 2023a; Wang et al.; Snell et al., 2017a; Yao et al., 2020). These approaches operate under the strong assumption that information for all classes is available simultaneously, which renders them *ineffective for class incremental learning scenarios.*

Zero-Shot Classification: As emerging classes continue to grow in dynamic environments, interest in a related field called "no-data learning" is surging. However, the existing approaches (Wang et al., 2021b; 2023b;

Hanouti & Borgne, 2022; Lu et al., 2018; Wan et al., 2019b; Song et al., 2018) suffer from two key limitations: (1) Many of these methods assume access to unlabeled instances of unseen classes during training, limiting their generalizability and (2) They typically only classify test instances into the set of unseen classes, which isn't practical. In computer vision (Verma et al., 2019; Wu et al., 2023), some approaches have addressed these issues and even integrated incremental learning successfully. However, *similar advancements in the graph domain are lacking.*

Class Incremental Learning: also known as lifelong learning has been extensively studied across various computer-vision tasks (Li & Hoiem, 2018; Rebuffi et al., 2016; Hou et al., 2019). However, these approaches often assume access to extensive labeled datasets during streaming sessions, which is impractical. Few-shot class incremental learning (FSCIL) has been introduced in the realms of image classification in (Tao et al., 2020; Cheraghian et al., 2021). Unlike images, graph data exhibits non-i.i.d characteristics, making incremental learning more challenging. Most recent works (Lu et al., 2022; Tan et al., 2022) have addressed the graph few-shot class incremental learning framework. However, a common but naive assumption in these approaches is the *abundant availability of base classes, which often isn't the case in practice.* Our proposed framework aims to bridge the gap by directly addressing the limitations found in various existing works.

3 Methodology

In this section, we begin by presenting the problem and explaining the key terms related to it. Then, we introduce some foundational concepts that will help build our formulation. Finally, we outline several crucial modules and provide detailed explanations for each.

3.1 Problem Statement

We denote an attributed graph as $G^t(V^t, E^t, X^t)$, where $V^t = \{v_1^t, v_2^t, \ldots, v_n^t\}$ is the vertex set and $E^t \subseteq V^t \times V^t$ is the edge set. $X^t = \{x_1, x_2, \ldots, x_{|V^t|}\} \in \mathbb{R}^{|V^t| \times d}$, is the node feature matrix where d is the feature dimension. In the base training stage, we have a base graph G^{base} with $|C^{base}|$ number of classes. Due to weak supervision, the number of labeled samples corresponding to C^{base} is extremely limited. In the streaming sessions, evolving graphs are presented $\{G^1, G^2, \ldots, G^T\}$ with $\{C^1, C^2, \ldots, C^T\}$ sets of classes. In the GFSCIL framework, every streaming session introduces δC^i new classes, each represented by k-shots. It is essential to note that $\delta C^i \cap \delta C^j = \emptyset$ and $C^t = C^{\text{base}} + \sum_{i=1}^{t} \delta C^i$.

Problem definition: ***Graph Class Incremental Learning under weak supervision***

In each streaming session, G^t introduces δC^i new classes, which are divided into two categories: $\delta C^{i,f}$ and $\delta C^{i,z}$. $\delta C^{i,f}$ classes, termed as seen classes, have few training instances (typically k-shots), while $\delta C^{i,z}$ classes, referred to as unseen classes lack any training instances. During the streaming session, we encounter both $\delta C^{i,f}$ and $\delta C^{i,z}$ classes, forming the class set denoted as $C^t = C^{t,S} \cup C^{t,U}$, where S stands for seen classes and U denotes unseen classes at time "t". Specifically, $C^{t,S} = C^{base} + \sum_{i=1}^{t} \delta C^{i,f}$ and $C^{t,U} = \sum_{i=1}^{t} \delta C^{i,z}$. Additional information, in terms of class semantics descriptions (CSDs), is provided for all the classes. The CSD matrix is denoted as $A_s = \{a_{s1}, a_{s2}, \ldots, a_{sC^t}\} = A_s^S \cup A_s^U$, with each row containing description of a class. Class semantics descriptions (CSDs) have been extensively studied in Wang et al. (2023b); Hanouti & Borgne (2022); Wang et al. (2021b); Ju et al. (2023). Throughout the paper, we interchangeably refer to "class semantics descriptions (CSDs)" and "semantic attributes". The goal is to classify all the unlabeled nodes (belonging to both seen and unseen classes) into C^t classes encountered so far.

Labeled training instances are called "support sets" ($\mathcal{S}$), while unlabeled testing instances are termed "query sets" ($\mathcal{Q}$). Unseen classes, which lack training instances, have their unlabeled instances presented only during inference.

3.2 Preliminaries: Label smoothness and Poisson Learning (Random walk perspective)

To enhance classification accuracy in scenarios with extremely low labeled data, leveraging additional samples is crucial. Semi-supervised learning, which combines labeled and unlabeled data, has shown significant

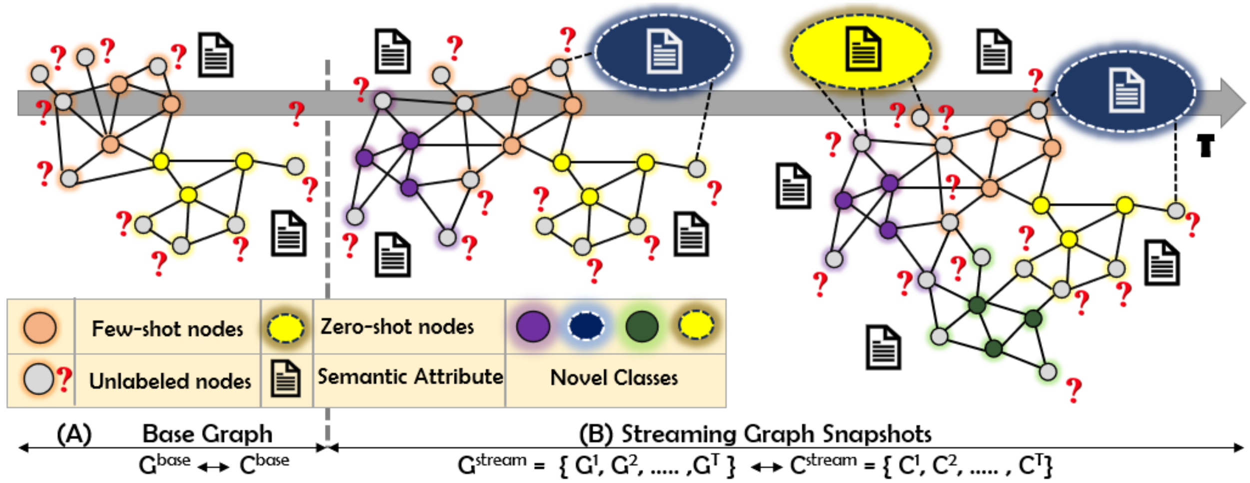


Figure 1: **Graph Class Incremental Learning under Weak Supervision**:(A) In the base graph G^{base}, the base classes C^{base} have extremely limited labeled instances.(B) In the streaming sessions, graph G^t has C^t number of classes. Depending upon the availability of the training instances, the classes are further classified as $C^{t,S}$ (seen classes) and $C^{t,U}$ (unseen classes). Seen classes are represented with k-shots, along with semantic attributes (CSDs). For unseen classes, only CSD information is available. The goal, is to classify the unlabeled instances into C^t classes encountered so far (Lu et al., 2022; Tan et al., 2022).

improvements by utilizing the topological structure of the data (Zhu et al., 2003; Zhou & Schölkopf, 2004; Zhou et al., 2003). Methods such as Poisson learning (Calder et al., 2020) have further extended the concept by incorporating structure-based information on graphs through random walks. The underlying assumption is that samples that are close to each other can potentially share similar classes. Previous research (Solomon et al., 2014; Belkin et al., 2006; Kalofolias, 2016), has emphasized the importance of the smoothness assumption for label propagation in scenarios with extremely low label rates. These findings form the basis of our proposed approach, which extends support sets through random walks without requiring extensive labeled nodes. This extended support set enhances the representation of prototypes for each class, leading to improved classification performance.

3.3 Prototype representation

Definition: The prototype of a class corresponds to a representative embedding vector which captures the overall characteristics of a class in the attribute space. Prototype representation has been extensively studied across various works (Snell et al., 2017b; Rebuffi et al., 2017; Lu et al., 2022; Tan et al., 2022) in the domain of few-shot representation learning.

The foundational works, (Snell et al., 2017b; Rebuffi et al., 2017) suggested using the mean of the support samples for prototype representation. Building upon this foundation, subsequent studies (Lu et al., 2022; Tan et al., 2022) introduced attention-based prototype generation techniques. These approaches were designed to address challenges such as class imbalance and mitigate biases arising from noisy support sets. Recent studies, (Wang et al.; 2023a) highlighted the significance of neighborhood sampling techniques, such as Poisson learning and Personalised Page Rank (PPR), for obtaining a more informed support set.

These approaches (Snell et al., 2017b; Rebuffi et al., 2017; Lu et al., 2022; Tan et al., 2022; Wang et al.; 2023a) suffer due to extremely weak supervision setting. Hence, we additionally leverage the label smoothness principle to gather the local neighborhood of the support nodes. The extended support set contains the labeled support nodes and the unlabeled neighbors gathered through random walks. The final node set for a class can be represented as $\mathcal{S}_{x,C} = \mathcal{S}_C \cup \mathcal{V}_C$, where $\mathcal{S}_C$, corresponds to nodes with labels and $\mathcal{V}_C$ is the sampled unlabeled node set for class C belonging to seen classes. The prototype thus obtained will be the

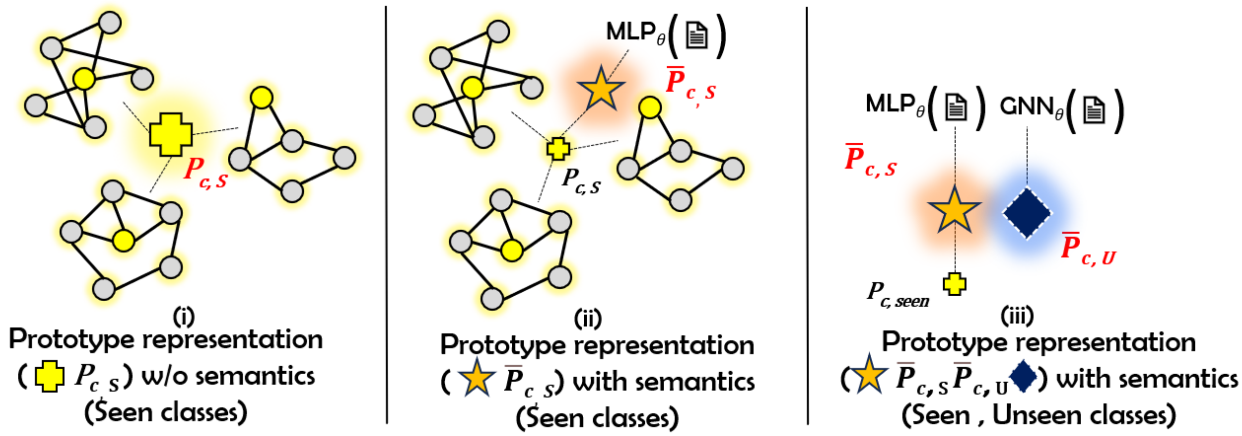


Figure 2: **Prototype representation**: For the GFSCIL task, we propose representing prototypes ($P_{c,S}$) using the averaged extended support set, as illustrated in (i). As demonstrated in (ii), we integrate semantic attributes (CSDs) to enhance the prototypes ($\overline{P}_{c,S}$) in TAGs. For GCL tasks with classes having no training instances, the semantic attributes (CSDs) are encoded as prototypes ($\overline{P}_{c,U}$).

average of the embeddings of all the nodes within the extended support set $\mathcal{S}_{x,C}$ represented as:

$$P_{C,S} := \frac{1}{|\mathcal{S}_{x,C}|} \sum_{i=1}^{|\mathcal{S}_{x,C}|} \mathrm{GNN}_\theta\left(v_i\right) \tag{1}$$

where, $\mathrm{GNN}_\theta\left(v_i\right)$ corresponds to embeddings of the node v_i, which are generated by aggregating information from its neighbors. Refer to Figure 2(i) for further details. In Text Attributed Graphs (TAGs), refer to Figure 2(ii), we enhance the prototype representation by incorporating semantic attributes associated with each class. The semantic loss, which will be elaborated upon later, facilitates the integration of attribute and semantic space. The encoded semantic attributes $\mathrm{MLP}_\phi\left(a_{sC}\right)$, for the same seen class C are merged with the original prototype to obtain the *new prototype representation*: $\bar{P}_{C,S} = \frac{P_{C,S}+\mathrm{MLP}_\phi(a_{sC})}{2}$. For the GFSCIL framework, the prototype set will consist of $P_S = \{P_{C,S}, \bar{P}_{C,S}\}$, depending on the type of graph used as input. For a class $\hat{C}$, with no training examples (where $\hat{C} \in C^{t,U}$), we rely on additional information in terms of semantic attributes. Refer to Figure 2(iii) for further insights into this representation. The prototype representation for such unseen classes is: $\bar{P}_{\hat{C},U} = \mathrm{GNN}_\theta\left(a_{s\hat{C}}\right)$, where $\mathrm{GNN}_\theta\left(a_{s\hat{C}}\right)$, represents the vector representation of the semantic attributes, where each attribute connects only to itself (self-loop) in its adjacency. In the GCL scenario, the prototype representation set is denoted as $\bar{P} = \{\bar{P}_{C,S}, \bar{P}_{\hat{C},U}\}$.

3.4 Transferable Metric Space Learning

Graphs constantly change, posing a challenge in how different node classes are positioned in metric space. As information for all classes isn't simultaneously available, it's crucial to establish criteria to prevent overlap between old and novel classes. Furthermore, representing novel classes under seen and unseen categories adds complexity to their representability, with some classes having limited samples and others only appearing during inference. To tackle this, a model is trained using the following loss functions:

Intra-class clustering loss: The goal here is to group instances that belong to the same class together. Several data augmentation strategies have been suggested previously (Li et al., 2018; Verma et al., 2020; Ding et al., 2018; Qiu et al., 2020), which generate consistent samples without affecting the semantic label. Despite their merits, we opt for a straightforward approach: sampling the neighborhood of the original node to obtain its correlated view. In our case, the original nodes correspond to the labeled k-shot representative nodes for each seen class. Neighbor nodes are sampled with the approach discussed in section: 3.2. The class prototype is responsible for grouping these nodes, employing the clustering loss defined as:

$$L_{cls,S} := \frac{1}{|C^{t,S}|} \sum_{j \in C^{t,S}} \left(\sum_{i \in n_j} \frac{\max(\|\mathrm{GNN}_\theta\,(v_i) - P_{j,S}\| - \gamma, 0)}{\sum_{k \in n_j} \max(\|\mathrm{GNN}_\theta\,(v_k) - P_{j,S}\| - \gamma, 0)} \right) \tag{2}$$

Here, $C^{t,S}$ refers to the set of seen classes encountered up to the current streaming session at a time "t". The number of samples for each class from the extended support set is represented by "n_j". The parameter "γ" defines the boundary from the prototype. Samples for a certain class are encouraged to stay within this boundary. The max(.) function ensures that only samples outside the boundary contribute to the loss.

Inter-class segregation loss: Unlike the augmentation strategies (Li et al., 2018; Verma et al., 2020; Ding et al., 2018; Qiu et al., 2020) employed to generate correlated pairs within a class, negative pairs are not explicitly sampled. Instead, we aim to increase dissimilarity among samples belonging to different classes. We achieve this by using representative embedding vectors for each class, referred to as class prototypes, to enhance separability among different classes. This prevents class overlap among all classes (C^t) encountered up to the current streaming session at a time "t". The segregation loss is defined as follows:

$$L_{seg} := \frac{-1}{|C^t|} \sum_{j \in C^t} \sum_{p \in C^t, p \neq j} \log \|\bar{P}_j - \bar{P}_p\| \tag{3}$$

Here, $C^t = \{C^{t,S} \cup C^{t,U}\}$ is the set of all classes, including both seen and unseen classes. Similarly, the prototypes belong to the prototype representation set $\bar{P} = \{\bar{P}_{C,S}, \bar{P}_{\hat{C},U}\}$. This loss function applies to both seen and unseen classes.

Semantic manipulation loss: Each modality offers a unique perspective on class representation, contributing to a more comprehensive view of prototypes. While extensively explored in the image domain (Zhang et al., 2023; Xing et al., 2019; Xu & Le, 2022; Guan et al., 2021), graphs provide an additional advantage by incorporating structural information (orientation) associated with each class within the graph. Class-semantic descriptors (CSDs) or semantic attributes, derived from class names and descriptions, are encoded and represented as $\mathrm{MLP}_\phi\,(a_{sC})$ for $C \in C^{t,S}$. The objective is to align the encoded semantics with the prototypes of the seen classes. The corresponding loss function is expressed as follows:

$$L_{sem,S} := \sum_{j \in C^{t,S}} \|\mathrm{MLP}_\phi\,(a_{sj}) - P_{j,S}\| \tag{4}$$

This loss function is responsible for integrating the attribute and semantic space. The newly learned semantic embeddings are later merged to obtain a new prototype representation (discussed previously). This loss function is specifically applied only to seen classes.

Knowledge refinement through experience: As the graph evolves incrementally, the learner model may tend to forget previously learned information when exposed to new knowledge, leading to catastrophic forgetting. To address this, it's crucial to preserve the previously acquired knowledge while integrating new information. This process is known as knowledge distillation. Among various techniques (Zhang et al., 2020; Rezayi et al., 2021; Feng et al., 2022), we opt for the teacher-student approach. The teacher model distills both attribute and semantic information using the following loss function:

$$L_{emb,S} = \frac{1}{n_{C^{(t-1),S}}} \sum_{i \in n_{C^{(t-1),S}}} \left\| \mathrm{GNN}_\theta^{teacher}\,(v_i) - \mathrm{GNN}_\theta^{student}\,(v_i) \right\| \tag{5}$$

$$L_{align,S} = \frac{1}{|C^{(t-1),S}|} \sum_{j \in C^{(t-1),S}} \left(1 - \frac{\mathrm{MLP}_\phi^{teacher}\,(a_{sj}) \cdot \mathrm{MLP}_\phi^{student}\,(a_{sj})}{\left\|\mathrm{MLP}_\phi^{teacher}\,(a_{sj})\right\| \left\|\mathrm{MLP}_\phi^{student}\,(a_{sj})\right\|} \right) \tag{6}$$

The total loss is: $L_{KD,S} := \lambda_1 \cdot L_{emb,S} + \lambda_2 \cdot L_{align,S}$. For the GFSCIL problem, where text attributes are not available knowledge distillation is solely performed across the node-embeddings (attribute information). This loss function applies only to seen classes.

4 Proposed Algorithm

In this section, we present our proposed framework: ***Graph Orientation Through Heuristics And Meta-learning (GOTHAM)****. At any given time t, we have a graph G^t as input, where the total classes $C^t = C^{t,S} \cup C^{t,U}$ encompasses both seen (few-shot) and unseen (zero-shot) class representation. The choice of framework type depends on the input, as illustrated in Figure:3. Once the input is understood, the following procedure is employed to perform node classification:*

(1) Episodic Learning: Once the choice of framework is decided, tasks ($\mathcal{T}$) are sampled for the corresponding graph G^t. Each task $\mathcal{T}^i \sim p(\mathcal{T})$, drawn from the task distribution, consists of an extended support set ($\mathcal{S}_x^i$) and a query set ($\mathcal{Q}^i$) required for episodic learning. Episodic learning, which has demonstrated great promise in the area of few-shot learning (Rebuffi et al., 2017; Tan et al., 2022; Huang & Zitnik, 2020; Vinyals et al., 2016; Zhou et al., 2019b), involves sampling tasks and learning from them, rather than directly training and then fine-tuning over batches of data. **(2) Prototype representation:** For each support set ($\mathcal{S}_x^i$), prototypes are generated for all the classes. If the class set C^t contains samples only from the seen classes (i.e. $C^t = C^{t,S}$), the prototype set will be P_S and the problem becomes a GFSCIL setting. Furthermore, if TAGs are given as an input, which offer additional semantic attribute information, a new prototype representation. For the GCL setting where ($C^t = C^{t,S} \cup C^{t,U}$) the prototype representation set is denoted as $\bar{P} = \{\bar{P}_{C,S}, \bar{P}_{\hat{C},U}\}$. **(3) Meta-learning and Finetuning**: After obtaining prototypes, different loss functions are applied along with prototypes on the support sets for meta-learning. In the GFSCIL scenario without TAGs, the model is trained using clustering loss and separability loss. With TAGs in both GFSCIL and GCL settings, semantic loss is also incorporated. The corresponding meta-training loss is defined as: $L_{train} := \alpha_1 \cdot L_{cls,S} + \alpha_2 \cdot L_{seg} + \alpha_3 \cdot L_{sem,S}$. Meta-learning is performed on the base graph. Once the model is trained on the support set ($\mathcal{S}_x^i$), its performance is validated on the corresponding query set ($\mathcal{Q}^i$). The model is then frozen and used for meta-finetuning. During meta-finetuning, the loss is defined as: $L_{finetune} := \alpha_1 \cdot L_{cls,S} + \alpha_2 \cdot L_{seg} + \alpha_3 \cdot L_{sem,S} + \alpha_4 \cdot L_{KD,S}$. **(4) Knowledge distillation:** During finetuning, knowledge distillation preserves previously learned class representations. The corresponding loss is defined as: $L_{KD,S} := \lambda_1 \cdot L_{emb,S} + \lambda_2 \cdot L_{align,S}$. **(5) Node classification:** For any graph G^t as input at time t, the model ultimately performs C^t-way node classification.

Algorithm 1 GOTHAM III.o

```
Input: G^t, A_s, C^t
Output: Label prediction on query nodes in Q^i ∈ T^i
Initialise θ, φ; Sample T^i ~ p(T)
Base Training (t = 0, C^t = C^base)
    for T^i = {S_x^i ∪ Q^i} ∈ C^base:
        Prototype (P̄_{C^base,S}) using eq:1
        Compute L_train on S_x^i
        Obtain node labels for Q^i
        Update θ, φ using gradient descent
    end
    Freeze the trained model
Finetuning (t > 0, C^t = C^base + Σ_{i=1}^t δC^i)
Load pre-trained model; create student, teacher models for knowledge distillation
    for T^i = {S_x^i ∪ Q^i} ∈ C^t:
        Prototype (P̄_{C^t}) using eq:1
        Compute L_finetune on S_x^i
        Obtain node labels for Q^i
        Update θ, φ using gradient descent
    end
```

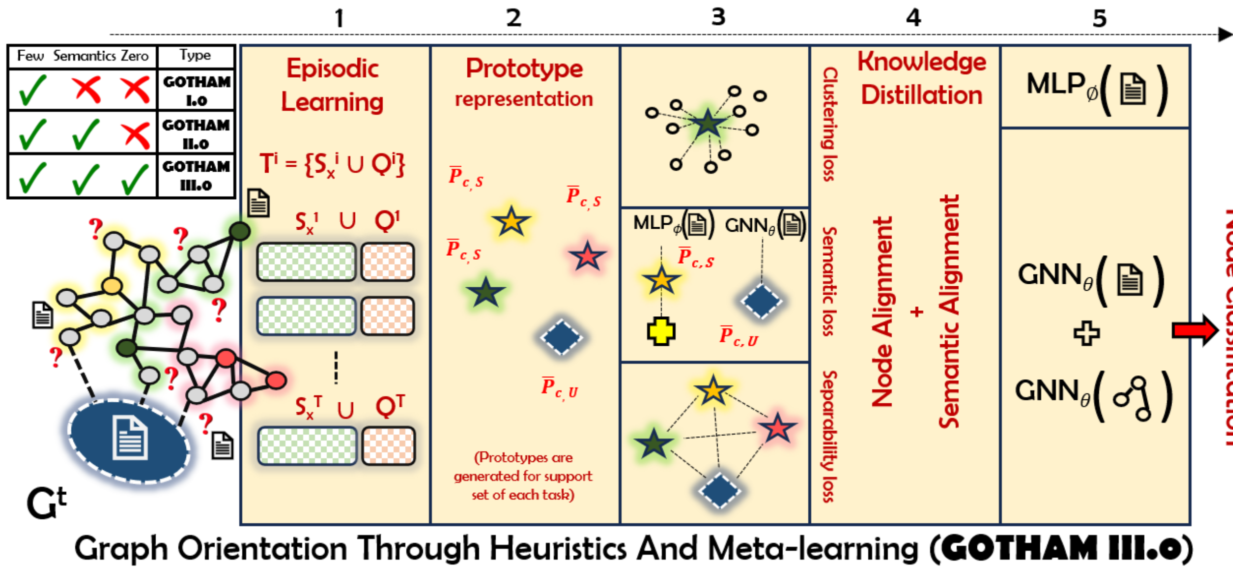


Figure 3: **GOTHAM III.o**: At any time t, the framework uses the graph G^t as input. The total classes are $C^t = C^{t,S} \cup C^{t,U}$. The steps are: (1) Create tasks (T) with support sets ($\mathcal{S}$) and query sets ($\mathcal{Q}$) for episodic learning. (2) Obtain prototype representations for each support set ($\mathcal{S}_x^i$). (3) Apply loss functions. (4) Use knowledge distillation to transfer knowledge from the teacher model to the student model. (5) Perform node classification.

5 Experiments

Datasets: We assess the performance of our proposed framework, GOTHAM, on three real-world datasets- Cora-ML, Amazon, and OBGN-Arxiv. We summarize the statistics of the datasets in Table 1. For more details about the dataset refer to the ***Appendix***.

Table 1: Statistics of datasets used in the experiments

Dataset	Nodes	Features	Classes	Class Labels	Tasks
Cora-ML	2,708	1,433	7	Neural Network, Rule Learning, Reinforcement Learning, Probabilistic Methods, Theory, Genetic Algorithms, Case-based	GFSCIL, GCL
Amazon	13,752	767	10	Label names Unavailable	GFSCIL
OBGN-Arxiv	169,343	128	40	Arxiv cs na, Arxiv cs mm, Arxiv cs lo, Arxiv cs cy, Arxiv cs cr, Arxiv cs dc, Arxiv cs hc, Arxiv cs cv, Arxiv cs ai, ...	GFSCIL, GCL

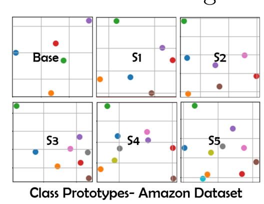
Experiment settings: We partition the dataset into base stage and multiple streaming sessions respectively. We assess our framework across two main problem settings: (1) Graph Few-shot Class Incremental Learning (GFSCIL) and (2) Graph Few-shot Class Incremental Learning under Weak Supervision (GCL). Cora-ML and OBGN-Arxiv are Text-Attributed Graphs (TAGs), enriched with semantic attributes. We generate semantic attributes/ Class Semantics Descriptors (CSDs) using "word2vec" (Mikolov et al., 2013), which transform textual descriptors into word embeddings. To simplify computation, we utilize Label-CSDs (Wang et al., 2021b). Initially, we evaluate all datasets under the GFSCIL setting. For the Cora-ML and Amazon dataset, we choose five classes as the novel classes and keep the rest as base classes, and adopt *1-way, 5-shot* setting, which means we have 6 sessions (1 base sessions + 5 novel sessions). For the OBGN-Arxiv dataset, we keep ten classes as base classes and the rest as novel, employing a *3-way, 10-shot* setting (totaling 11 sessions). Our framework seamlessly integrates semantic attribute information in Cora-ML and OBGN-Arxiv. Finally, we assess our framework under the GCL setting, focusing on Cora-ML and OBGN-Arxiv to demonstrate its effectiveness. During each streaming session, one class is designated as zero-shot, lacking training instances. Unlabeled instances for these classes are only available during inference. All the experiments are performed five times to ensure reproducibility. The top results are highlighted in **bold**, while the second best ones are underlined.

Baseline methods: In the GFSCIL setting, we benchmark our results against several state-of-the-art frameworks for few-shot class incremental learning and few-shot node classification, including: Meta-GNN (Zhou et al., 2019b), GPN (Ding et al., 2020a), iCaRL (Rebuffi et al., 2017), HAG-Meta (Tan et al., 2022), Geometer (Lu et al., 2022) and CPCA (Ren et al., 2023). Unlike previous methods, during base training, we provide only a limited number (*5- shots for Cora-ML and Amazon and 10-shots for OBGN-Arxiv*) of labeled instances for each class. In streaming sessions, novel classes receive k-shot representations. In the GCL setting, where novel classes have both few-shot and zero-shot representations, we compare against zero-shot learning frameworks with inductive learning as baselines. These approaches include DCDFL (741, 2024), GraphCEN (Ju et al., 2023), (CDVSc, BMVSc, WDVSc) (Wan et al., 2019a) and Random guess, introduced as a naive baseline. Unlike the traditional approach, the seen classes have only limited labeled instances (*5-shots for Cora-ML and 10-shots for OBGN-Arxiv*) available for training, and unseen classes have semantic attributes only. Under the GCL setting, the unlabeled instances will be classified into C^t classes encountered, resembling a *generalized zero-shot with inductive learning framework*. A detailed summary of the baseline methods and hyper-parameters employed is available in the ***Appendix***.

Graph Few-shot Class Incremental Learning (GFSCIL): We conducted experiments on the Amazon dataset, focusing on the GFSCIL problem as previously described. The results, detailed in Table 2, show an average improvement of around 6% across various streams. Visualizing the class prototypes generated by the GOTHAM framework reveals distinct separations among classes across streams, ensuring consistent performance.

Table 2: **GFSCIL setting:** (**Left**) Model performance (%) on the Amazon dataset under GFSCIL setting. **(Right)** Visualization of class prototypes for the Amazon dataset across different streaming sessions.

Amazon (1-way 5-shot GFSCIL setting)						
Stream	Base	S_1	S_2	S_3	S_4	S_5
Meta-GNN	**99.60**	86.33	82.43	77.75	70.82	67.94
GPN	93.56	85.23	74.88	73.40	66.17	63.36
iCaRL	66.20	47.33	39.13	35.75	29.84	29.66
HAG-Meta	95.43	88.76	75.67	69.56	67.21	61.86
GEOMETER	95.44	90.05	77.36	74.27	73.08	**74.36**
CPCA	95.37	87.88	83.72	77.13	76.37	69.32
GOTHAM I.o	96.61	**90.91**	**88.89**	**84.55**	**78.82**	73.81
%gain	**-03.00**	**00.00**	**06.17**	**08.74**	**03.20**	**00.00**



We extend our experiments to the Cora-ML and OBGN-Arxiv datasets, both Text-Attributed Graphs (TAGs) enriched with semantic attributes. Following the previously outlined experimental conditions, GOTHAM achieved an average improvement ranging from 6.4% to 13.5% over the baseline methods. We explored two variants of the framework: GOTHAM I.o, which solely relies on feature-based information, and GOTHAM II.o, which integrates semantic attributes. Table 3 demonstrates that incorporating semantic attributes in GOTHAM II.o notably enhances performance for both datasets.

Table 3: **GFSCIL with semantics:** Node classification accuracy (%) in the GFSCIL setting- leveraging semantic attributes for enhanced class representation on Cora-ML and OBGN-Arxiv datasets with GOTHAM.

Cora-ML (1-way 5-shot GFSCIL setting)						
Stream	Base	S_1	S_2	S_3	S_4	S_5
Meta-GNN	100	79.19	61.37	60.40	51.51	36.76
GPN	95.58	**91.89**	77.95	68.57	70.53	62.53
iCaRL	93.00	69.13	53.81	47.20	42.86	38.60
HAG-Meta	96.08	87.81	73.96	70.12	66.19	60.17
GEOMETER	96.46	89.91	77.58	70.20	54.50	62.76
CPCA	97.67	90.68	77.38	75.38	69.50	59.86
GOTHAM I.o	100	90.15	87.83	83.66	76.56	**75.03**
GOTHAM II.o	**100**	91.43	**88.69**	**84.00**	**76.92**	72.40
%gain	**00.00**	**00.00**	**13.78**	**11.43**	**09.06**	**19.55**

OBGN-Arxiv (3-way 10-shot GFSCIL setting)						
Stream	Base	S_1	S_2	S_6	S_9	S_{10}
Meta-GNN	76.60	66.10	57.38	36.55	29.77	28.82
GPN	78.38	68.21	57.88	35.77	28.78	30.12
iCaRL	62.80	39.54	35.22	21.97	15.68	16.45
HAG-Meta	77.17	68.19	58.22	37.13	28.28	24.68
GEOMETER	80.08	**70.68**	**61.07**	38.13	29.65	26.22
CPCA	69.71	56.96	50.39	33.35	25.76	24.88
GOTHAM I.o	72.33	59.94	47.84	30.31	25.12	25.44
GOTHAM II.o	**82.91**	70.20	60.26	**40.53**	**31.38**	**32.38**
%gain	**03.53**	**00.00**	**00.00**	**06.29**	**05.41**	**07.50**

Graph Class Incremental Learning under Weak Supervision (GCL): In a broader problem setting where novel classes have both few-shot and zero-shot representation, we conducted extensive experiments on the Cora-ML and OBGN-Arxiv datasets. The results in Table 4 indicate an average improvement of 7% to 54% across various streams over the baselines, showcasing the effectiveness of our framework.

Table 4: **GCL setting:** Node classification accuracy (%) on the OBGN-Arxiv dataset under the GCL setting. In each streaming session, one class is designated as zero-shot, lacking any training examples.

OBGN-Arxiv (2-way 10-shot, 1-way 0-shot GCL setting)											
Stream	Base	S_1	S_2	S_3	S_4	S_5	S_6	S_7	S_8	S_9	S_{10}
Random guess	18.10	14.62	12.50	09.47	08.64	08.00	06.43	05.81	05.59	04.86	05.00
CDVSc	68.35	50.86	43.02	37.68	29.28	26.03	22.12	19.78	15.33	11.48	10.73
BMVSc	68.38	51.54	44.28	36.78	30.30	27.22	21.98	20.12	19.78	10.56	09.34
WDVSc	67.22	50.98	45.02	35.78	29.54	26.77	21.67	18.34	16.88	11.56	07.86
GraphCEN	77.13	62.37	51.76	38.42	28.92	18.80	15.36	10.56	08.92	08.56	06.53
DCDFL	64.66	53.42	45.06	32.76	30.48	30.57	28.44	25.89	24.18	22.00	19.62
GOTHAM III.o	**82.91**	**68.11**	**59.90**	**47.68**	**43.67**	**43.31**	**38.76**	**36.41**	**34.62**	**32.55**	**30.28**
%gain	**07.49**	**09.20**	**15.73**	**24.10**	**43.27**	**41.67**	**36.29**	**40.63**	**43.20**	**47.95**	**54.33**

Ablation Study: We conducted a detailed analysis of our framework across three different aspects: **(A) Contribution of different loss functions:** Various loss functions contribute differently to optimal model performance. For this analysis, we selected the Cora-ML dataset, and the corresponding plot is available in Figure 4 (A). **(B) Support set sampling:** We categorized the dataset into small, moderate, and large-sized graphs. To ensure generalizability across other datasets, we performed support set sampling using k-hop random walks, with the ideal hop length observed between 2-4 hops from the labeled nodes. Refer to Figure 4 (B) for more details. **(C) GNN backbones:** We examined the role of different GNN architectures within the GOTHAM framework for the Cora-ML and Amazon datasets. Interestingly, performance remained consistent across different architectures, indicating model-agnostic behavior. Refer to Figure 4 (C) for more details.

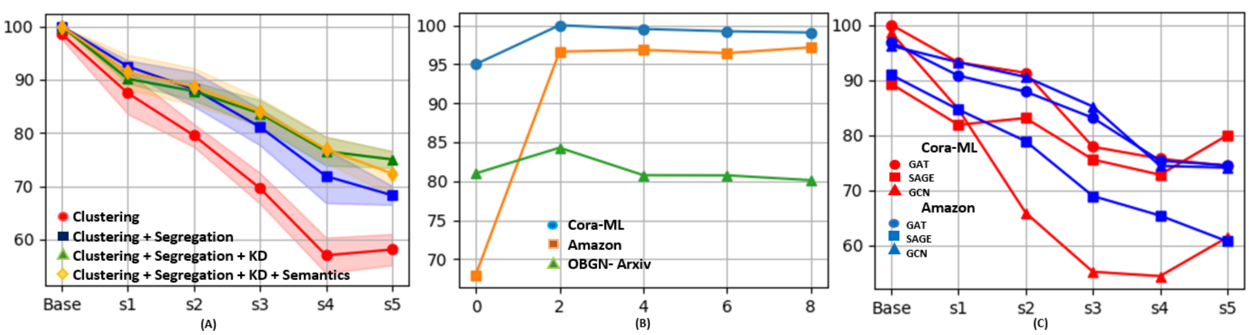
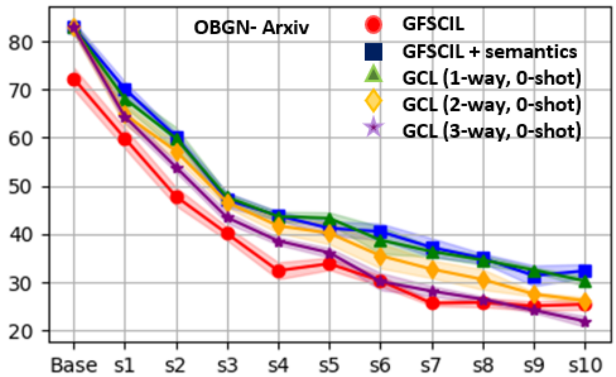


Figure 4: (A) Contribution of different loss functions on the Cora-ML dataset. (B) Support set sampling: determining ideal random-walk length. (C) Different GNN backbones on Cora-ML and Amazon datasets. (A) and (C) displays performance vs streaming sessions, while (B) shows performance vs random-walk length.

Figure (5) presents a detailed analysis of the Graph Class Incremental Learning under Weak Supervision (GCL) setting, showcasing the performance of our model across different variants of GOTHAM for various tasks on the Cora-ML and OBGN-Arxiv datasets. The plots offer an overview of GOTHAM's performance across different representations encountered during few-shot and zero-shot learning scenarios. Notably, the model maintains consistent performance even when faced with a heavy influence of unseen classes during streaming sessions. To simplify understanding: the experimental setup for base training remains consistent throughout. During streaming sessions, where we adopt an n-way, k-shot strategy, we experiment with different values of n while setting k to zero.

6 Conclusion

In this study, we introduced GOTHAM, a class incremental learning framework operating under weak supervision. We initially addressed the GFSCIL problem setting, where access to labeled data during base training is limited. Our experiments demonstrated the advantages of incorporating semantic attributes for Text-Attributed Graphs (TAGs). We then extended our focus to a broader objective, Graph Class



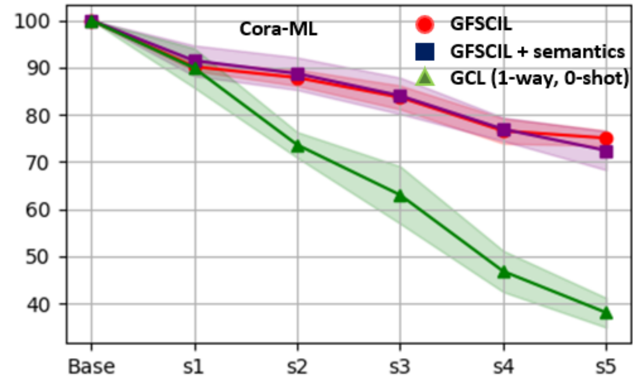


Figure 5: Performance analysis of GOTHAM framework on OBGN-Arxiv and Cora-ML datasets. **(Left)**: GCL with a 3-way k-shot setting shows consistent performance, even in zero-shot learning cases. **(Right)**: GCL with the 1-way k-shot setting on Cora-ML.

Incremental Learning under Weak Supervision (GCL), where novel classes have both a few-shot and zero-shot representation. Through extensive experiments, we conclusively established the generalizability and effectiveness of our framework across a wide range of tasks.

7 Estimating Lower Bound on Representation Distortion

Part 2: Establishing a theoretical understanding presented in the paper (2) for our work

Lower Bound Estimation for Representation Distortion:

The manuscript adopts a theoretical framework as cited in [1], for understanding the evolution and stability of prototypes in a graph neural network (GNN) model. The prototype $\mathbf{P}_{C,S}$ for class C is computed by aggregating node representations from class C using the GNN, as described by Equation (1) in the manuscript:

$$\mathbf{P}_{C,S} := \frac{1}{|\mathcal{S}_{x,C}|} \sum_{i=1}^{|\mathcal{S}_{x,C}|} \mathrm{GNN}_\theta\left(v_i\right) \tag{7}$$

where $\mathcal{S}_{x,C}$ is the set of nodes in class C, and $\mathrm{GNN}_\theta(v_i)$ is the output of the GNN for node v_i with parameters θ.

Assumptions

1. Graph Evolution Process: The graph evolves over time, with new nodes added and connected to existing ones. Specifically:

- The feature matrix X_0 is drawn from a continuous probability distribution supported on $\mathbb{R}^{n\times d}$.
- At each time t, a new node indexed $n+t$ is added to the graph, with positive probability of connecting to existing nodes. Edges in the graph do not vanish.
- The feature vector of a new node $\mathbf{x}_{n+t}$ is centered around zero, i.e., $\mathbb{E}[\mathbf{x}_{n+t} \mid G_0, \ldots, G_{t-1}] = \mathbf{0}_d$, ensuring that the features have zero mean conditioned on previous graph states.

2. Perturbed GCN Model: The GNN model is parameterized as follows:

- Each parameter θ_j of the model is drawn from a uniform distribution $U(\theta_j^*, \xi)$, centered at θ_j^* with perturbation range ξ.

- The model uses a Leaky ReLU activation function with slope β for negative inputs, influencing the response of the model to parameter perturbations.

Lower Bound Estimation

The adopted equation updated for our case is:

$$\sum_{C \in C_0 \cap C_T} \ell_\tau(\mathbf{P}_{C,S}) \geq \frac{N\beta^2\xi^4}{9} \sum_{C \in C_0 \cap C_T} \mathbb{E}\left[\left(\frac{1}{d_\tau(\mathbf{P}_{C,S})} - \frac{1}{d_0(\mathbf{P}_{C,S})}\right)^2 \frac{1}{|\mathcal{S}_{x,C}|} \sum_{k \in N_0(\mathbf{P}_{C,S})} \|S_{x,k}\|^2\right] \quad (8)$$

The proof for this equation is provided in the appendix. This equation quantifies the deviation in prototype representations due to graph dynamics and model parameter perturbations. It incorporates the following:

- $d_\tau(\mathbf{P}_{C,S})$, $d_0(\mathbf{P}_{C,S})$: The degrees of the prototype at times τ and 0, representing the number of nodes contributing to the prototype at those respective times.
- $N_0(\mathbf{P}_{C,S})$: The neighborhood of the prototype, i.e., the set of nodes that form the prototype.
- $C_0 \cap C_T$: The intersection of classes present at the initial time (C_0) and at the final time (C_T), indicating the classes that persist throughout the evolution of the graph.
- β is the slope ratio for negative values in the Leaky ReLU, N is the model width, and ξ is the perturbation range.

Remark: In our case, the expected deviation of prototype representations increases over time due to the evolving graph structure and model perturbations. This effect is more pronounced when the model's width N is large. The assumptions of a continuous probability distribution and an ever-growing graph are key to understanding the model's long-term behavior, reflecting the increasing complexity in large-scale models.

We kindly request the reviewer's guidance on whether this should be included in the final version of the manuscript.

References

Boosting zero-shot node classification via dependency capture and discriminative feature learning, 2024. URL `https://sigport.org/documents/boosting-zero-shot-node-classification-dependency-capture-and-discriminative-feature`.

Kian Ahrabian, Yishi Xu, Yingxue Zhang, Jiapeng Wu, Yuening Wang, and Mark Coates. Structure aware experience replay for incremental learning in graph-based recommender systems. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pp. 2832–2836, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450384469. doi: 10.1145/3459637.3482193. URL `https://doi.org/10.1145/3459637.3482193`.

Mikhail Belkin, Partha Niyogi, and Vikas Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research*, 7(85):2399–2434, 2006. URL `http://jmlr.org/papers/v7/belkin06a.html`.

Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. *arXiv: Machine Learning*, 2017. URL `https://api.semanticscholar.org/CorpusID:4630420`.

Jeff Calder, Brendan Cook, Matthew Thorpe, and Dejan Slepcev. Poisson learning: Graph based semi-supervised learning at very low label rates, 2020.

Shaosheng Cao, Wei Lu, and Qiongkai Xu. Deep neural networks for learning graph representations. *Proceedings of the AAAI Conference on Artificial Intelligence*, 30(1), Feb. 2016. doi: 10.1609/aaai.v30i1.10179. URL `https://ojs.aaai.org/index.php/AAAI/article/view/10179`.

Ali Cheraghian, Shafin Rahman, Pengfei Fang, Soumava Kumar Roy, Lars Petersson, and Mehrtash Tafazzoli Harandi. Semantic-aware knowledge distillation for few-shot class-incremental learning. *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 2534–2543, 2021. URL `https://api.semanticscholar.org/CorpusID:232147521`.

Daniel Cummings and Marcel Nassar. Structured citation trend prediction using graph neural networks. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, May 2020. doi: 10.1109/icassp40776.2020.9054769. URL `http://dx.doi.org/10.1109/ICASSP40776.2020.9054769`.

Angel Daruna, Mehul Gupta, Mohan Sridharan, and Sonia Chernova. Continual learning of knowledge graph embeddings, 2021. URL `https://arxiv.org/abs/2101.05850`.

Kaize Ding, Jianling Wang, Jundong Li, Kai Shu, Chenghao Liu, and Huan Liu. Graph prototypical networks for few-shot learning on attributed networks. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, pp. 295–304, New York, NY, USA, 2020a. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3411922. URL `https://doi.org/10.1145/3340531.3411922`.

Kaize Ding, Jianling Wang, Jundong Li, Kai Shu, Chenghao Liu, and Huan Liu. Graph prototypical networks for few-shot learning on attributed networks, 2020b.

Kaize Ding, Jianling Wang, Jundong Li, James Caverlee, and Huan Liu. Robust graph meta-learning for weakly-supervised few-shot node classification, 2022.

Ming Ding, Jie Tang, and Jie Zhang. Semi-supervised learning on graphs with generative adversarial nets. In *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, CIKM '18, pp. 913–922, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450360142. doi: 10.1145/3269206.3271768. URL `https://doi.org/10.1145/3269206.3271768`.

Kaituo Feng, Changsheng Li, Ye Yuan, and Guoren Wang. Freekd: Free-direction knowledge distillation for graph neural networks. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, pp. 357–366, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539320. URL `https://doi.org/10.1145/3534678.3539320`.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks, 2017.

Lukas Galke, Benedikt Franke, Tobias Zielke, and Ansgar Scherp. Lifelong learning of graph neural networks for open-world node classification. *2021 International Joint Conference on Neural Networks (IJCNN)*, pp. 1–8, 2020. URL `https://api.semanticscholar.org/CorpusID:237599305`.

Jiechao Guan, Zhiwu Lu, Tao Xiang, Aoxue Li, An Zhao, and Ji-Rong Wen. Zero and few shot learning with semantic feature synthesis and competitive learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 43(7):2510–2523, 2021. doi: 10.1109/TPAMI.2020.2965534.

Zhichun Guo, Chuxu Zhang, Wenhao Yu, John Herr, Olaf Wiest, Meng Jiang, and Nitesh V. Chawla. Few-shot graph learning for molecular property prediction. In *Proceedings of the Web Conference 2021*, WWW '21, pp. 2559–2567, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383127. doi: 10.1145/3442381.3450112. URL `https://doi.org/10.1145/3442381.3450112`.

Kourosh Hakhamaneshi, Marcel Nassar, Mariano Phielipp, Pieter Abbeel, and Vladimir Stojanovic. Pretraining graph neural networks for few-shot analog circuit modeling and design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 42(7):2163–2173, 2023. doi: 10.1109/TCAD.2022.3217421.

William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pp. 1025–1035, Red Hook, NY, USA, 2017. Curran Associates Inc. ISBN 9781510860964.

Yi Han, Shanika Karunasekera, and Christopher Leckie. Graph neural networks with continual learning for fake news detection from social media. *ArXiv*, abs/2007.03316, 2020. URL `https://api.semanticscholar.org/CorpusID:220380817`.

Celina Hanouti and Hervé Le Borgne. Learning semantic ambiguities for zero-shot learning, 2022.

Mikael Henaff, Joan Bruna, and Yann LeCun. Deep convolutional networks on graph-structured data. *ArXiv*, abs/1506.05163, 2015. URL `https://api.semanticscholar.org/CorpusID:10443309`.

Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 831–839, 2019. URL `https://api.semanticscholar.org/CorpusID:195453293`.

Yifan Hou, Jian Zhang, James Cheng, Kaili Ma, Richard T. B. Ma, Hongzhi Chen, and Ming-Chang Yang. Measuring and improving the use of graph information in graph neural networks. In *International Conference on Learning Representations*, 2020. URL `https://openreview.net/forum?id=rkeIIkHKvS`.

Kexin Huang and Marinka Zitnik. Graph meta learning via local subgraphs. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 5862–5874. Curran Associates, Inc., 2020. URL `https://proceedings.neurips.cc/paper_files/paper/2020/file/412604be30f701b1b1e3124c252065e6-Paper.pdf`.

Wei Ju, Yifang Qin, Siyu Yi, Zhengyang Mao, Kangjie Zheng, Luchen Liu, Xiao Luo, and Ming Zhang. Zero-shot node classification with graph contrastive embedding network. *Transactions on Machine Learning Research*, 2023. ISSN 2835-8856. URL `https://openreview.net/forum?id=8wGXnjRLSy`.

Vassilis Kalofolias. How to learn a graph from smooth signals, 2016.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL `https://openreview.net/forum?id=SJU4ayYgl`.

Xiaoyu Kou, Yankai Lin, Shaobo Liu, Peng Li, Jie Zhou, and Yan Zhang. Disentangle-based continual graph representation learning. *ArXiv*, abs/2010.02565, 2020. URL `https://api.semanticscholar.org/CorpusID:222142503`.

Lin Lan, Pinghui Wang, Xuefeng Du, Kaikai Song, Jing Tao, and Xiaohong Guan. Node classification on graphs with few-shot novel labels via meta transformed network embedding. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS'20, Red Hook, NY, USA, 2020. Curran Associates Inc. ISBN 9781713829546.

Qimai Li, Zhichao Han, and Xiao-Ming Wu. Deeper insights into graph convolutional networks for semi-supervised learning. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018. ISBN 978-1-57735-800-8.

Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2018. doi: 10.1109/TPAMI.2017.2773081.

Zemin Liu, Yuan Fang, Chenghao Liu, and Steven C.H. Hoi. Relative and absolute location embedding for few-shot node classification on graph. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(5):4267–4275, May 2021. doi: 10.1609/aaai.v35i5.16551. URL `https://ojs.aaai.org/index.php/AAAI/article/view/16551`.

Bin Lu, Xiaoying Gan, Lina Yang, Weinan Zhang, Luoyi Fu, and Xinbing Wang. Geometer: Graph few-shot class-incremental learning via prototype representation. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, KDD '22, pp. 1152–1161, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450393850. doi: 10.1145/3534678.3539280. URL https://doi.org/10.1145/3534678.3539280.

Zhiwu Lu, Jiechao Guan, Aoxue Li, Tao Xiang, An Zhao, and Ji-Rong Wen. Zero and few shot learning with semantic feature synthesis and competitive learning, 2018.

Tomas Mikolov, Kai Chen, Gregory S. Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. In *International Conference on Learning Representations*, 2013. URL https://api.semanticscholar.org/CorpusID:5959482.

Guo-Jun Qi, Charu Aggarwal, Qi Tian, Heng Ji, and Thomas Huang. Exploring context and content links in social media: A latent space method. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 34(5):850–862, 2012. doi: 10.1109/TPAMI.2011.191.

Jiezhong Qiu, Qibin Chen, Yuxiao Dong, Jing Zhang, Hongxia Yang, Ming Ding, Kuansan Wang, and Jie Tang. Gcc: Graph contrastive coding for graph neural network pre-training. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery amp; Data Mining*, KDD '20. ACM, August 2020. doi: 10.1145/3394486.3403168. URL http://dx.doi.org/10.1145/3394486.3403168.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, G. Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 5533–5542, 2016. URL https://api.semanticscholar.org/CorpusID:206596260.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning, 2017.

Yixin Ren, Li Ke, Dong Li, Hui Xue, Zhao Li, and Shuigeng Zhou. Incremental graph classification by class prototype construction and augmentation. *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, 2023. URL https://api.semanticscholar.org/CorpusID:264350245.

Saed Rezayi, Handong Zhao, Sungchul Kim, Ryan Rossi, Nedim Lipka, and Sheng Li. Edge: Enriching knowledge graph embeddings with external text. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy, Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou (eds.), *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 2767–2776, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.221. URL https://aclanthology.org/2021.naacl-main.221.

Aditya Hemant Shahane, Saripilli Venkata Swapna Manjiri, Ankesh Jain, and Sandeep Kumar. Graph of circuits with GNN for exploring the optimal design space. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=VNjJAWjuEU.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In I. Guyon, U. Von Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017a. URL https://proceedings.neurips.cc/paper_files/paper/2017/file/cb8da6767461f2812ae4290eac7cbc42-Paper.pdf.

Jake Snell, Kevin Swersky, and Richard S. Zemel. Prototypical networks for few-shot learning, 2017b.

Justin Solomon, Raif M. Rustamov, Leonidas Guibas, and Adrian Butscher. Wasserstein propagation for semi-supervised learning. In *Proceedings of the 31st International Conference on International Conference on Machine Learning - Volume 32*, ICML'14, pp. I–306–I–314. JMLR.org, 2014.

Jie Song, Chengchao Shen, Yezhou Yang, Yang Liu, and Mingli Song. Transductive unbiased embedding for zero-shot learning. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 1024–1033, 2018. doi: 10.1109/CVPR.2018.00113.

Aravind Subramanian, Pablo Tamayo, Vamsi Mootha, Sayan Mukherjee, Benjamin Ebert, Michael Gillette, Amanda Paulovich, Scott Pomeroy, Todd Golub, Eric Lander, and Jill Mesirov. Subramanian a, tamayo p, mootha vk, mukherjee s, ebert bl, gillette ma, paulovich a, pomeroy sl, golub tr, lander es, mesirov jpgene set enrichment analysis: a knowledge-based approach for interpreting genome-wide expression profiles. proc natl acad sci usa 102(43): 15545-15550. *Proceedings of the National Academy of Sciences of the United States of America*, 102:15545–50, 11 2005. doi: 10.1073/pnas.0506580102.

Zhen Tan, Kaize Ding, Ruocheng Guo, and Huan Liu. Graph few-shot class-incremental learning. In *Proceedings of the Fifteenth ACM International Conference on Web Search and Data Mining*, WSDM '22, pp. 987–996, New York, NY, USA, 2022. Association for Computing Machinery. ISBN 9781450391320. doi: 10.1145/3488560.3498455. URL https://doi.org/10.1145/3488560.3498455.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pp. 990–998, New York, NY, USA, 2008. Association for Computing Machinery. ISBN 9781605581934. doi: 10.1145/1401890.1402008. URL https://doi.org/10.1145/1401890.1402008.

Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018. URL https://openreview.net/forum?id=rJXMpikCZ.

Vikas Verma, Meng Qu, Kenji Kawaguchi, Alex Lamb, Yoshua Bengio, Juho Kannala, and Jian Tang. Graphmix: Improved training of gnns for semi-supervised learning, 2020.

Vinay Kumar Verma, Dhanajit Brahma, and Piyush Rai. A meta-learning framework for generalized zero-shot learning, 2019.

Oriol Vinyals, Charles Blundell, Timothy Lillicrap, koray kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016. URL https://proceedings.neurips.cc/paper_files/paper/2016/file/90e1357833654983612fb05e3ec9148c-Paper.pdf.

Ziyu Wan, Dongdong Chen, Yan Li, Xingguang Yan, Junge Zhang, Yizhou Yu, and Jing Liao. Transductive zero-shot learning with visual structure constraint. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019a. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/5ca359ab1e9e3b9c478459944a2d9ca5-Paper.pdf.

Ziyu Wan, Yan Li, Min Yang, and Junge Zhang. Transductive zero-shot learning via visual center adaptation. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press, 2019b. ISBN 978-1-57735-809-1. doi: 10.1609/aaai.v33i01.330110059. URL https://doi.org/10.1609/aaai.v33i01.330110059.

Chen Wang, Yuheng Qiu, Dasong Gao, and Sebastian Scherer. Lifelong graph learning, 2022a. URL https://arxiv.org/abs/2009.00647.

Junshan Wang, Guojie Song, Yi Wu, and Liang Wang. Streaming graph neural networks via continual learning. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, CIKM '20, pp. 1515–1524, New York, NY, USA, 2020a. Association for Computing Machinery. ISBN 9781450368599. doi: 10.1145/3340531.3411963. URL `https://doi.org/10.1145/3340531.3411963`.

Song Wang, Yushun Dong, Kaize Ding, Chen Chen, and Jundong Li. Few-shot node classification with extremely weak supervision. *Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining*. doi: 10.1145/3539597.3570435. URL `https://par.nsf.gov/biblio/10414116`.

Song Wang, Xiao Huang, Chen Chen, Liang Wu, and Jundong Li. Reform: Error-aware few-shot knowledge graph completion. In *Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, CIKM '21, pp. 1979–1988, New York, NY, USA, 2021a. Association for Computing Machinery. ISBN 9781450384469. doi: 10.1145/3459637.3482470. URL `https://doi.org/10.1145/3459637.3482470`.

Song Wang, Yushun Dong, Xiao Huang, Chen Chen, and Jundong Li. Faith: Few-shot graph classification with hierarchical task graphs. In Lud De Raedt (ed.), *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence, IJCAI-22*, pp. 2284–2290. International Joint Conferences on Artificial Intelligence Organization, 7 2022b. doi: 10.24963/ijcai.2022/317. URL `https://doi.org/10.24963/ijcai.2022/317`. Main Track.

Song Wang, Yushun Dong, Xiao Huang, Chen Chen, and Jundong Li. Faith: Few-shot graph classification with hierarchical task graphs. pp. 2259–2265, 07 2022c. doi: 10.24963/ijcai.2022/314.

Song Wang, Zhen Tan, Huan Liu, and Jundong Li. Contrastive meta-learning for few-shot node classification. *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2023a. URL `https://api.semanticscholar.org/CorpusID:259262390`.

Xiao Wang, Meiqi Zhu, Deyu Bo, Peng Cui, Chuan Shi, and Jian Pei. Am-gcn: Adaptive multi-channel graph convolutional networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pp. 1243–1253, New York, NY, USA, 2020b. Association for Computing Machinery. ISBN 9781450379984. doi: 10.1145/3394486.3403177. URL `https://doi.org/10.1145/3394486.3403177`.

Zheng Wang, Jialong Wang, Yuchen Guo, and Zhiguo Gong. Zero-shot node classification with decomposed graph prototype network. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery amp; Data Mining*, KDD '21. ACM, August 2021b. doi: 10.1145/3447548.3467230. URL `http://dx.doi.org/10.1145/3447548.3467230`.

Zhengbo Wang, Jian Liang, Zilei Wang, and Tieniu Tan. Exploiting semantic attributes for transductive zero-shot learning, 2023b.

Yanan Wu, Tengfei Liang, Songhe Feng, Yi Jin, Gengyu Lyu, Haojun Fei, and Yang Wang. Metazscil: A meta-learning approach for generalized zero-shot class incremental learning. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(9):10408–10416, Jun. 2023. doi: 10.1609/aaai.v37i9.26238. URL `https://ojs.aaai.org/index.php/AAAI/article/view/26238`.

Louis-Pascal Xhonneux, Meng Qu, and Jian Tang. Continuous graph neural networks. In Hal Daumé III and Aarti Singh (eds.), *Proceedings of the 37th International Conference on Machine Learning*, volume 119 of *Proceedings of Machine Learning Research*, pp. 10432–10441. PMLR, 13–18 Jul 2020. URL `https://proceedings.mlr.press/v119/xhonneux20a.html`.

Chen Xing, Negar Rostamzadeh, Boris Oreshkin, and Pedro O O. Pinheiro. Adaptive cross-modal few-shot learning. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL `https://proceedings.neurips.cc/paper_files/paper/2019/file/d790c9e6c0b5e02c87b375e782ac01bc-Paper.pdf`.

Jingyi Xu and Hieu M. Le. Generating representative samples for few-shot classification. *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 8993–9003, 2022. URL `https://api.semanticscholar.org/CorpusID:248562924`.

Keyulu Xu, Weihua Hu, Jure Leskovec, and Stefanie Jegelka. How powerful are graph neural networks? In *International Conference on Learning Representations*, 2019. URL `https://openreview.net/forum?id=ryGs6iA5Km`.

Yishi Xu, Yingxue Zhang, Wei Guo, Huifeng Guo, Ruiming Tang, and Mark Coates. Graphsail: Graph structure aware incremental learning for recommender systems. In *Proceedings of the 29th ACM International Conference on Information amp; Knowledge Management*, CIKM '20. ACM, October 2020. doi: 10.1145/3340531.3412754. URL `http://dx.doi.org/10.1145/3340531.3412754`.

Huaxiu Yao, Chuxu Zhang, Ying Wei, Meng Jiang, Suhang Wang, Junzhou Huang, Nitesh Chawla, and Zhenhui Li. Graph few-shot learning via knowledge transfer. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):6656–6663, Apr. 2020. doi: 10.1609/aaai.v34i04.6142. URL `https://ojs.aaai.org/index.php/AAAI/article/view/6142`.

Jiaxuan You, Tianyu Du, and Jure Leskovec. Roland: Graph learning framework for dynamic graphs. *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, 2022. URL `https://api.semanticscholar.org/CorpusID:251518327`.

Wenchao Yu, Wei Cheng, Charu C. Aggarwal, Kai Zhang, Haifeng Chen, and Wei Wang. 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*, KDD '18, pp. 2672–2681, New York, NY, USA, 2018. Association for Computing Machinery. ISBN 9781450355520. doi: 10.1145/3219819.3220024. URL `https://doi.org/10.1145/3219819.3220024`.

Yuyao Zhai, Liang Chen, and Minghua Deng. Generalized cell type annotation and discovery for single-cell rna-seq data. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(4):5402–5410, Jun. 2023. doi: 10.1609/aaai.v37i4.25672. URL `https://ojs.aaai.org/index.php/AAAI/article/view/25672`.

Hai Zhang, Junzhe Xu, Shanlin Jiang, and Zhenan He. Simple semantic-aided few-shot learning, 2023.

Wentao Zhang, Xupeng Miao, Yingxia Shao, Jiawei Jiang, Lei Chen, Olivier Ruas, and Bin Cui. Reliable data distillation on graph convolutional network. In *Proceedings of the 2020 ACM SIGMOD International Conference on Management of Data*, SIGMOD '20, pp. 1399–1414, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367356. doi: 10.1145/3318464.3389706. URL `https://doi.org/10.1145/3318464.3389706`.

Xikun ZHANG, Dongjin Song, and Dacheng Tao. CGLB: Benchmark tasks for continual graph learning. In *Thirty-sixth Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. URL `https://openreview.net/forum?id=5wNiiIDynDF`.

Dengyong Zhou and Bernhard Schölkopf. Learning from labeled and unlabeled data using random walks. In Carl Edward Rasmussen, Heinrich H. Bülthoff, Bernhard Schölkopf, and Martin A. Giese (eds.), *Pattern Recognition*, pp. 237–244, Berlin, Heidelberg, 2004. Springer Berlin Heidelberg. ISBN 978-3-540-28649-3.

Dengyong Zhou, Jason Weston, Arthur Gretton, Olivier Bousquet, and Bernhard Schölkopf. Ranking on data manifolds. In S. Thrun, L. Saul, and B. Schölkopf (eds.), *Advances in Neural Information Processing Systems*, volume 16. MIT Press, 2003. URL `https://proceedings.neurips.cc/paper_files/paper/2003/file/2c3ddf4bf13852db711dd1901fb517fa-Paper.pdf`.

Fan Zhou, Chengtai Cao, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Ji Geng. Meta-gnn: On few-shot node classification in graph meta-learning. In *Proceedings of the 28th ACM International Conference on Information and Knowledge Management*, CIKM '19, pp. 2357–2360, New York, NY, USA, 2019a. Association for Computing Machinery. ISBN 9781450369763. doi: 10.1145/3357384.3358106. URL `https://doi.org/10.1145/3357384.3358106`.

Fan Zhou, Chengtai Cao, Kunpeng Zhang, Goce Trajcevski, Ting Zhong, and Ji Geng. Meta-gnn: On few-shot node classification in graph meta-learning, 2019b.

Xiaojin Zhu, Zoubin Ghahramani, and John D. Lafferty. Combining active learning and semi-supervised learning using gaussian fields and harmonic functions. In *International Conference on Machine Learning*, 2003. URL https://api.semanticscholar.org/CorpusID:1052837.

Yanqiao Zhu, Yichen Xu, Feng Yu, Qiang Liu, Shu Wu, and Liang Wang. Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, WWW '21, pp. 2069–2080, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450383127. doi: 10.1145/3442381.3449802. URL https://doi.org/10.1145/3442381.3449802.

8 Appendix

8.1 Datasets

We assess the performance of our proposed framework, GOTHAM, on three real-world datasets- Cora-ML, Amazon, and OBGN-Arxiv. The detailed description is in Table 5:

Table 5: Statistics of datasets used in the experiments

Dataset	Nodes	Features	Classes	Class Labels	Tasks
Cora-ML	2,708	1,433	7	Neural Network, Rule Learning, Reinforcement Learning, Probabilistic Methods, Theory, Genetic Algorithms, Case-based	GFSCIL, GCL
Amazon	13,752	767	10	Label names Unavailable	GFSCIL
OBGN-Arxiv	169,343	128	40	Arxiv cs na, Arxiv cs mm, Arxiv cs lo, Arxiv cs cy, Arxiv cs cr, Arxiv cs dc, Arxiv cs hc, Arxiv cs cv, Arxiv cs ai, ...	GFSCIL, GCL

Cora-ML (Bojchevski & Günnemann, 2017): This is an academic network of machine learning papers. The dataset contains 7 classes, with each node representing a paper and each edge representing a citation between papers.

Amazon (Hou et al., 2020): This dataset represents segments of the Amazon co-purchase e-commerce network. Each node is an item, and each edge denotes a co-purchase relationship by a common user. Node features are bag-of-words encoded product reviews, and class labels correspond to product categories.

OBGN-Arxiv (Subramanian et al., 2005): This dataset is a directed graph representing the citation network of Computer Science arXiv papers indexed by MAG. Each node is an arXiv paper, and each directed edge indicates a citation from one paper to another. Each paper has a 128-dimensional feature vector, created by averaging the embeddings of words in its title and abstract.

8.2 Baseline methods

In the GFSCIL setting, we benchmark our results against several state-of-the-art frameworks for few-shot class incremental learning and few-shot node classification, including:

8.2.1 Few-shot node classification

Meta-GNN (Zhou et al., 2019b): Meta-GNN addresses few-shot node classification in graph meta-learning. It learns from numerous similar tasks to classify nodes from new classes with few labeled samples. Meta-GNN is versatile and can be easily integrated into any state-of-the-art GNN.

Graph Prototypical Network (GPN) (Ding et al., 2020a): GPN is an advanced method for few-shot node classification. It uses graph neural networks and meta-learning on attributed networks for metric-based few-shot learning.

8.2.2 Class incremental learning

Incremental classifier and representation learning (iCaRL) (Rebuffi et al., 2017): iCaRL is a class-incremental method for image classification. We enhance it by replacing the feature extractor with a two-layer GAT network.

Hierarchical-Attention-based Graph Meta-learning (HAG-Meta) (Tan et al., 2022): HAG-Meta follows the graph pseudo-incremental learning approach, allowing the model to learn new classes incrementally by cyclically adopting them from base classes. It also tackles class imbalance using hierarchical attention modules.

Graph Few-Shot Class-Incremental Learning via Prototype Representation (Geometer)(Lu et al., 2022): Geometer predicts a node's label by finding the nearest class prototype in the metric space and adjusting the prototypes based on geometric proximity, uniformity, and separability of novel classes. To address catastrophic forgetting and unbalanced labeling, it uses teacher-student knowledge distillation and biased sampling.

Class Prototype Construction and Augmentation (CPCA) (Ren et al., 2023): CPCA is a method that constructs class prototypes in the embedding space to capture rich topological information of nodes or graphs, representing past data for future learning. To enhance the model's adaptability to new classes, CPCA uses class prototype augmentation (PA) to create virtual classes by combining current prototypes.

In the GCL setting, where novel classes have both few-shot and zero-shot representations, we compare against zero-shot learning frameworks with inductive learning as baselines. These approaches include:

8.2.3 Zero-shot learning

DCDFL (741, 2024): In DCDFL, a model for zero-shot node classification captures dependencies and learns discriminative features. It uses a relation-aware network to leverage long-range dependencies between nodes and employs a domain-invariant adversarial loss to reduce domain bias and promote domain-insensitive feature representations. Additionally, it enhances the representation by utilizing inter-class separability within the metric space.

GraphCEN (Ju et al., 2023): GraphCEN constructs an affinity graph to model class relations and uses node- and class-level contrastive learning (CL) to jointly learn node embeddings and class assignments. The two levels of CL are optimized to enhance each other.

(CDVSc, BMVSc, WDVSc) (Wan et al., 2019a): Based on the observation that visual features of test instances form distinct clusters, a new visual structure constraint on class centers for transductive ZSL is proposed to improve the generality of the projection function and alleviate domain shift issues. Three strategies—symmetric Chamfer distance, bipartite matching distance, and Wasserstein distance—are used to align the projected unseen semantic centers with the visual cluster centers of test instances.

Random guess: Randomly guessing an unseen label, introduced as a naive baseline.

8.3 Parameter settings

In our proposed framework, various sets of hyper-parameters are involved. These are summarized in Table 6 below. The code implementation is available here: `https://shorturl.at/2VCEc`

Table 6: Parameter settings

Parameter	Value	Parameter	Value	Parameter	Value	Parameter	Value
word2vec	512	$\{\alpha_1, \alpha_2, \alpha_3\}$	$\{1, 0.25, 1\}$	meta_lr	$\{1e^{-3}, 1e^{-5}\}$	Hidden layer (MLP)	512
Random walk	$\{2, 3, 4\}$	Hidden channels (GNNs)	512	ft_lr	$\{1e^{-3}, 1e^{-5}\}$	Out channels (MLP)	512
Boundary (γ)	0.01	Out channels (GNNs)	512	weight decay	$5e^{-3}$	$\{\lambda_1, \lambda_2\}$	$\{1, 1\}$