COS-META: Enhancing Few-shot Node Classification with Contrastive Meta-Learning

Md. Sirajum Munir Prince[1] and Renata Dividino[1]

Brock University, St Catharines, ON, L2S 3A1, Canada
{mprince, rdividino}@brocku.ca

Abstract. Graph neural networks have shown significant progress in node classification tasks. However, their performance declines when learning from only a few examples per class or when adapting to unseen classes. Meta-learning helps overcome this limitation by enabling models to generalize from limited examples, and adapt to novel classes not encountered during training. In this setting, algorithms train on diverse meta-tasks consisting of a support set containing a few nodes from specific classes and a query set with unseen nodes from those classes, which are used to evaluate their performance based on their ability to adapt to novel data distributions. Current approaches often augment meta-tasks with additional data such as the neighboring nodes of target nodes are used to enrich both support and query sets. However, the structural complexity of graphs introduces challenges in designing effective meta-tasks, as variations in graph structures across tasks can hinder consistent feature representations. To address these challenges, this research explores augmentation strategies in combination with contrastive learning to extract node and class characteristics by identifying instances based on similarity. Evaluation of the effectiveness of this approach through a comprehensive comparison with state-of-the-art methods on benchmark node classification datasets has been demonstrated in this work. This project's source code is publicly available at https://github.com/sirajummprince/COS-META.

Keywords: Meta-Learning, Few-shot Learning, Node Classification, Graph Neural Networks, Contrastive Learning, Task Augmentation

1 Introduction

Graph Neural Networks (GNNs) have emerged as a powerful framework specifically designed to process graph-structured data, where nodes represent entities and edges capture their relationships. GNNs transform graph topologies into low-dimensional node representations through iterative message-passing operations that aggregate, and transform neighborhood information. These learnt representations effectively preserve both local structure and global connectivity patterns, enabling state-of-the-art performance across diverse downstream tasks including node classification, link prediction, and graph classification. GNNs

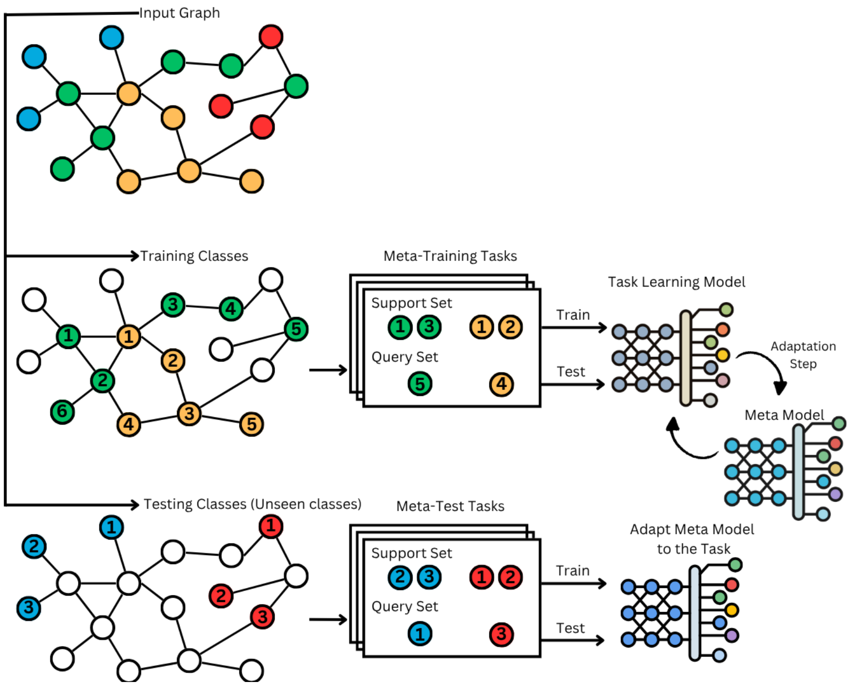

Fig. 1: Graph Meta-Learning Setup: Meta-training and meta-test tasks are designed based on the training and test classes, respectively. In each task, $\mathcal{K}$ nodes are sampled from $\mathcal{N}$ classes to form the support and query sets.

excel in domains with inherent relational structures such as social networks, molecular interactions, recommendation systems, and knowledge graphs.

However, GNNs typically require substantial amounts of labeled data to learn high quality node representations. This requirement creates significant challenges in practical scenarios where obtaining labeled samples for every class is difficult. For example, a classification model may predict user preferences and interests based on user profiles and social connections in networks, where nodes represent users and posts and edges denote interactions. However, when GNNs are used to classify users engaging with emerging topics, only a small subset of users may have direct connections to these new themes, potentially limiting the model's effectiveness. Consequently, GNN performance often deteriorates substantially when faced with limited labeled samples. This limitation has driven research interest in few-shot node classification, which aims to classify nodes effectively using only a small number of labeled samples as references.

Meta-learning [7] is a machine learning (ML) paradigm that enables models to *learn how to learn* by leveraging prior knowledge to adapt quickly to new tasks. It extracts transferable knowledge from tasks with abundant labeled data and applies it to tasks with limited labeled data, effectively addressing few-shot learning challenges. A key component of this approach is the episodic training framework, which simulates learning conditions with few labeled samples during training. This strategy enhances generalization by repeatedly exposing the model to adaptation scenarios. In meta-learning approaches with graphs [21, 9, 20, 8, 23, 14, 3, 19], the model is trained over multiple episodes to learn a shared node embedding space across both training and test meta-tasks. During training, a small set of labeled nodes (the support set) is sampled as a reference for learning unlabeled test nodes (the query set). Typically those approaches employ data augmentation to enhance generalization, such that for each support node, one or more subgraphs centered on it are sampled, generating enriched representations of nodes and their neighborhoods.

Recent approaches leverage contrastive learning (CL) [22, 13, 27] to exploit information granularity and self-supervision, enhancing the transfer of knowledge between training and unseen classes. The COSMIC [9] model captures relationships at both node- and class-levels, optimizing node embeddings and class assignments simultaneously. Task-aware CL [12] models the relationship between node representations and downstream tasks (e.g., node classification). However, these frameworks often struggle with the semantic gap between local structural patterns and global class semantics, particularly when node neighborhoods exhibit high structural similarity across different semantic categories. Building on these ideas, we propose COS-META, a contrastive meta-learning framework for few-shot node classification. Our method employs graph contrastive learning at the node ego-graph level, using topology-aware subgraph augmentation to enrich local structural knowledge and preserve multi-hop neighborhood semantics. Additionally, we incorporate a classifier GNN, which is jointly optimized with contrastive learning. This dual approach allows the GNN to benefit from both the similarity of node embeddings learned through contrastive meta-learning and the available labels for improved classification performance.

2 Background

A brief of the formal definitions and terminologies used in this work are provided in this section. Let a graph be represented as $G = (V, E, X, A, Y)$ with set of nodes $v_1, ..., v_N \in V$ and edges $(v_i, v_j) \in E$. $X \in \mathrm{R}^{N \times n}$ is the feature matrix where X_i corresponds to the n-dimensional feature vector of node $v_i \in V$, and an adjacency matrix $A \in \{0, 1\}^{N \times N}$ in which A_{ij} is 1 if there exists an edge between any two nodes $v_i \in V$ and $v_j \in V$ and 0 otherwise. $Y = y_1, ..., y_M$ is denoted as a set of M distinct node labels.

2.1 Graph Neural Networks

GNNs have enabled the training of deep learning models on graph-structured data for various tasks such as node classification, link prediction or graph classification. GNNs learn a multiple-step mapping $f : G \rightarrow Z \in \mathrm{R}^{N \times n'}$, where Z is an updated feature matrix of a graph G, to represent nodes and edges of G into low-dimensional vector representations (i.e., embeddings). Each step typically consists of two phases: 1) AGGREGATE: in the aggregation phase, information about the neighborhood of every node is gathered; 2) UPDATE: in the update phase, the features of every node is updated based on its current feature vector and the aggregated neighborhood information. These steps are applied as many times as the number of layers in the GNN. The aggregation and update functions are as such:

$$h_{\mathrm{N}(v)} = \mathrm{AGGREGATE}(\{h_u^{(k-1)} : u \in \mathrm{N}(v)\})$$

where the embedding of node v at k-th layer is represented as h_v^k, and $\mathrm{N}(v)$ represent the set of neighbors of the node v. AGGREGATE is a differentiable, permutation-invariant neighborhood aggregation function. A number of architectures for AGGREGATE have been proposed in [6, 10, 25, 5], and therefore can take various forms, such as mean, sum, or max pooling. UPDATE is a differentiable function that takes the aggregated embeddings as input and produces a new embedding for each node.

$$h_v^{(k)} = \mathrm{UPDATE}(h^{(k-1)}, h_{\mathrm{N}(v)})$$

Examples of UPDATE include one-layer perceptrons [6] and MLPs (Multi-Layer Perceptrons) [25]. The input to the first layer is $X^0 = X$, i.e., the initial node features. The output of the final layer Z^k can then be used for downstream tasks, such as node classification or link prediction. Node classification aims to specify a GNN that can accurately map nodes $\in V$ to labels $\in Y$.

2.2 Few-shot Node Classification

Given a graph G and the set of labels Y of its containing nodes, Y can be divided into two disjoint sets: the training classes Y_{tr} and the test classes Y_{ts} (i.e., $Y_{tr} \cap Y_{ts} = \emptyset$). It is to be assumed that there exist abundant labeled nodes in G for each class in Y_{tr} and extremely few labeled nodes for each class in Y_{ts}. The goal is to develop a classifier that can predict the categories of unlabeled nodes from node classes in Y_{ts} by transferring the meta-knowledge obtained from Y_{tr} during the meta-training stage, thus enabling the model to adapt to new classes (Y_{ts}) with limited labeled examples.

The meta-learning framework addresses this challenge by learning how to learn from few examples through a two-stage process: meta-training and meta-testing. During the meta-training phase, multiple learning episodes (or tasks) is constructed from the training classes Y_{tr}. In each episode, at first $\mathcal{N}$ classes from Y_{tr} are randomly sampled, then for each selected class, $\mathcal{K}$ labeled nodes are sampled ($\mathcal{N}$-way $\mathcal{K}$-shot learning) to form the support set and additional nodes from the same classes to construct the query set. This episodic training paradigm ensures that the model repeatedly practices the exact scenario, which it will encounter during meta-testing: learning to adapt to new classes with limited examples. The support set in each episode serves as the few-shot training data, while the query set provides a way to evaluate, and optimize the model's adaptation capability. Through multiple episodes, the model learns generalizable strategies for feature extraction, and adaptation that can be effectively transferred to novel classes in Y_{ts}(see Figure 1).

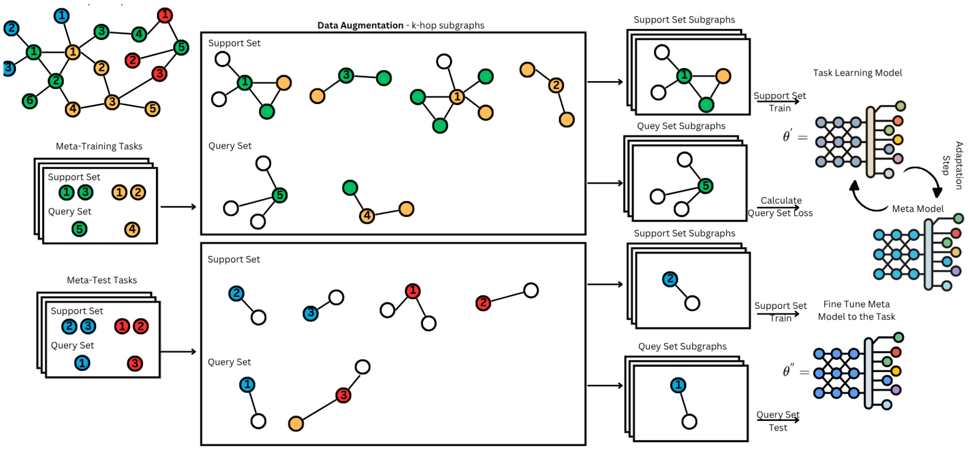

Fig. 2: Graph Meta-Learning Setup with Data Augmentation: To enhance generalization, data augmentation is applied, such as k-hop subgraph extraction (e.g., for $k = 1$) for each node in the support and query set, which captures local structural information and enriches node representations.

Formally, let $\mathcal{T}_{tr} = \{T_1^{tr}, T_2^{tr}, ..., T_n^{tr}\}$ be a set of meta-training tasks, where each task T_i consists of a support set S_i^{tr} and a query set Q_i^{tr}. The meta-learning objective aims to find optimal meta model parameters θ that minimize the loss across all meta-training tasks $T_i \in \mathcal{T}_{tr}$, where each task contributes through the loss function $\mathcal{L}(f_{\theta_i}(Q_i); \theta)$. During meta-testing, the model's ability is evaluated to learn from few examples on entirely new classes from Y_{ts}. Let $\mathcal{T}_{ts} = \{T_1^{ts}, T_2^{ts}, ..., T_m^{ts}\}$ be a set of meta-test tasks, where each task T_i^{ts} consists of a support set S_i^{ts} and a query set Q_i^{ts}. Each support set S_i^{ts} contains $\mathcal{K}$ labeled examples per class for $\mathcal{N}$ test classes, and the model must leverage its meta-learned knowledge to quickly adapt, and make accurate predictions on the unlabeled nodes in the query set.

2.3 Contrastive Learning

Contrastive learning (CL) is a ML technique, particularly in unsupervised and self-supervised settings, for learning representations without explicit labels. It enforces similarity between related data points while pushing apart dissimilar ones. CL relies on contrastive loss, which minimizes the distance between *positive* pairs (similar items), and maximizes it for *negative* pairs (dissimilar items). This approach enhances the robustness and generalization of learned representations in various tasks. Different frameworks have been proposed in [16, 11, 2] such as the Triplet Margin Loss [11] objective function, which optimizes embeddings by ensuring that an anchor (a) is closer to a similar example (p) than to a dissimilar one (n) by a given margin α. This function can be expressed as $\mathcal{L}_{triplet} = \max(0, d(a, p) - d(a, n) + \alpha)$, where $d(\cdot, \cdot)$ represents a distance metric between embeddings, typically Euclidean or cosine distance. The margin parameter α enforces a minimum separation between positive and negative pairs, helping to create more discriminative embeddings.

3 Related Works

Few Shot Learning (FSL) for Node Classification

Meta-GNN [28], the first framework to incorporate principles into GNNs for FSL node classification, leverages Model-Agnostic Meta-Learning (MAML) [4] to learn a good parameter initialization that can quickly adapt to new node classification tasks with only $\mathcal{K}$ labeled samples per class. An attribute-matching framework, AMM-GNN, based on MAML was proposed in [19], using attribute-level attention to extract task-specific information on attributed networks and enhance knowledge transfer. GPN [3], also designed for few-shot node classification on attributed networks, leverages prototypical learning. It implements an encoder that captures both node attributes and topological structure, and a node valuator that estimates the informativeness of each labeled node by considering its centrality and importance within the network. Similarly leveraging prototype learning, G-META's [8] core innovation is representing each node with its local subgraph. Prototypes computed from these embeddings guide adaptation via meta-gradients to improve cross-task knowledge transfer. Meta-GPS [14], proposed for heterophilic graphs, utilizes prototype-based initialization and scaling transformations. The framework adapts representations to varying task distributions using a network encoder and the S^2 transformation. X-FNC [21] generates pseudo-labeled nodes to address weak supervision in meta-training, using Poisson Label Propagation to enhance label propagation. The information bottleneck principle is applied to filter out irrelevant features, improving task adaptation. TENT [20] reduces distribution shifts between meta-training and meta-testing by incorporating node-level, class-level, and task-level adaptations. It reduces connectivity pattern variance by constructing subgraphs for each class using a virtual class node, ensuring structural coherence. TENT uses a task-adaptive matching strategy to preserve mutual information, enhancing task-level performance. I-GNN [15] was proposed to handle the observation that meta-learning approaches with graphs suffer significant performance degradation in the more realistic inductive setting. I-GNN freezes a pre-trained encoder and only fine-tunes a new linear classifier on few-shot tasks. Finally, Pro-MC [23] uses proxy subgraphs to balance noisy and smooth subgraph biases, improving robustness and knowledge transfer. Pro-MC generates proxy subgraphs to calibrate manifold smoothness and enhance knowledge transfer. Our approach follows some of the same ideas of related work. COS-META follows MAML and leverages local subgraphs around target nodes, but without modifying subgraph elements and processes them through a GNN and dual objectives to adapt to the tasks.

3.1 FSL with Constrastive Learning (CL)

The work in [27] introduces multimodal meta-learning into graph CL, proposing a method that effectively adapts to unseen novel classes through bi-level meta-optimization. The framework does not rely on graph augmentations or negative examples. GraphCEN [9] builds an affinity graph to model class relationships, leveraging both node- and class-level CL to jointly optimize node embeddings and class assignments. By combining class semantics with GNNs to create a joint matrix, it enables node-level CL in the row space for effective embeddings and class-level CL in the column space for class consistency, improving both representation learning and class assignment. Task-aware CL [12] enhance downstream task performance by maximizing mutual information between node representations and the downstream task (e.g., node classification). The core of Task-aware CL is the Task-Aware Contrastive Loss (XTCL), which uses the XGBoost sampler to select positive examples, optimizing XTCL for improved generalization. This method captures valuable graph signals in node representations, enhancing model performance. The proposed approach in this research, similar to COS-MIC, uses node-level contrastive learning on the support set to update the GNN model, incorporating structural knowledge via subgraphs. However, in contrast to COLA, the proposed model only use nodes from training classes to create meta-training tasks, consistent with most of the related works. Additionally, like Task-aware CL, CL is applied at the node level, but cross-entropy loss has been utilized for the task/classification level instead.

4 Methodology

This research presents COS-META, a novel framework for FSL on graphs that addresses the challenge of node classification with limited labeled data. COS-META employs a Model-Agnostic Meta-Learning (MAML) [4] approach combined with subgraph-based contrastive learning to learn generalizable node representations that can rapidly adapt to new tasks with minimal supervision.

4.1 COS-META Architecture

COS-META consists of three key components: (1) ego-subgraph extraction to preserve local structure, (2) an encoder to learn node representations, and (3) a graph pooling mechanism to produce graph-level embeddings (see Figure 2).

Ego-Subgraph Extraction For each node v in either the support or query set, we extract its k-hop ego-subgraph $\mathcal{G}_v = (\mathcal{V}_v, \mathcal{E}_v)$, which captures the local neighborhood structure around v. This subgraph-based approach enables the model to focus on relevant local patterns while maintaining computational efficiency. The ego-subgraph provides rich structural context that is crucial for effective node classification in few-shot scenarios.

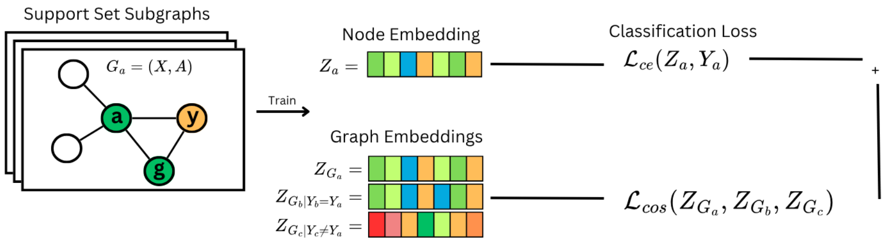

Fig. 3: Few-Shot Node Classification with Contrastive Learning: The proposed approach extends cross-entropy loss with contrastive loss on the graph embeddings that enhances class separation by enforcing a margin between positive and negative graph pairs. This improves the model's ability to learn discriminative node representations in low-data scenarios.

Encoder The GNN encoder f_θ processes the ego-subgraph to generate both node-level and graph-level representations. For a given ego-subgraph $\mathcal{G}_v$ with node features $\mathbf{X}_v$ and edge connectivity $\mathbf{A}_v$, the encoder produces:

$$\mathbf{Z}, \mathbf{Z}_{\mathcal{G}_v} = f_\theta(\mathbf{X}_v, \mathbf{A}_v)$$

where $\mathbf{Z}$ represents the node embeddings within the subgraph and $\mathbf{Z}_{\mathcal{G}_v}$ is the pooled graph-level embedding obtained through mean pooling:

$$\mathbf{Z}_{\mathcal{G}_v} = \frac{1}{|\mathcal{V}_v|} \sum_{u \in \mathcal{V}_v} \mathbf{Z}_u$$

The target node embedding $\mathbf{Z}_v$ is extracted from $\mathbf{Z}$ and used for classification through a linear classifier:

$$\hat{\mathbf{y}}_v = \mathbf{W}\mathbf{Z}_v + \mathbf{b}$$

4.2 Joint Loss Function

Our model is trained using a joint objective that combines supervised classification with contrastive learning to enhance both predictive accuracy and representation quality (see Figure 3):

$$\mathcal{L} = \lambda \mathcal{L}_{\text{CE}} + (1 - \lambda)\mathcal{L}_{\text{CL}}$$

where $\mathcal{L}_{\text{CE}}$ is the cross-entropy loss for classification and $\mathcal{L}_{\text{CL}}$ is the contrastive loss, with λ controlling the balance between the two objectives.

Cross-Entropy Loss The classification loss is computed using standard cross-entropy:

$$\mathcal{L}_{\text{CE}} = -\sum_{i} y_i \log(\text{softmax}(\hat{\mathbf{y}}_i))$$

where y_i is the true label and $\hat{\mathbf{y}}_i$ are the predicted logits for node i.

Contrastive Loss The contrastive component employs InfoNCE loss [16] to learn discriminative graph-level representations. For an anchor node v with graph embedding $\mathbf{Z}_{\mathcal{G}_v}$, we identify positive samples (nodes from the same class) and negative samples (nodes from different classes) within the current task. The contrastive loss is formulated as:

$$\mathcal{L}_{\text{CL}} = -\log \frac{\exp(\text{sim}(\mathbf{Z}_{\mathcal{G}_v}, \mathbf{Z}_{\mathcal{G}_v^+})/\tau)}{\exp(\text{sim}(\mathbf{Z}_{\mathcal{G}_v}, \mathbf{Z}_{\mathcal{G}_v^+})/\tau) + \sum_k \exp(\text{sim}(\mathbf{Z}_{\mathcal{G}_v}, \mathbf{Z}_{\mathcal{G}_v^-})/\tau)}$$

where $\text{sim}(\cdot, \cdot)$ denotes cosine similarity, $\mathbf{Z}_{\mathcal{G}_v^+}$ is a positive example from the same class, $\mathbf{Z}_{\mathcal{G}_v^-}$ represents negative examples from different classes, and τ is a temperature parameter that controls the concentration of the distribution.

To improve learning efficiency, we employ hard negative mining by selecting the most challenging negative samples i.e., those with highest similarity to the anchor, encouraging the model to learn more discriminative representations.

4.3 Meta-Learning Framework

Meta-Training Let $\mathcal{T}_{tr} = \{T_1^{tr}, T_2^{tr}, ..., T_n^{tr}\}$ denote a set of meta-training tasks sampled from disjoint class sets. Each task $T_i^{tr} \in \mathcal{T}_{tr}$ consists of a support set S_i^{tr} and a query set Q_i^{tr}, following an $\mathcal{N}$-way $\mathcal{K}$-shot paradigm where $\mathcal{N}$ classes are randomly selected and $\mathcal{K}$ labeled examples per class form the support set.

The meta-learning objective seeks to find optimal initial parameters θ^* such that, after few gradient steps of adaptation on any support set S_i^{tr}, the model generalizes well to the corresponding query set Q_i^{tr}. This is formalized as:

$$\theta^* = \arg\min_{\theta} \sum_{T_i^{tr} \in \mathcal{T}_{tr}} \mathcal{L}_{query}(f_{\theta_i'}(Q_i^{tr}), y_{Q_i^{tr}})$$

where θ_i' represents the adapted parameters obtained through inner-loop gradient steps on S_i^{tr}:

$$\theta_i' = \theta - \alpha \nabla_\theta \mathcal{L}_{support}(f_\theta(S_i^{tr}), y_{S_i^{tr}})$$

Here, α is the inner learning rate, and for simplicity, we show one gradient step (the formulation naturally extends to multiple steps). $y_{S_i^{tr}}$ and $y_{Q_i^{tr}}$ represent the ground truth labels for the support and query sets, respectively.

Meta-Testing During meta-testing, the learned meta-parameters θ^* are adapted to novel tasks $\mathcal{T}_{ts}$ containing previously unseen classes. For each test task T_j^{ts} with support set S_j^{ts} and query set Q_j^{ts}, the model performs rapid adaptation through gradient updates on S_j^{ts} to obtain task-specific parameters θ'_j, which are then evaluated on Q_j^{ts}.

4.4 Training Procedure

The training follows the standard MAML protocol with bi-level optimization. The inner loop adapts model parameters on each task's support set through gradient descent:

$$\theta'_i = \theta - \alpha \nabla_\theta \mathcal{L}_{support}(f_\theta(S_i^{tr}), y_{S_i^{tr}})$$

The outer loop updates the meta-parameters based on the query set performance using the adapted parameters:

$$\theta \leftarrow \theta - \beta \nabla_\theta \sum_{T_i^{tr} \in \mathcal{T}_{tr}} \mathcal{L}_{query}(f_{\theta'_i}(Q_i^{tr}), y_{Q_i^{tr}})$$

where β is the meta-learning rate and θ'_i represents the inner-loop adapted parameters for task T_i^{tr}. This bi-level optimization enables the model to learn initial parameters that facilitate rapid adaptation to new tasks while maintaining good generalization performance across diverse graph learning scenarios.

5 Evaluation

In the following, the datasets used in this work are outlined, model architecture, and evaluation metrics employed to assess the effectiveness of learning unseen classes against state of the art frameworks. This project's source code is publicly available in a Github repository [1].

Datasets

In this study, we evaluate the proposed method on three benchmark graph datasets for node classification: Cora and Citeseer [26], which are considered small- to medium-scale datasets, and CoraFull [1], a large-scale variant of Cora. Evaluating across datasets of varying sizes is essential to assess the generalizability and scalability of the model, ensuring its effectiveness on both lightweight and complex graph learning tasks. All three datasets are citation networks, where the goal is to predict the category of a publication or paper. The table below summarizes the key statistics of these datasets.

Dataset	Nodes	Edges	Classes	Features
Cora	2708	5429	7	1433
Citeseer	3327	4552	6	3703
CoraFull	19793	126842	70	8710

[1] https://github.com/sirajummprince/COS-META

Baselines

The experimental results of the proposed framework have been compared with the experimental results with those of the state-of-the-art FSL node classification methods, including MAML [4], ProtoNet [17], Meta-GNN [28], GPN [3], AMM-GNN [19], G-Meta [8], TENT [20], InfoGCL [24] & COSMIC [22].

Implementation Details

All experiments were conducted in a personal workspace with the following configuration: Ubuntu 24.04.2 LTS, Linux kernel 6.11.0-19-generic, 13$^{\text{th}}$ Gen Intel Core i7-13700HX processor, 32 GiB RAM, and an NVIDIA GeForce RTX 4050 Laptop GPU (6 GB). For episodic training, 2-hop subgraphs have been used to maximize the information in the embeddings of the datasets. All experiments were conducted under various $\mathcal{N}$-way $\mathcal{K}$-shot settings. The hyperparameters used for training and testing the GNN are provided below:

Parameter	**Value**
Hidden Dimension	128
Number of Epochs	200
Learning Rate	0.001
Weight Decay	1×10^{-5}
Number of Training Tasks	100(Cora) 100(CiteSeer) 1000(CoraFull)
Number of Validation Tasks	20(Cora) 20(CiteSeer) 200(CoraFull)
Number of Testing Tasks	40(Cora) 40(CiteSeer) 400(CoraFull)
Batch Size	50
Negative Sampling for Training	1 - 3
λ	0.5

Results

Table 1 shows the performance of the proposed framework evaluated against baseline models on the few-shot node classification task under 2-way 1-shot and 2-way 5-shot configurations. Additionally, Table 2 presents results for 3-way 3-shot and 3-way 5-shot configurations, which are not covered by the baselines.

The results presented in Tables 1 demonstrate the superior performance of the proposed COS-META model across a variety of few-shot node classification settings on the Cora and CiteSeer datasets. In the standard 2-way 5-shot setting, COS-META significantly outperforms all prior models, achieving accuracies of 77.50% and 92.99% on Cora, and 63.99% and 84.41% on CiteSeer for the 2-way 1-shot and 2-way 5-shot settings, respectively. These results surpass the best-performing baselines, such as G-Meta (67.03% and 82.30% on Cora) and TENT (62.75% and 72.95% on CiteSeer).

In extended few-shot configurations in Table 2, COS-META continues to deliver consistently high performance, achieving better results in more complex tasks such as 2-way 5-shot with scores of 92.99% on Cora, 84.41% on CiteSeer 74.13% on CoraFull. These results highlight the model's strong generalization

Model	**Cora**		**CiteSeer**	
FSL Settings	2-way 1-shot	2-way 5-shot	2-way 1-shot	2-way 5-shot
MAML [4]	53.13 ± 2.26	57.39 ± 2.23	52.39 ± 2.20	54.13 ± 2.18
ProtoNet [17]	53.04 ± 2.36	57.92 ± 2.34	52.51 ± 2.44	55.69 ± 2.27
Meta-GNN [28]	65.27 ± 2.93	72.51 ± 1.91	56.14 ± 2.62	67.34 ± 2.10
GPN [3]	62.61 ± 2.71	76.39 ± 2.33	53.10 ± 2.39	63.09 ± 2.50
AMM-GNN [19]	65.23 ± 2.67	82.30 ± 2.07	54.53 ± 2.51	62.93 ± 2.42
G-Meta [8]	67.03 ± 3.22	80.05 ± 1.98	55.15 ± 2.68	64.53 ± 2.35
TENT [20]	53.05 ± 2.78	62.15 ± 2.13	62.75 ± 3.23	72.95 ± 2.13
I-GNN [15]	54.45 ± 3.13	65.18 ± 2.21	58.70 ± 3.17	65.60 ± 2.59
COS-META	**77.5 ± 4.71**	**92.99 ± 1.44**	**63.99 ± 2.76**	**84.41 ± 3.65**

Table 1: Performance comparison (Mean Accuracy(%)) on the Cora and CiteSeer datasets under various $\mathcal{N}$-way $\mathcal{K}$-shot few-shot settings, as reported in [15, 18]. The best score is in bold and the second best is underlined. COS-META, the proposed approach, consistently outperforms the baselines.

Dataset	**Model**	**2-Way**			**3-Way**	
		1-Shot	**3-Shot**	**5-Shot**	**3-Shot**	**5-Shot**
Cora	COS-META	77.5 ± 4.71	88.50 ± 1.77	92.99 ± 1.44	76.33 ± 2.11	76.66 ± 7.20
CiteSeer	COS-META	63.99 ± 2.76	83.50 ± 3.38	84.41 ± 3.65	71.33 ± 5.79	76.83 ± 2.90
CoraFull	COS-META	67.23 ± 1.92	76.40 ± 2.01	74.13 ± 2.76	65.85 ± 2.91	65.45 ± 3.90

Table 2: Additional performance analysis (Mean Accuracy ± std (%)) of COS-META on the Cora, CiteSeer and CoraFull datasets for various $\mathcal{N}$-way $\mathcal{K}$-shot few-shot settings.

capability as the number of classes and examples increases. Even in the more challenging 3-shot settings, where limited examples are available per class, COS-META maintains high accuracy (e.g., 76.66% on Cora, 76.83% on CiteSeer and 65.45% on CoraFull in 3-way 5-shot), showcasing its robustness.

On Cora, COS-META achieves its highest accuracy in the 2-way 5-shot setting with 92.99%, showing clear benefits from increased support examples. Accuracy improves steadily with more shots, jumping from 77.5% (1-shot) to 88.50% and 92.99% in the 3-shot and 5-shot setting. A similar trend appears in the 3-way setting, though with slightly lower performance due to increased task complexity. On CiteSeer, accuracy rises from 63.99% (1-shot) to 84.41% (5-shot, 2-way), with the 3-way 5-shot setting reaching 76.83%, reflecting the model's generalizability. CoraFull, a large-scale and more complex dataset, presents a greater challenge. Interestingly, performance slightly decreases in the 5-shot settings compared to the 3-shot case, suggesting potential overfitting or limitations in modeling more diverse or dense node relationships in larger graphs.

Overall, COS-META shows strong performance across few-shot settings, maintaining accuracy as task complexity grows. Results indicate that smaller $\mathcal{N}$ values typically yield better accuracy, as fewer classes reduce classification com-

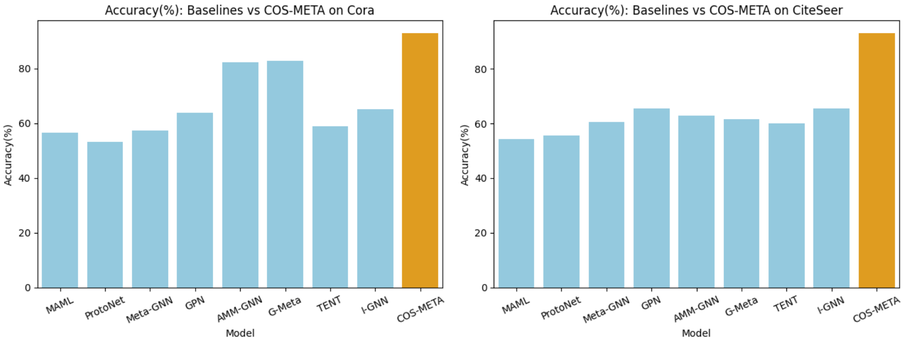

Fig. 4: Accuracies(%) of the baseline models vs. COS-META. COS-META outperform the baseline models on Cora, and CiteSeer datasets.

plexity. Despite this, COS-META avoids major performance drops thanks to its contrastive meta-learning strategy, which ensures semantic consistency between support and query pairs. Additionally, higher $\mathcal{K}$ values consistently improve results, showing the model's ability to leverage more support examples effectively.

Figures 4 provides an overall summary of COS-META's performance compared to existing baseline models across Cora, CiteSeer datasets. These results reflect improvements of at least 10.47%, 10.69%, and 11.46% over the best-performing baselines (e.g., G-Meta and TENT).

These findings highlight two important insights: (1) FSL strategies are highly competitive even to traditional supervised GNNs and (2) COS-META, with its meta-learning-based design, achieves state-of-the-art performance, demonstrating both efficiency and scalability in low-data regimes.

6 Conclusion

This paper addresses the problem of few-shot node classification in graph data mining by proposing a contrastive meta-learning framework, COS-META. The framework employs a MAML approach, combined with subgraph-based contrastive learning and a node-level cross-entropy objective, to learn robust and generalizable representations. This two-step optimization process encourages the learning of transferable features, enhancing the effectiveness of meta-learning when only a small amount of labeled data is available. Extensive experiments are conducted on three benchmark citation networks, i.e., Cora, CiteSeer, and CoraFull, under various $\mathcal{N}$-way $\mathcal{K}$-shot settings. The results demonstrate the effectiveness of the proposed approach. For future work, the aim to develop learnable subgraph generation mechanisms that adaptively select the most informative structures for a given task or dataset, potentially leveraging reinforcement learning. While the approached method demonstrated strong performance on citation networks, its effectiveness on heterogeneous graphs with multiple node and edge types remains to be explored.

Limitation

Despite promising results, this approach has several limitations that highlight directions for future work. One key challenge is the computational complexity associated with processing contrastive structures, especially in large graphs. Although several optimization mechanisms have been implemented, there remains a need for more efficient subgraph extraction and processing algorithms. Due to limited CUDA resources, it was not possible to evaluate the proposed method on larger datasets and other $\mathcal{N}$-way $\mathcal{K}$-shots scenarios such as ogbn-arxiv [22], Coauthor-CS [22], DBLP [22] etc. Nonetheless, COS-META framework outperformed baselines even with these constraints.

References

1. Bojchevski, A., Günnemann, S.: Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. arXiv preprint arXiv:1707.03815 (2017)
2. Chuang, C.Y., Robinson, J., Lin, Y.C., Torralba, A., Jegelka, S.: Debiased contrastive learning. Advances in neural information processing systems **33**, 8765–8775 (2020)
3. Ding, K., Wang, J., Li, J., Shu, K., Liu, C., Liu, H.: Graph prototypical networks for few-shot learning on attributed networks. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 295–304 (2020)
4. Finn, C., Abbeel, P., Levine, S.: Model-agnostic meta-learning for fast adaptation of deep networks. In: International conference on machine learning. pp. 1126–1135. PMLR (2017)
5. Gilmer, J., Schoenholz, S.S., Riley, P.F., Vinyals, O., Dahl, G.E.: Neural message passing for quantum chemistry. In: International conference on machine learning. pp. 1263–1272. PMLR (2017)
6. Hamilton, W., Ying, R., Leskovec, J.: Inductive representation learning on large graphs. Advances in neural information processing systems **30** (2017)
7. Hospedales, T., Antoniou, A., Micaelli, P., Storkey, A.: Meta-learning in neural networks: A survey. IEEE transactions on pattern analysis and machine intelligence **44**(9), 5149–5169 (2021)
8. Huang, K., Zitnik, M.: Graph meta learning via local subgraphs. Advances in neural information processing systems **33**, 5862–5874 (2020)
9. Ju, W., Qin, Y., Yi, S., Mao, Z., Zheng, K., Liu, L., Luo, X., Zhang, M.: Zero-shot node classification with graph contrastive embedding network. Transactions on Machine Learning Research (2023)
10. Kipf, T.N., Welling, M.: Semi-supervised classification with graph convolutional networks. In: International Conference on Learning Representations (2016)
11. Le-Khac, P.H., Healy, G., Smeaton, A.F.: Contrastive representation learning: A framework and review. IEEE Access **8**, 193907–193934 (2020). https://doi.org/10.1109/ACCESS.2020.3031549
12. Lin, Y.C., Neville, J.: Improving node representation by boosting target-aware contrastive loss. arXiv preprint arXiv:2410.03901 (2024)
13. Liu, H., Feng, J., Kong, L., Tao, D., Chen, Y., Zhang, M.: Graph contrastive learning meets graph meta learning: A unified method for few-shot node tasks. In: Proceedings of the ACM on Web Conference 2024. pp. 365–376 (2024)

14. Liu, Y., Li, M., Li, X., Giunchiglia, F., Feng, X., Guan, R.: Few-shot node classification on attributed networks with graph meta-learning. In: Proceedings of the 45th international ACM SIGIR conference on research and development in information retrieval. pp. 471–481 (2022)
15. Mathavan, H., Tan, Z., Mudiam, N., Liu, H.: Inductive linear probing for few-shot node classification. In: International Conference on Social Computing, Behavioral-Cultural Modeling and Prediction and Behavior Representation in Modeling and Simulation. pp. 274–284. Springer (2023)
16. Oord, A.v.d., Li, Y., Vinyals, O.: Representation learning with contrastive predictive coding. arXiv preprint arXiv:1807.03748 (2018)
17. Snell, J., Swersky, K., Zemel, R.: Prototypical networks for few-shot learning. Advances in neural information processing systems **30** (2017)
18. Tan, Z., Wang, S., Ding, K., Li, J., Liu, H.: Transductive linear probing: A novel framework for few-shot node classification. In: Learning on Graphs Conference. pp. 4–1. PMLR (2022)
19. Wang, N., Luo, M., Ding, K., Zhang, L., Li, J., Zheng, Q.: Graph few-shot learning with attribute matching. In: Proceedings of the 29th ACM International Conference on Information & Knowledge Management. pp. 1545–1554 (2020)
20. Wang, S., Ding, K., Zhang, C., Chen, C., Li, J.: Task-adaptive few-shot node classification. In: Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 1910–1919 (2022)
21. Wang, S., Dong, Y., Ding, K., Chen, C., Li, J.: Few-shot node classification with extremely weak supervision. In: Proceedings of the Sixteenth ACM International Conference on Web Search and Data Mining. pp. 276–284 (2023)
22. Wang, S., Tan, Z., Liu, H., Li, J.: Contrastive meta-learning for few-shot node classification. In: Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining. pp. 2386–2397 (2023)
23. Wang, Z., Cao, L., Lin, W., Jiang, M., Tan, K.C.: Robust graph meta-learning via manifold calibration with proxy subgraphs. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 37, pp. 15224–15232 (2023)
24. Xu, D., Cheng, W., Luo, D., Chen, H., Zhang, X.: Infogcl: Information-aware graph contrastive learning. Advances in Neural Information Processing Systems **34**, 30414–30425 (2021)
25. Xu, K., Hu, W., Leskovec, J., Jegelka, S.: How powerful are graph neural networks? In: International Conference on Learning Representations (2018)
26. Yang, Z., Cohen, W., Salakhudinov, R.: Revisiting semi-supervised learning with graph embeddings. In: International Conference on Machine Learning. pp. 40–48. PMLR (2016)
27. Zhao, F., Wang, D.: Multimodal graph meta contrastive learning. In: Proceedings of the 30th ACM International Conference on Information & Knowledge Management. pp. 3657–3661 (2021)
28. Zhou, F., Cao, C., Zhang, K., Trajcevski, G., Zhong, T., Geng, J.: Meta-gnn: On few-shot node classification in graph meta-learning. In: Proceedings of the 28th ACM International Conference on Information and Knowledge Management. pp. 2357–2360 (2019)