LOW-RANK FEW-SHOT NODE CLASSIFICATION BY NODE-LEVEL GRAPH DIFFUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

In this paper, we propose a novel node-level graph diffusion method with low-rank feature learning for few-shot node classification (FSNC), termed Low-Rank Few-Shot Graph Diffusion Model or LR-FGDM. LR-FGDM first employs a novel Few-Shot Graph Diffusion Model (FGDM) as a node-level graph generative method to generate an augmented graph with an enlarged support set, then performs low-rank transductive classification to obtain the few-shot node classification results. Our graph diffusion model, FGDM, comprises two components, the Hierarchical Graph Autoencoder (HGAE) with an efficient hierarchical edge reconstruction method and a new prototypical regularization, and the Latent Diffusion Model (LDM). The low-rank regularization is robust to the noise inherently introduced by the diffusion model and empirically inspired by the Low Frequency Property. We also provide a strong theoretical guarantee justifying the low-rank regularization for the transductive classification in few-shot learning. Extensive experimental results evidence the effectiveness of LR-FGDM for few-shot node classification, which outperforms the current state-of-the-art. The code of the LR-FGDM is available at https://anonymous.4open.science/r/LR-FGDM/.

1 INTRODUCTION

Graph Neural Networks (GNNs) (Kipf & Welling, 2016b; Hamilton et al., 2017) are widely used for semi-supervised node classification (Veličković et al., 2018), but their effectiveness relies on ample labeled data, which is often costly or impractical to obtain. This challenge motivates few-shot node classification (FSNC), where only a few labeled nodes per class are available. Most FSNC methods (Zhou et al., 2019; Ding et al., 2020; Wang et al., 2022; Huang & Zitnik, 2020; Qian et al., 2021; Lan et al., 2020; Liu et al., 2021b) follow a meta-learning framework (Finn et al., 2017; Snell et al., 2017) to generalize across tasks. More recent approaches (Tan et al., 2022; Huang & Zitnik, 2020) leverage self-supervised Graph Contrastive Learning (GCL) (Mo et al., 2022; Jin et al., 2021) to train simple classifiers from pre-trained embeddings, achieving superior performance despite using only unlabeled data. However, all existing methods remain constrained by the limited support set size. Although techniques like mix-up (Liu et al., 2025b) and random perturbation (Wu et al., 2022) offer marginal gains, the potential of generative models to synthesize support nodes remains underexplored. Building on the success of diffusion models in vision, recent works have extended them to synthetic graph generation. Methods like EDP-GNN (Niu et al., 2020) and its successors (Jo et al., 2022; Haefeli et al., 2022; Vignac et al., 2023; Limnios et al., 2023) adapt diffusion models to capture both discrete and continuous graph properties, generating realistic structures that align well with real-world networks. However, these approaches focus on graph-level generation and do not support structured node- or edge-level synthesis. Node-level graph augmentation typically relies on GANs (Jia et al., 2023; Wu et al., 2023; Wang et al., 2018; Liang et al., 2020; Yang et al., 2019) to generate minority class nodes in imbalanced graphs, despite known issues with GAN training instability and poor distributional matching (Dhariwal & Nichol, 2021).

In this work, we propose a novel node-level graph diffusion method with low-rank feature learning for FSNC, termed Low-Rank Few-Shot Graph Diffusion Model or LR-FGDM. LR-FGDM employs a novel Few-Shot Graph Diffusion Model (FGDM) to generate an augmented graph with an enlarged support set. The FGDM in LR-FGDM consists of two components, including the Hierarchical Graph Autoencoder (HGAE) with an efficient hierarchical edge reconstruction method and the Latent Diffusion Model (LDM). The HGAE learns compact latent node features for LDM by incorporating a prototypical regularization to encourage semantic structure in the latent space. The hierarchical edge reconstruction method enables efficient reconstruction of the edges connecting to a node from the latent space in a hierarchical manner to avoid the quadratic complexity in edge reconstruction of the regular GAE (Kipf & Welling, 2016a). Given a FSNC task, the FGDM first generates the synthetic graph structure, consisting of the synthetic support nodes and the edges connecting to the original graph. The synthetic graph structure is then incorporated into the original graph, forming an augmented graph with an enlarged support set consisting of the original support nodes and the synthetic support nodes.

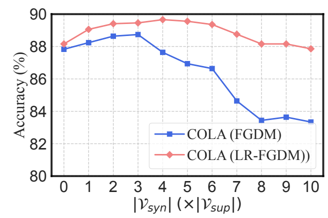

Figure 1: 5-way 5-shot code classification accuracies on Cora-Full trained with different numbers of synthetic support nodes added. $\mathcal{V}_{\text{syn}}$ and $\mathcal{V}_{\text{sup}}$ denote the synthetic and original support sets. COLA (LR-FGDM) is trained on the augmented graph by FGDM with the low-rank regularization, while COLA (FGDM) is trained without the regularization.

Although prior methods enlarge the support set via random perturbations (Gao et al., 2023b) or mix-up (Liu et al., 2025b), they fail to generate faithful graph structures, often assigning edges to synthetic nodes by reusing neighbors of real nodes. In contrast, our FGDM jointly encodes nodes and edges into a semantically regularized latent space for training the latent diffusion model, capturing the true joint distribution of features and structure. As shown by the Frechet Node Distance (FND) and Frechet Edge Distance (FED) in Section F.8, FGDM produces more realistic synthetic nodes and edges than existing augmentation methods. Let $\mathcal{V}_{\text{syn}}$ and $\mathcal{V}_{\text{sup}}$ denote the set of synthetic support nodes and the original support set. As shown in Figure 1, while adding synthetic support nodes with FGDM improves COLA's performance when $|\mathcal{V}_{\text{syn}}| \leq 3|\mathcal{V}_{\text{sup}}|$, further increasing the number of synthetic nodes beyond this threshold leads to a sharp performance drop. This is due to inherent noise in the diffusion generation process (Ho et al., 2020; Fu et al., 2024; Azizi et al., 2023; He et al., 2022), including randomness in the forward process (Ho et al., 2020) and errors in class conditioning (Fu et al., 2024). To this end, we propose a low-rank learning method inspired by the widely-studied Low Frequency Property (LFP) (Rahaman et al., 2019; Arora et al., 2019; Cao et al., 2021; Choraria et al., 2022; Wang et al., 2024; 2025), which suggests that the projection of the ground truth class labels mostly concentrates on the top eigenvectors of the kernel gram matrix of the model, to be detailed in Section 3.4. Motivated by LFP, the truncated nuclear norm (TNN) is added as a low-rank regularization term to the training loss function of the transductive few-shot node classifier on the augmented graph. It is observed from Figure 1 that the COLA trained with the low-rank regularization performs much better than the regular COLA when the synthetic nodes added in the augmented graph are more than $3|\mathcal{V}_{\text{sup}}|$.

Existing graph few-shot learning methods (Huang & Zitnik, 2020; Wang et al., 2023; Ma et al., 2025; Zhao et al., 2025) show that training graph encoders without label supervision yields better generalization to novel classes. However, the diffusion-based generator DoG (Wang et al., 2025) relies on class labels as conditioning signals during both training and generation, which is problematic in few-shot settings where test-time labels are disjoint from training. Although semi-supervised K-means (Basu et al., 2002; Bair, 2013) can be applied on the node attributes to obtain pseudo labels for conditioning, as illustrated in Figure 2 (a), it often leads to semantic drift and unreliable conditioning due to entangled base and novel class semantics. In contrast, our LR-FGDM conditions the diffusion model on cluster prototypes jointly learned with the prototypical regularization, as illustrated in Figure 2 (b), rather than pseudo labels, to avoid this issue. As another significant difference from DoG (Wang et al., 2025), a new prototypical regularization is introduced to HGAE to improve cluster separability in the latent space, making the prototype-based conditioning semantically aligned and robust.

Contributions. The contributions of this paper are presented as follows.

First, we propose the Low-Rank Few-Shot Graph Diffusion Model (LR-FGDM), a novel generative framework for FSNC tasks by synthesizing labeled support nodes and the associated edges through a node-level graph diffusion model, Few-Shot Graph Diffusion Model (FGDM). The FGDM features a new Hierarchical Graph Autoencoder (HGAE) that incorporates prototypical regularization to structure the latent space semantically, and our LDM uses the prototypes instead of class labels such as those in DoG (Wang et al., 2025) as the conditioning features. To mitigate the inherent inefficiency of the quadratic complexity in edge reconstruction over the entire graph, FGDM also introduces a hierarchical edge reconstruction method, which hierarchically reconstructs the edges connecting to each synthetic support node. While prior methods (Liu et al., 2025b; Wu et al., 2022) have shown promising results in enlarging the support set for FSNC, Table 7 in Section F.2 shows that LR-FGDM substantially outperforms existing support set augmentation approaches. Moreover, we introduce the Frechet Node Distance (FND) and the Frechet Edge Distance (FED) to validate the faithfulness of the synthetic support nodes and the associated edges in Section F.8 of the appendix.

Second, we introduce a low-rank regularization method in LR-FGDM for the transductive node classifier trained on the augmented graph, which is empirically motivated by the widely studied Low Frequency Property (LFP) in deep learning (Rahaman et al., 2019; Arora et al., 2019; Cao et al., 2021; Choraria et al., 2022; Wang et al., 2024; 2025) and theoretically justified by a novel generalization bound for the transductive few-shot node classifier in Theorem 3.1. The low-rank regularization promotes lower kernel complexity (KC) thus leads to a lower generalization bound for the test loss of the transductive classifier. Section F.3 demonstrates the much lower KC and the upper bound for the test loss of LR-FGDM compared to the baseline without low-rank regularization. Furthermore, as shown in Table 1 in Section 4.2, Table 9 in Section F.4, and Table 5 in Section F.1 of the appendix, LR-FGDM consistently outperforms state-of-the-art FSNC methods across multiple graph benchmarks.

2 Related Works

2.1 Few-Shot Node Classification (FSNC)

While GNNs for node classification are commonly trained in a semi-supervised fashion (Kipf & Welling, 2016b), many efforts (Sun et al., 2020; Hamilton et al., 2017; Veličković et al., 2018) aim to reduce label reliance; however, they struggle with unseen classes at inference, motivating the study of FSNC. Most previous FSNC methods (Zhou et al., 2019; Finn et al., 2017; Yao et al., 2020; Snell et al., 2017; Huang & Zitnik, 2020; Wang et al., 2022; Wu et al., 2024; Zhang et al., 2025a) adopt a meta-learning framework by training the FSNC model through a series of meta tasks. More recently, several works have incorporated contrastive learning into meta-learning to enhance task-specific representation learning. (Liu et al., 2021a) and CPLAE (Gao et al., 2021) apply supervised contrastive losses within meta-tasks using augmented views, while PsCo (Jang et al., 2023) and MetaContrastive (Ni et al., 2021) introduce the momentum encoder for unsupervised contrastive meta-learning. COLA (Huang & Zitnik, 2020) contrasts support and query prototypes to promote class-level consistency, and COSMIC (Wang et al., 2023) leverages multi-view contrastive regularization between structural and feature-based representations.

2.2 Graph Diffusion Models and Generative Data Augmentation on Graph

Score-based diffusion models (Song et al., 2021b) have achieved state-of-the-art performance in diverse generative tasks (Ho et al., 2020; Song & Ermon, 2019; Gao et al., 2023a; Rombach et al., 2022; Baranchuk et al., 2022; Song et al., 2021c;a; Song & Ermon, 2020; Rombach et al., 2022). Graph diffusion models have emerged for synthetic graph generation (Niu et al., 2020; Haefeli et al., 2022; Jo et al., 2022; Zhou et al., 2024), with early works (Jo et al., 2022; Haefeli et al., 2022; Vignac et al., 2023) designing discrete diffusion processes over adjacency matrices. SaGess (Limnios et al., 2023) performs conditional generation of graphs inspired by latent diffusion models (LDMs). However, these models primarily target graph-level generation, limiting their utility in node-level tasks such as FSNC. To enhance the performance of GNNs, node-level data augmentation has been applied to structure (Zhao et al., 2021b; Rong et al., 2020; Feng et al., 2022; Lai et al., 2024), features (You et al., 2020; Kong et al., 2022; Azad & Fang, 2024), and labels (Han et al., 2022; Wang et al., 2021; Verma et al., 2021; Zhao et al., 2024b). In FSNC, recent methods enhance the support set by perturbing node features and leveraging pseudo-labeled queries (Wu et al., 2022), or by using LLM-based prompting to generate synthetic support nodes for text-attributed graphs (Zhang et al., 2025b). Generative data augmentation has been used to enhance GNN performance by generating synthetic nodes and edges to address class imbalance and enrich minority class features and connectivity (Zhao et al., 2021b; Zhou et al., 2024; Qu et al., 2021; Zhao et al., 2021a; Hsu et al., 2024; Gao et al., 2023b; Hsu et al., 2023). However, these approaches often rely on GANs (Jia et al., 2023; Wu et al., 2023; Wang et al., 2018; Liang et al., 2020; Yang et al., 2019), which suffer from training instability and poor alignment with real data distributions (Dhariwal & Nichol, 2021). To the best of our knowledge, our proposed FGDM is among the first to synthesize synthetic graph structures via diffusion models in a principled manner for FSNC.

3 FORMULATION

We aim to boost the performance of existing FSNC methods by augmenting the support set in a few-shot task, thereby alleviating the data scarcity in each novel class.

The Pipeline of Integrating LR-FGDM with an Existing Few-Shot Learning Method. The proposed LR-FGDM serves as a plug-in module to enhance existing FSNC methods like COSMIC (Wang et al., 2023) and COLA (Huang & Zitnik, 2020) by augmenting the support set. LR-FGDM consists of three steps for FSNC: (1) training FGDM, which includes learning a Hierarchical Graph Autoencoder (HGAE) on the original graph and a Latent Diffusion Model (LDM) on its latent space; (2) generating an augmented graph by injecting synthetic support nodes and their edges into the original graph; and (3) applying an existing FSNC method to learn node embeddings, followed by training a low-rank transductive classifier on the augmented support set. Figure 2 (b) illustrates the training of the LR-FGDM, and Figure 4 (b) in the appendix illustrates the generation of the synthetic support nodes by LR-FGDM. LR-FGDM improves few-shot classification by expanding the support set, leveraging the benefits of stronger supervision and reduced overfitting as shown in prior augmentation studies (Wu et al., 2022; Liu et al., 2025b).

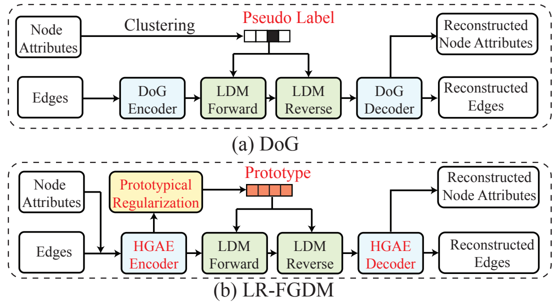

Figure 2: Figure (a) and (b) illustrate the training of the DoG (Wang et al., 2025) and the training of the LR-FGDM, respectively, with the difference between LR-FGDM and DoG marked in red. Figure 4 in the appendix illustrates the generation of the synthetic support nodes by LR-FGDM and DoG (Wang et al., 2025).

Our LR-FGDM contains two components, which are the generation of an augmented graph with synthetic support data by FGDM, and few-shot learning with low-rank transductive classification on the augmented graph. Our FGDM features a novel Hierarchical Graph Autoencoder (HGAE) with an efficient hierarchical edge reconstruction method and a new prototypical regularization, detailed in Section 3.2. Then, the generation of an augmented graph with the synthetic support nodes and edges is explained in Section 3.3. The low-rank transductive linear classifier for few-shot classification with theoretical guarantee is detailed in Section 3.4.

3.1 PRELIMINARIES

Few-Shot Node Classification (FSNC). FSNC assumes disjoint label sets across splits, denoted as $\mathcal{C}_{\text{train}}$, $\mathcal{C}_{\text{val}}$, and $\mathcal{C}_{\text{test}}$ (Huang & Zitnik, 2020; Luo et al., 2024; Wang et al., 2023; Zhao et al., 2024a). An n-way task requires the model to classify nodes into n distinct classes randomly sampled from $\mathcal{C}_{\text{test}}$, with only k labeled instances per class provided in the support set. Each task consists of a labeled support set of $n \times k$ nodes and an unlabeled query set. The support set guides the model to learn a transductive classifier to predict the labels of nodes in the query set.

Attributed Graph and Notations. An attributed graph with N nodes is denoted by $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{X})$. Here, $\mathcal{V} = \{v_1, v_2, \ldots, v_N\}$ represents the nodes, and $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ represents the edges. Node attributes are given by $\mathbf{X} \in \mathbb{R}^{N \times D}$, where each row $\mathbf{X}_i \in \mathbb{R}^D$ corresponds to the attributes of node v_i. The adjacency matrix $\mathbf{A} \in \{0,1\}^{N \times N}$ of graph $\mathcal{G}$ defines connections. Each row $\mathbf{A}_i$ represents the connections of node v_i. The neighborhood $\mathcal{N}(i) = \{j \mid \tilde{\mathbf{A}}_{i,j} = 1\}$ includes node v_i itself and all nodes connected to v_i. The notation $[N]$ denotes all natural numbers from 1 to N inclusive. $[\mathbf{A}]_i$ stands for the i-th row of a matrix $\mathbf{A}$. $\|\cdot\|_p$ denotes the p-norm of a vector or a matrix.

3.2 FEW-SHOT GRAPH DIFFUSION MODEL (FGDM)

Hierarchical Graph Autoencoder (HGAE) with Prototypical Regularization. To encode a node v_i, we first generate a latent feature of the node attribute $\mathbf{X}_i$ as $f(\mathbf{X}_i)$, where $f(\cdot)$ is a Multi-Layer Perceptron (MLP) layer. To incorporate the information from the edges connected to v_i, we add positional embeddings (Ma et al., 2021; You et al., 2019) to the node attributes of v_i's neighbors. For each neighbor $j \in \mathcal{N}(i)$, we modify the node attributes as $\mathbf{X}'_j = \mathbf{X}_j + \text{pos}(j)$, where $\text{pos}(\cdot)$ is a function converting the position index into an embedding vector (Vaswani et al., 2017). We apply two Graph Attention Network (GAT) (Veličković et al., 2018) layers to aggregate the information in $\{\mathbf{X}_j^{'} \mid j \in \mathcal{N}(i)\}$ into a single latent feature $\mathbf{Z}_i^{'}$. Next, we concatenate $\mathbf{Z}_i^{'}$ with $f(\mathbf{X}_i)$ to obtain the latent feature of node v_i by $\mathbf{Z}_i = f'(\mathbf{Z}_i^{'} \| f(\mathbf{X}_i))$, where f' is another MLP layer encoding the concatenated features to the latent space of LDM with lower dimension D'. After encoding a node v_i in the graph to a latent feature $\mathbf{Z}_i$, the decoder of the HGAE reconstructs the node attribute $\widehat{\mathbf{X}}_i$ by three consecutive MLP layers and its associated edges $\widehat{\mathbf{A}}_i$ by the hierarchical edge reconstruction method to be introduced later.

Prototypical Regularization. Existing FSNC methods (Snell et al., 2017; Laenen & Bertinetto, 2021; Ding et al., 2020; Lin et al., 2022) show that prototypical learning improves node embeddings by promoting intra-class compactness and inter-class separability. To align latent features of semantically similar nodes, we add a prototypical regularization to the HGAE loss, encouraging nodes within the same cluster to approach shared prototypes. The cluster assignments are obtained via semi-supervised K-means (Basu et al., 2002; Bair, 2013) utilizing labeled nodes to guide clustering while incorporating unlabeled nodes for better generalization to unseen classes. Let $\mathbf{p}_c \in \mathbb{R}^{D'}$ represent the prototype of cluster $c \in [K]$, where K is the number of prototypes. The prototypical regularization loss is defined as $\mathcal{L}_{\text{proto}} = \sum_{i=1}^{N} \left\|\mathbf{Z}_i - \mathbf{p}_{\pi(i)}\right\|^2$, where $\mathbf{p}_{\pi(i)} = \frac{1}{|\mathcal{V}_{\pi(i)}|} \sum_{j \in \mathcal{V}_{\pi(i)}} \mathbf{Z}_j$. $\pi(i)$ is the cluster index of node v_i and $\mathcal{V}_{\pi(i)}$ is the set of nodes in the cluster $\pi(i)$. Unlike prior works (Ding et al., 2020; Lin et al., 2022) that use prototypes for query classification in FSNC, we employ prototypes as a regularization to learn semantically coherent latent features for improved conditional sampling in few-shot generation.

To address the quadratic complexity inherent in conventional GAEs (Zhai et al., 2018; Kipf & Welling, 2016a), we introduce a prototype-guided hierarchical edge reconstruction framework designed to promote efficient edge decoding. The edge reconstruction is conducted hierarchically based on clusters induced by learned prototype representations. We define an inter-cluster neighbor map $\mathbf{C} \in \{0,1\}^{N \times K}$, where $\mathbf{C}_{ik} = 1$ indicates that node v_i connects to at least one node within the cluster represented by prototype k. Additionally, an intra-cluster neighbor map $\mathbf{M} \in \{0,1\}^{N \times K \times M}$ is constructed, where $\mathbf{M}_{ikm} = 1$ signifies that node v_i is connected to the m-th node within cluster k, with M denoting the maximum number of nodes in any cluster. In contrast to the Bi-Level Neighborhood Decoder (BLND) employed in DoG (Wang et al., 2025) using balanced K-means applied to node attributes, our method leverages prototype clusters learned jointly with the encoder, because nodes in the same prototype cluster have similar latent features, thus tend to connect with each other. The structure of the network used for the hierarchical edge reconstruction in the HGAE with prototypical regularization is illustrated in Figure 5 in Section E of the appendix. Furthermore, Table 4 in Section E of the appendix demonstrates the superior performance of the hierarchical edge reconstruction when compared with the BLND.

Training the HGAE with Prototypical Regularization. For each node v_i, the hierarchical edge reconstruction method first reconstructs its inter-cluster neighbor map $\widehat{\mathbf{C}}_i$ with one MLP layer. After that, the predicted cluster indices $\mathcal{C}(i) = \{k \in [K] | \mathbf{C}_{ik} = 1\}$ are separately fed to an embedding layer to generate a set of class-conditional features $\mathcal{Z}(i) = \left\{g(k) \in \mathbb{R}^{D'} | k \in \mathcal{C}(i)\right\}$ using the class-conditional embedding method in Classifier-Free Guidance (Ho & Salimans, 2022), where g contains one text embedding layer followed by an MLP layer. Next, each of the class-conditional features $g(k) \in \mathcal{Z}(i)$ is concatenated with the latent feature of the other branch for decoding the intra-cluster neighbor map by $\widehat{\mathbf{M}}_{ik} = g'(\mathbf{Z}_i \| g(k))$, where g' is another MLP layer. The HGAE is trained by minimizing the sum of the node reconstruction loss, the hierarchical edge reconstruction loss, and the prototypical loss $\mathcal{L}_{\text{proto}}$ as follows,

$$\mathcal{L}_{\text{HGAE}} = \underbrace{\left\|\mathbf{X} - \widehat{\mathbf{X}}\right\|_2^2}_{\text{Node Reconstruction Loss}} + \underbrace{\left(\left\|\mathbf{C} - \widehat{\mathbf{C}}\right\|_2^2 + \left\|\mathbf{M} - \widehat{\mathbf{M}}\right\|_2^2\right)}_{\text{Hierarchical Edge Reconstruction Loss}} + \mathcal{L}_{\text{proto}}, \tag{1}$$

where $\|\cdot\|_2$ denotes the Euclidean norm. We perform a detailed complexity analysis of the hierarchical edge reconstruction method in Section B of the appendix. Table 12 in Section F.6 of the appendix demonstrates the improved efficiency of the proposed hierarchical edge reconstruction method compared to the decoder in a regular GAE.

Training the LDM. Once the HGAE with the hierarchical edge reconstruction method is trained, we obtain a set of latent representations $\mathbf{Z} = \{\mathbf{Z}_i \in \mathbb{R}^{D'} \mid v_i \in \mathcal{V}\}$ encoding both node attributes and edges. Traditional class-conditional diffusion models typically condition on class labels, including DoG (Wang et al., 2025), as illustrated in Figure 2 (a). However, in FSNC, the classes in the support and query sets are novel and disjoint from those used during training. As the diffusion model is trained prior to test-time adaptation, it cannot directly condition on these unseen class labels. To overcome this limitation, we leverage the prototypical regularization introduced in the HGAE, which encourages the latent representations $\mathbf{Z}_i$ to cluster around their respective prototype representations. These prototypes, computed as the mean latent representation of each cluster, serve as semantically meaningful and continuous conditioning signals. Instead of relying on discrete class labels, the LDM is conditioned directly on the corresponding prototype representation for each training node, enabling prototype-based conditional generation. As illustrated in Figure 2 (b), each latent feature is paired with its assigned prototype as the conditioning input under the Classifier-Free Guidance (CFG) framework (Ho & Salimans, 2022). This design allows the LDM to learn to generate latent features aligned with the semantic structure of the data without requiring access to class labels. The training algorithm of FGDM is presented in Algorithm 1 in Section G of the appendix.

3.3 Generation of Augmented Graph with Synthetic Support Data by FGDM

Generation of Synthetic Graph Structures with FGDM. Once the FGDM is trained, we aim to generate synthetic graph structures consisting of synthetic support nodes and edges connecting to the original graph. We first obtain the cluster label of each of the support nodes obtained from the prototypical regularization, which is used to get the prototype representation for the conditional generation of the synthetic graph structure, as illustrated in Figure 4 (b) in the appendix. Let $\mathcal{V}_{\text{sup}}$ be the original support nodes and $\mathcal{V}_{\text{syn}}$ be the synthetic support nodes. Let the node attributes of $\mathcal{V}_{\text{syn}}$ be $\mathbf{X}_{\text{syn}}$ and the affinity matrix encoding edges between the synthetic nodes and real nodes be $\mathbf{A}_{\text{syn}}$. Then the synthetic graph structure is denoted as $(\mathcal{V}_{\text{syn}}, \mathbf{X}_{\text{syn}}, \mathbf{A}_{\text{syn}})$. Let M be the number of nodes in the synthetic graph structure. The adjacency matrix of the augmented graph is $\mathbf{A}_{\text{aug}} = [\mathbf{A}\ \mathbf{A}_{\text{syn}}; \mathbf{A}_{\text{syn}}\ \mathbf{A}] \in \mathbb{R}^{(N+M)\times(N+M)}$, and the node attributes of the augmented graph is $\mathbf{X}_{\text{aug}} = [\mathbf{X}; \mathbf{X}_{\text{syn}}] \in \mathbb{R}^{(N+M)\times D}$. The augmented graph, which is the combination of the original graph $\mathcal{G}$ and the synthetic graph structure, is then denoted by $\mathcal{G}_{\text{aug}} = (\mathcal{V} \cup \mathcal{V}_{\text{syn}}, \mathbf{X}_{\text{aug}}, \mathbf{A}_{\text{aug}})$. Let $\mathcal{V}_{\mathcal{L}} = \mathcal{V}_{\text{sup}} \cup \mathcal{V}_{\text{syn}}$ denote the augmented support set. In practice, we generate the synthetic graph structures, consisting of $M = q \times n \times k$ synthetic support nodes and their edges connecting to the original graph, where q denotes the number of synthetic nodes generated per real support node. The value of q for different tasks on different datasets is selected by cross-validation as detailed in Section F.5. The augmented support set $\mathcal{V}_{\mathcal{L}}$ then consists of $(q+1)nk$ support nodes with $(q+1)k$ nodes in each of the n novel classes. The augmented graph $\mathcal{G}_{\text{aug}}$ is then encoded using existing few-shot graph encoders, such as COSMIC (Wang et al., 2023) and COLA (Huang & Zitnik, 2020), yielding the representation for all the nodes in the augmented graph, which is denoted as $\mathbf{H} \in \mathbb{R}^{(N+M)\times d}$. The generation of the augmented graph is described in Algorithm 2 in Section G of the appendix.

3.4 Low-Rank Transductive Linear Classifier for Few-Shot Learning

Due to the inherent stochasticity of diffusion models (Ho et al., 2020; Rombach et al., 2022), the synthetic graph structures generated by LR-FGDM may introduce noise, leading to semantic mismatches between synthetic support nodes and their labels (Azizi et al., 2023; He et al., 2022). To address this, we follow prior FSNC methods (Wang et al., 2023; Huang & Zitnik, 2020) by training a transductive node classifier on embeddings from a few-shot graph encoder. Motivated by the Low Frequency Property (LFP) (Rahaman et al., 2019; Arora et al., 2019; Cao et al., 2021; Choraria et al., 2022; Wang et al., 2024; 2025), which suggests that class labels concentrate on top eigenvectors of the model's kernel gram matrix, we introduce a novel low-rank regularization for the classifier with theoretical guarantees.

Notation Definition. Let $\mathbf{u} \in \mathbb{R}^{N'}$ be a vector, we use $[\mathbf{u}]_{\mathcal{A}}$ to denote a vector formed by elements of $\mathbf{u}$ with indices in $\mathcal{A}$ for $\mathcal{A} \subseteq [N']$. If $\mathbf{u}$ is a matrix, then $[\mathbf{u}]_{\mathcal{A}}$ denotes a submatrix formed by rows of $\mathbf{u}$ with row indices in $\mathcal{A}$. $\|\cdot\|_{\text{F}}$ denotes the Frobenius norm, and $\|\cdot\|_p$ denotes the p-norm. Let $\mathcal{V}_{\text{FS}}$ denote all the nodes from the n novel classes in an n-way k-shot task, and let $\mathcal{V}_{\mathcal{L}}$ and $\mathcal{V}_{\mathcal{U}}$ denote the labeled support set and the unlabeled query set in $\mathcal{V}_{\text{FS}}$. Let N denote the number of nodes in $\mathcal{V}_{\text{FS}}$. Let $\mathcal{V}_{\text{FS}} = \{v'_1, v'_2, \ldots, v'_N\}$, where v'_i is the i-th node in $\mathcal{V}_{\text{FS}}$. Let $\mathbf{y}_i \in \mathbb{R}^n$ be the ground-truth one-hot class label vector for v'_i in $\mathcal{V}_{\text{FS}}$, and define $\mathbf{Y}_{\text{FS}} \coloneqq [\mathbf{y}_1; \mathbf{y}_2; \ldots \mathbf{y}_N] \in \mathbb{R}^{N\times n}$ be the ground-truth label matrix defined on the n novel classes for all the nodes in $\mathcal{V}_{\text{FS}}$. Let $\mathbf{H}_{\text{FS}} \in \mathbb{R}^{N\times d}$ be the representations of all the nodes in $\mathcal{V}_{\text{FS}}$. We define $\mathbf{F}(\mathbf{W}) = \mathbf{H}_{\text{FS}}\mathbf{W}$ as the linear output of the transductive few-shot classifier with $\mathbf{W} \in \mathbb{R}^{d \times n}$ being the weight matrix. Let $\mathbf{K}$ be the gram matrix of the node representations $\mathbf{H}_{\text{FS}}$, which is calculated by $\mathbf{K} = \mathbf{H}_{\text{FS}}^{\top}\mathbf{H}_{\text{FS}} \in \mathbb{R}^{N \times N}$. Let $\left\{\widehat{\lambda}_i\right\}_{i=1}^{N}$ with $\widehat{\lambda}_1 \ge \widehat{\lambda}_2 \ldots \ge \widehat{\lambda}_{\min\{N,d\}} \ge \widehat{\lambda}_{\min\{N,d\}+1} = \ldots, = 0$ be the eigenvalues of $\mathbf{K}$.

Low-Rank Transductive Few-Shot Node Classification. In order to encourage the features $\mathbf{H}_{\text{FS}}$ or the gram matrix $\mathbf{K} = \mathbf{H}_{\text{FS}}^{\top}\mathbf{H}_{\text{FS}}$ to be low-rank, we explicitly add the truncated nuclear norm $\|\mathbf{K}\|_{r_0} := \sum_{r=r_0+1}^{N} \widehat{\lambda}_i$ to the loss function of the transductive few-shot node classifier. The starting rank $r_0 < \min(N, d)$ is the rank of the features $\mathbf{H}_{\text{FS}}$ we aim to keep in the node representation, that is, if $\|\mathbf{K}\|_{r_0} = 0$, then $\operatorname{rank}(\mathbf{K}) = r_0$. Therefore, the overall loss function is

$$\min_{\mathbf{W}} L(\mathbf{W}) = \frac{1}{m} \sum_{v_i' \in \mathcal{V}_{\mathcal{L}}} \text{KL}\left(\mathbf{y}_i, [\text{softmax}\left(\mathbf{H}_{\text{FS}}\mathbf{W}\right)]_i\right) + \tau \|\mathbf{K}\|_{r_0}, \tag{2}$$

where KL is the KL divergence. $\tau > 0$ is the weighting parameter for the truncated nuclear norm $\|\mathbf{K}\|_{r_0}$. We use a regular gradient descent to optimize (2) with a learning rate $\eta \in (0, \frac{1}{\widehat{\lambda}_1})$. $\mathbf{W}$ is initialized by $\mathbf{W}^{(0)} = \mathbf{0}$, and at the t-th iteration of gradient descent for $t \ge 1$, $\mathbf{W}$ is updated by $\mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} - \eta \nabla_{\mathbf{W}} L(\mathbf{W})|_{\mathbf{W}=\mathbf{W}^{(t-1)}}$. The optimal rank r_0 on different datasets is decided by cross-validation as detailed in Section 4.1 of the appendix.

Motivation of the Low-Rank Regularization. We study how the information of the ground-truth class label defined on the novel classes are distributed on different eigenvectors of the feature gram matrix $\mathbf{K} = \mathbf{H}_{\text{FS}}^{\top}\mathbf{H}_{\text{FS}}$ by performing eigen-projection. We first compute the eigenvectors $\mathbf{U}$ of the feature gram matrix $\mathbf{K}$. Let $\mathbf{U}^{(1:r)} \in \mathbb{R}^{N \times r}$ be the top r-eigenvectors of $\mathbf{K}$ and $\mathbf{U}^{(r)}$ be the r-th eigenvector of $\mathbf{K}$. Then, the eigen-projection value of the ground-truth label $\mathbf{Y}_{\text{FS}}$ on $\mathbf{U}^{(r)}$ is computed by $p_r = \frac{1}{n}\sum_{c=1}^{n} \left\|{\mathbf{U}^{(r)}}^{\top}\mathbf{Y}_{\text{FS}}^{(c)}\right\|_2^2 / \left\|\mathbf{Y}_{\text{FS}}^{(c)}\right\|_2^2$ for $r \in [N]$, where $\mathbf{Y}_{\text{FS}}^{(c)}$ is the c-th column of $\mathbf{Y}_{\text{FS}}$. We let $\mathbf{p} = [p_1, \ldots, p_N] \in \mathbb{R}^N$. The eigen-projection p_r reflects the amount of the signal in the label projected onto the r-th eigenvector of $\mathbf{K}$, and the signal concentration ratio of a rank r reflects the proportion of signal projected onto the top r eigenvectors of $\mathbf{K}$. The signal concentration ratio for rank r is computed by $\|\mathbf{p}(1:r)\|_1$, where $\mathbf{p}(1:r)$ contains the first r elements of $\mathbf{p}$.

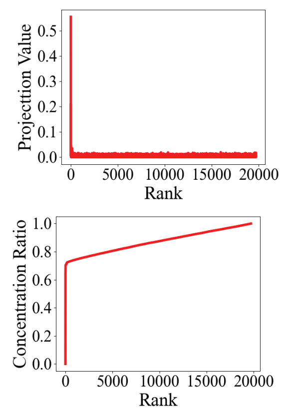

Figure 3: Eigen-projection (first row) and concentration ratio (second row) of the ground truth label on Cora-Full. By the rank $r = r_0 = 0.2\min\{N, D'\}$, the signal concentration ratio of the ground truth label is 0.792.

It is observed from the curves in the first row of Figure 3 that the projection of the ground truth labels for the novel classes mostly concentrates on the top eigenvectors of $\mathbf{K}$, known as the Low Frequency Property (LFP) widely studied in other areas of machine learning (Rahaman et al., 2019; Arora et al., 2019; Cao et al., 2021; Choraria et al., 2022; Wang et al., 2024; 2025). *We remark that the low-rank regularization ensures that mostly only the low-rank part of the node representations* $\mathbf{H}_{\text{FS}}$ *is used for the FSNC, so that our transductive node classifier trained by (2) is free of the noise in the high-rank part of the* $\mathbf{H}_{\text{FS}}$*, thus being robust to the noise in the synthetic graph structures introduced by LR-FGDM*. The low-rank learning is also theoretically justified by Theorem 3.1, showing that the low-rank learning reduces the kernel complexity and renders a tighter bound for the test loss.

Theoretical Justification for the Low-Rank Regularization. We have the following theoretical result, Theorem 3.1, on the Mean Squared Error (MSE) loss of the unlabeled query nodes $\mathcal{U}$ measured by the gap between $[\mathbf{F}(\mathbf{W}, t)]_{\mathcal{U}}$ and $[\mathbf{Y}_{\text{FS}}]_{\mathcal{U}}$ when using the low-rank feature $\mathbf{H}_{\text{FS}}$ with $r_0 \in [N]$, which is the generalization error bound for the linear transductive classifier using $\mathbf{F}(\mathbf{W}) = \mathbf{H}_{\text{FS}}\mathbf{W}$ to predict the labels of the qurey nodes. Similar to existing works such as (Kothapalli et al., 2023) that use the Mean Squared Error (MSE) to analyze the optimization and the generalization of GNNs, we employ the MSE loss to provide the generalization error of the node classifier in the following theorem. It is remarked that the MSE loss is necessary for the generalization analysis of transductive learning using transductive local Rademacher complexity (Tolstikhin et al., 2014; Yang, 2025).

Theorem 3.1. Let $m \ge cN$ for a constant $c \in (0, 1)$, and $r_0 \in [N]$. Assume that a set $\mathcal{L}$ with $|\mathcal{L}| = m$ is sampled uniformly without replacement from $[N]$, and the remaining nodes $\mathcal{V}_{\mathcal{U}} = \mathcal{V}_{\text{FS}} \setminus \mathcal{V}_{\mathcal{L}}$ are the test nodes. Then for every $x > 0$, with probability at least $1 - \exp(-x)$, after the t-th iteration of gradient descent on the training loss $L(\mathbf{W})$ for all $t \geq 1$, we have

$$\mathcal{U}_{\text{test}}(t) \coloneqq \frac{1}{u} \|[\mathbf{F}(\mathbf{W}, t) - \mathbf{Y}_{\text{FS}}]_{\mathcal{U}}\|_{\text{F}}^2 \leq \frac{2c_0 L(\mathbf{K}, \mathbf{Y}_{\text{FS}}, t)}{m} + c_0 \text{KC}(\mathbf{K}) + \frac{c_0 x}{u}, \tag{3}$$

where c_0 is a positive number depending on $\mathbf{U}$, $\left\{\widehat{\lambda}_i\right\}_{i=1}^{r_0}$, and τ_0 with $\tau_0^2 = \max_{i \in [N]} \mathbf{K}_{ii}$. $L(\mathbf{K}, \mathbf{Y}_{\text{FS}}, t) \coloneqq \left\| \left(\mathbf{I}_m - \eta \left[\mathbf{K}\right]_{\mathcal{L},\mathcal{L}}\right)^t \left[\mathbf{Y}_{\text{FS}}\right]_{\mathcal{L}} \right\|_{\text{F}}^2$, KC is the kernel complexity of the kernel gram matrix $\mathbf{K} = \mathbf{H}_{\text{FS}} \mathbf{H}_{\text{FS}}^\top$ defined by $\text{KC}(\mathbf{K}) = \min_{r_0 \in [N]} r_0 \left(\frac{1}{u} + \frac{1}{m}\right) + \sqrt{\|\mathbf{K}\|_{r_0}} \left(\frac{1}{\sqrt{u}} + \frac{1}{\sqrt{m}}\right)$.

This theorem is proved in Section A of the appendix. Detailed explanation about Theorem 3.1 is deferred to Section A.1 of the appendix.

4 Experiments

We evaluate the performance of the LR-FGDM for shot augmentation combined with the low-rank regularization for FSNC. In Section 4.1, we present the implementation details of the proposed LR-FGDM. In Section 4.2, we present the results for different FSNC settings. An ablation study on the effectiveness of the prototypical regularization in the HGAE and low-rank regularization on the few-shot classifier is performed in Section 4.3. Additional experiment results are presented in Section F of the appendix. In Section F.1, we present the results for FSNC on three additional graph datasets. We also compare the LR-FGDM against existing state-of-the-art shot augmentation methods in Section F.2. In Section F.3, we study the effectiveness of LR-FGDM in reducing the kernel complexity of the kernel gram matrix and the upper bound for the test loss of the transductive linear classifier in LR-FGDM. In Section F.4, we study the effectiveness of LR-FGDM on a heterophilic graph dataset, the Roman-Empire dataset (Platonov et al., 2023). In Section F.5 of the appendix, we present the details about cross-validation used to select the number of synthetic support nodes. In Section F.6, we perform the efficiency analysis of LR-FGDM. In Section F.7, we perform the sensitivity analysis of the hyperparameters τ, r_0, K, and q. We have proposed the Frechet Node Distance (FND) and the Frechet Edge Distance (FED) to validate the faithfulness of the synthetic support nodes and the associated edges with comparison to existing shot augmentation methods in Section F.8 of the appendix. The statistical significance of the improvements achieved by LR-FGDM in Section 4.2 and Section 4.3 is validated by the student t-test detailed in Section F.9. In our experiments, we apply LR-FGDM on top of existing FSNC methods, COSMIC (Wang et al., 2023) and COLA (Huang & Zitnik, 2020), which are the most recent state-of-the-art FSNC methods with the best performance.

4.1 Implementation Details

We conduct experiments for FSNC on CoraFull (Bojchevski & Günnemann, 2018), ogbn-arxiv (Hu et al., 2020), Coauthor-CS (Shchur et al., 2018), DBLP (Tang et al., 2008), Roman-Empire (Platonov et al., 2023), Amazon-Computers, Amazon-Photo (Shchur et al., 2018), and Citeseer (Sen et al., 2008), with details in Section C.1 of the appendix. The training settings of LR-FGDM and the hyper-parameter tuning are described in Section C.2 of the appendix.

4.2 Results

We compare the performance of the proposed LR-FGDM with state-of-the-art FSNC methods, including ProtoNet (Snell et al., 2017), Meta-GNN (Zhou et al., 2019), GPN (Ding et al., 2020), G-Meta (Huang & Zitnik, 2020), TENT (Wang et al., 2022), KD-FSNC (Wu et al., 2024), NormProp (Zhang et al., 2025a), COSMIC (Wang et al., 2023), COLA (Huang & Zitnik, 2020), and STAR (Liu et al., 2025a). We also compare LR-FGDM with the diffusion-based synthetic graph structure generation method DoG (Wang et al., 2025). Since DoG requires label-conditioning during training and generation, which is unavailable for unseen classes in few-shot settings, we employ semi-supervised K-means (Basu et al., 2002; Bair, 2013) to obtain pseudo labels as conditioning signals for DoG. The number of clusters and the number of synthetic nodes are all decided by cross-validation. We integrate LR-FGDM into COSMIC and COLA, resulting in two variants, denoted as COSMIC (LR-FGDM) and COLA (LR-FGDM). The experiments are conducted for 2-way and 5-way classification tasks, each with 1-shot and 5-shot settings following (Huang & Zitnik, 2020; Wang et al., 2023). The mean accuracy and standard deviation across 20 independent runs for each setting are reported. It is observed in Table 1 that LR-FGDM consistently improves the performance of COSMIC and COLA on all the datasets. For example, COLA (LR-FGDM) outperforms COLA by 2.29% on Coauthor-CS for the 5-way 5-shot FSNC. The results on the heterophilic graph dataset, Roman-Empire, and three additional graph datasets, Amazon-Computers, Amazon-Photo, and Citeseer, are deferred to Table 9 in Section F.4 and Table 5 in Section F.1 of the appendix.

Table 1: The overall FSNC results of all methods under different settings. The best result is in bold, and the second-best result is underlined. The statistical significance of the results is deferred to Table 17 of the appendix.

Dataset	CoraFull				ogbn-arxiv			
Task	2-way 1-shot	2-way 5-shot	5-way 1-shot	5-way 5-shot	2-way 1-shot	2-way 5-shot	5-way 1-shot	5-way 5-shot
ProtoNet (Snell et al., 2017)	57.10 ± 2.47	72.71 ± 2.55	32.43 ± 1.61	51.54 ± 1.68	62.56 ± 2.86	75.82 ± 2.79	37.30 ± 2.00	53.31 ± 1.71
Meta-GNN (Zhou et al., 2019)	75.28 ± 3.85	84.59 ± 2.89	55.33 ± 2.43	70.50 ± 2.02	62.52 ± 3.41	70.15 ± 2.68	27.14 ± 1.94	31.52 ± 1.71
GPN (Ding et al., 2020)	74.29 ± 3.47	85.58 ± 2.53	52.75 ± 2.32	72.82 ± 1.88	64.00 ± 3.71	76.78 ± 3.50	37.81 ± 2.34	50.50 ± 2.13
G-Meta (Huang & Zitnik, 2020)	78.23 ± 3.41	89.49 ± 2.04	60.44 ± 2.48	75.84 ± 1.70	63.03 ± 3.32	76.56 ± 2.89	31.48 ± 1.70	47.16 ± 1.73
TENT (Wang et al., 2022)	77.75 ± 3.29	88.20 ± 2.61	55.44 ± 2.08	70.10 ± 1.73	70.30 ± 2.85	81.35 ± 2.77	48.26 ± 1.73	61.38 ± 1.72
KD-FSNC (Wu et al., 2024)	83.92 ± 2.68	94.08 ± 2.42	74.55 ± 2.47	85.89 ± 2.15	74.86 ± 3.15	84.67 ± 2.39	52.74 ± 2.13	64.91 ± 1.70
NormProp (Zhang et al., 2025a)	83.61 ± 2.64	93.87 ± 2.39	74.21 ± 2.52	85.47 ± 2.14	74.33 ± 3.10	84.36 ± 2.41	52.37 ± 2.11	64.28 ± 1.72
STAR (Liu et al., 2025a)	85.22 ± 1.69	94.95 ± 1.48	75.85 ± 1.72	87.31 ± 1.55	76.45 ± 2.03	86.11 ± 2.10	54.82 ± 1.75	66.98 ± 1.25
DoG (Wang et al., 2025)	85.10 ± 1.98	94.35 ± 1.82	75.13 ± 1.56	86.47 ± 1.13	77.33 ± 2.31	86.89 ± 2.21	53.42 ± 1.47	65.69 ± 1.85
COSMIC (Wang et al., 2023)	84.32 ± 2.75	94.51 ± 2.47	74.93 ± 2.49	86.34 ± 2.17	75.71 ± 3.17	85.19 ± 2.35	53.28 ± 2.19	65.42 ± 1.69
COLA (Huang & Zitnik, 2020)	85.83 ± 1.92	95.17 ± 1.85	76.47 ± 2.12	87.83 ± 1.89	77.12 ± 2.36	86.42 ± 2.28	55.24 ± 2.04	67.52 ± 1.75
COSMIC (LR-FGDM)	$\underline{86.21 \pm 2.38}$	$\underline{96.74 \pm 2.11}$	$\underline{76.93 \pm 2.15}$	$\underline{88.81 \pm 1.93}$	$\underline{77.68 \pm 2.75}$	$\underline{87.24 \pm 2.13}$	$\underline{55.48 \pm 2.01}$	$\underline{67.59 \pm 1.52}$
COLA (LR-FGDM)	$\mathbf{87.54 \pm 1.74}$	$\mathbf{97.38 \pm 1.67}$	$\mathbf{78.52 \pm 1.94}$	$\mathbf{89.66 \pm 1.72}$	$\mathbf{79.02 \pm 2.18}$	$\mathbf{88.34 \pm 2.10}$	$\mathbf{57.28 \pm 1.86}$	$\mathbf{69.63 \pm 1.57}$

Dataset	Coauthor-CS				DBLP			
Task	2-way 1-shot	2-way 5-shot	5-way 1-shot	5-way 5-shot	2-way 1-shot	2-way 5-shot	5-way 1-shot	5-way 5-shot
ProtoNet (Snell et al., 2017)	59.92 ± 2.70	71.69 ± 2.51	32.13 ± 1.52	49.25 ± 1.50	60.97 ± 2.56	72.81 ± 2.73	31.31 ± 1.58	52.26 ± 1.88
Meta-GNN (Zhou et al., 2019)	85.90 ± 2.96	90.11 ± 2.17	52.86 ± 2.14	68.59 ± 1.49	82.60 ± 3.23	86.15 ± 3.29	67.24 ± 2.72	72.15 ± 2.40
GPN (Ding et al., 2020)	84.31 ± 2.73	90.36 ± 1.90	60.66 ± 2.07	81.79 ± 1.18	79.55 ± 3.46	85.85 ± 2.61	59.38 ± 2.40	75.46 ± 1.87
G-Meta (Huang & Zitnik, 2020)	84.19 ± 2.97	91.02 ± 1.61	59.68 ± 2.16	74.18 ± 1.29	80.46 ± 3.29	88.53 ± 2.36	63.32 ± 2.70	75.82 ± 2.11
TENT (Wang et al., 2022)	87.85 ± 2.48	91.75 ± 1.60	63.70 ± 1.88	76.90 ± 1.19	84.40 ± 2.73	90.05 ± 2.34	61.56 ± 2.23	74.84 ± 2.04
KD-FSNC (Wu et al., 2024)	89.78 ± 2.36	93.21 ± 2.01	67.05 ± 1.66	84.42 ± 1.17	91.81 ± 2.41	94.37 ± 1.70	74.83 ± 2.15	83.75 ± 1.91
NormProp (Zhang et al., 2025a)	89.34 ± 2.41	93.62 ± 1.97	67.48 ± 1.68	84.61 ± 1.14	91.52 ± 2.45	94.05 ± 1.72	75.39 ± 2.18	84.12 ± 1.89
STAR (Liu et al., 2025a)	91.28 ± 1.15	95.41 ± 1.85	69.25 ± 1.23	87.60 ± 1.33	93.10 ± 1.47	95.52 ± 1.55	77.14 ± 1.35	87.10 ± 1.05
DoG (Wang et al., 2025)	91.10 ± 1.84	94.88 ± 1.53	68.96 ± 1.80	87.35 ± 1.41	93.55 ± 1.35	96.05 ± 1.22	78.87 ± 1.37	87.59 ± 1.25
COSMIC (Wang et al., 2023)	90.29 ± 2.30	94.32 ± 1.93	68.21 ± 1.63	85.47 ± 1.11	92.35 ± 2.52	94.82 ± 1.69	76.52 ± 2.24	85.31 ± 1.92
COLA (Huang & Zitnik, 2020)	91.53 ± 2.03	95.78 ± 1.84	70.46 ± 1.57	87.54 ± 1.19	93.48 ± 2.17	95.92 ± 1.68	78.18 ± 2.05	87.23 ± 1.87
COSMIC (LR-FGDM)	$\underline{92.48 \pm 2.01}$	$\underline{96.71 \pm 1.67}$	$\underline{70.41 \pm 1.48}$	$\underline{87.72 \pm 1.03}$	$\underline{94.78 \pm 2.29}$	$\underline{96.95 \pm 1.53}$	$\underline{78.66 \pm 2.03}$	$\underline{87.44 \pm 1.71}$
COLA (LR-FGDM)	$\mathbf{93.84 \pm 1.85}$	$\mathbf{97.91 \pm 1.56}$	$\mathbf{72.93 \pm 1.41}$	$\mathbf{89.83 \pm 1.11}$	$\mathbf{95.89 \pm 2.03}$	$\mathbf{97.98 \pm 1.47}$	$\mathbf{80.16 \pm 1.88}$	$\mathbf{89.51 \pm 1.65}$

4.3 Ablation Study

To thoroughly study the effectiveness of the prototypical regularization in the HGAE and low-rank regularization on the classifier, we conduct an ablation study on CoraFull, ogbn-arxiv, Coauthor-CS, and DBLP under the 5-way 5-shot setting for FSNC. We evaluate three variants of the COLA (LR-FGDM), which are the COLA (LR-FGDM) without the prototypical regularization, COLA (LR-FGDM) without the low-rank regularization, and COLA (LR-FGDM) without both the low-rank regularization and the prototypical regularization. It is observed from Table 2 that both the low-rank regularization on training the few-shot node classifier and the prototypical regularization on training the HGAE play important roles in improving the performance of the baseline method.

Table 2: Ablation study on the low-rank regularization and the prototypical regularization. The statistical significance of the results is deferred to Table 18 of the appendix.

Method	CoraFull	ogbn-arxiv	Coauthor-CS	DBLP
COLA (Huang & Zitnik, 2020)	87.83	67.52	87.54	87.23
COLA (LR-FGDM) w/o both low-rank and prototypical regularization	88.12	67.91	87.93	87.55
COLA (LR-FGDM) w/o low-rank regularization	88.74	$\underline{68.60}$	88.72	88.28
COLA (LR-FGDM) w/o prototypical regularization	$\underline{88.79}$	68.45	$\underline{89.02}$	$\underline{88.64}$
COLA (LR-FGDM)	**89.66**	**69.63**	**89.83**	**89.51**

5 Conclusion

In this paper, we propose a novel node-level graph diffusion method with low-rank feature learning for FSNC, termed Low-Rank Few-Shot Graph Diffusion Model or LR-FGDM. LR-FGDM addresses the limitation of data scarcity in few-shot settings by augmenting the support set through a novel node-level graph diffusion model and enforcing low-rank regularization on the training of the few-shot node classifier. FGDM integrates a Hierarchical Graph Autoencoder (HGAE) with a hierarchical edge reconstruction method and a Latent Diffusion Model (LDM). The low-rank regularization is motivated by the Low Frequency Property (LFP) and theoretically justified by a theorem to show lower generalization error. Extensive experiments on multiple graph benchmark datasets show that LR-FGDM significantly improves the performance of few-shot node classifiers, demonstrating superior generalization capabilities compared to state-of-the-art methods.

REFERENCES

Sanjeev Arora, Simon S. Du, Wei Hu, Zhiyuan Li, and Ruosong Wang. Fine-grained analysis of optimization and generalization for overparameterized two-layer neural networks. In *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 322–332. PMLR, 2019.

Amitoz Azad and Yuan Fang. A learned generalized geodesic distance function-based approach for node feature augmentation on graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 49–58, 2024.

Shekoofeh Azizi, Simon Kornblith, Chitwan Saharia, Mohammad Norouzi, and David J Fleet. Synthetic data from diffusion models improves imagenet classification. *Transactions on Machine Learning Research*, 2023.

Eric Bair. Semi-supervised clustering methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(5):349–361, 2013.

Dmitry Baranchuk, Andrey Voynov, Ivan Rubachev, Valentin Khrulkov, and Artem Babenko. Label-efficient semantic segmentation with diffusion models. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022.

Sugato Basu, Arindam Banerjee, and Raymond J Mooney. Semi-supervised clustering by seeding. In *Proceedings of the nineteenth international conference on machine learning*, pp. 27–34, 2002.

Aleksandar Bojchevski and Stephan Günnemann. Deep gaussian embedding of graphs: Unsupervised inductive learning via ranking. In *ICLR*, 2018.

Andrew Brock, Jeff Donahue, and Karen Simonyan. Large scale GAN training for high fidelity natural image synthesis. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019.

Yuan Cao, Zhiying Fang, Yue Wu, Ding-Xuan Zhou, and Quanquan Gu. Towards understanding the spectral bias of deep learning. In Zhi-Hua Zhou (ed.), *International Joint Conference on Artificial Intelligence*, pp. 2205–2211. ijcai.org, 2021.

Moulik Choraria, Leello Tadesse Dadi, Grigorios Chrysos, Julien Mairal, and Volkan Cevher. The spectral bias of polynomial neural networks. In *International Conference on Learning Representations*. OpenReview.net, 2022.

Prafulla Dhariwal and Alexander Nichol. Diffusion models beat gans on image synthesis. *Advances in neural information processing systems*, 34:8780–8794, 2021.

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 and Knowledge Management*, pp. 295–304, 2020.

DC Dowson and BV666017 Landau. The fréchet distance between multivariate normal distributions. *Journal of multivariate analysis*, 1982.

Wenzheng Feng, Yuxiao Dong, Tinglin Huang, Ziqi Yin, Xu Cheng, Evgeny Kharlamov, and Jie Tang. GRAND+: scalable graph random neural networks. In *WWW '22: The ACM Web Conference 2022, Virtual Event, Lyon, France, April 25 - 29, 2022*, pp. 3248–3258. ACM, 2022.

Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *International conference on machine learning*, pp. 1126–1135. PMLR, 2017.

Hengyu Fu, Zhuoran Yang, Mengdi Wang, and Minshuo Chen. Unveil conditional diffusion models with classifier-free guidance: A sharp statistical theory. *arXiv preprint arXiv:2403.11968*, 2024.

Sicheng Gao, Xuhui Liu, Bohan Zeng, Sheng Xu, Yanjing Li, Xiaoyan Luo, Jianzhuang Liu, Xiantong Zhen, and Baochang Zhang. Implicit diffusion models for continuous super-resolution. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2023, Vancouver, BC, Canada, June 17-24, 2023*, pp. 10021–10030. IEEE, 2023a.

Xinyi Gao, Wentao Zhang, Tong Chen, Junliang Yu, Hung Quoc Viet Nguyen, and Hongzhi Yin. Semantic-aware node synthesis for imbalanced heterogeneous information networks. In *Proceedings of the 32nd ACM International Conference on Information and Knowledge Management*, pp. 545–555, 2023b.

Yizhao Gao, Nanyi Fei, Guangzhen Liu, Zhiwu Lu, and Tao Xiang. Contrastive prototype learning with augmented embeddings for few-shot learning. In *Uncertainty in Artificial Intelligence*, pp. 140–150. PMLR, 2021.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. Generative adversarial networks. *Communications of the ACM*, 63(11):139–144, 2020.

Kilian Konstantin Haefeli, Karolis Martinkus, Nathanaël Perraudin, and Roger Wattenhofer. Diffusion models for graphs benefit from discrete state spaces. *CoRR*, abs/2210.01549, 2022. doi: 10.48550/ARXIV.2210.01549.

Will Hamilton, Zhitao Ying, and Jure Leskovec. Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30, 2017.

Xiaotian Han, Zhimeng Jiang, Ninghao Liu, and Xia Hu. G-mixup: Graph data augmentation for graph classification. In *International Conference on Machine Learning*, pp. 8230–8248. PMLR, 2022.

Ruifei He, Shuyang Sun, Xin Yu, Chuhui Xue, Wenqing Zhang, Philip Torr, Song Bai, and Xiaojuan Qi. Is synthetic data from generative models ready for image recognition? *arXiv preprint arXiv:2210.07574*, 2022.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint arXiv:2207.12598*, 2022.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Hung Chun Hsu, Bo-Jun Wu, Ming-Yi Hong, Che Lin, and Chih-Yu Wang. A GAN approach for node embedding in heterogeneous graphs using subgraph sampling. *CoRR*, abs/2312.06519, 2023.

Hung Chun Hsu, Ting-Le Lin, Bo-Jun Wu, Ming-Yi Hong, Che Lin, and Chih-Yu Wang. Fincgan: A gan framework of imbalanced node classification on heterogeneous graph neural network. In *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5750–5754. IEEE, 2024.

Weihua Hu, Matthias Fey, Marinka Zitnik, Yuxiao Dong, Hongyu Ren, Bowen Liu, Michele Catasta, and Jure Leskovec. Open graph benchmark: Datasets for machine learning on graphs. *Advances in neural information processing systems*, 33:22118–22133, 2020.

Kexin Huang and Marinka Zitnik. Graph meta learning via local subgraphs. *Advances in neural information processing systems*, 33:5862–5874, 2020.

Huiwon Jang, Hankook Lee, and Jinwoo Shin. Unsupervised meta-learning via few-shot pseudo-supervised contrastive learning. *arXiv preprint arXiv:2303.00996*, 2023.

Nan Jia, Xiaolin Tian, Wenxing Gao, and Licheng Jiao. Deep graph-convolutional generative adversarial network for semi-supervised learning on graphs. *Remote Sensing*, 15(12):3172, 2023.

Ming Jin, Yizhen Zheng, Yuan-Fang Li, Chen Gong, Chuan Zhou, and Shirui Pan. Multi-scale contrastive siamese networks for self-supervised graph representation learning. *arXiv preprint arXiv:2105.05682*, 2021.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based generative modeling of graphs via the system of stochastic differential equations. In *International Conference on Machine Learning, ICML*, 2022.

Thomas N. Kipf and Max Welling. Variational graph auto-encoders. *CoRR*, abs/1611.07308, 2016a.

Thomas N Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. *arXiv preprint arXiv:1609.02907*, 2016b.

Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*. OpenReview.net, 2017.

Kezhi Kong, Guohao Li, Mucong Ding, Zuxuan Wu, Chen Zhu, Bernard Ghanem, Gavin Taylor, and Tom Goldstein. Robust optimization as data augmentation for large-scale graphs. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 60–69. IEEE, 2022.

Vignesh Kothapalli, Tom Tirer, and Joan Bruna. A neural collapse perspective on feature evolution in graph neural networks. In Alice Oh, Tristan Naumann, Amir Globerson, Kate Saenko, Moritz Hardt, and Sergey Levine (eds.), *Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023*, 2023.

Steinar Laenen and Luca Bertinetto. On episodes, prototypical networks, and few-shot learning. *Advances in Neural Information Processing Systems*, 34:24581–24592, 2021.

Yurui Lai, Xiaoyang Lin, Renchi Yang, and Hongtao Wang. Efficient topology-aware data augmentation for high-degree graph neural networks. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1463–1473, 2024.

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. *Advances in Neural Information Processing Systems*, 33:16520–16531, 2020.

Ming Liang, Yao Meng, Jiyu Wang, David L Lubkeman, and Ning Lu. Feedergan: Synthetic feeder generation via deep graph adversarial nets. *IEEE Transactions on Smart Grid*, 12(2):1163–1173, 2020.

Stratis Limnios, Praveen Selvaraj, Mihai Cucuringu, Carsten Maple, Gesine Reinert, and Andrew Elliott. Sagess: Sampling graph denoising diffusion model for scalable graph generation. *CoRR*, abs/2306.16827, 2023. doi: 10.48550/ARXIV.2306.16827.

Xixun Lin, Zhao Li, Peng Zhang, Luchen Liu, Chuan Zhou, Bin Wang, and Zhihong Tian. Structure-aware prototypical neural process for few-shot graph classification. *IEEE Transactions on Neural Networks and Learning Systems*, 35(4):4607–4621, 2022.

Chen Liu, Yanwei Fu, Chengming Xu, Siqian Yang, Jilin Li, Chengjie Wang, and Li Zhang. Learning a few-shot embedding model with contrastive learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 8635–8643, 2021a.

Yonghao Liu, Fausto Giunchiglia, Ximing Li, Lan Huang, Xiaoyue Feng, and Renchu Guan. Enhancing unsupervised graph few-shot learning via set functions and optimal transport. In Yizhou Sun, Flavio Chierichetti, Hady W. Lauw, Claudia Perlich, Wee Hyong Tok, and Andrew Tomkins (eds.), *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining, V.1, KDD 2025, Toronto, ON, Canada, August 3-7, 2025*, pp. 871–882. ACM, 2025a.

Yonghao Liu, Mengyu Li, Fausto Giunchiglia, Lan Huang, Ximing Li, Xiaoyue Feng, and Renchu Guan. Dual-level mixup for graph few-shot learning with fewer tasks. In *Proceedings of the ACM on Web Conference 2025*, pp. 2646–2656, 2025b.

Zemin Liu, Yuan Fang, Chenghao Liu, and Steven CH Hoi. Relative and absolute location embedding for few-shot node classification on graph. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 4267–4275, 2021b.

Yihong Luo, Yuhan Chen, Siya Qiu, Yiwei Wang, Chen Zhang, Yan Zhou, Xiaochun Cao, and Jing Tang. Fast graph sharpness-aware minimization for enhancing and accelerating few-shot node classification. *arXiv preprint arXiv:2410.16845*, 2024.

Liheng Ma, Reihaneh Rabbany, and Adriana Romero-Soriano. Graph attention networks with positional embeddings. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pp. 514–527. Springer, 2021.

Xiuhui Ma, Junhai Zhai, and Hao Chen. Contextual learning and feature tuning for few-shot node classification. *Information Sciences*, 719:122494, 2025.

Yujie Mo, Liang Peng, Jie Xu, Xiaoshuang Shi, and Xiaofeng Zhu. Simple unsupervised graph representation learning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 36, pp. 7797–7805, 2022.

Renkun Ni, Manli Shu, Hossein Souri, Micah Goldblum, and Tom Goldstein. The close relationship between contrastive learning and meta-learning. In *International Conference on Learning Representations*, 2021.

Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation invariant graph generation via score-based generative modeling. In *The 23rd International Conference on Artificial Intelligence and Statistics, AISTATS 2020, 26-28 August 2020, Online [Palermo, Sicily, Italy]*, volume 108 of *Proceedings of Machine Learning Research*, pp. 4474–4484. PMLR, 2020.

Oleg Platonov, Denis Kuznedelev, Michael Diskin, Artem Babenko, and Liudmila Prokhorenkova. A critical look at the evaluation of gnns under heterophily: Are we really making progress? In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Yiyue Qian, Yiming Zhang, Yanfang Ye, Chuxu Zhang, et al. Distilling meta knowledge on heterogeneous graph for illicit drug trafficker detection on social media. *Advances in Neural Information Processing Systems*, 34:26911–26923, 2021.

Liang Qu, Huaisheng Zhu, Ruiqi Zheng, Yuhui Shi, and Hongzhi Yin. Imgagn: Imbalanced network embedding via generative adversarial graph networks. In Feida Zhu, Beng Chin Ooi, and Chunyan Miao (eds.), *KDD '21: The 27th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, Virtual Event, Singapore, August 14-18, 2021*, pp. 1390–1398. ACM, 2021.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pp. 5301–5310. PMLR, 09–15 Jun 2019.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022*, pp. 10674–10685. IEEE, 2022.

Yu Rong, Wenbing Huang, Tingyang Xu, and Junzhou Huang. Dropedge: Towards deep graph convolutional networks on node classification. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net, 2020.

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

Oleksandr Shchur, Maximilian Mumme, Aleksandar Bojchevski, and Stephan Günnemann. Pitfalls of graph neural network evaluation. *Relational Representation Learning Workshop, NeurIPS 2018*, 2018.

Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. *Advances in neural information processing systems*, 30, 2017.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021a.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada*, pp. 11895–11907, 2019.

Yang Song and Stefano Ermon. Improved techniques for training score-based generative models. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Yang Song, Conor Durkan, Iain Murray, and Stefano Ermon. Maximum likelihood training of score-based diffusion models. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021, December 6-14, 2021, virtual*, pp. 1415–1428, 2021b.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *9th International Conference on Learning Representations, ICLR*, 2021c.

Ke Sun, Zhouchen Lin, and Zhanxing Zhu. Multi-stage self-supervised learning for graph convolutional networks on graphs with few labeled nodes. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 5892–5899. AAAI Press, 2020.

Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 2818–2826, 2016.

Zhen Tan, Song Wang, Kaize Ding, Jundong Li, and Huan Liu. Transductive linear probing: A novel framework for few-shot node classification. *arXiv preprint arXiv:2212.05606*, 2022.

Jie Tang, Jing Zhang, Limin Yao, Juanzi Li, Li Zhang, and Zhong Su. Arnetminer: extraction and mining of academic social networks. In *SIGKDD*, 2008.

Ilya O. Tolstikhin, Gilles Blanchard, and Marius Kloft. Localized complexities for transductive learning. In Maria-Florina Balcan, Vitaly Feldman, and Csaba Szepesvári (eds.), *Conference on Learning Theory*, volume 35 of *JMLR Workshop and Conference Proceedings*, pp. 857–884. JMLR.org, 2014.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *International Conference on Learning Representations*, 2018.

Vikas Verma, Meng Qu, Kenji Kawaguchi, Alex Lamb, Yoshua Bengio, Juho Kannala, and Jian Tang. Graphmix: Improved training of gnns for semi-supervised learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 10024–10032, 2021.

Clément Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

Hongwei Wang, Jia Wang, Jialin Wang, Miao Zhao, Weinan Zhang, Fuzheng Zhang, Xing Xie, and Minyi Guo. Graphgan: Graph representation learning with generative adversarial nets. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32, 2018.

Kuansan Wang, Zhihong Shen, Chiyuan Huang, Chieh-Han Wu, Yuxiao Dong, and Anshul Kanakia. Microsoft academic graph: When experts are not enough. *Quantitative Science Studies*, 2020a.

Ning Wang, Minnan Luo, Kaize Ding, Lingling Zhang, Jundong Li, and Qinghua Zheng. Graph few-shot learning with attribute matching. In *Proceedings of the 29th ACM International Conference on Information & Knowledge Management*, pp. 1545–1554, 2020b.

Song Wang, Kaize Ding, Chuxu Zhang, Chen Chen, and Jundong Li. Task-adaptive few-shot node classification. In *Proceedings of the 28th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 1910–1919, 2022.

Song Wang, Zhen Tan, Huan Liu, and Jundong Li. 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.

Yancheng Wang, Rajeev Goel, Utkarsh Nath, Alvin C. Silva, Teresa Wu, and Yingzhen Yang. Learning low-rank feature for thorax disease classification. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang (eds.), *Advances in Neural Information Processing Systems*, 2024.

Yancheng Wang, Changyu Liu, and Yingzhen Yang. Diffusion on graph: Augmentation of graph structure for node classification. *Trans. Mach. Learn. Res.*, 2025, 2025.

Yiwei Wang, Wei Wang, Yuxuan Liang, Yujun Cai, and Bryan Hooi. Mixup for node and graph classification. In *Proceedings of the Web Conference 2021*, pp. 3663–3674, 2021.

Cheng Wu, Chaokun Wang, Jingcao Xu, Ziyang Liu, Kai Zheng, Xiaowei Wang, Yang Song, and Kun Gai. Graph contrastive learning with generative adversarial network. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 2721–2730, 2023.

Zongqian Wu, Peng Zhou, Guoqiu Wen, Yingying Wan, Junbo Ma, Debo Cheng, and Xiaofeng Zhu. Information augmentation for few-shot node classification. In *Proceedings of the Thirty-First International Joint Conference on Artificial Intelligence (IJCAI-22)*, pp. 3601–3607. International Joint Conferences on Artificial Intelligence Organization, 2022.

Zongqian Wu, Yujie Mo, Peng Zhou, Shangbo Yuan, and Xiaofeng Zhu. Self-training based few-shot node classification by knowledge distillation. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pp. 15988–15995, 2024.

Carl Yang, Peiye Zhuang, Wenhan Shi, Alan Luu, and Pan Li. Conditional structure generation through graph variational generative adversarial nets. *Advances in neural information processing systems*, 32, 2019.

Yingzhen Yang. A new concentration inequality for sampling without replacement and its application for transductive learning. In *International Conference on Machine Learning*, 2025.

Huaxiu Yao, Chuxu Zhang, Ying Wei, Meng Jiang, Suhang Wang, Junzhou Huang, Nitesh Chawla, and Zhenhui Li. Graph few-shot learning via knowledge transfer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pp. 6656–6663, 2020.

Jiaxuan You, Rex Ying, and Jure Leskovec. Position-aware graph neural networks. In Kamalika Chaudhuri and Ruslan Salakhutdinov (eds.), *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, volume 97 of *Proceedings of Machine Learning Research*, pp. 7134–7143. PMLR, 2019.

Yuning You, Tianlong Chen, Yongduo Sui, Ting Chen, Zhangyang Wang, and Yang Shen. Graph contrastive learning with augmentations. In Hugo Larochelle, Marc'Aurelio Ranzato, Raia Hadsell, Maria-Florina Balcan, and Hsuan-Tien Lin (eds.), *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*, 2020.

Junhai Zhai, Sufang Zhang, Junfen Chen, and Qiang He. Autoencoder and its various variants. In *2018 IEEE international conference on systems, man, and cybernetics (SMC)*, pp. 415–419. IEEE, 2018.

Baoming Zhang, MingCai Chen, Jianqing Song, Shuangjie Li, Jie Zhang, and Chongjun Wang. Normalize then propagate: Efficient homophilous regularization for few-shot semi-supervised node classification. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2025a.

Chenhui Zhang, Huanjing Zhao, Beining Yang, Yukuo Cen, Junyu Ren, Yuxiao Dong, Evgeny Kharlamov, Shu Zhao, and Jie Tang. Leveraging large language models for node generation in few-shot learning. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence (AAAI)*, 2025b.

Huanjing Zhao, Beining Yang, Yukuo Cen, Junyu Ren, Chenhui Zhang, Yuxiao Dong, Evgeny Kharlamov, Shu Zhao, and Jie Tang. Pre-training and prompting for few-shot node classification on text-attributed graphs. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4467–4478, 2024a.

Shiman Zhao, Wei Chen, Tengjiao Wang, Jiahui Yao, Dawei Lu, and Jiabin Zheng. Less is enough: Relation graph guided few-shot learning for multi-label aspect category detection. In *2025 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2025, Hyderabad, India, April 6-11, 2025*, pp. 1–5. IEEE, 2025.

Tianxiang Zhao, Xiang Zhang, and Suhang Wang. Graphsmote: Imbalanced node classification on graphs with graph neural networks. In *Proceedings of the 14th ACM international conference on web search and data mining*, pp. 833–841, 2021a.

Tong Zhao, Yozen Liu, Leonardo Neves, Oliver Woodford, Meng Jiang, and Neil Shah. Data augmentation for graph neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pp. 11015–11023, 2021b.

Wentao Zhao, Qitian Wu, Chenxiao Yang, and Junchi Yan. Geomix: Towards geometry-aware data augmentation. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pp. 4500–4511, 2024b.

Cai Zhou, Xiyuan Wang, and Muhan Zhang. Latent graph diffusion: A unified framework for generation and prediction on graphs. *CoRR*, abs/2402.02518, 2024. doi: 10.48550/ARXIV.2402.02518.

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*, pp. 2357–2360, 2019.

Zhaocheng Zhu, Zuobai Zhang, Louis-Pascal Xhonneux, and Jian Tang. Neural bellman-ford networks: A general graph neural network framework for link prediction. *Advances in Neural Information Processing Systems*, 2021.

A Theoretical Results

We present the proof of Theorem 3.1 in this section.

Proof of Theorem 3.1. It can be verified that at the t-th iteration of gradient descent for $t \ge 1$, we have

$$\mathbf{W}^{(t)} = \mathbf{W}^{(t-1)} - \eta \left[\mathbf{H}_{\text{FS}}\right]_{\mathcal{L}}^{\top} \left[\mathbf{H}_{\text{FS}} \mathbf{W}^{(t-1)} - \mathbf{Y}_{\text{FS}}\right]_{\mathcal{L}}. \tag{4}$$

It follows by (4) that

$$\left[\mathbf{H}_{\text{FS}}\right]_{\mathcal{L}} \mathbf{W}^{(t)} = \left[\mathbf{H}_{\text{FS}}\right]_{\mathcal{L}} \mathbf{W}^{(t-1)} - \eta \mathbf{K}_{\mathcal{L},\mathcal{L}} \left[\mathbf{H}_{\text{FS}} \mathbf{W}^{(t-1)} - \mathbf{Y}_{\text{FS}}\right]_{\mathcal{L}}, \tag{5}$$

where $\mathbf{K}_{\mathcal{L},\mathcal{L}} \coloneqq \left[\mathbf{H}_{\text{FS}}\right]_{\mathcal{L}} \left[\mathbf{H}_{\text{FS}}\right]_{\mathcal{L}}^{\top} \in \mathbb{R}^{m \times m}$. With $\mathbf{F}(\mathbf{W}, t) = \mathbf{H}_{\text{FS}} \mathbf{W}^{(t)}$, it follows by (5) that

$$\left[\mathbf{F}(\mathbf{W}, t) - \mathbf{Y}_{\text{FS}}\right]_{\mathcal{L}} = \left(\mathbf{I}_m - \eta \left[\mathbf{K}\right]_{\mathcal{L},\mathcal{L}}\right) \left[\mathbf{F}(\mathbf{W}, t-1) - \mathbf{Y}_{\text{FS}}\right]_{\mathcal{L}}.$$

It follows from the above equality and the recursion that

$$\left[\mathbf{F}(\mathbf{W}, t) - \mathbf{Y}_{\text{FS}}\right]_{\mathcal{L}} = -\left(\mathbf{I}_m - \eta \left[\mathbf{K}\right]_{\mathcal{L},\mathcal{L}}\right)^{t} \left[\mathbf{Y}_{\text{FS}}\right]_{\mathcal{L}}. \tag{6}$$

We apply (Yang, 2025, Corollary 3.7) to obtain the following bound for the test loss $\frac{1}{u}\|[\mathbf{F}(\mathbf{W},t)-\mathbf{Y}_{\rm FS}]_{\mathcal{U}}\|_{\rm F}^2$:

$$\frac{1}{u}\|[\mathbf{F}(\mathbf{W},t)-\mathbf{Y}_{\rm FS}]_{\mathcal{U}}\|_{\rm F}^2 \le \frac{c_0}{m}\|[\mathbf{F}(\mathbf{W},t)-\mathbf{Y}_{\rm FS}]_{\mathcal{L}}\|_{\rm F}^2 + c_0 \min_{0\le Q\le n} r(u,m,Q) + \frac{c_0 x}{u}, \tag{7}$$

with

$$r(u,m,Q) := Q\left(\frac{1}{u}+\frac{1}{m}\right) + \left(\sqrt{\frac{\sum\limits_{q=Q+1}^{N}\widehat{\lambda}_q}{u}} + \sqrt{\frac{\sum\limits_{q=Q+1}^{N}\widehat{\lambda}_q}{m}}\right),$$

where c_0 is a positive constant depending on $\mathbf{U}$, $\left\{\widehat{\lambda}_i\right\}_{i=1}^{r}$, and τ_0 with $\tau_0^2 = \max_{i\in[N]}\mathbf{K}_{ii}$.

It follows from (6) and (7) that for every $r_0 \in [N]$, we have

$$\begin{aligned}
&\frac{1}{u}\|[\mathbf{F}(\mathbf{W},t)-\mathbf{Y}_{\rm FS}]_{\mathcal{U}}\|_{\rm F}^2\\
&\le \frac{c_0}{m}\left\|\left(\mathbf{I}_m - \eta\,[\mathbf{K}]_{\mathcal{L},\mathcal{L}}\right)^t [\mathbf{Y}_{\rm FS}]_{\mathcal{L}}\right\|_{\rm F}^2 + c_0 r_0\left(\frac{1}{u}+\frac{1}{m}\right) + c_0\left(\sqrt{\frac{\sum\limits_{q=r_0+1}^{N}\widehat{\lambda}_q}{u}} + \sqrt{\frac{\sum\limits_{q=r_0+1}^{N}\widehat{\lambda}_q}{m}}\right) + \frac{c_0 x}{u}\\
&\overset{①}{\le} \frac{2c_0}{m}\left\|\left(\mathbf{I}_m - \eta\,[\mathbf{K}]_{\mathcal{L},\mathcal{L}}\right)^t [\mathbf{Y}_{\rm FS}]_{\mathcal{L}}\right\|_{\rm F}^2 + c_0 r_0\left(\frac{1}{u}+\frac{1}{m}\right) + c_0\sqrt{\|\mathbf{K}\|_{r_0}}\left(\sqrt{\frac{1}{u}}+\sqrt{\frac{1}{m}}\right) + \frac{c_0 x}{u},
\end{aligned} \tag{8}$$

where ① follows from the Cauchy-Schwarz inequality, (6), and $\sum_{q=r_0+1}^{N}\widehat{\lambda}_q = \|\mathbf{K}\|_{r_0}$. (3) then follows directly from (8). □

A.1 Further Explanation of Theorem 3.1

Define $\mathbf{F}(\mathbf{W},t) := \mathbf{H}_{\rm FS}\mathbf{W}^{(t)}$ as the output of the classifier after the t-th iteration of gradient descent for $t \ge 1$. It is noted that $\mathcal{U}_{\rm test}(t)$ is the test loss of the unlabeled query nodes measured by the distance between the classifier output $\mathbf{F}(\mathbf{W},t)$ and $\mathbf{Y}_{\rm FS}$. There are two terms on the upper bound for the test loss in (3), $L(\mathbf{K},\mathbf{Y}_{\rm FS},t)$ and $\mathrm{KC}(\mathbf{K})$, which are explained as follows. $L(\mathbf{K},\mathbf{Y}_{\rm FS},t)$ corresponds to the training loss of the node classifier with the ground-truth label for the novel classes. $\mathrm{KC}(\mathbf{K})$ is the kernel complexity (KC), which measures the complexity of the kernel gram matrix from the node representation $\mathbf{H}_{\rm FS}$. We remark that the TNN $\|\mathbf{K}\|_{r_0}$ appears on the RHS of the upper bound (3), theoretically justifying why we learn the low-rank features $\mathbf{K}$ for FSNC by adding the TNN $\|\mathbf{K}\|_{r_0}$ to the training loss. Moreover, when the low frequency property holds, $L(\mathbf{K},\mathbf{Y}_{\rm FS},t)$ would be very small with enough iteration number t. $\mathbf{K} = \mathbf{H}_{\rm FS}^\top\mathbf{H}_{\rm FS}$ is approximately a low-rank matrix of rank r_0 since $\mathbf{H}_{\rm FS}$ is approximately a rank-r_0 matrix with its TNN optimized through the optimization of the encoder of the HGAE. A smaller $\|\mathbf{K}\|_{r_0}$ is obtained by optimizing the training loss in Equation (2), which in turn ensures a smaller kernel complexity (KC) defined in Theorem 3.1, contributing to a smaller generalization bound for transductive node classification.

B Complexity Analysis of the Hierarchical Edge Reconstruction Method

In our work, we have proposed an efficient hierarchical edge reconstruction method to reconstruct the edges connected to a node in the graph. To show its efficiency, we analyze the inference time complexity and the parameter size of the HGAE with the hierarchical edge reconstruction method. For comparison, we also analyze the inference time complexity and the parameter size of GAE, where the hierarchical edge reconstruction method is replaced by a regular edge decoder that directly reconstructs the adjacency matrix $\mathbf{A}$ (Kipf & Welling, 2016a). For ease of comparison, we denote the number of parameters and inference cost of all the MLP and GAT layers except the hierarchical edge reconstruction process as S_{MLP} and C_{MLP}, respectively. For a node v_i in the graph, let $d_i = \sum_{k=1}^{K} \widehat{\mathbf{C}}_{ik}$ be the number of clusters predicted to be connected to v_i. Let D' be the dimension of the input feature for the hierarchical edge reconstruction. The inference time complexity of HGAE with hierarchical edge reconstruction is $\mathcal{O}(KD' + d_i D' M + C_{\text{MLP}})$, where $\mathcal{O}(KD')$ is the additional complexity for computing the inter-cluster neighbor map and encoding the cluster indices. $\mathcal{O}(d_i D' M)$ is the computation cost for computing the intra-cluster neighbor map. In contrast, the inference time complexity of a regular GAE with a regular edge decoder is $\mathcal{O}(D'KM + C_{\text{MLP}})$. We note that d_i is upper bounded by the degree of the node v_i. In most graph datasets, the average degree of nodes is usually very small. For instance, on CoraFull, where the average node degree is 6.41, we have $d_i \leq 6.41$. As a result, $D'(K + d_i M) \ll D'KM$. For example, setting $K = 200$ and $M = 100$ on Pubmed, we find that the inference time complexity of HGAE with hierarchical edge reconstruction is $\mathcal{O}(841D' + C_{\text{MLP}})$, which is much more efficient than the regular edge decoder whose inference time complexity is $\mathcal{O}(20000D' + C_{\text{MLP}})$. In general, the inference time complexity of HGAE with hierarchical edge reconstruction is much lower than that of GAE with a regular edge decoder.

Table 3: Statistics of the graph datasets.

Dataset	# Nodes	# Edges	# Features	# Classes
CoraFull	19,793	63,421	8,710	70
ogbn-arxiv	169,343	1,166,243	128	40
Coauthor-CS	18,333	81,894	6,805	15
DBLP	40,672	144,135	7,202	137
Roman-Empire	22,662	32,927	64	18
Citeseer	3,327	4,732	3,703	6
Amazon-Computers	13,752	245,861	767	10
Amazon-Photo	7,650	119,081	745	8

C Additional Experiment Details

C.1 Datasets

To evaluate the performance of our method on FSNC, we conduct experiments on eight widely used real-world benchmark datasets, which are CoraFull (Bojchevski & Günnemann, 2018), ogbn-arxiv (Hu et al., 2020), Coauthor-CS (Shchur et al., 2018), DBLP (Tang et al., 2008), Roman-Empire (Platonov et al., 2023), Amazon-Computers, Amazon-Photo (Shchur et al., 2018), and Citeseer (Sen et al., 2008) with their statistics summarized in Table 3. CoraFull is an extended version of the Cora dataset, constructed from the entire citation network, where nodes represent papers and edges denote citation links; node classes correspond to paper topics. ogbn-arxiv is a directed citation graph derived from the arXiv Computer Science category in the Microsoft Academic Graph (MAG) (Wang et al., 2020a), where nodes are arXiv papers and edges represent citation relations. Node labels are based on 40 CS subject areas. Coauthor-CS is a co-authorship graph extracted from MAG during the KDD Cup 2016 challenge, where nodes denote authors and edges indicate co-authorship. Node features are derived from paper keywords, and node classes correspond to the authors' most active research fields. DBLP is another citation network in which nodes represent papers and edges denote citation links. Node features are based on paper abstracts, and labels correspond to publication venues. Roman-Empire is a synthetic dependency graph designed to simulate extreme heterophily where adjacent nodes often belong to different classes. Nodes represent words and edges reflect syntactic dependencies, with class labels assigned based on grammatical roles. Amazon-Computers and Amazon-Photo are two product co-purchase networks from the Amazon dataset, where nodes represent products and edges indicate frequently co-purchased items. Node features are extracted from product reviews, and class labels represent product categories. Citeseer is a citation network where nodes are scientific publications and edges represent citation links. Node features are TF-IDF weighted word vectors, and classes correspond to research topics.

C.2 Training Settings of HGAE and LDM

The training of the HGAE is divided into two phases. In the first phase, we pre-train the HGAE by only minimizing the node reconstruction loss and the edge reconstruction loss in Equation (1) for 500 epochs. In the second phase, we minimize $\mathcal{L}_{\text{HGAE}}$ with the prototypical loss for another 500 epochs. We use the Adam optimizer with a learning rate of 0.001 for the training. The weight decay is set to 1×10^{-5}. We train the LDM in the LR-FGDM after finishing the training of the HGAE. We use the Adam optimizer with a learning rate of 0.0002 to train the LDM for 1000 epochs.

The rank parameter r_0 and the weighting parameter τ associated with the TNN loss are selected through cross-validation tailored to each dataset. We define the rank as $r_0 = \lceil \gamma \min\{N, d\} \rceil$, where γ represents the rank ratio and d is the dimension of the learned node representations. The hyperparameter γ is searched over the set $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$, while the TNN weight τ is chosen from $\{0.05, 0.1, 0.15, 0.2, 0.25, 0.3, 0.35, 0.4, 0.45, 0.5\}$. The number of the prototype clusters, K, is selected from $\{5, 10, 15, 20, 25\}$.

D Illustration of the Synthetic Support Node Generation by LR-FGDM

Figure 4 illustrates the generation of the synthetic support nodes and the associated synthetic edges by the LR-FGDM and the DoG (Wang et al., 2025), with the difference between LR-FGDM and DoG marked in red.

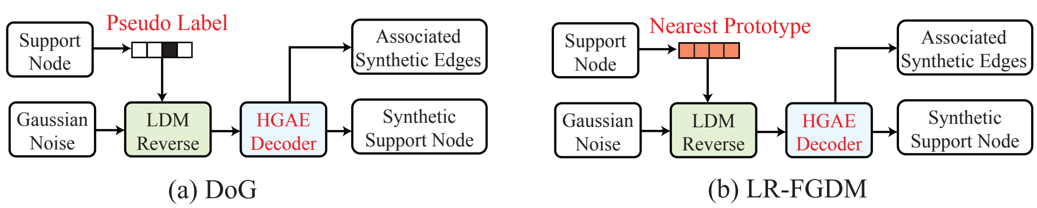

Figure 4: Figure (a) illustrates the generation of the synthetic support nodes and the associated synthetic edges by the DoG (Wang et al., 2025). Figure (b) illustrates the generation of the synthetic support nodes and the associated synthetic edges by the LR-FGDM.

E Details and Studies on the Hierarchical Edge Reconstruction Method

Figure 5 illustrates the structure of the network used for the hierarchical edge reconstruction in the HGAE with prototypical regularization. In contrast to the Bi-Level Neighborhood Decoder (BLND) employed in DoG (Wang et al., 2025) using balanced K-means on node attributes, our method leverages prototype cluster assignment and prototype representations learned jointly with the encoder of the HGAE with prototypical regularization, because nodes in the same prototype cluster have similar latent features, thus tend to connect with each other.

To validate the effectiveness of the hierarchical edge reconstruction method compared to the BLND proposed in DoG (Wang et al., 2025), we perform an ablation study by comparing the performance of the LR-FGDM with an ablation model where the hierarchical edge reconstruction module is replaced by the BLND in DoG (Wang et al., 2025). The ablation model is denoted as LR-FGDM (BLND). The study is performed for the 5-way 5-shot FSNC task on CoraFull, ogbn-arxiv, Coauthor-CS, and DBLP. It is observed in Table 4 that LR-FGDM consistently outperforms LR-FGDM (BLND) across all datasets. For example, LR-FGDM achieves a 1.16% improvement on ogbn-arxiv, highlighting the superiority of the proposed hierarchical edge reconstruction method in capturing meaningful structural patterns for few-shot learning. These results underscore the benefits of leveraging prototype-guided inter-cluster and intra-cluster connectivity over purely attribute-based neighborhood decoders like BLND.

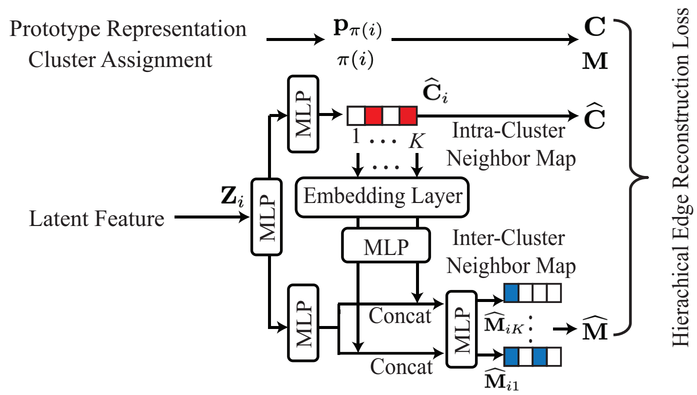

Figure 5: Illustration of the network architecture for the hierarchical edge reconstruction method in the HGAE with prototypical regularization.

Table 4: Performance comparison between the proposed hierarchical edge reconstruction method and the Bi-Level Neighborhood Decoder (BLND) in DoG (Wang et al., 2025) for the 5-way 5-shot FSNC task.

Data	CoraFull	ogbn-arxiv	Coauthor-CS	DBLP
LR-FGDM (BLND)	88.52	68.47	88.96	88.37
LR-FGDM	**89.66**	**69.63**	**89.83**	**89.51**

F Additional Experiment Results

F.1 Few-Shot Node Classification on Citeseer, Amazon Computers, and Amazon Photos

In this section, we further validate the effectiveness of LR-FGDM for FSNC on additional datasets, including Citeseer (Sen et al., 2008), Amazon Computers (Shchur et al., 2018), and Amazon Photos (Shchur et al., 2018). Due to the limited number of classes in these three datasets, we follow (Wu et al., 2024) and only perform FSNC under the 2-way 1-shot and the 2-way 5-shot settings. It is observed in Table 5 that LR-FGDM consistently improves the performance of COSMIC and COLA on all three datasets and significantly outperforms all competing FSNC methods. To demonstrate the statistical significance of the improvements achieved by LR-FGDM over the baseline methods, we perform t-tests on the few-shot classification accuracies obtained from 20 independent few-shot tasks for each setting and each dataset. It is observed in Table 6 that models enhanced by LR-FGDM consistently yield statistically significant improvements over the corresponding methods across all few-shot settings with p-values $p < 0.05$.

Table 5: The overall FSNC results of all methods under different settings for Amazon-Computers, Amazon-Photo, and Citeseer. The node classification accuracy and its standard deviation are in %. The best result under each setting is in bold, and the second-best result is underlined.

Method	Amazon-Computers		Amazon-Photo		Citeseer	
	2-way 1-shot	2-way 5-shot	2-way 1-shot	2-way 5-shot	2-way 1-shot	2-way 5-shot
ProtoNet (Snell et al., 2017)	56.67 ± 2.54	63.11 ± 2.60	66.74 ± 2.08	72.64 ± 1.94	67.39 ± 1.65	79.02 ± 2.33
Meta-GNN (Zhou et al., 2019)	60.54 ± 2.79	68.36 ± 2.15	69.34 ± 2.03	76.20 ± 1.87	67.41 ± 1.60	79.08 ± 2.27
GPN (Ding et al., 2020)	63.85 ± 2.31	71.02 ± 2.07	72.35 ± 1.92	77.88 ± 1.74	69.12 ± 1.68	80.02 ± 2.14
G-Meta (Huang & Zitnik, 2020)	62.56 ± 3.11	71.47 ± 2.97	70.18 ± 2.10	77.45 ± 1.81	65.53 ± 1.58	78.01 ± 1.80
TENT (Wang et al., 2022)	77.74 ± 3.16	86.06 ± 2.16	84.62 ± 2.78	86.53 ± 2.00	75.03 ± 2.81	85.31 ± 2.42
KD-FSNC (Wu et al., 2024)	86.92 ± 1.74	95.30 ± 0.85	91.08 ± 2.17	96.60 ± 0.41	79.48 ± 2.62	86.43 ± 1.32
NormProp (Zhang et al., 2025a)	85.10 ± 2.08	94.35 ± 1.30	90.42 ± 2.21	96.15 ± 0.53	78.41 ± 2.37	85.60 ± 1.61
COSMIC (Wang et al., 2023)	87.12 ± 1.82	95.60 ± 1.01	91.54 ± 2.04	96.12 ± 0.42	79.77 ± 2.20	86.23 ± 1.53
COLA (Huang & Zitnik, 2020)	87.52 ± 1.78	95.89 ± 1.02	91.74 ± 1.04	96.38 ± 0.33	80.13 ± 2.11	87.02 ± 1.30
COSMIC (LR-FGDM)	<u>88.63 ± 1.70</u>	<u>96.74 ± 0.97</u>	<u>92.38 ± 1.91</u>	<u>97.22 ± 0.29</u>	<u>80.92 ± 2.01</u>	<u>87.62 ± 1.27</u>
COLA (LR-FGDM)	$\mathbf{89.14 \pm 1.66}$	$\mathbf{97.21 \pm 0.89}$	$\mathbf{93.41 \pm 1.75}$	$\mathbf{97.45 \pm 0.26}$	$\mathbf{81.73 \pm 1.84}$	$\mathbf{88.21 \pm 1.23}$

F.2 Comparison with Existing Shot Augmentation Methods

In this section, we compare LR-FGDM with existing shot augmentation methods. IA-FSNC (Wu et al., 2022) incorporates confidently predicted query nodes as additional labeled instances and in-

Table 6: p-values from t-tests comparing COSMIC (LR-FGDM) and COLA (LR-FGDM) against their corresponding baseline methods, COSMIC and COLA, in Table 5.

Dataset	Amazon-Computers		Amazon-Photo		Citeseer	
Task	2-way 1-shot	2-way 5-shot	2-way 1-shot	2-way 5-shot	2-way 1-shot	2-way 5-shot
COSMIC (LR-FGDM)	0.026	0.018	0.021	0.012	0.034	0.017
COLA (LR-FGDM)	0.008	0.004	0.006	0.003	0.010	0.005

troduces noise-based perturbations to the node features. We also compare LR-FGDM with the best-performing GAN-based synthetic graph structure generation method, Semantic-aware Node Synthesis (SNS) (Gao et al., 2023b), which is originally proposed to generate synthetic nodes in the minority class for imbalanced datasets with a Generative Adversarial Network (GAN) (Goodfellow et al., 2020). SNS is adapted to the FSNC scenario as a baseline for shot augmentation by generating synthetic support nodes. We further compare LR-FGDM with another shot augmentation method SMILE (Liu et al., 2025b), which augments the support set using a mix-up strategy. For a fair comparison, we apply IA-FSNC, SNS, and SMILE to augment the support set in COLA for 5-way 5-shot FSNC in the same manner as LR-FGDM. It is observed in Table 7 that LR-FGDM always outperforms all the competing shot augmentation methods across multiple graph datasets and few-shot settings.

Table 7: Comparison of LR-FGDM with existing shot augmentation methods on 5-way 5-shot node classification. All methods are applied to augment the COLA (Huang & Zitnik, 2020). The statistical significance of the results is deferred to Table 19 of the appendix.

Method	CoraFull	ogbn-arxiv	Coauthor-CS	DBLP
COLA (Huang & Zitnik, 2020)	87.83	67.52	87.54	87.23
COLA (SMILE) (Liu et al., 2025b)	88.21	68.01	88.02	88.07
COLA (IA-FSNC) (Wu et al., 2022)	88.36	<u>68.17</u>	88.21	88.16
COLA (SNS) (Gao et al., 2023b)	<u>88.49</u>	68.32	<u>88.34</u>	<u>88.28</u>
COLA (LR-FGDM)	**89.66**	**69.63**	**89.83**	**89.51**

F.3 Study on the Kernel Complexity (KC) and the Upper Bound for the Test Loss in Theorem 3.1

In this section, we study the effectiveness of LR-FGDM in reducing the upper bound for the test loss in Equation (3) and the two terms in it, including the kernel complexity of the kernel gram matrix, $\mathrm{KC}(\mathbf{K})$, and the training loss of the node classifier with the ground-truth label, $L(\mathbf{K}, \mathbf{Y}_{\mathrm{FS}}, t)$. The study is performed for 5-way 5-shot node classification tasks on Cora-Full and Coauthor-CS with two baseline models, COSMIC (Wang et al., 2023) and COLA (Huang & Zitnik, 2020), as well as the corresponding models augmented by LR-FGDM, which are COSMIC (LR-FGDM) and COLA (LR-FGDM). It is observed from Table 8 that the upper bound for the test loss for the few-shot node classifiers trained by LR-FGDM is significantly lower than that of the baseline without low-rank regularization. Furthermore, LR-FGDM achieves substantially lower values in both $\mathrm{KC}(\mathbf{K})$ and $L(\mathbf{K}, \mathbf{Y}_{\mathrm{FS}}, t)$ in the upper bound, validating the theoretical motivation behind low-rank regularization and confirming the robustness and generalization benefits of LR-FGDM in FSNC.

F.4 Effectiveness of LR-FGDM on Heterophilic Graphs

While most existing FSNC studies have primarily targeted homophilous graphs, where connected nodes tend to share similar labels, many real-world graphs exhibit heterophily, where neighboring nodes often belong to different classes. In such cases, standard GNN-based few-shot methods face fundamental limitations, as neighborhood aggregation mechanisms become less effective or even detrimental to learning discriminative node representations. This poses an even greater challenge under few-shot conditions, where only a handful of labeled nodes per class are available to guide the model. To assess the generalization capability of our proposed LR-FGDM in this challenging scenario, we conduct experiments on the Roman-Empire dataset (Platonov et al., 2023) following COLA (Huang & Zitnik, 2020), which is a syntactic word dependency graph characterized by extreme heterophily. In this graph, node labels reflect grammatical roles rather than local connectivity, making the graph structure highly non-homophilous. To demonstrate the statistical significance of the improvements achieved by LR-FGDM over the baseline methods, we perform t-tests on the

Table 8: Comparisons on $L(\mathbf{K}, \mathbf{Y}_{\text{FS}}, t)$, $\text{KC}(\mathbf{K})$ and the value of the upper bound for the test loss from Theorem 3.1. The lowest values for each dataset in the table are bold, and the second-lowest values are underlined. A baseline without "(LR-FGDM)" indicates that the low-rank regularization is not used for training the transductive linear classifier in , and a method "(LR-FGDM)" indicates that the low-rank regularization is used.

Datasets		COSMIC (Wang et al., 2023)	COLA (Huang & Zitnik, 2020)	**COSMIC (LR-FGDM)**	**COLA (LR-FGDM)**
CoraFull	$L(\mathbf{K}, \mathbf{Y}_{\text{FS}}, t)$	6.44	6.38	<u>3.72</u>	**3.65**
	KC	0.35	0.40	<u>0.20</u>	**0.18**
	Upper Bound	10.80	11.25	<u>7.05</u>	**6.74**
ogbn-arxiv	$L(\mathbf{K}, \mathbf{Y}_{\text{FS}}, t)$	4.54	4.69	<u>4.02</u>	**3.95**
	KC	0.47	0.50	<u>0.24</u>	**0.21**
	Upper Bound	9.40	9.84	<u>8.20</u>	**7.97**
Coauthor-CS	$L(\mathbf{K}, \mathbf{Y}_{\text{FS}}, t)$	4.26	3.95	<u>3.38</u>	**3.40**
	KC	0.52	0.66	<u>0.30</u>	**0.28**
	Upper Bound	7.99	7.63	<u>6.25</u>	**6.16**
DBLP	$L(\mathbf{K}, \mathbf{Y}_{\text{FS}}, t)$	4.63	4.41	<u>3.75</u>	**3.58**
	KC	0.48	0.53	<u>0.26</u>	**0.23**
	Upper Bound	8.25	8.01	<u>6.85</u>	**6.50**

few-shot classification accuracies obtained from 20 independent few-shot tasks for each setting. It is observed in Table 10 that models enhanced by LR-FGDM consistently yield statistically significant improvements over the corresponding methods across all few-shot settings with p-values $p < 0.05$.

Table 9: FSNC results on the Roman-Empire dataset, which features extreme heterophily. Accuracy and standard deviation are reported in %. The best result for each setting is in bold, and the second-best is underlined.

Method	2-way 1-shot	2-way 5-shot	5-way 1-shot	5-way 5-shot
MAML (Finn et al., 2017)	42.83 ± 2.31	50.12 ± 2.24	19.45 ± 1.20	25.73 ± 1.41
ProtoNet (Snell et al., 2017)	48.67 ± 2.65	61.34 ± 2.47	28.52 ± 1.69	44.10 ± 1.72
Meta-GNN (Zhou et al., 2019)	63.45 ± 3.20	73.28 ± 2.85	44.80 ± 2.14	60.33 ± 2.06
GPN (Ding et al., 2020)	62.10 ± 3.14	74.01 ± 2.63	42.73 ± 2.21	63.45 ± 1.90
AMM-GNN (Wang et al., 2020b)	65.02 ± 3.05	76.48 ± 2.13	48.92 ± 2.39	66.22 ± 1.84
G-Meta (Huang & Zitnik, 2020)	66.74 ± 3.22	78.36 ± 2.14	50.14 ± 2.43	66.40 ± 1.75
TENT (Wang et al., 2022)	66.23 ± 3.08	77.29 ± 2.39	45.73 ± 2.01	61.78 ± 1.81
KD-FSNC (Wu et al., 2024)	69.15 ± 2.59	80.16 ± 2.11	58.92 ± 2.34	73.28 ± 1.90
NormProp (Zhang et al., 2025a)	69.02 ± 2.67	80.13 ± 2.14	57.84 ± 2.30	72.46 ± 1.92
COSMIC (Wang et al., 2023)	71.84 ± 2.71	82.35 ± 2.26	60.25 ± 2.42	75.33 ± 2.05
COLA (Huang & Zitnik, 2020)	70.96 ± 2.44	81.48 ± 2.09	59.31 ± 2.37	74.02 ± 1.88
COSMIC (LR-FGDM)	<u>73.38 ± 2.56</u>	<u>83.79 ± 2.18</u>	<u>61.45 ± 2.34</u>	<u>76.52 ± 1.91</u>
COLA (LR-FGDM)	$\mathbf{75.62 \pm 2.35}$	$\mathbf{85.41 \pm 2.02}$	$\mathbf{63.18 \pm 2.11}$	$\mathbf{78.23 \pm 1.76}$

Table 10: p-values from t-tests comparing COSMIC (LR-FGDM) and COLA (LR-FGDM) against their corresponding baseline methods, COSMIC and COLA, on Roman-Empire.

Task	2-way 1-shot	2-way 5-shot	5-way 1-shot	5-way 5-shot
COSMIC (LR-FGDM)	0.034	0.024	0.028	0.012
COLA (LR-FGDM)	0.019	0.022	0.013	0.002

F.5 Cross-Validation on the Number of Synthetic Nodes

The number of synthetic nodes generated for each novel class given each support node, denoted as q, plays a crucial role in determining the effectiveness of LR-FGDM. While generating more synthetic nodes can potentially enrich the support set and provide stronger supervision signals, it may also introduce redundancy or noise if excessive synthetic samples are added. In our experiments, we select the value of q for different datasets using 5-fold cross-validation over the base training classes. The value of q is selected from a range of candidate values, including $\{1, 2, 3, 4, 5, 6, 7, 8, 9, 10\}$.

F.6 Training Efficiency Analysis

To study the efficiency of the FGDM, we compare the training time between our HGAE with hierarchical edge reconstruction and the regular GAE without hierarchical edge reconstruction. In addition, we also compare the time for the generation of our FGDM and FGDM without hierarchical edge reconstruction in its HGAE. All evaluations are conducted using a single Nvidia A100

Table 11: The selected number of synthetic nodes per support node (q) for each dataset and few-shot setting.

Dataset	2-way 1-shot	2-way 5-shot	5-way 1-shot	5-way 5-shot
CoraFull	2	3	3	5
ogbn-arxiv	5	8	5	7
Coauthor-CS	4	6	4	5
DBLP	5	3	4	5

GPU. It is observed from the results in Table 12 that the hierarchical edge reconstruction method significantly reduces the computation cost of the training and synthetic graph structure generation. For instance, the training of GAE without hierarchical edge reconstruction takes over five times the training time of our GAE with hierarchical edge reconstruction on ogbn-arxiv. In addition, the hierarchical edge reconstruction method also significantly reduces the time for synthetic graph structure generation. For instance, the data generation without the hierarchical edge reconstruction method takes over four times the data generation time of our FGDM with the hierarchical edge reconstruction method on ogbn-arxiv.

Table 12: Time for the training of GAE and LDM in FGDM and data generation with FGDM on different datasets.

Datasets	Training Time (minutes)			Generation Time (s/sample)	
	HGAE	GAE without hierarchical edge reconstruction	LDM	FGDM	FGDM without hierarchical edge reconstruction
CoraFull	41	129	154	0.067	0.073
Coauthor CS	52	145	179	0.074	0.088
ogbn-arxiv	301	1690	315	0.130	0.426
DBLP	11	16	39	0.049	0.066

F.7 Sensitivity Analysis on the Hyperparameters τ, r_0, K, and q

In this section, we first conduct a sensitivity analysis τ, which is the weighting parameter for the TNN $\|\mathbf{K}\|_{r_0}$ in Equation 2. The study is performed using COLA (LR-FGDM) on the CoraFull dataset for the 5-way 5-shot node classification task. We evaluate the performance of the COLA (LR-FGDM) with τ varying in $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. As shown in Table 13, although the best performance is achieved at $\tau = 0.6$, the performance of COLA (LR-FGDM) remains stable across different values of τ. Even the lowest performing setting, $\tau = 0.1$, results in only a marginal 0.18% decrease in accuracy compared to the best result. In addition, we perform an ablation study to examine the influence of the rank parameter $r_0 = \lceil \gamma \min N, d \rceil$, where $\gamma \in (0, 1]$ controls the effective rank used in the truncated nuclear norm. We evaluate COLA (LR-FGDM) with γ varying from 0.05 to 0.5. As shown in Table 13, the accuracy is robust to different values of γ, with the highest performance observed at $\gamma = 0.2$. We also conduct an ablation study on the hyperparameter K, which denotes the number of clusters used for prototype regularization in the HGAE. We vary K from 5 to 50 with a step size of 5. As shown in Table 13, the performance remains stable across different values of K, with a slight peak at $K = 10$. This suggests that the model is not sensitive to the choice of cluster number, K.

Table 13: Sensitivity analysis on the weighting parameter τ for the TNN, the rank ratio γ used in $r_0 = \lceil \gamma \min\{N, d\} \rceil$, and the number of clusters K for prototype computation in COLA (LR-FGDM) for the 5-way 5-shot node classification task on CoraFull.

τ	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9
Accuracy	89.48	89.55	89.59	89.57	89.63	89.66	89.62	89.65	89.58

γ	0.05	0.1	0.15	0.2	0.25	0.3	0.35	0.4	0.45	0.5
Accuracy	89.27	89.44	89.58	89.66	89.62	89.55	89.64	89.60	89.60	89.55

K	5	10	15	20	25	30	35	40	45	50
Accuracy	89.30	89.66	89.58	89.64	89.53	89.59	89.62	89.56	89.48	89.44

Furthermore, we perform a sensitivity analysis to evaluate the effect of the number of synthetic nodes per support node, q, on FSNC performance under the 5-way 5-shot setting. For each dataset, we vary q from 1 to 10 and report the mean classification accuracy over 20 tasks. As shown in Table 14, increasing q generally leads to consistent improvements in performance up to a certain threshold, beyond which the gains tend to saturate or marginally decline. This trend highlights the benefit of augmenting the support set with a moderate number of synthetic nodes, which helps enhance generalization by enriching the local representation space. Notably, while further increasing q beyond the optimal value does not continue to improve performance, the resulting degradation is marginal.

Table 14: Sensitivity analysis on the number of synthetic nodes per support node (q) for four datasets under the 5-way 5-shot FSNC setting.

q	1	2	3	4	5	6	7	8	9	10
CoraFull	88.71	89.02	89.37	89.51	89.66	89.53	89.59	89.44	89.41	89.40
ogbn-arxiv	67.80	68.21	68.84	69.08	69.33	69.42	69.63	69.59	69.45	69.20
Coauthor-CS	88.33	88.82	89.08	89.31	89.83	89.62	89.54	89.48	89.30	89.12
DBLP	87.93	88.44	88.87	89.02	89.51	89.35	89.29	89.10	88.94	88.91

F.8 Quality Evaluation of the Augmented Graph

This paper introduces a novel node-level graph diffusion model named FGDM, which synthesizes the graph structures. The synthetic graph structure, which consists of the synthetic support nodes and the associated edges, generated by the FGDM, is subsequently combined with the original graph to form an augmented graph. In Section 4.2, we have shown that the FSNC methods trained on the augmented graph achieve significantly better performance. In this section, we directly evaluate the data quality of the synthetic graph structures generated by the FGDM. In the visual domain, the Frechet Inception Distance (FID) is a widely used metric to evaluate the quality of the synthetic images generated by the generative models (Brock et al., 2019; Ho et al., 2020). The FID score measures the similarity between the distribution of the generated images and the distribution of the real images. To compute the FID score, the pre-trained Inception v3 (Szegedy et al., 2016) is used to extract the features from both the real images and the generated images, which are then modeled as the multivariate Gaussian distributions. The FID score is then calculated using the Frechet Distance (FD) (Brock et al., 2019) between the two multivariate Gaussian distributions modeling the real and the generated images (Dowson & Landau, 1982). A lower FID score indicates that the generated images are more similar to the real images, suggesting better quality.

Quality Evaluation of the Synthetic Nodes. Although the Inception model cannot be applied to the graph data, we can replace the Inception model in the computation of the FID score with the pre-trained GCN (Kipf & Welling, 2017) for extracting node features to adapt the metric to evaluate the quality of synthetic nodes generated by the FGDM. To this end, we define the Frechet Node Distance (FND), which is the FD between the multivariate Gaussians modeling the node features extracted by pre-trained GCN. We randomly split the nodes from the novel classes in the original graph into two partitions of equal size, which are the base partition and the test partition. To mitigate the influence of the randomness, we compute the FND scores with 10 different random splits and report the mean and the standard deviation of the FND scores across different runs. The FND computed between the nodes in the test partition and the base partition of the original graph establishes the baseline of the expected FND score for high-quality support nodes. By computing the FND score between the features of the synthetic support nodes in the synthetic graph structures generated by the FGDM and the features of the nodes in the base partition of the original graph, we evaluate the quality of the synthetic support nodes. For simplicity, we refer to the FND score for the synthetic support nodes as the FND between their features and the features of the nodes in the base partition of the original graph. To show the effectiveness of the prototypical regularization in the training of the HGAE for the PGDM, we also compute the FND for the nodes in the synthetic graph structures generated by the PGDM without the prototypical regularization. To demonstrate the advantages of the FGDM over the vanilla diffusion model, the DDPM (Ho et al., 2020), we train a baseline DDPM model on the input node attributes of the original graph and synthesize the same number of synthetic nodes as the FGDM. The synthetic edges are then generated by connecting each synthetic node to its K-nearest neighbors in the original graph using the K-nearest neighbors (KNN) algorithm with $K = \lceil d_{\text{ave}} \rceil$, where d_{ave} is the average degree of the original graph. The synthetic graph structures, including the synthetic nodes and edges generated by the baseline DDPM model, are combined with the original graph to form the augmented graph. Next, we compute the FND score for the nodes in the synthetic graph structures generated by the baseline DDPM model. In addition, we also compute the FND for the support nodes generated by three competing shot augmentation methods, including SMILE (Liu et al., 2025b), IA-FSNC (Wu et al., 2022), and SNS (Gao et al., 2023b). The ablation study is performed for the 5-way 5-shot setting of the FSNC. The lower FND scores indicate that the node features are more similar to the features of nodes in the base partition of the original graph. It is observed in Table 15 that the FND score of the nodes in the synthetic graph structures generated by the FGDM is closest to the FND score of the original graph, which demonstrates that the FGDM generates faithful synthetic nodes.

Table 15: Frechet Node Distance (FND) to the nodes in the base partition of the original graph. The mean and standard deviation of the FND scores computed with 10 different random splits of the base partition and the test partition in the original graph are reported. The evaluation is performed for the 5-way 5-shot setting of the FSNC. The FND score for the original graph is computed between the nodes in the test partition and the nodes in the base partition of the original graph.

Data	CoraFull	ogbn-arxiv	Coauthor-CS	DBLP
Baseline DDPM	13.41±0.43	8.95±0.28	10.32±0.41	7.34±0.39
SMILE (Liu et al., 2025b)	10.21±0.39	6.03±0.32	8.21±0.43	6.48±0.34
IA-FSNC (Wu et al., 2022)	9.92±0.41	5.87±0.29	7.88±0.36	6.12±0.37
SNS (Gao et al., 2023b)	9.73±0.40	5.69±0.31	7.45±0.34	6.01±0.33
FGDM (w/o Prototypical Regularization)	9.24±0.48	5.23±0.30	7.13±0.47	5.95±0.36
FGDM	**8.10±0.27**	**4.39±0.30**	**5.29±0.21**	**4.08±0.42**
Original Graph	7.95±0.32	4.33±0.27	4.21±0.26	3.84±0.33

Quality Evaluation of the Synthetic Edges. Similar to the design of the FND score for evaluating the quality of synthetic nodes, we replace the Inception model in the computation of FID with the pre-trained GNN (Zhu et al., 2021) for edge feature extraction to adapt the metric to evaluate the quality of the synthetic edges generated by the FGDM. To this end, we define Frechet Edge Distance (FED), which is the FD between the multivariate Gaussians modeling the edge features extracted by a pre-trained GNN. Similar to the evaluation of the FND, we randomly split the edges in the original graph into two partitions of equal size, which are the base partition and the test partition. To mitigate the influence of the randomness, we compute the FED scores with 10 different random splits and report the mean and the standard deviation of the FED across different runs. The FED computed between the edges in the test partition and the edges in the base partition of the original graph establishes the baseline of the expected FED score for the high-quality edges. By computing the FED between the features of edges in the synthetic graph structures generated by the FGDM and the features of edges in the base partition of the original graph, we evaluate the quality of the edges in the synthetic graph structures. For simplicity, we refer to the FED score for the synthetic edges as the FED between their features and the features of edges in the base partition of the original graph. Similar to the evaluation of the synthetic nodes, we also compute the FED score for the edges in the synthetic graph structures generated by the FGDM without the prototypical regularization. We compute the FED score for the edges in the synthetic graph structures generated by the baseline DDPM model. Since the edges in the synthetic graph structures are generated by the KNN algorithm, we evaluate the baseline DDPM models using different values of K from $\{\lceil d_{\text{ave}}/4\rceil, \lceil d_{\text{ave}}/2\rceil, \lceil d_{\text{ave}}\rceil, 2\times\lceil d_{\text{ave}}\rceil, 4\times\lceil d_{\text{ave}}/4\rceil\}$. In addition, we also compute the FED for the edges generated by three competing shot augmentation methods, including SMILE (Liu et al., 2025b), IA-FSNC (Wu et al., 2022), and SNS (Gao et al., 2023b). The FED score for edges in the original graph is also computed. We use the same NBFNet (Zhu et al., 2021) pre-trained on the original graph to extract the edge features for computing the FED score. The ablation study is performed for the 5-way 5-shot setting of the FSNC. Lower FED scores indicate that the edge features are more similar to the features of edges in the base partition of the original graph. It is observed in Table 16 that the FED score of the edges in the synthetic graph structures generated by the FGDM is closest to the FED score of the original graph, which demonstrates that the FGDM generates more faithful synthetic edges compared to the competing methods.

Table 16: Frechet Edge Distance (FED) to the edges in the base partition of the original graph. The mean and standard deviation of the FED scores computed with 10 different random splits of the base partition and the test partition in the original graph are reported. The evaluation is performed for the 5-way 5-shot setting of the FSNC. The FED score for the original graph is computed between the nodes in the test partition and the nodes in the base partition of the original graph.

Data	CoraFull	ogbn-arxiv	Coauthor-CS	DBLP
Baseline DDPM ($K = \lceil d_{\text{ave}}/4 \rceil$)	12.17±0.47	8.85±0.30	10.89±0.38	8.05±0.41
Baseline DDPM ($K = \lceil d_{\text{ave}}/2 \rceil$)	11.34±0.36	8.53±0.28	9.43±0.34	7.32±0.33
Baseline DDPM ($K = \lceil d_{\text{ave}} \rceil$)	10.48±0.39	8.01±0.31	9.04±0.36	6.88±0.32
Baseline DDPM ($K = \lceil 2 \times d_{\text{ave}} \rceil$)	10.51±0.35	7.96±0.29	9.17±0.37	6.94±0.34
Baseline DDPM ($K = \lceil 4 \times d_{\text{ave}} \rceil$)	10.92±0.42	8.05±0.33	9.41±0.39	7.10±0.36
SMILE (Liu et al., 2025b)	10.41±0.36	7.89±0.30	8.93±0.34	6.67±0.31
IA-FSNC (Wu et al., 2022)	10.96±0.34	7.80±0.29	8.62±0.33	6.39±0.30
SNS (Gao et al., 2023b)	10.18±0.35	7.77±0.28	8.66±0.32	6.46±0.29
FGDM (w/o Prototypical Regularization)	10.13±0.38	7.70±0.27	8.79±0.35	6.33±0.30
FGDM	**8.29±0.30**	**5.33±0.24**	**6.56±0.26**	**5.39±0.27**
Original Graph	8.10±0.27	5.14±0.22	6.37±0.25	5.19±0.26

Table 17: p-values from t-tests comparing COSMIC (LR-FGDM) and COLA (LR-FGDM) against their corresponding baseline methods, COSMIC and COLA, in Table 1.

Dataset	CoraFull				ogbn-arxiv			
Task	2-way 1-shot	2-way 5-shot	5-way 1-shot	5-way 5-shot	2-way 1-shot	2-way 5-shot	5-way 1-shot	5-way 5-shot
COSMIC (LR-FGDM)	0.031	0.022	0.018	0.014	0.027	0.039	0.044	0.017
COLA (LR-FGDM)	0.009	0.005	0.006	0.004	0.012	0.007	0.011	0.003

Dataset	Coauthor-CS				DBLP			
Task	2-way 1-shot	2-way 5-shot	5-way 1-shot	5-way 5-shot	2-way 1-shot	2-way 5-shot	5-way 1-shot	5-way 5-shot
COSMIC (LR-FGDM)	0.013	0.015	0.034	0.025	0.019	0.011	0.027	0.014
COLA (LR-FGDM)	0.007	0.003	0.009	0.004	0.006	0.002	0.005	0.003

Table 18: p-values from t-tests comparing the COLA (LR-FGDM) with the second-best ablation model in Table 2.

Dataset	CoraFull	ogbn-arxiv	Coauthor-CS	DBLP
p-values	0.012	0.009	0.043	0.008

Table 19: p-values from t-tests comparing the COLA (LR-FGDM) with the second-best ablation model in Table 7.

Dataset	CoraFull	ogbn-arxiv	Coauthor-CS	DBLP
p-values	0.025	0.032	0.026	0.018

F.9 Improvement Statistical Significance Analysis

To demonstrate the statistical significance of the improvements achieved by LR-FGDM over the baseline methods, we perform t-tests on the few-shot classification accuracies obtained from 20 independent few-shot tasks for each setting. For results in Table 1, we compare COSMIC (LR-FGDM) and COLA (LR-FGDM) against their corresponding baseline methods, COSMIC and COLA. It is observed in Table 17 that models enhanced by LR-FGDM consistently yield statistically significant improvements over the corresponding methods across all datasets and settings with p-values $p < 0.05$. In addition, we further validate the statistical significance of the improvements of COLA (LR-FGDM) over the ablation models in Table 2 and models augmented by other support set augmentation methods in Table 7. It is observed from Table 18 and Table 19 that COLA (LR-FGDM) significantly outperforms all competing variants, with all p-values below 0.05, further supporting the statistical significance of the improvements by LR-FGDM over existing methods.

Algorithm 1 Training FGDM (Training the HGAE and the LDM)

Input: The input attribute matrix $\mathbf{X}$, adjacency matrix $\mathbf{A}$, the training epochs of the HGAE t_{HGAE}, the labels Y_{base} of the labeled nodes $\mathcal{V}_{\text{base}}$ in the base training set, the training epochs of the LDM t_{LDM}, and the learning rate η

Output: The parameters of the HGAE $\boldsymbol{\omega}$ and the parameters of the LDM $\boldsymbol{\theta}$

1: Obtain the inter-cluster neighbor map $\mathbf{C}$ and the intra-cluster neighbor map $\mathbf{M}$ by applying balanced K-Means clustering on $\mathbf{X}$
2: Initialize the parameter $\boldsymbol{\omega}$ of the HGAE
3: **for** $t \leftarrow 1$ to t_{HGAE} **do**
4:     Update $\boldsymbol{\omega}$ by $\boldsymbol{\omega} \leftarrow \boldsymbol{\omega} - \eta \nabla_{\boldsymbol{\omega}} L_{\text{HGAE}}$ with L_{HGAE} from Eq.(1)
5: **end for**
6: Compute cluster prototypes $\{\mathbf{p}_c\}$ from the latent representations $\mathbf{Z}$ using semi-supervised K-means clustering (Bair, 2013)
7: Assign each node to its cluster prototype based on the clustering result
8: Initialize the parameter $\boldsymbol{\theta}$ of the LDM
9: Map the node attributes $\mathbf{X}$ and the adjacency matrix $\mathbf{A}$ to the latent space using the encoder g_e of the HGAE as $\mathbf{H} = g_e(\mathbf{X}, \mathbf{A})$
10: Train the LDM with CFG (Ho & Salimans, 2022) on latent pairs $(\mathbf{Z}_i, \mathbf{p}_{\pi(i)})$ where $\mathbf{p}_{\pi(i)}$ is the prototype associated with node v_i
11: **return** The parameters of the HGAE $\boldsymbol{\omega}$ and the parameters of the LDM $\boldsymbol{\theta}$

Algorithm 2 Generation of the Augmented Graph $\mathcal{G}_{\text{aug}}$

Input: The input attribute matrix $\mathbf{X}$, the adjacency matrix $\mathbf{A}$, the set of support nodes $\mathcal{V}_{\text{sup}} = \{v_1, \ldots, v_S\}$, number of synthetic nodes per support node q, and the prototype assignments $\{\mathbf{p}_{\pi(i)}\}_{i=1}^{S}$

Output: The augmented graph $\mathcal{G}_{\text{aug}} = (\mathcal{V} \cup \mathcal{V}_{\text{syn}}, \mathbf{X}_{\text{aug}}, \mathbf{A}_{\text{aug}})$

1: Let $M = q \times |\mathcal{V}_{\text{sup}}|$, total number of synthetic nodes
2: Initialize counter $m \leftarrow 1$
3: **for** $i \leftarrow 1$ to $|\mathcal{V}_{\text{sup}}|$ **do**
4:     **for** $j \leftarrow 1$ to q **do**
5:         Sample noise $\boldsymbol{\varepsilon} \sim \mathcal{N}(\mathbf{0}, \mathbf{I})$
6:         Condition on prototype $\mathbf{p}_{\pi(i)}$ of support node v_i
7:         Generate latent feature $\widehat{\mathbf{Z}}_m$ using LDM conditioned on $\mathbf{p}_{\pi(i)}$
8:         $m \leftarrow m + 1$
9:     **end for**
10: **end for**
11: Decode $\widehat{\mathbf{Z}} = \{\widehat{\mathbf{Z}}_i\}_{i=1}^{M}$ to $\mathbf{X}_{\text{syn}}$ and $\mathbf{A}_{\text{syn}}$ with the decoder of the HGAE $(\mathbf{X}_{\text{syn}}, \mathbf{A}_{\text{syn}}) = g_d(\widehat{\mathbf{Z}})$
12: Form augmented attribute matrix: $\mathbf{X}_{\text{aug}} = [\mathbf{X}; \mathbf{X}_{\text{syn}}]$
13: Form augmented adjacency matrix: $\mathbf{A}_{\text{aug}} = [\mathbf{A}\ \ \mathbf{A}_{\text{syn}}; \mathbf{A}_{\text{syn}}\ \ \mathbf{A}]$
14: **return** $\mathcal{G}_{\text{aug}} = (\mathcal{V} \cup \mathcal{V}_{\text{syn}}, \mathbf{X}_{\text{aug}}, \mathbf{A}_{\text{aug}})$

G Algorithms for the Training of LR-FGDM

We present the training algorithm of the FGDM in Algorithm 1, which comprises two steps. The first step, which is from Line 1 to Line 5 in Algorithm 1, describes the training of the HGAE. The second step, which is from Line 6 to Line 10, describes the training of the LDM. Algorithm 2 describes the generation process for the augmented graph $\mathcal{G}_{\text{aug}}$.