





Graph Neural Networks With Triple Attention for Few-Shot Learning

Hao Cheng , Joey Tianyi Zhou , Senior Member, IEEE, Wee Peng Tay , Senior Member, IEEE, and Bihan Wen , Member, IEEE

Abstract—Recent advances in Graph Neural Networks (GNNs) have achieved superior results in many challenging tasks, such as few-shot learning. Despite its capacity to learn and generalize a model from only a few annotated samples, GNN is limited in scalability, as deep GNN models usually suffer from severe over-fitting and over-smoothing. In this work, we propose a novel GNN framework with a *triple-attention mechanism*, i.e., node self-attention, neighbor attention, and layer memory attention, to tackle these challenges. We provide both theoretical analysis and illustrations to explain why the proposed attentive modules can improve GNN scalability for few-shot learning tasks. Our experiments show that the proposed Attentive GNN model outperforms the state-of-the-art few-shot learning methods using both GNN and non-GNN approaches. The improvement is consistent over the mini-ImageNet, tiered-ImageNet, CUB-200-2011, and Flowers-102 benchmarks, using both ConvNet-4 and ResNet-12 backbones, and under both the inductive and transductive settings. Furthermore, we demonstrate the superiority of our method for few-shot fine-grained and semi-supervised classification tasks with extensive experiments.

Index Terms—Graph neural network, self-attention mechanism, few-shot classification, meta learning.

I. INTRODUCTION

THE success of deep learning lies with the promises of training deep neural networks from a large-scale dataset in a supervised manner. However, conventional deep learning methods may suffer from sample inefficiency, thus the trained models can hardly generalize to new tasks with limited supervision. Such task is known as *few-shot learning* [1], which attempts

Manuscript received 19 August 2022; revised 20 November 2022 and 20 December 2022; accepted 21 December 2022. Date of publication 2 January 2023; date of current version 8 December 2023. This work was supported in part by Joey Tianyi Zhou's A*STAR SERC Central Research Fund (Use-inspired Basic Research) and Singapore Government's Research, Innovation and Enterprise 2020 Plan (Advanced Manufacturing and Engineering domain) under Grant A18A1b0045, in part by Singapore Ministry of Education Academic Research Fund Tier 2 under Grant MOE-T2EP20220-0002, and in part by AcRF Tier 1 under Grant RG61/22 and Start-Up Grant. The associate editor coordinating the review of this manuscript and approving it for publication was Mr. Jingkuan Song. (Corresponding author: Bihan Wen.)

Hao Cheng, Wee Peng Tay, and Bihan Wen are with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore 639798 (e-mail: hao006@e.ntu.edu.sg; wptay@ntu.edu.sg; bihan.wen@ntu.edu.sg).

Joey Tianyi Zhou is with the A*STAR Centre for Frontier AI Research, Singapore 138632 (e-mail: zhouty@cfar.a-star.edu.sg).

The code for this work is publicly available at <https://github.com/chenghao94/AGNN>.

Digital Object Identifier 10.1109/TMM.2022.3233442

to learn a classifier predicting the novel labels of query samples using only a few labeled support samples of each class. To tackle the few-shot learning challenges, various methods have been recently proposed [2], [3], [4], [5], including the popular *meta-learning* framework [2] based on *episodic training*.

Classic few-shot methods [2], [3], [4], [6] applied Convolutional Neural Networks (CNNs) for image classification. More recent works proposed to apply the Graph Neural Networks (GNNs) [5], [7], [8], [9] or Graph Convolutional Networks (GCNs) [10], [11] to process data with rich relational structures in few-shot scenarios. Compared with CNNs, Graph Networks are more powerful in exploiting the intra- and inter-class relationships amongst samples, which are thus more effective for few-shot learning. The current GNN-based few-shot methods improve the accuracy and generalizability from the perspective of node/edge update [8], [9], [12], [13] and graph structure design [5], [14], [15]. In general, graph-based methods model the feature embeddings of samples as vertices in a graph and propagate label information between nodes by performing node or edge feature aggregation from neighbor nodes with graph convolution. Unlike general-purpose GNN models for other tasks, adjacency matrices have no structural priors in few-shot scenarios. Therefore, existing few-shot GNN methods usually construct fully connected graph models and utilize neighbor similarity information for graph updates.

GNNs have achieved superior performance on many tasks such as node classification [16], skeleton action recognition [17], point cloud classification [18], and video classification [19]. However, several works [20], [21], [22], [23] reported over-fitting and over-smoothing issues when learning deeper GNN models (i.e., poor scalability) as shown in Fig. 1, as applying GCN or GNN is a special form of Laplacian smoothing, which averages the neighbors of the target nodes. Furthermore, graph-based few-shot methods usually model each task as a fully connected graph, i.e., each node is adjacent to all other nodes, making it more prone to this issue. Some recent works [21], [23], [24], [25], [26] have been proposed to alleviate the above issues and designed deeper graph layers. DropEdge [21] attempted to alleviate these obstacles via randomly dropping graph edges in training, showing improvement for node classification tasks. DeepGCNs [23] borrowed the ideas from popular CNNs (e.g., ResNet [27] and DenseNet [24]) and adopted the residual/dense connections and dilated convolutions to deep GCN layers for point cloud semantic segmentation tasks. To the best

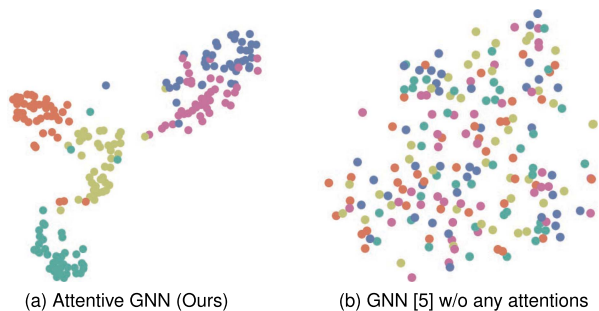


Fig. 1. T-SNE visualization of the image features extracted from the selected classes in deeper GNN layers (here we select the output feature of the 8-th GNN layer). Different colors mean different classes.

of our knowledge, no work to date has addressed these issues for few-shot learning using the graph attention mechanism.

In this work, we propose an attentive graph neural network (AGNN) with a novel triple-attention mechanism, i.e., node self-attention, neighbor attention, and layer memory attention for highly scalable and effective few-shot learning. Specifically, node self-attention exploits inter-node and inter-class correlation beyond CNN-based features and class information. Neighbor attention imposes sparsity on the adjacency matrices to attend to the most related neighbor nodes across layers. Layer memory attention applies dense connection to earlier-layer “memory” of node features and adjacency matrices. Furthermore, we explain how the attentive modules help GNN generate discriminative features and alleviate over-smoothing and over-fitting with feature visualization and theoretical analysis. We conduct extensive experiments demonstrating that the proposed AGNN outperforms state-of-the-art methods over four datasets, including two standard few-shot classification benchmarks, mini-ImageNet and tiered-ImageNet, and two fine-grained datasets, CUB and Flowers-102.

The contributions of this paper are summarized as follows,

- We propose an AGNN model which contains a triple-attention mechanism to tackle the over-fitting and over-smoothing problem and improve the few-shot performance for GNN models.
- We provide both theoretical analysis and visualizations to explain the effectiveness of the proposed AGNN for alleviating over-smoothing and over-fitting problems in few-shot scenarios for GNN-based models.
- Extensive experiments are conducted over four benchmarks for standard, fine-grained, and semi-supervised few-shot learning tasks under both inductive and transductive settings. Results show that our AGNN method achieves state-of-the-art performance over all benchmarks under all settings.

This paper is an extension of our recent conference work [28] that briefly investigated three attention mechanisms on GNNs for few-shot classification. Compared with this earlier work, here we provide more details and discussions about the relation to existing graph-based models. We also improve the performance of AGNN by modifying the proposed triple attentions. Specifically, we adopt the self-attention transformer block to

replace the self-correlation computation to provide a more flexible task-specific embedding for graph initialization. We combine the latter two attention mechanisms to propose a novel enhanced layer-wise sparsity mechanism. Compared with the previously proposed neighbor attention mechanism with a fixed sparse rate, each AGNN layer adopts a variable sparsity rate, which provides flexible relationships between nodes across different AGNN layers. For the layer memory attention mechanism, in addition to using the output feature as the layer memory, the output adjacency matrix of the current layer is also considered as part of the layer memory, which is passed to the next layer as edge knowledge. Furthermore, we include more extensive experimental results to illustrate the properties of the proposed method with extensive evaluation and comparisons on additional datasets under more challenging settings, e.g., fine-grained few-shot classification, and semi-supervised few-shot classification.

The remainder of this article is organized as follows. Section II summarizes the related work, including few-shot learning, graph neural network, and attention mechanism on graph models. Section III gives a brief overview of the few-shot learning task and the general GNN model. Section IV describes the proposed AGNN with triple attention mechanisms, how to apply it to the few-shot task, and the relation to existing graph-based models. Section V gives a theoretical analysis of the proposed attention mechanisms on why it helps to alleviate the over-smoothing and over-fitting problems for few-shot learning with graph models. Section VI demonstrates the effectiveness of the proposed AGNN model for few-shot classification over four benchmarks under standard, fine-grained, and semi-supervised few-shot settings. Section VII concludes this article.

II. RELATED WORK

A. Few-Shot Learning

Few-shot learning is a challenging task that aims to recognize novel categories with limited labeled examples of each class. Following the meta-learning framework [2], existing methods can be generally divided into three groups: gradient-based methods [29], [30], [31], [32], data augmentation-based methods [33], [34], [35], and metric-based methods [2], [3], [4], [6], [11], [36], [37], [38], [39], [40], [41], [42], [43]. Recently, a rising trend is to apply attention mechanisms to solve few-shot tasks. For example, CAN [44] generated cross attention maps for each pair of nodes to highlight the object regions for classification. Inspired by non-local block, Binary Attention Network [45] considered a non-local attention module to learn the similarity between node embeddings globally. Considering the attention between query samples with each support class, CTM [37] found task-relevant features based on both intra-class commonality and inter-class uniqueness. FEAT [11] utilized a self-attention Transformer to learn task-specific adaptive instance embeddings. RENet [46] employed self-correlation and cross-correlation modules to extract relational feature embeddings within and between images. SET-RCL [42] proposed a style-aware episodic training strategy with robust contrastive learning to learn a style-invariant feature representation for

cross-domain few-shot learning tasks. CubMeta [43] introduced the concept of curriculum learning into meta-learning and proposed an effective self-paced meta-learning method to obtain stronger meta-learners for few-shot classifications. DUAL ATT-NET [47] adopted a dual-attention to explicitly model the crucial relation of fine-grained parts and implicitly captures discriminative while subtle fine-grained details. While these methods are all based on CNNs for feature embedding, most recent works exploited GNN for more effective modeling of inter- and intra-class relations in few-shot classification. It is unclear how these attention schemes can be extended to GNN frameworks.

B. Graph Neural Network

GNN [48], [49] was first proposed for learning with graph-structured data and has proved to be a powerful technique for aggregating information from neighboring vertices in the graph. Recently, there is growing interest in GNNs [5], [7], [8], [9], [12], [13], [14], [15] to handle the few-shot learning task. GNN was first used for few-shot learning in [5], which aims to learn a complete graph network of nodes with both feature and class information. Based on the episodic training mechanism, meta-graph parameters were trained to predict the label of a query node on the graph. Later, TPN [12] introduced the transductive setting into few-shot learning and constructed a top- k graph to propagate labels from support set to query set in the graph. Besides node label information, EGNN [13] exploited edge information for the directed graph by defining both class and edge labels for fully exploring the internal information of the graph. DPGN [14] constructed a dual complete graph network to combine instance-level and distribution-level relations. MCGN [15] combined the GNN and conditional random field (CRF) as a unified model and models the graph affinity as the pair-wise marginal probabilities in the CRF for feature update. TLRM [7] proposed a sample-to-task relation module to capture the task-level relation representations in each GNN layer. TRPN-D [8] adopted the decoupling training strategy to preserve the diversity across different few-shot tasks to enhance the generalizability of GNN models. GCLR [9] applied a VAE-based encoder-decoder module to enrich the node representations in the latent feature space. DR-CapsGNN [19] extended the capsule network to the few-shot video classification task and explored local-global relations while preserving the detailed properties of videos.

C. Attention Mechanism on Graph Models

Attention Mechanism [50] aims to focus on image regions that are more task-related by learning a binary matrix or a weighted matrix. In particular, self-attention [47], [51], [52], [53] considers the inherent correlation (attention) of the input features itself, which is mostly applied in deep models. In GCN scenarios, GAT [54] used a graph attention layer to learn a weighted parameter vector based on entire neighborhoods to update node representation. ReGAT [55] modeled multi-type visual object relations via a graph attention mechanism to learn question-adaptive relation representations for VQA tasks. SAGPool [56] selected

the top- k percentage of nodes based on the self-attention score to generate a mask matrix for graph pooling. Despite the promising results achieved by these methods, no attention-based GNN is proposed specifically for few-shot learning. In this work, a novel triple-attention mechanism in GNN is introduced to alleviate the over-fitting and over-smoothing challenges in few-shot classification.

III. PRELIMINARIES

We first provide the formal problem definition of few-shot classification tasks, followed by an overview of the general GNN models.

A. Problem Definition

A general few-shot classification task consists of a large-scale and labeled training set with classes \mathcal{C}_{train} and a few-shot testing set with classes \mathcal{C}_{test} , which are mutually exclusive, i.e., $\mathcal{C}_{train} \cap \mathcal{C}_{test} = \emptyset$, few-shot image classification algorithms aim to train a classification model over the training set, which could be applied to the testing set with only a few labeled information of each given class. For the testing set, each few-shot test task \mathcal{T} follows the N -way K -shot task setting, where N is the number of selected classes and K is the number of labeled samples which is often set from 1 to 5, i.e., the testing set contains a labeled N -class support set $\mathcal{S} = \{x_i, y_i\}_{i=1}^{N \times K}$ with K samples of each class, and a query set $\mathcal{Q} = \{x_j, y_j\}_{j=1}^Q$ with unlabeled Q query samples also from these N classes to be predicted, denoted as $\mathcal{T} = \mathcal{S} \cup \mathcal{Q}$. The values of N and K are both very small for few-shot learning.

A popular and effective way is to apply the meta-learning framework to exploit information in the training set and improve generalizability. Specifically, meta-learning methods separate the training set \mathcal{C}_{train} into various few-shot training tasks $\mathcal{C}_{train} = \{\mathcal{T}_{train}^l\}_{l=1}^L$ to mimic the test setting, and apply episodic training [2] to learn model parameters from a large number of simulated meta-tasks by minimizing the classification error over the query set \mathcal{Q} of L meta-tasks on the training set \mathcal{C}_{train} as

$$\theta^* = \arg \min_{\theta} \sum_{l=1}^L \sum_{j=1}^Q \ell(f_{\theta}(x_j; \mathcal{T}_{train}^l), y_j), \quad (1)$$

where ℓ denotes the cross-entropy loss function and f_{θ} represents the model $f(\cdot)$ with the parameters θ .

B. General GNN Models

GNNs [48], [49], [57] are neural networks for learning with graph-structured data. Similar to the classic CNNs that exploit the local features (e.g., image patch textures, sparsity) for representation, researchers designed GNNs to mimic the behavior of CNNs to handle graph-structured data. In a GNN model, we consider a graph $\mathbf{G} = (V, E)$ with nodes \mathbf{V} and edges \mathbf{E} . Each sample data (e.g., image) is represented as a node in the graph, and GNN mines the neighborhood information of each node

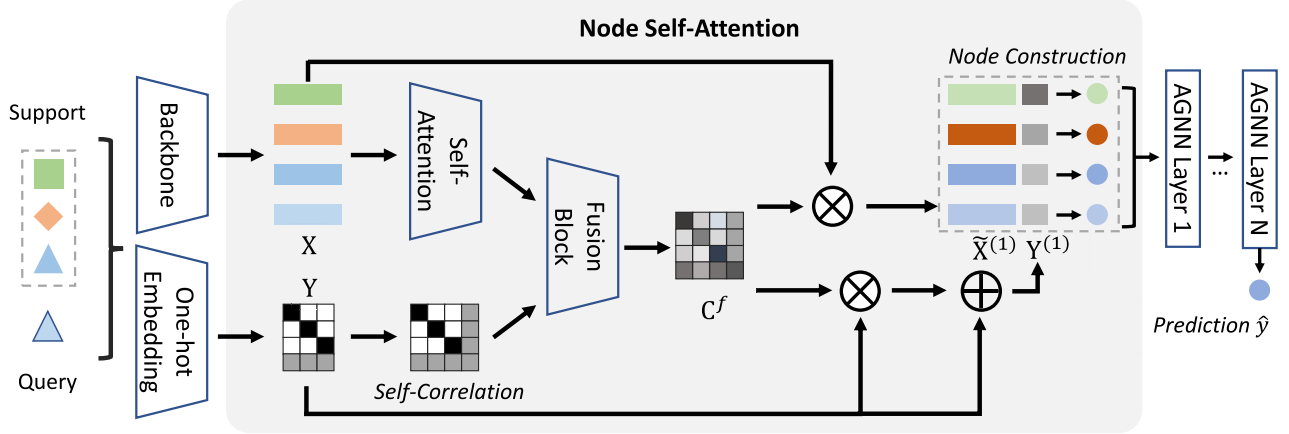


Fig. 2. The overall framework of the proposed Attentive GNN (AGNN) framework for the few-shot learning task. This figure shows an example of a 3-way 1-shot setting with a query sample. For each support and query sample, the color and shape of the sample represent its corresponding class. X and Y denote the feature embedding extracted from the backbone and one-hot label embedding, respectively. The grey box denotes the node self-attention module. Specifically, the node self-attention module first applies self-attention and self-correlation blocks on features and labels to generate attention maps, which are then fed into a fusion block to generate the attention map C^f for graph initialization. AGNN then predicts the query sample after N AGNN layers. Detailed information on each AGNN layer is shown in Fig. 3.

based on the graph structure, which is crucial for building discriminative and generalization features for many tasks, e.g., node classification, graph classification, etc. To be specific, considering a multi-layer GNN model, following the previous work [58], [59], the output of the k -th GNN layer can be represented as:

$$X^{(k+1)} = F_k(X^{(k)}, \mathbf{W}^{(k)}) = \rho(\hat{A}^{(k)} X^{(k)} \mathbf{W}^{(k)}), \quad (2)$$

where $X^{(k)} = \{x_1^{(k)}, x_2^{(k)}, \dots, x_V^{(k)}\} \in \mathbb{R}^{V \times d_k}$ denotes the input feature and $x_i^{(k)}$ denotes the feature of node i in the k -th layer, with V and d_k being the number of nodes and feature dimension at the k -th layer. Besides, $\hat{A}^{(k)} \in \mathbb{R}^{V \times V}$ is called the weighted adjacency matrix, $\mathbf{W}^{(k)} \in \mathbb{R}^{d_k \times d_{k+1}}$ is the trainable linear transformation, and ρ denotes a non-linear function, e.g., ReLU or Leaky-ReLU.

There are different ways to construct the adjacency matrix $A^{(k)}$. For example, in the classic GCN [48], $A_{i,j}^{(k)}$ indicates whether node i and j are directly connected in the graph. Besides, $A_{i,j}^{(k)}$ can be the similarity or distance matrix between nodes i and j in [2], [5], i.e., $A_{i,j}^{(k)} = f_\theta(\phi(x_i^{(k)}), \phi(x_j^{(k)}))$, where ϕ denotes the node feature embedding, and the parameters θ of the distance metric function f can be fixed or learned. One classic example is to apply cosine correlation as the similarity metric, while a more flexible method is to learn a multi-layer perceptron (MLP) as the metric, i.e., $f_\theta(\phi(x_i^{(k)}), \phi(x_j^{(k)})) = \text{MLP}(|\phi(x_i^{(k)}) - \phi(x_j^{(k)})|)$, where $|\cdot|$ denotes the element-wise absolute function. More recent works applied the Gaussian similarity function to construct the adjacency matrix, e.g., TPN [12] proposed the similarity function as $A_{i,j} = \exp(-0.5d(\phi(x_i)/\sigma_i, \phi(x_j)/\sigma_j))$, with σ being an example-wise length-scale parameter learned by a relation network of nodes used for normalization.

IV. ATTENTIVE GRAPH NEURAL NETWORKS

Based on the GNN model, we propose an AGNN model containing triple attentive mechanisms: node self-attention, neighbor attention, and layer memory attention. Fig. 2 shows the pipeline of AGNN for few-shot learning, and Fig. 3 illustrates the details of one AGNN layer. We discuss each attention mechanism, followed by how AGNN is applied for few-shot learning.

A. Node Self-Attention

Denote the feature of each sample (i.e., node) i as $\mathbf{x}_i \in \mathbb{R}^d$, and the one-hot vector of its corresponding label as $\mathbf{y}_i \in \mathbb{R}^N$, where d is the feature dimension, N is the total number of classes and $1 \leq i \leq V$. The one-hot vector sets only the element corresponding to the ground-truth category to be 1, while the others are all set to 0. Note that the one-hot encoding of the query sample is initialized with the uniform distribution (i.e., all values in the vector are set to $1/N$). To obtain a task-specific feature as a suitable graph initialization, we propose **node self-attention** to exploit the sample correlation in the initial stage at the feature and category levels, respectively. Denote the sample matrices and label matrices as:

$$\begin{aligned} X &= [\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_V]^T \in \mathbb{R}^{V \times d}, \\ Y &= [\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_V]^T \in \mathbb{R}^{V \times N}. \end{aligned} \quad (3)$$

We first consider the self-attention between feature embeddings of nodes in a graph. Inspired by the popular and powerful Transformer architecture [50], we employ two linear projection layers with mapping function $W_Q, W_K \in \mathbb{R}^{d \times d_l}$ and compute the self-attention matrix as:

$$C^x = \text{softmax}((XW_Q)(XW_K)^T), \quad (4)$$

where d_l is the feature dimension of the latent space. For simplicity, we set $d_l = d$ in this paper. $\text{softmax}(\cdot)$ denotes a row-wise softmax operator for label correlation matrices. For

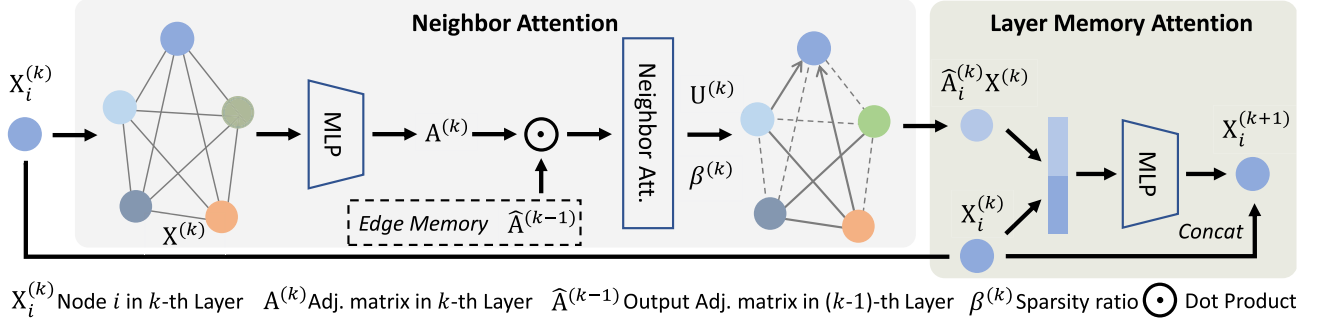


Fig. 3. Illustration of k -th AGNN layer. The grey and green boxes denote neighbor attention and layer memory attention, respectively. Specifically, in the k -th layer, the proposed AGNN method first adapts an MLP block to encode the node similarity and generate the adjacency (Adj.) matrix $A^{(k)}$. Then before applying sparsity constraint to construct the optimal Adj. matrix $\hat{A}^{(k)}$, AGNN reweights $A^{(k)}$ as $U^{(k)}$ by considering the “earlier edge memory” of the last GNN layer, i.e., the optimal Adj. matrix $\hat{A}^{(k-1)}$. Then neighbor attention applies sparsity constraints with a variable sparsity ratio $\beta^{(k)}$ to attend to the most related nodes and generate the output Adj. matrix $\hat{A}^{(k)}$. AGNN then applies the layer memory attention module to update the node features of the graph.

$\mathbf{P} = (\mathbf{XW}_Q)(\mathbf{XW}_K)^T$, the row-wise softmax operator is defined as:

$$\mathbf{C}^x(i, j) = \exp\{\mathbf{P}(i, j)\} / \sum_{k \in \mathcal{N}_i} \exp\{\mathbf{P}(i, k)\}, \quad (5)$$

where \mathcal{N}_i denotes the set of nodes that are connected to the node x_i . We then calculate the label correlation matrices as:

$$\mathbf{C}^y = \text{softmax}(\mathbf{Y}\mathbf{Y}^T). \quad (6)$$

Note that the element (i, j) of the self-correlation function $\mathbf{Y}\mathbf{Y}^T$ for one-hot label matrix \mathbf{Y} indicates if sample i and j are in the same class, which means that the matrix $\mathbf{Y}\mathbf{Y}^T$ is a binary matrix. However, refer to the study in [60], a binary correlation vector for each sample (i.e., each row of the matrix $\mathbf{Y}\mathbf{Y}^T$) only contains information about the correct class but no information about other classes. To solve this problem and achieve a good initialization of the graph, we enlarge the correlation weight of the inter-class samples and reduce the correlation weight of the intra-class samples by introducing the softmax function. In detail, for the matrix $\mathbf{Y}\mathbf{Y}^T$, if a row has a common class with many other samples (corresponding to the K -shot with $K > 1$), the softmax will result in small equal weights. In this way, each sample combines the information from all neighbors.

The proposed node self-attention module exploits the correlation amongst both image features and label vectors, which should share the information from different perspectives for the same node. The next step is to fuse \mathbf{C}^x and \mathbf{C}^y using trainable 1×1 kernels as:

$$\mathbf{C}^f = f_\tau([\mathbf{C}^x, \mathbf{C}^y]) \in \mathbb{R}^{V \times V}, \quad (7)$$

where $[\mathbf{C}^x, \mathbf{C}^y]$ denotes the concatenated attention map, and f_τ is a 1×1 convolution layer. With the fused self-attention map, both the feature and the label vectors are updated on the nodes:

$$\tilde{\mathbf{X}}^{(1)} = \mathbf{C}^f \mathbf{X}, \quad \mathbf{Y}^{(1)} = \alpha \mathbf{Y} + (1 - \alpha) \mathbf{C}^f \mathbf{Y}, \quad (8)$$

where $\alpha \in [0, 1]$ is a weighting parameter. Unlike the feature update, the label update preserves the initial labels, which are the ground truth, in the support set, using the weighting parameter α to regularize the label update. The updated sample features

$\tilde{\mathbf{X}}^{(1)}$ and labels $\mathbf{Y}^{(1)}$ are concatenated to form the node features $\mathbf{X}^{(1)} \in \mathbb{R}^{V \times (d+N)}$ in the first AGNN layer.

B. Neighbor Attention Via Sparsity

Similar to various successful GNN frameworks, the proposed AGNN applies an MLP to learn the adjacency matrix A_{ij} for feature updates. When the GNN model becomes deeper, the risk of over-smoothing increases as GNN tends to mix information from all neighbor nodes and eventually converge to a stationary point in training. To tackle this challenge, we propose a novel **neighbor attention** via two strategies, i.e., sparsity constraint and memory attention, to attend to the most related nodes as illustrated in Fig. 3.

To exploit the neighbor information of the graph across all AGNN layers, we consider the relationship between the two nodes in both current and previous layers when computing the adjacency matrix in each layer. Specifically, when calculating the weight $U^{(k)}(i, j)$ between two nodes $(\mathbf{x}_i^{(k)}, \mathbf{x}_j^{(k)})$ in the k -th layer using an $\text{MLP}^{(k)}$, we also consider the relationship between them in the previous $(k-1)$ -th layer as:

$$U^{(k)}(i, j) = \text{MLP}^{(k)}\left(\left|\mathbf{x}_i^{(k)} - \mathbf{x}_j^{(k)}\right|\right) \odot \hat{A}^{(k-1)}(i, j), \forall i, j, \quad (9)$$

where $\hat{A}^{(k-1)}$ is the output adjacency matrix in the last layer and $\text{MLP}^{(k)}(\cdot)$ contains two convolutional blocks with a sigmoid layer at last.

Note that each element $\hat{A}^{(k-1)}(i, j)$ in $\hat{A}^{(k-1)}$ is between $[0, 1]$, indicating the similarity between node i and node j in the previous layer. Hence the (9) can be regarded as a regularized MLP function that reweights the similarity between two nodes according to the “edge memory” obtained from the previous layer.

With $U^{(k)}(i, j)$, we then apply sparsity constraint to attend to the most related nodes by solving the following sparse problem:

$$\hat{A}^{(k)} = \arg \min_{A^{(k)}} \left\| A^{(k)} - U^{(k)} \right\|_F, \quad s.t. \left\| A_i^{(k)} \right\|_0 \leq \beta^{(k)} V, \forall i, \quad (10)$$

where $A_i^{(k)} \in \mathbb{R}^{1 \times V}$ denotes the i -th row of $A^{(k)}$, $\beta^{(k)} \in (0, 1]$ denotes the ratio of nodes maintained for feature update in the k -th AGNN layer. With the ℓ_0 constraint, the adjacency matrix $\hat{A}^{(k)}$ has up to $\beta^{(k)}V$ non-zeros in each row, corresponding to the **attended neighbor nodes**.

Different from setting a fixed sparsity ratio for all AGNN layers, we change the sparsity ratio in each layer following $\beta^{(k)} = 1 - 0.1k$. Such a dynamic layer-wise version brings more flexibility to each AGNN layer. Specifically, we do not impose strong sparsity constraints in the first few layers to aggregate more neighbor information. As AGNN layers go deeper, the stronger sparse constraint with a lower maintenance ratio forces the model to capture information only from the most relevant nodes, i.e., nodes with a higher probability of belonging to the same class.

The solution to (10) is achieved using the projection onto an ℓ_0 unit ball, i.e., keeping the $\beta^{(k)}V$ elements of each $U_i^{(k)}$ with the largest magnitudes [61]. Since the solution to (10) is non-differentiable, we apply alternating projection for training, i.e., in each epoch, $U^{(k)}$ is first updated using back-propagation by (9), followed by (10) to update $\hat{A}^{(k)}$ which is constrained to be sparse. For simplicity, we keep the top- m value for each row of $A^{(k)}$ and set the others to 0 to construct the sparse matrix with $m = \beta^{(k)}V$.

C. Layer Memory Attention

To avoid the over-smoothing and over-fitting issues due to “over-mixing” neighboring nodes’ information, another approach is to attend to the “earlier memory” of intermediate features, including both edge and node features at previous layers. Inspired by DenseNet [24], JKNet [62], GFCN [63] and few-shot GNN [5], we densely connect the output of each GNN layer, as the intermediate GNN-node features maintain more consistent and general representation across different GNN layers. During the adjacency matrix (i.e., edge feature) update process, AGNN treats the output adjacency matrix of the previous layer as “edge memory” and adopts it according to (9) to balance the relationship among neighbors in each AGNN layer before adopting the sparsity constraint. Then, AGNN applies the transition function based on (2) to update node features. In addition, we utilize graph self-loop, i.e., identity matrix \mathbf{I} to incorporate self information. Thus the update rule of the AGNN in the k -th layer is formulated as:

$$F_k(X^{(k)}, \mathbf{W}^{(k)}) = \rho \left(\left[\hat{A}^{(k)} X^{(k)} \parallel \mathbf{I} X^{(k)} \right] \mathbf{W}^{(k)} \right), \quad (11)$$

where \parallel means row-wise feature concatenation and $\mathbf{W}^{(k)} \in \mathbb{R}^{2d_k \times m}$. Furthermore, instead of using $F_k(X^{(k)}, \mathbf{W}^{(k)}) \in \mathbb{R}^{V \times m}$ directly as the input node feature at the $(k+1)$ -th layer, we propose to attend to the “early memory” in a similar way as [5] by concatenating the node feature at the k -th layer as:

$$X^{(k+1)} = \left[X^{(k)}, F_k \left(X^{(k)}, \mathbf{W}^{(k)} \right) \right] \in \mathbb{R}^{V \times (d + N + km)}. \quad (12)$$

Equation (12) shows that the output feature size of k -th layer $X^{(k+1)}$, the size of MLP block for computation of the adjacency matrix in (10) and the corresponded transform matrix $\mathbf{W}^{(k)}$ are

all positively correlated with the number of layers. Thus, as the number of AGNN layers k increases, it needs more memory to store the features and parameters for each GNN layer. There are only $V \times m$ new features introduced in a new layer, while the node features of earlier layer $X^{(k)}$ are attended to the early memory.

D. AGNN for Few-Shot Learning

Following the same strategy of episodic training [2] with the meta-learning framework, we simulate N -way K -shot tasks $\mathcal{C}_{train} = \{\mathcal{T}_{train}^l\}_{l=1}^L$ which are randomly sampled from the training set, in which the support set includes K labeled samples (e.g., images) from each of the N classes and the query set includes unlabeled samples from the same N classes. Each task is modeled as a graph [5], in which each node represents an image sample with its label. The objective is to learn the parameters of the AGNN model using the simulated tasks, which are generalizable for an unseen few-shot task.

Loss Function: We adopt the single-stage training scheme without pre-training the feature extractor and jointly train the AGNN model combined with the backbone network. For each simulated few-shot task \mathcal{T}_{train}^l with its query set $\mathcal{Q} = \{(\mathbf{x}_i, y_i)\}_{i=1}^Q$, the parameters of the backbone feature extractor, node self-attention block f_τ , and M AGNN layers $\{\text{MLP}_i^{(k)}, \mathbf{W}_i^{(k)}\}_{i=1}^M$ are trained by minimizing the summation of the cross-entropy loss of classes over all query samples from each layer as:

$$\mathcal{L}_{cls} = - \sum_{l=1}^M \sum_{i=1}^Q y_i \log P(\hat{y}_i^l = y_i | \mathcal{T}_{train}^l), \quad (13)$$

where \hat{y}_i^l denotes the predicted labels of the query sample \mathbf{x}_i in the l -th AGNN layer and y_i is the corresponding ground-truth labels. We evaluate the proposed AGNN for the few-shot task using both **inductive** and **transductive** settings, which correspond to $Q = 1$, and $Q = Nq$ with $q \geq 1$, respectively. For each query sample in the N -way K -shot task, we initialize the one-hot feature y with a uniform distribution, i.e., each value is set to $1/N$.

E. Relation to Existing Graph-Based Models

GAT [54]: Different from the classic GNNs, GAT [54] exploited attention mechanism amongst all neighbor nodes in the feature domain after the linear transformation $\mathbf{W}^{(k)}$ and computes the weights α based on attention coefficients for graph update as:

$$x_i^{(k+1)} = \rho \left(\sum_{j \in \mathcal{N}_i} \alpha_{ij} x_j^{(k)} \mathbf{W}^{(k)} \right), \quad (14)$$

where \mathcal{N}_i denotes the set of the neighbor (i.e., connected) nodes of x_i . Similar to our proposed AGNN model, GAT also considers self-attention on the nodes. However, unlike our proposed method, which applies the node self-attention mechanism before the GNN layer, GAT applies a self-attention mechanism after the linear transformation \mathbf{W} . With a shared attention mechanism parametrized by a weight vector $\vec{\alpha}$, GAT allows all neighbor

nodes to attend to the target node with attention coefficients as:

$$\alpha_{ij} = \frac{\exp\left(\rho\left(\vec{\mathbf{a}}^T\left[x_i^{(k)}\mathbf{W}^{(k)}\|x_j^{(k)}\mathbf{W}^{(k)}\right]\right)\right)}{\sum_{p\in\mathcal{N}_i}\exp\left(\rho\left(\vec{\mathbf{a}}^T\left[x_i^{(k)}\mathbf{W}^{(k)}\|x_p^{(k)}\mathbf{W}^{(k)}\right]\right)\right)}. \quad (15)$$

However, GAT only considers the relationship among neighbors in the same layer while it fails to utilize the layer-wise information, which may lead to over-smoothing. Furthermore, GAT just applies self-attention based on node features while ignoring label information.

TPN [12]: TPN [12] first introduced a transductive mechanism to utilize the entire query set for transductive inference in few-shot learning. A graph construction module is proposed to exploit the manifold structure of the novel class space using the union of support set and query set. Specifically, TPN applies a relation module to learn the example-wise length-scale parameter σ , which is then used to compute the node similarity matrix A as:

$$A_{ij} = \exp\left(-\frac{1}{2}d\left(\frac{x_i}{\sigma_i}, \frac{x_j}{\sigma_j}\right)\right), \quad (16)$$

where $d(\cdot, \cdot)$ is the Euclidean distance function.

Similar to the proposed neighbor attention mechanism, TPN only keeps top- k values in each row of A to construct a sparse k -nearest neighbor graph. With this graph, instead of iterative label propagation, TPN applies a closed-form solution to propagate the labels from the support set to the query set:

$$F^* = (I - \alpha S)^{-1}Y, \quad (17)$$

where $S = D^{-1/2}AD^{-1/2}$ is the normalized symmetric matrix in which D is a diagonal matrix with its (i, i) -element equal to the sum of the i -th row of affinity matrix A . TPN can predict the class of query samples by regressing directly from support features to query features in closed form without large-scale learnable parameters. However, the graph structure is fixed upon the computation of the sparse similarity matrix A for each iteration. Moreover, TPN does not consider the label information for graph construction and the relationship among different layers, which limits its performance.

DPGN [14]: Unlike the single graph-based methods mentioned above, DPGN [14] proposes a dual-graph architecture including a point graph and distribution graph to leverage both instance-level and distribution-level representation to propagate label information better. The point graph contains node features of images and follows the same steps in (2) to update graphs, identical to our proposed AGNN model and other GNN-based methods. However, the strategy of updating the adjacency matrix in DPGN is different from our proposed method. Our proposed AGNN learns a sparse adjacency matrix $A^{(k)}$ with an MLP in (10) by considering both the node feature information of the current layer and the adjacency information of the previous layer, while DPGN constructs a dual distribution graph by gathering 1-vs- n relation on each node to refine the point graph by delivering distribution relations between each pair of samples.

V. WHY IT WORKS

A. Discriminative Sample Representation

It is critical to obtain the initial feature representation of the samples that are sufficiently discriminative (i.e., samples of different classes are separated) for the GNN models in few-shot tasks. However, most of the existing GNN models work with generic features using a CNN-based backbone and fail to capture the task-specific structure. The proposed node self-attention module exploits the cross-sample correlation and can thus effectively guide the feature representation for each few-shot task.

B. Alleviation of Over-Smoothing and Over-Fitting Problems

Over-fitting arises when learning an over-parametric model from the limited training data, and it is extremely severe as the objective of few-shot learning is to generalize the knowledge from the training set for few-shot tasks. On the other hand, *over-smoothing* phenomenon refers to the case where the features of all (connected) nodes converge to similar values as the model depth increases. We provide theoretical analysis to show that the proposed triple-attention mechanism can alleviate both over-fitting and over-smoothing in GNN training.

Lemma 1: The node self-attention module is equivalent to a GNN layer if $\alpha = 0$ as

$$X^{(k)} = [X, Y], \quad A^{(k)} = C^f, \quad \mathbf{W}^{(k)} = \mathbf{I}. \quad (18)$$

Proposition 1: Applying the node self-attention module to replace a GNN layer in AGNN reduces the trainable-parameter complexity from $\mathcal{O}\{d_x(d_x + L)\}$ to $\mathcal{O}\{d_x d_e\}$, where d_x and d_e represents the input feature dimension and the projected dimension in the latent space, respectively. L denotes the depth of MLP for generating the adjacency metric.

The node self-attention module only involves two linear projection layers and the 1×1 kernels that are trainable.

Lemma 1 and Proposition 1 prove that the node self-attention module involves much fewer trainable parameters than a normal GNN layer. Thus, applying node self-attention instead of another GNN layer will reduce the model complexity, thus lowering the risk of over-fitting.

Next, we show that using neighbor attention can help alleviate over-smoothing for training GNN models. The analysis is based on the recent works on DropEdge [21] and GNN information loss [64]. They proved that a sufficiently deep GNN model will always suffer from “ ϵ -smoothing” [64], where ϵ is defined as the error bound of the maximum distance among node features. Another concept is the “information loss” [64] of a graph model \mathbf{G} , i.e., the dimensionality reduction of the node feature-space after T layers of GNNs, denoted as $\Theta_{T, \mathbf{G}}$. We use these two concepts to quantify the over-smoothing issue in our analysis.

Theorem 1: Denote the same multi-layer GNN model with and without neighbor attention as $\tilde{\mathbf{G}}$ and \mathbf{G} , respectively. Besides, denote the number of GNN layers for them to encounter the ϵ -smoothing [64] as $T(\tilde{\mathbf{G}}, \epsilon)$ and $T(\mathbf{G}, \epsilon)$, respectively. With sufficiently small β in the neighbor attention module, either (i) $T(\tilde{\mathbf{G}}, \epsilon) \leq T(\mathbf{G}, \epsilon)$, or (ii) $\Theta_{T(\mathbf{G}, \epsilon), \mathbf{G}} > \Theta_{T(\tilde{\mathbf{G}}, \epsilon), \tilde{\mathbf{G}}}$, will hold.

Remarks: The result shows that the GNN model with the neighbor attention (i) increases the maximum number of layers to encounter over-smoothing, or if the number of layers remains, (ii) the over-smoothing phenomenon is alleviated.

For the above results, we provide the full proofs in Section V-C.

C. Proofs of the Proposed Attention Mechanism

We present the detailed proofs of Lemma 1, Proposition 1 regarding the proposed node self-attention, and Theorem 1 regarding the neighbor attention.

Proof of Lemma 1: First of all, we analyze the proposed node self-attention, whose feature and label vector updates are

$$\tilde{\mathbf{X}}^{(1)} = \mathbf{C}^f \mathbf{X}, \quad \mathbf{Y}^{(1)} = \alpha \mathbf{Y} + (1 - \alpha) \mathbf{C}^f \mathbf{Y}, \quad (19)$$

where \mathbf{C}^f denotes the attention map, \mathbf{X} and \mathbf{Y} (resp. $\mathbf{X}^{(1)}$ and $\mathbf{Y}^{(1)}$) denote the input (resp. output) feature and label vectors, respectively.

We prove Lemma 1, which shows that the proposed node self-attention can alleviate *Over-fitting* by reducing the model complexity compared to adding more GNN layers. The output of the general k -th GNN layer can be represented as

$$\mathbf{X}^{(k+1)} = \mathbf{F}_k(\mathbf{X}^{(k)}, \mathbf{W}^{(k)}) = \rho(\hat{\mathbf{A}}^{(k)} \mathbf{X}^{(k)} \mathbf{W}^{(k)}). \quad (20)$$

With the condition for equivalence, the output of the k -th GNN layer becomes

$$\mathbf{X}^{(k+1)} = \mathbf{F}_k(\mathbf{X}^{(k)}, \mathbf{I}) = \mathbf{C}^f \mathbf{X}^{(k)} \mathbf{I} = \mathbf{C}^f \mathbf{X}^{(k)}. \quad (21)$$

Thus, (21) is equivalent to putting the node self-attention to replace the k -th GNN layer, with $\mathbf{X}^{(k+1)} = \mathbf{X}^{(1)}$ and $\mathbf{X}^{(k)} = [\mathbf{X}, \mathbf{Y}]$. ■

Proof of Proposition 1: Next, we prove Proposition 1, which shows the model complexity decrease from a trainable GNN layer to the proposed node self-attention module.

For a GNN layer following (2), both $\mathbf{W}^{(k)}$ and the $\text{MLP}^{(k)}$ are trainable, corresponding to free parameters scale as $\mathcal{O}\{d_x^2\}$ and $\mathcal{O}\{d_x L\}$, respectively. On the contrary, based on Lemma 1, the proposed node self-attention is equivalent to a GNN layer, with the $\mathbf{W}^{(k)}$ and the $\text{MLP}^{(k)}$ fixed. The only trainable parameters are linear layers to project features into the latent space for attention computation and the 1×1 kernels to fuse the $\mathbf{C}^{\mathbf{X}}$ and $\mathbf{C}^{\mathbf{Y}}$, with the complexity scales as $\mathcal{O}\{d_x d_e\}$ and $\mathcal{O}\{1\}$, respectively. ■

Proofs of Theorem 1: Next, we show that using neighbor attention can help alleviate over-smoothing for training GNN. We first quantify the degree of over-smoothing using the definitions from [21] and [64].

Definition 1 (Feature Subspace): Denote the M -dimensional subspace $\mathcal{M} = \{\mathbf{U}\Sigma \mid \mathbf{U} \in \mathbb{R}^{V \times M}, \mathbf{U}^T \mathbf{U} = \mathbf{I}_M, \Sigma \in \mathbb{R}^{M \times d}\}$ as the feature space, with $M \leq V$.

Definition 2 (Projection Loss): Denote the operator of projection $\mathbf{X} \in \mathbb{R}^{V \times d_x}$ onto a M -dimensional subspace as $\mathbb{P}_{\mathcal{M}}: \mathbb{R}^{V \times d_x} \rightarrow \mathbb{R}^{V \times d_x}$ as

$$\mathbb{P}_{\mathcal{M}}(\mathbf{X}) = \arg \min_{\mathbf{Z} \in \mathcal{M}} \|\mathbf{X} - \mathbf{Z}\|_F. \quad (22)$$

Denote the projection loss $\theta_{\mathcal{M}}(\mathbf{X})$ as

$$\theta_{\mathcal{M}}(\mathbf{X}) = \|\mathbf{X} - \mathbb{P}_{\mathcal{M}}(\mathbf{X})\|_F = \min_{\mathbf{Z} \in \mathcal{M}} \|\mathbf{X} - \mathbf{Z}\|_F. \quad (23)$$

Definition 3 (ϵ -smoothing): The GNN layer that suffers from ϵ -smoothing if $\theta_{\mathcal{M}}(\mathbf{X}) < \epsilon$. Given a multi-layer GNN \mathbf{G} with each the feature output of each layer as $\mathbf{X}^{(k)}$, we define the ϵ -smoothing layer as the minimal value k that encounters ϵ -smoothing, i.e.,

$$T(\mathbf{G}, \epsilon) = \min_k \{\theta_{\mathcal{M}}(\mathbf{X}) < \epsilon\}. \quad (24)$$

Definition 4 (Dimensionality Reduction): Suppose the dimensionality reduction of the node feature-space after T layers of GNNs is denoted as $\Theta_{T, \mathbf{G}} = d_x - T(\mathbf{G}, \epsilon)$.

With these definitions from [21] and [64], we can now prove Theorem 1 for the **neighbor attention** in (9) and (10).

Given the original $\mathbf{U}^{(k)}$, the solution to (10) is achieved using the projection onto a ℓ_0 unit ball, i.e., keeping the βV elements of each $\mathbf{U}_i^{(k)}$ with the largest magnitudes [61], i.e.,

$$\hat{\mathbf{A}}_i^{(k)}(j) = \begin{cases} \mathbf{U}_i^{(k)}(j) & , j \in \Omega_{\beta^{(k)}V}^i \\ 0 & , j \in \bar{\Omega}_{\beta^{(k)}V}^i \end{cases} \quad (25)$$

Here, the set $\Omega_{\beta^{(k)}V}^i = \text{supp}(\hat{\mathbf{A}}_i^{(k)})$ indexes the top- $\beta^{(k)}V$ elements of largest magnitude in $\mathbf{U}_i^{(k)}$, and $\bar{\Omega}_{\beta^{(k)}V}^i$ denotes the complement set of $\Omega_{\beta^{(k)}V}^i$. When $\hat{\mathbf{A}}_i^{(k)}(j) = 0$, it is equivalent to remove the edge connecting the i -th node and j -th node. Thus, $|\bar{\Omega}_{\beta^{(k)}V}^i|$ equals the number of edges been dropped by the neighbor attention, and $|\bar{\Omega}_{\beta^{(k)}V}^i| \rightarrow V$ as $\beta^{(k)} \rightarrow 0$.

Therefore, when $\beta^{(k)}$ is sufficiently small, there are a sufficient number of edges being dropped by the neighbor attention. Based on the **Theorem 1** in [21], we have either of the two to alleviate the over-smoothing phenomenon:

- The number of layers without ϵ -smoothing increases by neighbor attention via sparsity, i.e., $T(\tilde{\mathbf{G}}, \epsilon) \leq T(\mathbf{G}, \epsilon)$.
- The information loss (i.e., dimensionality reduction by feature embedding) decreases by neighbor attention via sparsity, i.e., $\Theta_{T(\mathbf{G}, \epsilon), \mathbf{G}} > \Theta_{T(\tilde{\mathbf{G}}, \epsilon), \tilde{\mathbf{G}}}$. ■

VI. EXPERIMENTS

To evaluate the performance of our proposed AGNN method for few-shot classification, we conducted various experiments on four benchmarks. In this section, we first describe the dataset information and implementation details of our network. Then we conduct extensive experiments under several extended few-shot classification tasks, including standard classification, fine-grained classification, and semi-supervised classification under both transductive and inductive settings to evaluate the generalizability of the AGNN model. Finally, we perform ablation studies to analyze the effectiveness of each attention mechanism.

A. Dataset Description

We conduct few-shot image classification experiments on four benchmarks, including Mini-ImageNet [2], Tiered-ImageNet [65], Flowers-102 [36], [66], and Caltech-UCSD Birds 200-2011 [67].

Mini-ImageNet: Mini-ImageNet contains 60000 images of 100 different classes extracted from the ILSVRC-12 challenge [68]. We follow the dataset splits proposed by [31], i.e., 64, 16, and 20 classes for training, validation, and testing, respectively.

Tiered-ImageNet: Tiered-ImageNet dataset is a more challenging data subset also from the ILSVRC-12 challenge [68], which contains more classes that are organized in a hierarchical structure, i.e., 608 classes from 34 top categories. We follow the setups proposed by [65] and split 34 top categories (resp. 608 classes) into 20 (resp. 351), 6 (resp. 97), and 8 (resp. 160) classes for training, validation, and testing, respectively.

Flowers-102: Flowers-102 [66] was initially proposed for fine-grained image classification of flowers. It contains 102 different flowers with 8189 images, and each image size is $84 \times 84 \times 3$. There are large variations of scale, pose, and light of flower images. In addition, some categories have significant variations within classes. Following the split in [36], we split 102 classes into 52, 25, and 25 for training, validation, and testing, respectively.

Caltech-UCSD Birds 200-2011: CUB [67] was also initially proposed for fine-grained image classification. It contains 200 different birds with 11788 images, and each image size is $84 \times 84 \times 3$. Compared with the classic image classification task, we need to find the minor difference between classes, making it a more challenging problem. Following the split in [69], we split 200 classes into 100, 50, and 50 for training, validation, and testing, respectively.

B. Implementation Details

We follow most of the DNN-based few-shot learning schemes [2], [3], [5] and first apply the popular ConvNet-4 as the backbone feature extractor, with 3×3 convolution kernels, numbers of channels as [64,96,128,256] at corresponding layers, a batch normalization layer, a max pooling layer, and a LeakyReLU activation layer. Besides, two dropout layers are adapted to the last two convolution blocks to alleviate over-fitting [5]. Furthermore, to compare with the more complicated CNN-based methods, we also apply ResNet-12 as the backbone, following a similar setup in [38]. On this basis, a fully-connected layer with batch normalization is added to the end for dimensionality reduction. We conducted both 5-way 1-shot and 5-way 5-shot experiments, under both inductive and transductive settings [12]. We use only one query sample for the inductive and one query sample per class for the transductive experiments on the ConvNet-4 and ResNet-12 backbone. Our models are all trained using Adam optimizer with an initial learning rate of 1×10^{-3} . For the ConvNet-4 backbone, the weight decay is set to 10^{-6} and the mini-batch sizes are set to 40 for all settings. For the ResNet-12 backbone, the weight decay is

TABLE I

FEW-SHOT CLASSIFICATION ACCURACY AVERAGED OVER MINI-IMAGE NET AND TIERED-IMAGE NET DATASETS WITH THE CONVNET-4 BACKBONE. UNDER EACH SETTING (I.E., TRANSDUCTIVE OR INDUCTIVE, 1-SHOT OR 5-SHOT), THE BEST AND SECOND BEST RESULTS UNDER EACH DATASET ARE HIGHLIGHTED AS RED AND BLUE, RESPECTIVELY. [†] INDICATES THAT THE SETFEAT METHOD ADOPTS THE CONVNET4 BACKBONE WITH ADDITIONAL 10 SELF-ATTENTION MODULES. [‡] DENOTES THAT THE GNN RESULT IS OUR IMPLEMENTATION BASED ON PUBLIC CODE. [‡] SHOWS THAT MCGN HAS A DIFFERENT TRANSDUCTIVE SETTING FROM OTHER GNN BASED METHODS

Method	Trans	mini-ImageNet		tiered-ImageNet	
		1-shot	5-shot	1-shot	5-shot
<i>Non-GNN based methods</i>					
Matching-Net [2]	✗	46.60	55.30	-	-
ProtoNet [3]	✗	49.42	68.20	48.58	69.57
Relation-Net [4]	✗	50.44	65.32	-	-
FEAT [11]	✗	55.15	71.61	-	-
IEPT [70]	✗	56.26	73.91	58.25	75.63
MCL-Katz [71]	✗	57.88	74.03	57.63	73.96
SetFeat [72] [†]	✗	59.10	74.97	-	-
<i>GNN based methods</i>					
GNN [5]	✗	50.33	66.41	54.97	70.92
EGNN [13]	✗	52.86	66.85	-	70.98
Ours	✗	63.49	76.62	64.60	82.43
<i>Non-GNN based methods</i>					
MAML [29]	✓	48.70	63.11	51.67	70.30
MAML [29]	✓	50.83	66.19	53.23	70.83
Relation-Net [4]	✓	51.38	67.07	54.48	71.31
FEAT [11]	✓	57.04	72.89	-	-
ECKPN [73]	✓	68.89	83.59	70.45	86.74
<i>GNN based methods</i>					
TPN [12]	✓	51.94	67.55	59.91	73.30
TPN [‡] [12]	✓	53.75	69.43	57.53	72.85
GNN [‡] [5]	✓	54.14	70.38	65.11	76.40
EGNN [13]	✓	59.63	76.34	63.52	80.24
DPGN [14]	✓	66.01	82.83	69.43	85.92
MCGN [‡] [15]	✓	67.32	83.03	71.21	85.98
Ours	✓	76.99	85.20	74.54	87.99

10^{-5} , and the mini-batch sizes are set to 28. We reduce the learning rate to 0.1 every 15 K and 18 K epochs over mini-ImageNet and tiered-ImageNet, respectively. The output feature dimension of two backbones is 128, and the number of GNN layers is set to 5. The weighting parameter α in Eq (8) is set to 0.5 and 0.9 for mini-ImageNet and tiered-ImageNet, respectively.

C. Standard Few-Shot Image Classification

We compare the proposed AGNN to state-of-the-art CNN- and GNN-based methods, using the ConvNet-4 and ResNet-12 backbone, and Tables I, II list the average accuracy of the few-shot image classification, respectively. Table I shows that the proposed AGNN has achieved state-of-the-art performances under 5-way 1-shot and 5-shot settings and outperforms GNN-based methods by about 9.67% and 3.33% over mini-ImageNet and tiered-ImageNet datasets under the ConvNet-4 backbone under the 1-shot setting. When adopting a deeper backbone network (i.e., ResNet-12), we can observe a consistent result, which demonstrates the effectiveness of the AGNN approach. Furthermore, when comparing the results of the two backbones on the same tiered-ImageNet dataset, we can find that the performance improvement of the GNN-based methods is not obvious.

TABLE II
FEW-SHOT CLASSIFICATION ACCURACY AVERAGED OVER TIERED-IMAGENET WITH THE RESNET-BASED BACKBONE. THE BEST (RESP. SECOND BEST) RESULTS ARE HIGHLIGHTED AS RED (RESP. BLUE)

Method	Backbone	tiered-ImageNet	
		5-way 1-shot	5-way 5-shot
<i>Non-GNN based methods</i>			
ProtoNet [3]	ResNet-12	65.65 ± 0.92	83.40 ± 0.65
MetaOptNet [30]	ResNet-12	65.99 ± 0.72	81.56 ± 0.53
CTM [37]	ResNet-18	68.41 ± 0.39	84.28 ± 1.73
Meta-Baseline [74]	ResNet-12	68.62 ± 0.27	83.29 ± 0.18
CAN [44]	ResNet-12	69.89 ± 0.51	84.23 ± 0.37
FEAT [11]	ResNet-12	70.80 ± 0.23	84.79 ± 0.16
DeepEMD [6]	ResNet-12	71.16 ± 0.87	86.03 ± 0.58
CAN+Trans [44]	ResNet-12	73.21 ± 0.58	84.93 ± 0.38
ECKPN [73]	ResNet-12	73.59 ± 0.45	88.13 ± 0.28
MCL-Katz [71]	ResNet-12	73.62	86.29
STL DeepBDC [75]	ResNet-12	73.82 ± 0.47	89.00 ± 0.30
<i>GNN based methods</i>			
TPN [12]	ResNet-12	59.91 ± 0.94	73.30 ± 0.75
DPGN [14]	ResNet-12	72.45 ± 0.51	87.24 ± 0.39
Ours	ResNet-12	76.73 ± 0.38	90.00 ± 0.41

TABLE III
FEW-SHOT CLASSIFICATION ACCURACY AVERAGED OVER CUB AND FLOWERS-102 DATASETS WITH THE CONVNET-4 BACKBONE. THE BEST AND SECOND BEST RESULTS UNDER EACH SETTING AND DATASET ARE HIGHLIGHTED AS RED AND BLUE, RESPECTIVELY

Method	CUB		Flowers-102	
	1-shot	5-shot	1-shot	5-shot
<i>Non-GNN based methods</i>				
Meta-Baseline [74]	59.29	78.70	72.97	85.91
ProtoNet [3]	63.72	81.50	64.4	80.2
Matching-Net [2]	67.73	79.00	66.0	82.0
COMET [36]	67.90	85.30	70.4	86.7
VFD [76]	68.42	82.42	-	-
FEAT [11]	68.87	82.90	-	-
Dual ATT-Net [47]	72.89	86.60	-	-
<i>GNN based methods</i>				
DPGN [14]	72.97	83.81	73.27	86.34
GNN [5]	73.72	82.60	75.74	86.74
Ours	75.81	88.22	84.10	90.28

One possible explanation is that GNN methods mainly exploit graph structure to collect and exploit important information from neighboring nodes. However, deeper feature extractors can only help better initialization, which is less relevant to GNN layers. Another observation is that for the same method, the accuracy of transductive learning is typically better than that of inductive learning, by further exploiting the correlation amongst the multiple query samples.

D. Fine-Grained Few-Shot Classification

We also evaluate the proposed AGNN method on two fine-grained datasets (i.e., CUB and Flowers-102) under the few-shot setting. Compared with classification tasks on standard datasets such as the mini-ImageNet dataset, the few-shot fine-grained classification task is more challenging due to the significant intra-class variance and inter-class similarity. Table III

TABLE IV
COMPARISON OF FULLY SUPERVISED AND SEMI-SUPERVISED FEW-SHOT CLASSIFICATION ON THE TIERED-IMAGENET BENCHMARK UNDER THE 5-WAY SETTING. HERE “FS” MEANS THE TYPICAL FULLY SUPERVISED FEW-SHOT SETTING. THE PERCENTAGE UNDER THE SEMI-SUPERVISED SETTING CORRESPONDS TO THE PROPORTION OF LABEL SAMPLES IN EACH CLASS OF SUPPORT SET UNDER THE 5-SHOT SETTING

Method	Fs		Semi-supervised	
	1-shot	20%	80%	5-shot
ProtoNet [3]	53.11	53.11	67.47	71.73
ProtoNet (w/ unlabeled)	53.11	53.96	69.65	71.73
Cluster-FSL [77]	52.70	54.45	66.18	67.38
GNN [5]	65.11	66.08	75.47	76.40
EGNN [13]	63.52	64.61	74.46	80.15
DPGN [14]	69.43	74.21	80.68	85.92
TLRM [7]	61.43	69.50	74.02	77.80
Ours	74.54	77.24	80.28	87.99

summarizes the 5-way classification results with the ConvNet-4 backbone over CUB and Flowers-102 datasets. It can be seen that our proposed AGNN achieves state-of-the-art performance on both datasets for both 1-shot and 5-shot settings. Our proposed AGNN improves GNN by a large margin ranging from 2.09% to 8.36% on both target datasets, which validates the effectiveness of the proposed attention mechanisms. Notably, we observe that GNN-based few-shot methods perform better than other few-shot methods, proving that GNN can help exploit the intra-class and inter-class relationships between samples, especially for the fine-grained classification task.

E. Semi-Supervised Few-Shot Classification

For the semi-supervised experiment, we follow the typical 5-way 5-shot setting with only a partially labeled support set [5], [13]. We conduct the experiments over the tiered-ImageNet benchmark, and the result is presented in Table IV. For each class, we set the same labeled ratio of the support samples, e.g., 20% labeled ratio corresponds to one labeled support sample and four unlabeled support samples. We perform the ProtoNet method [3] as the baseline for comparison and report the results of two versions according to whether the method utilizes the unlabeled support samples. Here the original version of the ProtoNet method means that we ignore unlabeled support samples and only use the partially labeled support samples of each class to compute prototype for classification, which is equivalent to the corresponding few-shot setting, i.e., 20% (resp. 80%) in the semi-supervised setting is equal to the fully supervised 5-way 1-shot (resp. 4-shot) setting. In contrast, we also implement a new version of the ProtoNet denoted as “ProtoNet w/ unlabeled,” i.e., considering these unlabeled support samples as extra query samples during training for semi-supervised learning.

As shown in Table IV, we can observe that semi-supervised learning increases the performance of all methods in comparison to the results under the typical fully-supervised few-shot setting with the same number of labeled support samples (0.85%, 0.97%, 1.09%, 2.70%, 4.78%, 8.07% for ProtoNet, GNN, EGNN, AGNN, DPGN, and TLRM, respectively).

Furthermore, we find that the difference between the ProtoNet baseline and GNN-based methods under the semi-supervised setting becomes larger. This indicates that GNN-based methods can obtain a more discriminative relationship representation via graph structure. The proposed AGNN outperforms the GNN, EGNN, and TLRM in all cases. Furthermore, AGNN achieves better performance than DPGN under the 20% semi-supervised setting, while DPGN performs slightly better under the 80% semi-supervised setting. A plausible explanation is that DPGN applies a dual-graph structure to encode distribution and instance information, which can better propagate label information when there is less unlabeled data (corresponding to the 80% semi-supervised setting) but may convey misinformation between the dual-graph when data missing is severe (corresponding to the 20% semi-supervised setting). Notably, the performance gap between the proposed AGNN and EGNN becomes larger under the semi-supervised setting, which proves the ability of proposed attention mechanisms to integrate more accurate and essential information via graph convolution. Moreover, there exists another popular semi-supervised few-shot setting, i.e., each few-shot task consists of additional unlabeled data for each category. Different from semi-supervised settings in GNN-based methods, methods of this type utilize additional information to help adjust the class distribution to reduce bias. To evaluate the effectiveness of the proposed method, we also compare the results with one of the state-of-the-art method Cluster-FSL [77] in our setting. For a fair comparison, we change the support and unlabeled size for Cluster-FSL to match our setting, e.g., for 80% semi-supervised setting, we set the support size to 4 and the unlabeled size to 1. Compared to Cluster-FSL, GNN-based methods perform better, which demonstrates the superiority of our proposed attention modules.

F. Robustness in Transductive Learning

Comparing to the typical inductive setting in few-shot learning, transductive few-shot learning is a novel setting first proposed in [12], which allows the model to utilize the whole unlabeled query instances in each few-shot task, leading to promising results. While the sampled query samples are always **uniformly** distributed for each class in the conventional transductive learning setting [12], such an assumption may not hold in practice, e.g., the query set contains the **random number** of samples for each class. This problem may be severe, especially for GNN-based models, which may learn the distribution of each class by graph convolution during meta-training. We study how robust the proposed AGNN is for such a setting by comparing it to the baseline GNN [5] only with layer memory attention, DPGN [14] and AGNN without specific attention mechanism. In the training phase, we simulate a fixed number of query sets (i.e., 25) and change the number of test samples for each class correspondingly for all methods under such a setting. Table V shows the image classification accuracy with 5-way 1-shot transductive learning with 5 query samples in each class, averaged over the tiered-ImageNet dataset. It can be found that the accuracies of all GNN-based methods decrease due to the different class distributions in each graph between training and testing

TABLE V
EFFECT OF QUERY SAMPLES DISTRIBUTION OVER TIERED-IMAGENET DATASET FOR THE 5-WAY 1-SHOT TASK UNDER THE TRANSDUCTIVE SETTING. THE TOTAL NUMBER OF QUERY SAMPLES UNDER THE TWO SETTINGS REMAINS THE SAME (I.E., 25)

Method	Random	Uniform
DPGN [14]	57.84	63.20
GNN [5]	59.77	65.11
Ours w/o Neighbor Att.	59.64	59.70
Ours w/o Node Self Att.	61.25	64.53
Ours	61.36	67.94

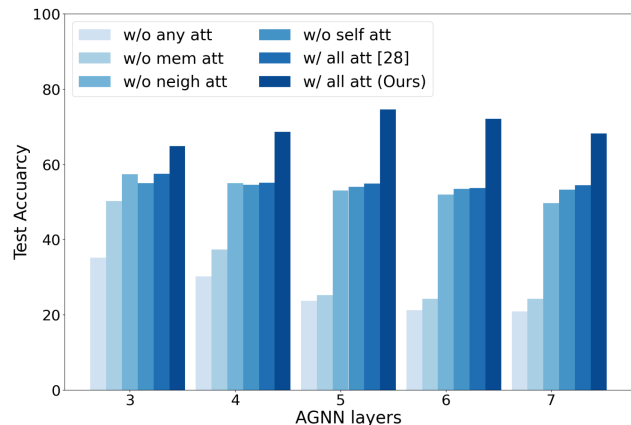


Fig. 4. Ablation study: Classification accuracies using AGNN (3 to 7 layers) and its variations over the tiered-ImageNet dataset.

tasks, especially for the DPGN method that delivers distribution relations between nodes in the distribution graph to refine the point graph for classification. In contrast, with the query-set samples of “random” labels, the proposed AGNN can still generate significantly better results compared to the vanilla GNN. We also observe that each proposed attention mechanism contributes to robustness. For example, neighbor attention can help prevent “over-mixing” with all nodes, as the sparse adjacency matrix can attend to the related nodes (i.e., nodes with the same class) in an adaptive way,

G. Ablation Study

We investigate the effectiveness of each proposed attention module by conducting the following ablation study.

1) *Impact of AGNN Layers*: Fig. 4 plots the image classification accuracy over the tiered-ImageNet dataset, with different variations of the proposed AGNN, by removing the node self-attention (self att), neighbor attention (neigh att), and layer memory attention (memory att) modules. We also compare with our previous work [28]. It is clear that all variants except our own method generate degraded results as GNN layers go deeper, and some even suffer from more severe over-smoothing, i.e., the accuracy of GNN without any attention mechanisms drops quickly as the number of GNN layers increases. Results also show that neighbor and layer memory attentions are more essential to alleviate the over-smoothing problem. This is consistent with our expectations as the node self-attention is only

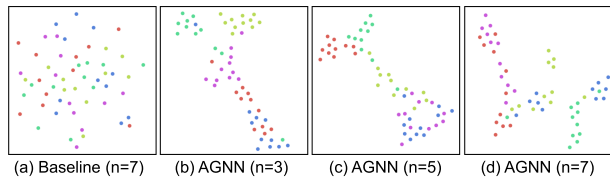


Fig. 5. T-SNE visualization of node features under 5-way 1-shot with 10 query samples in each class on the testing set of the mini-ImageNet dataset. From left to right: the baseline GNN method with $n = 7$ layers and the proposed AGNN method with different layers, i.e., $n = 3, 5$, and 7 .

adopted before the operation of graph convolution, which has less impact on the over-smoothing issue. Furthermore, with the comparison between the two AGNN versions, we conclude the proposed layer-wise neighbor attention can provide better affinity. A reasonable explanation is that the layer-wise neighbor attention mechanism adopts the previously learned neighbor information (i.e., adjacency matrix) as prior knowledge, which can better help learn more accurate affinity between nodes and avoid over-fitting to the features in the current layer. Moreover, the dynamic sparsity ratio can force GNNs to aggregate different degrees of neighbor information at different layers, thereby providing flexible graph structures for various tasks and alleviating over-fitting and over-smoothing issues.

We also plot the t-SNE visualizations of different layers for the proposed AGNN ($n = 3, n = 5$, and $n = 7$) and the baseline GNN method ($n = 7$) over mini-ImageNet, which is shown in Fig. 5. Results show that as the GNN layer goes deeper, i.e., $n = 7$, the baseline method appears to over-smoothing issues on the testing set. In this case, the few-shot GNN-based method failed to generalize to novel tasks without attention or regularized terms. In contrast, the proposed AGNN method can alleviate this issue even at very deep layers. Another observation is that as the number of layers increases, the AGNN performance first increases and then decreases, and when $n = 5$, the AGNN achieves the best t-SNE performance. A reasonable explanation is that the proposed attention modules can build a flexible graph model, which helps to alleviate the over-smoothing issue for deep GNN layers and generalize well to unseen tasks to avoid the over-fitting problem.

2) *Comparison with Self-Attention Transformer Mechanism:* As mentioned before, we propose a node self-attention module to exploit the relationship between samples by learning the correlation matrix from feature and label levels, respectively, and introducing a fusion strategy to combine the information. Instead of the self-attention mechanism we proposed, the self-attention transformer [11], [50] is also a popular module that leverages the relationship between samples to learn discriminative feature embeddings. To validate the effectiveness of our proposed node self-attention module, we experiment with these two kinds of designed modules for few-shot image classification. For a fair comparison, we add these two modules (our proposed node self-attention module and self-attention transformer module) before the AGNN layers to evaluate the performance, respectively. Note that we concatenate the embedding feature and one-hot label feature as the input of the self-attention transformer module. The output dimension of the linear mapping

TABLE VI
TEST ACCURACY OF DIFFERENT SELF-ATTENTION MECHANISMS OVER THE MINI-IMAGENET DATASET. * INDICATES THAT THE PROPOSED METHOD ADOPTS THE SELF-ATTENTION TRANSFORMER LAYER INSTEAD OF THE NODE SELF-ATTENTION MECHANISM

Method	5-way 1-shot	5-way 5-shot
ProtoNet [3]	49.42	68.20
ProtoNet + Transformer [11]	55.15	71.61
AGNN [28]	60.88	74.59
Ours w/ Transformer*	63.73	83.80
Ours	76.99	85.20

TABLE VII
COMPARISON WITH DIFFERENT DESIGNS (I.E., FIXED OR FLEXIBLE) OF THE SPARSITY RATIO $\beta^{(k)}$ IN THE NEIGHBOR ATTENTION MODULE OF THE AGNN METHOD UNDER THE 5-WAY 1-SHOT SETTING

Setting	mini-ImageNet	tiered-ImageNet
Fixed ($\beta^{(k)} = 1.0$)	64.18 \pm 0.51	69.89 \pm 0.45
Fixed ($\beta^{(k)} = 0.9$)	64.18 \pm 0.51	62.03 \pm 0.42
Fixed ($\beta^{(k)} = 0.8$)	63.17 \pm 0.54	54.78 \pm 0.46
Fixed ($\beta^{(k)} = 0.7$)	63.30 \pm 0.51	56.04 \pm 0.43
Fixed ($\beta^{(k)} = 0.6$)	63.00 \pm 0.53	64.91 \pm 0.52
Flexible ($\beta^{(k)} = 1.0 - 0.1k$)	76.99 \pm 0.37	74.54 \pm 0.46

function for the transformer is set the same as the input dimension. We apply ProtoNet [3] as the baseline method for comparison. We also incorporate FEAT [11], a few-shot method that applies the self-attention transformer as one kind of embedding adaptation function into comparison. As we can see from the results in Table VI, under our AGNN framework, our node self-attention mechanism can achieve an improvement of 13.26% and 1.40% over the self-attention transformer under 5-way 1-shot and 5-way 5-shot settings, respectively. It shows that node self-attention can implement rich relationships between samples from different levels and fuse them better. Moreover, we find that our proposed method with the transformer can also obtain a performance improvement compared with ProtoNet with the transformer, which validates the effectiveness of our proposed AGNN method.

3) *Design Choice of Sparsity Ratio:* To validate the effectiveness of our proposed flexible setting of sparsity ratio in the neighbor attention module, we consider two different sparsity ratio designs, i.e., a fixed value for all AGNN layers or different values for each AGNN layer. In both designs, the number of AGNN layers is set to 5, i.e., the integer k ranges from 0 to 4. Table VII shows the classification accuracy with two sparsity ratio designs in the neighbor attention module under the 5-way 1-shot setting. Results show that the classification accuracy of AGNN is affected by the sparsity ratio, and $\beta^{(k)} = 1.0$ is the optimal parameter setting considering the fixed sparsity ratio design under the 1-shot setting. Compared with the results of fixed sparsity design, we observe that our proposed variable sparsity ratios can significantly improve the performance of few-shot classification tasks. A plausible explanation is that this dynamic layer-wise sparsity ratio design brings more flexibility to each AGNN layer. Specifically, we need to aggregate more neighbor

TABLE VIII
INDUCTIVE ACCURACY ON MINI-IMAGENET AND TIERED-IMAGENET DATASETS UNDER 5-WAY 1-SHOT SETTING. “-” MEANS NOT APPLYING THE NODE SELF-ATTENTION MECHANISM

Hyper-Parameter Setting		5-way 1-shot	
α	m	mini-ImageNet	tiered-ImageNet
-	16	65.17 \pm 0.51	67.30 \pm 0.51
-	32	65.72 \pm 0.53	67.68 \pm 0.52
0.5	16	76.99 \pm 0.37	72.13 \pm 0.43
0.9	32	76.28 \pm 0.44	74.54 \pm 0.46
0.9	48	76.28 \pm 0.47	68.19 \pm 0.46

TABLE IX
ABLATION STUDY: EFFECTS OF THE PROPOSED ATTENTION MODULES OVER MINI-IMAGENET

Self att.	Neigh att.	Mem att.	5-way 1-shot	5-way 5-shot
-	-	-	61.55 \pm 0.52	75.91 \pm 0.50
-	✓	✓	72.36 \pm 0.44	83.99 \pm 0.33
✓	-	✓	65.62 \pm 0.51	79.30 \pm 0.45
✓	✓	-	69.72 \pm 0.40	79.19 \pm 0.37
✓	✓	✓	76.99 \pm 0.37	85.20 \pm 0.25

information in shallow AGNN layers to learn feature representations. Thus, there is no need to impose strong sparsity constraints in the first few layers, that is, no or less sparsity, i.e., a higher value of β . However, as AGNN goes deeper, we need to force the model to capture information only from the most relevant nodes for classification and avoid over-smoothing issues, i.e., nodes with a higher probability of belonging to the same class.

4) *Impact of Hyper-Parameters:* There are two hyper-parameters in the proposed AGNN method, namely α and m , corresponding to the ratio for label fusion and the output feature dimension of each AGNN layer, respectively. The weighted parameter α for label fusion ranges between 0 and 1. The number of dimension m is selected from $\{16, 32, 48\}$. Table VIII shows how varying these two parameters affect the test accuracy for image classification averaged over the mini-ImageNet dataset under the transductive setting. Besides, we also test the model when the label fusion mechanism is totally removed, denoted as “-” in the table. The results show that introducing the node self-attention mechanism before the AGNN layers can learn more flexible task-relevant features, thereby improving the performance of the proposed model. It is obvious that $m = 32$ is a proper ratio for all datasets. The empirical results also show that the mini-ImageNet dataset is not sensitive to the hyper-parameters while retaining a larger proportion of ground-truth label information (i.e., a larger value of α) is more conducive to improving performance for the tiered-ImageNet dataset.

5) *The Effects of the Proposed Attention Modules:* Table IX summarizes the effects of the proposed node self-attention (self att), neighbor attention (neigh att), and layer memory attention (mem att) modules over mini-ImageNet. Without node self-attention, the proposed method directly utilizes the output feature embeddings extracted from the backbone network as node representations of the AGNN model. Without neighbor attention, each AGNN layer adopts a fully connected

adjacency matrix to update node features. Without layer memory attention, the proposed method does not consider the dense connection between different AGNN layers, i.e., the node features of each layer are just the output of the current AGNN layer. Results show that all modules consistently improve the classification performance under both 5-way 1-shot and 5-shot settings over mini-ImageNet. Furthermore, we can observe that the effectiveness of neighbor and layer memory attentions is more solid than node self-attention. As neighbor attention via sparsity constructs a task-specific dynamic and flexible relationship between nodes, it improves the generalizability of unseen tasks in few-shot scenarios. In addition, the layer memory mechanism enables AGNN to aggregate the information of each layer, which also helps to improve performance. In contrast, the node self-attention mechanism combines image and category information to provide a task-specific feature initialization, which also helps to some extent.

VII. CONCLUSION

In this paper, we proposed a novel Attentive GNN model for few-shot learning. The proposed AGNN makes full use of the relationships between image samples for knowledge modeling and generalization. By introducing a triple-attention mechanism for graph initialization, graph update, and correlation across graph layers, the proposed AGNN model effectively alleviates over-smoothing and over-fitting issues when applying deep GNN models. Extensive experiments are conducted on both standard few-shot classification benchmarks and two more challenging scenarios (i.e., fine-grained and semi-supervised few-shot classification tasks), showing that our proposed AGNN achieved state-of-the-art results comparing to few-shot learning methods.

Limitations: Despite that the proposed AGNN performs well for few-shot learning tasks, GNN-based few-shot methods in general are usually limited by the graph size. To be specific, when the number of samples in each meta-task increases, GNN-based few-shot methods require more space and computational complexity for graph construction and feature update, which limits the extension of such type of methods to other applications. In future work, it is worth investigating how the proposed method can adapt to large-scale graphs for more classification tasks.

REFERENCES

- [1] L. Fei-Fei, R. Fergus, and P. Perona, “One-shot learning of object categories,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 4, pp. 594–611, Apr. 2006.
- [2] O. Vinyals et al., “Matching networks for one shot learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3630–3638.
- [3] J. Snell, K. Swersky, and R. Zemel, “Prototypical networks for few-shot learning,” in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4077–4087.
- [4] F. Sung et al., “Learning to compare: Relation network for few-shot learning,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 1199–1208.
- [5] V. G. Satorras and J. B. Estrach, “Few-shot learning with graph neural networks,” in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: <https://openreview.net/forum?id=BJj6qGbRW>
- [6] C. Zhang, Y. Cai, G. Lin, and C. Shen, “DeepEMD: Few-shot image classification with differentiable earth mover’s distance and structured classifiers,” in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 12203–12213.

- [7] Y. Guo, Z. Ma, X. Li, and Y. Dong, "TLRM: Task-level relation module for GNN-based few-shot learning," in *Proc. IEEE Int. Conf. Vis. Commun. Image Process.*, 2021, pp. 1–5.
- [8] Y. Ma et al., "Transductive relation-propagation with decoupling training for few-shot learning," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 33, no. 11, pp. 6652–6664, Nov. 2022.
- [9] X. Zhong, C. Gu, M. Ye, W. Huang, and C.-W. Lin, "Graph complemented latent representation for few-shot image classification," *IEEE Trans. Multimedia*, early access, Jan. 11, 2022, doi: [10.1109/TMM.2022.3141886](https://doi.org/10.1109/TMM.2022.3141886).
- [10] J. Zhang, M. Zhang, Z. Lu, and T. Xiang, "AdarGCN: Adaptive aggregation GCN for few-shot learning," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, 2021, pp. 3482–3491.
- [11] H.-J. Ye, H. Hu, D.-C. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 8808–8817.
- [12] Y. Liu et al., "Learning to propagate labels: Transductive propagation network for few-shot learning," in *Proc. Int. Conf. Learn. Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=SyVuRiC5K7>
- [13] J. Kim, T. Kim, S. Kim, and C. D. Yoo, "Edge-labeling graph neural network for few-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 11–20.
- [14] L. Yang et al., "DPGN: Distribution propagation graph network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2020, pp. 13390–13399.
- [15] S. Tang et al., "Mutual CRF-GNN for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 2329–2339.
- [16] W. L. Hamilton, R. Ying, and J. Leskovec, "Inductive representation learning on large graphs," in *Proc. 31st Int. Conf. Neural Inf. Process. Syst.*, 2017, pp. 1025–1035.
- [17] M. Li et al., "Actional-structural graph convolutional networks for skeleton-based action recognition," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 3595–3603.
- [18] C. Wang, B. Samari, and K. Siddiqi, "Local spectral graph convolution for point set feature learning," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 52–66.
- [19] Y. Feng, J. Gao, and C. Xu, "Learning dual-routing capsule graph neural network for few-shot video classification," *IEEE Trans. Multimedia*, early access, Mar. 07, 2022, doi: [10.1109/TMM.2022.3156938](https://doi.org/10.1109/TMM.2022.3156938).
- [20] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Proc. 32nd AAAI Conf. Artif. Intell.*, 2018, pp. 3538–3545.
- [21] Y. Rong, W. Huang, T. Xu, and J. Huang, "DropEdge: Towards deep graph convolutional networks on node classification," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: <https://openreview.net/forum?id=Hkx1qkrKPr>
- [22] Y. Luo et al., "Every node counts: Self-ensembling graph convolutional networks for semi-supervised learning," *Pattern Recognit.*, vol. 106, 2020, Art. no. 107451.
- [23] G. Li et al., "DeepGCNs: Making GCNs go as deep as CNNs," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Apr. 19, 2021, doi: [10.1109/TPAMI.2021.3074057](https://doi.org/10.1109/TPAMI.2021.3074057).
- [24] G. Huang, Z. Liu, L. V. D. Maaten, and K. Q. Weinberger, "Densely connected convolutional networks," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2017, pp. 4700–4708.
- [25] B. Chamberlain et al., "Beltrami flow and neural diffusion on graphs," in *Proc. Adv. Neural Inf. Process. Syst.*, 2021, vol. 34, pp. 1594–1609.
- [26] B. Chamberlain et al., "Grand: Graph neural diffusion," in *Proc. Int. Conf. Mach. Learn.*, 2021, pp. 1407–1418.
- [27] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 770–778.
- [28] H. Cheng, J. T. Zhou, W. P. Tay, and B. Wen, "Attentive graph neural networks for few-shot learning," in *Proc. IEEE 5th Int. Conf. Multimedia Inf. Process. Retrieval*, 2022, pp. 152–157.
- [29] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, 2017, vol. 70, pp. 1126–1135.
- [30] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 10657–10665.
- [31] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," in *Proc. Int. Conf. Learn. Representations*, 2017. [Online]. Available: <https://openreview.net/forum?id=rJY0-KcII>
- [32] A. Phaphuangwitayakul, Y. Guo, and F. Ying, "Fast adaptive meta-learning for few-shot image generation," *IEEE Trans. Multimedia*, vol. 24, pp. 2205–2217, 2022.
- [33] B. Hariharan and R. Girshick, "Low-shot visual recognition by shrinking and hallucinating features," in *Proc. IEEE Int. Conf. Comput. Vis.*, 2017, pp. 3018–3027.
- [34] Y.-X. Wang, R. Girshick, M. Hebert, and B. Hariharan, "Low-shot learning from imaginary data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2018, pp. 7278–7286.
- [35] S. Yang, L. Liu, and M. Xu, "Free lunch for few-shot learning: Distribution calibration," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=JWOiYxMG92s>
- [36] K. Cao, M. Brbic, and J. Leskovec, "Concept learners for few-shot learning," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=eJlJF3-LoZO>
- [37] H. Li, D. Eigen, S. Dodge, M. Zeiler, and X. Wang, "Finding task-relevant features for few-shot learning by category traversal," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, 2019, pp. 1–10.
- [38] B. Oreshkin, P. R. López, and A. Lacoste, "TADAM: Task dependent adaptive metric for improved few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 721–731.
- [39] H. Huang, J. Zhang, J. Zhang, J. Xu, and Q. Wu, "Low-rank pairwise alignment bilinear network for few-shot fine-grained image classification," *IEEE Trans. Multimedia*, vol. 23, pp. 1666–1680, 2021.
- [40] Y. Zhu, W. Min, and S. Jiang, "Attribute-guided feature learning for few-shot image recognition," *IEEE Trans. Multimedia*, vol. 23, pp. 1200–1209, 2021.
- [41] H. Zhang, H. Li, and P. Koniusz, "Multi-level second-order few-shot learning," *IEEE Trans. Multimedia*, early access, Jan. 13, 2022, doi: [10.1109/TMM.2022.3142955](https://doi.org/10.1109/TMM.2022.3142955).
- [42] J. Zhang, J. Song, L. Gao, and H. Shen, "Free-lunch for cross-domain few-shot learning: Style-aware episodic training with robust contrastive learning," in *Proc. 30th ACM Int. Conf. Multimedia*, 2022, pp. 2586–2594.
- [43] J. Zhang, J. Song, L. Gao, Y. Liu, and H. T. Shen, "Progressive meta-learning with curriculum," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 9, pp. 5916–5930, Sep. 2022.
- [44] R. Hou, H. Chang, M. Bingpeng, S. Shan, and X. Chen, "Cross attention network for few-shot classification," in *Proc. Adv. Neural Inf. Process. Syst.*, 2019, pp. 4005–4016.
- [45] L. Ke, M. Pan, W. Wen, and D. Li, "Compare learning: Bi-attention network for few-shot learning," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 2233–2237.
- [46] D. Kang, H. Kwon, J. Min, and M. Cho, "Relational embedding for few-shot classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 8822–8833.
- [47] S.-L. Xu, F. Zhang, X.-S. Wei, and J. Wang, "Dual attention networks for few-shot fine-grained recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, pp. 2911–2919.
- [48] J. Bruna, W. Zaremba, A. Szlam, and Y. LeCun, "Spectral networks and locally connected networks on graphs," in *Proc. Int. Conf. Learn. Representations*, CBLs, Apr. 2014.
- [49] M. Defferrard, X. Bresson, and P. Vandergheynst, "Convolutional neural networks on graphs with fast localized spectral filtering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3844–3852.
- [50] A. Vaswani et al., "Attention is all you need," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5998–6008.
- [51] J. Cheng, L. Dong, and M. Lapata, "Long short-term memory-networks for machine reading," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 551–561.
- [52] A. Parikh, O. Täckström, D. Das, and J. Uszkoreit, "A decomposable attention model for natural language inference," in *Proc. Conf. Empirical Methods Natural Lang. Process.*, 2016, pp. 2249–2255.
- [53] L. Guo, Z. Zha, S. Ravishanker, and B. Wen, "Exploiting non-local priors via self-convolution for highly-efficient image restoration," *IEEE Trans. Image Process.*, vol. 31, pp. 1311–1324, 2022.
- [54] P. Veličković et al., "Graph attention networks," in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: <https://openreview.net/forum?id=rJXmpikCZ>
- [55] L. Li, Z. Gan, Y. Cheng, and J. Liu, "Relation-aware graph attention network for visual question answering," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2019, pp. 10313–10322.
- [56] J. Lee, I. Lee, and J. Kang, "Self-attention graph pooling," in *Proc. 36th Int. Conf. Mach. Learn., Int. Mach. Learn. Soc.*, 2019, pp. 6661–6670.
- [57] A. Sperduti and A. Starita, "Supervised neural networks for the classification of structures," *IEEE Trans. Neural Netw.*, vol. 8, no. 3, pp. 714–735, May 1997.
- [58] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," in *Proc. Int. Conf. Learn. Representations*, 2017. [Online]. Available: <https://openreview.net/forum?id=SJU4ayYgI>

- [59] K. Xu, W. Hu, J. Leskovec, and S. Jegelka, "How powerful are graph neural networks?," in *Proc. Int. Conf. Learn. Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=ryGs6iA5Km>
- [60] G. Hinton, O. Vinyals, and J. Dean, "Distilling the knowledge in a neural network," in *Proc. NeurIPS workshop Deep Learn. Representation Learn.*, 2014.
- [61] B. Wen, S. Ravishanker, and Y. Bresler, "Structured overcomplete sparsifying transform learning with convergence guarantees and applications," *Int. J. Comput. Vis.*, vol. 114, no. 2/3, pp. 137–167, 2015.
- [62] K. Xu et al., "Representation learning on graphs with jumping knowledge networks," in *Proc. Int. Conf. Mach. Learn.*, 2018, pp. 5453–5462.
- [63] F. Ji, J. Yang, Q. Zhang, and W. P. Tay, "GFCN: A new graph convolutional network based on parallel flows," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 3332–3336.
- [64] K. Oono and T. Suzuki, "Graph neural networks exponentially lose expressive power for node classification," in *Proc. Int. Conf. Learn. Representations*, 2020. [Online]. Available: <https://openreview.net/forum?id=S11dO2EFPp>
- [65] M. Ren et al., "Meta-learning for semi-supervised few-shot classification," in *Proc. Int. Conf. Learn. Representations*, 2018. [Online]. Available: <https://openreview.net/forum?id=HJcSzz-CZ>
- [66] M.-E. Nilsback and A. Zisserman, "Automated flower classification over a large number of classes," in *Proc. IEEE 6th Indian Conf. Comput. Vis., Graph. Image Process.*, 2008, pp. 722–729.
- [67] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD Birds-200-2011 dataset," California Inst. Technol., Tech. Rep. CNS-TR-2011-001, 2011.
- [68] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
- [69] W.-Y. Chen, Y.-C. Liu, Z. Kira, Y.-C. F. Wang, and J.-B. Huang, "A closer look at few-shot classification," in *Proc. Int. Conf. Learn. Representations*, 2019. [Online]. Available: <https://openreview.net/forum?id=HkxLXnAcFQ>
- [70] M. Zhang et al., "IEPT: Instance-level and episode-level pretext tasks for few-shot learning," in *Proc. Int. Conf. Learn. Representations*, 2021. [Online]. Available: <https://openreview.net/forum?id=xzqLpqRzxLq>
- [71] Y. Liu et al., "Learning to affiliate: Mutual centralized learning for few-shot classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14411–14420.
- [72] A. Afrasiyabi, H. Larochelle, J.-F. Lalonde, and C. Gagné, "Matching feature sets for few-shot image classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 9014–9024.
- [73] C. Chen, X. Yang, C. Xu, X. Huang, and Z. Ma, "ECKPN: Explicit class knowledge propagation network for transductive few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2021, pp. 6596–6605.
- [74] Y. Chen, Z. Liu, H. Xu, T. Darrell, and X. Wang, "Meta-baseline: Exploring simple meta-learning for few-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 9062–9071.
- [75] J. Xie, F. Long, J. Lv, Q. Wang, and P. Li, "Joint distribution matters: Deep brownian distance covariance for few-shot classification," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 7972–7981.
- [76] J. Xu, H. Le, M. Huang, S. Athar, and D. Samarasinghe, "Variational feature disentangling for fine-grained few-shot classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis.*, 2021, pp. 8812–8821.
- [77] J. Ling, L. Liao, M. Yang, and J. Shuai, "Semi-supervised few-shot learning via multi-factor clustering," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, 2022, pp. 14564–14573.



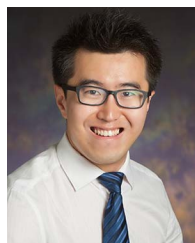
Hao Cheng received the B.Eng. degree from the School of Software, Dalian University of Technology, Dalian, China, in 2016, and the M.S. degree from the School of Computer Science and Technology, Tianjin University, Tianjin, China, in 2019. He is currently working toward the Ph.D. degree with Nanyang Technological University, Singapore. His research interests include few-shot learning, graph neural networks, and computer vision.



Joey Tianyi Zhou (Senior Member, IEEE) received the Ph.D. degree in computer science from Nanyang Technological University, Singapore. He is currently a Senior Scientist, Investigator and Group Manager with A*STAR Centre for Frontier AI Research (CFAR), Singapore. He is also holding an Adjunct Faculty position (adj. Assoc. Prof.) with the National University of Singapore, Singapore. Before working with CFAR, he was a Senior Research Engineer with SONY U.S. Research Center in San Jose, USA. His research interests mainly focuses on improving the efficiency and robustness of machine learning algorithms. Dr. Zhou organized ICDCS annual workshop on Efficient AI meets Edge Computing, ACML'16 workshop on Learning on Big Data workshop and IJCAI'19 workshop on Multi-output Learning. He is an Associate Editor for IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE and IEEE ACCESS, *IET Image Processing*, and TPC Chair in Mobimedia 2020. He was the recipient of the NeurIPS Best Reviewer Award in 2017.



Wee Peng Tay (Senior Member, IEEE) received the B.S. degree in electrical engineering and mathematics, and the M.S. degree in electrical engineering from Stanford University, Stanford, CA, USA, in 2002, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, MA, USA, in 2008. He is currently an Associate Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. Dr. Tay received the Tan Chin Tuan Exchange Fellowship in 2015. He is a co-author of the Best Student Paper Award at the Asilomar conference on Signals, Systems, and Computers in 2012 and the IEEE Signal Processing Society Young Author Best Paper Award in 2016. His research interests include signal and information processing over networks, distributed inference and estimation, statistical privacy, and robust machine learning. He was an Associate Editor for IEEE TRANSACTIONS ON SIGNAL PROCESSING (2015–2019), and is currently an Associate Editor for IEEE TRANSACTIONS ON SIGNAL AND INFORMATION PROCESSING OVER NETWORKS, the Editor for IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, and the Editor of IEEE OPEN JOURNAL OF VEHICULAR TECHNOLOGY.



Bihan Wen (Member, IEEE) received the B.Eng. degree in electrical and electronic engineering from Nanyang Technological University, Singapore, in 2012, and the M.S. and Ph.D. degrees in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2015 and 2018, respectively. He is currently a Nanyang Assistant Professor with the School of Electrical and Electronic Engineering, Nanyang Technological University. His research interests include machine learning, computational imaging, computer vision, image and video processing, and Big Data applications. Dr. Wen was the recipient of the 2016 Yee Fellowship and the 2012 Professional Engineers Board Gold Medal. He was also the recipient of the Best Paper Runner Up Award at the IEEE International Conference on Multimedia and Expo in 2020. He has been an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY since 2022, and an Associate Editor for *MDPI Micromachines* since 2021. He has also been the Guest Editor for *IEEE Signal Processing Magazine* in 2022.