Fuzzy Graph Neural Network for Few-Shot Learning

Tong Wei, Junlin Hou and Rui Feng
Shanghai Key Lab of Intelligent Information Processing
School of Computer Science, Fudan University
Shanghai, China
{18210240206, 18110240004, fengrui}@fudan.edu.cn

***Abstract*—Recent works have shown that graph neural networks (GNNs) can substantially improve the performance of few-shot learning benefitting from their natural ability to learn inter-class uniqueness and intra-class commonality. However, previous GNN methods have not achieved satisfactory performance due to the absence of a strong relational inductive bias which determines how entities interact and are isolated. In this paper, inspired by the fuzzy theory, we propose a novel meta-learning method called Fuzzy GNN (FGNN), which obtains superior relational inductive biases in each episode, for few-shot learning. Specifically, we employ an edge-focused GNN to perform the edge prediction by iteratively updating the edge-labels. According to the output of edge prediction, we design a fuzzy membership function to achieve more exact relationship representations for node classification. The parameters of the FGNN are learned by episodic training with mixed loss including node-label and edge-label. Extensive experimental evaluation clearly demonstrates the effectiveness of FGNN. The results show that our method achieves state-of-the-art performance and a significant improvement over other GNN methods on two few-shot learning benchmarks.**

I. Introduction

The recent success of deep neural networks [1], [2] has boosted research on many computer vision tasks such as image classification, object detection, and semantic segmentation. However, the power of deep models is partially attributed to the availability of large training data. This precondition not only limits the domain in which the models can be applied, but also does not conform to human cognitive process. They can rapidly learn a new concept from only one or a few examples based on their past experiences. Therefore, more and more researchers are turning their attention to few-shot learning [3]–[5]. The aim of few-shot learning is to learn new objects with only a few training examples for each of them. This is not too difficult for humans, but still a challenging problem for the machine.

Inspired by human learning, researchers have explored a meta-learning process for few-shot learning, which solves new tasks with few labeled data based on knowledge obtained from previous experiences. More specifically, meta-learning strategies can learn how to efficiently recognize unseen classes with few training data by leveraging a distribution of similar tasks. They learn an across-task meta-learner from multiple similar tasks to summarize a common representation, providing a better initialization for new tasks with unseen classes. Recent research [5]–[9] has successfully exploited this meta-learning paradigm to tackle the problem of few-shot image classification. Essentially, these methods learn a similarity measure and propagate the label information from the support sets of images to the query sets.

Since the full exploitation of relationships between a support set and a query is greatly required in few-shot learning [9]–[11], Graph Neural Networks (GNNs) have been introduced to handle rich relational structures on each recognition task. GNNs aggregate features from neighbors iteratively by message passing algorithm, and therefore express complex interactions among support and query instances. In particular, GNN methods in few-shot learning learn inter-class uniqueness and intra-class commonality by optimizing the nodes and edges update functions to achieve a better performance. For example, Garcia et al. [9] use a node-focused GNN to propagate messages between connected nodes to classify unlabeled samples. Kim et al. [11] update edge-labels iteratively for inferring a query association to an existing support set. However, each few-shot task is constructed as a fully connected graph with edge weights in the existing methods since they do not have intrinsic graph structures. This kind of structure results in a weak relational inductive bias for GNN, making it hard to learn the accurate relationship in the graph due to the inexact graph structure, which may propagate noise through edges between unrelated nodes.

The relationship in few-shot tasks is the similarity between samples. However, 'similar' is a fuzzy concept, which is not clearly defined. To solve the problem of the fuzzy relationship, we introduce fuzzy theory which is good at dealing with problems relating to ambiguous, subjective and imprecise judgments. In this paper, we propose a novel meta-learning method called Fuzzy Graph Neural Network (FGNN)[1] to obtain superior relational inductive biases in learning episodes for few-shot learning. Compared with the previous GNN methods, FGNN offers a new relationship representation strategy for graph construction instead of adopting the fully connected graph structure. In our model, we treat the relationship construction between nodes as a fuzzy problem(in Section 2) and design a membership function to compute the membership degree of each element from the universe of discourse to a fuzzy set. The universe of discourse is defined by the relation-

[1]Code: https://github.com/sadbb/few-shot-fgnn

978-1-7281-6926-2/20/$31.00 ©2020 IEEE

ship representations, that is, the edge feature in GNNs, which generated by the edge prediction. The membership degree can be understood as a more reasonable updated relationship representation and offer superior relational inductive biases for different learning episodes.

The proposed FGNN consists of a node-focused GNN and an edge-focused GNN. In the beginning, the graph is initialized to a fully connected structure to perform edge prediction. After edge-focused updating, some edges will be broken off according to membership degree computed by a designed membership function. Then the graph will be updated with the new structure for node classification. Ultimately, the node loss and the edge loss will be computed to update the parameters of the FGNN, and the total model is learned in meta-learning strategy [12], [13].

In the learning process of the edge prediction, we only use one graph instead of using two graphs [11] because embedding two opposing relationships in one graph can force the model to learn their difference, which is beneficial for learning inter-class uniqueness and intra-class commonality. We conduct experiments on two benchmark few-shot image classification datasets to show our performance and compare them with few-shot learning methods.

To sum up, our contributions are three-fold: 1) We propose a novel FGNN method that obtains superior relational inductive biases in learning episodes for solving few-shot learning tasks. 2) We develop a novel fuzzy strategy for generating more reasonable relationship representations. 3) We conduct extensive experiments on two challenging few-shot learning benchmarks, and FGNN outperforms other few-shot learning methods including the existing GNN methods.

In the following sections, we review the related work in Section II. It is followed by the elaboration of the proposed FGNN approach in Section III. We provide experimental results in Section IV, and finally we conclude in Section V.

II. Related work

A. Few-Shot Learning

The aim of few-shot learning is to classify samples from unseen classes with only a few labeled training examples. The classification of few-shot learning methods is not immutable, because some methods combine several mechanisms at the same time. We focus on the methods that most relevant to ours and divide them into two branches according to the emphasis of their works.

Works of the first branch are based on metric-learning. Obviously, these methods explore a similarity metric, which can be viewed as learning to compare. In 2015, a Siamese network [4] trains a parallel network in a supervised way and then compares the similarity between their extracted features for few-shot learning. A year later, Matching nets [5] based on memory and attention introduced the episodic training mechanism into few-shot learning for the first time, but it can only be used in the 1-shot tasks. To deal with this problem, the Prototypical network [6] takes the mean of each class in episodic training as its corresponding prototype representation and then treats few-shot learning as 1-shot learning. To explore the relationship between query images and support images, Sung et al.[6] proposed Relation Network to learn a deep non-linear measure.

Works of the second branch are based on meta-learning. The idea of meta-learning has been proposed more than a couple of decades ago [14]. For few-shot learning, these meta-learning methods aim to train a meta-learner on multiple few-shot tasks, in which only a few labeled samples are available, to classify samples from unseen classes. For example, Ravi et al. [12] trained an LSTM-based meta-learner, which learns weight initialization and an optimizer of the model weights. MM-Net [15] used memory slots to structure a contextual learner to predict the parameters of an embedding network for unlabeled images. But most of the meta-learning methods typically use shallow neural networks which limit their effectiveness and they always suffer from overfitting. Sun et al. [8] proposed a method called meta-transfer learning (MTL) which learned to adapt a deep CNN for few-shot learning tasks. MTL employs deep neural networks to improve performance and design two neuron operations to reduce the probability of overfitting. However, MTL and other meta-learning methods treat the samples in the same task as independent individuals which makes it difficult to learn the relationship between the samples. Therefore, several GNN methods are proposed to explore inter-class uniqueness and intra-class commonality in training episodes.

B. Graph neural network

Graph Neural Network (GNN), a deep learning architecture on graph-structured data, was first introduced by Gori et al. [16]. Due to the ability to exchange messages with neighbor nodes, some approaches have introduced GNN into few-shot learning [9], [11], [17], [18]. Works on GNNs can be divided into two categories: spectral-based methods [19] and spatial-based methods [20]–[22]. In few-shot learning, different tasks generate different graph structures which the spectral-based methods cannot adapt to [23], so we focus on spatial-based methods. Garcia et al. [9] for the first time, incorporated GNNs into few-shot learning and generalized several proposed few-shot learning models into the GNN framework. Kim et al. [11] adapted a deep neural network on the edge-labeling graph called EGNN, which is a concatenation of feature extractor and edge-focused GNN and use fully-connected structure. They use the output of edge prediction to evaluate the model. In FGNN, the model is evaluated by the output of node classification. The output of edge prediction is only used to generate the relational inductive bias with the membership function. This strategy offers a strong inductive bias for different episodes to optimize the propagation process.

C. Fuzzy theory

Since its inception in 1965 by Zadeh [24], the theory of fuzzy sets has advanced in a variety of disciplines. Not all concepts in the world, such as 'young', are crisp which

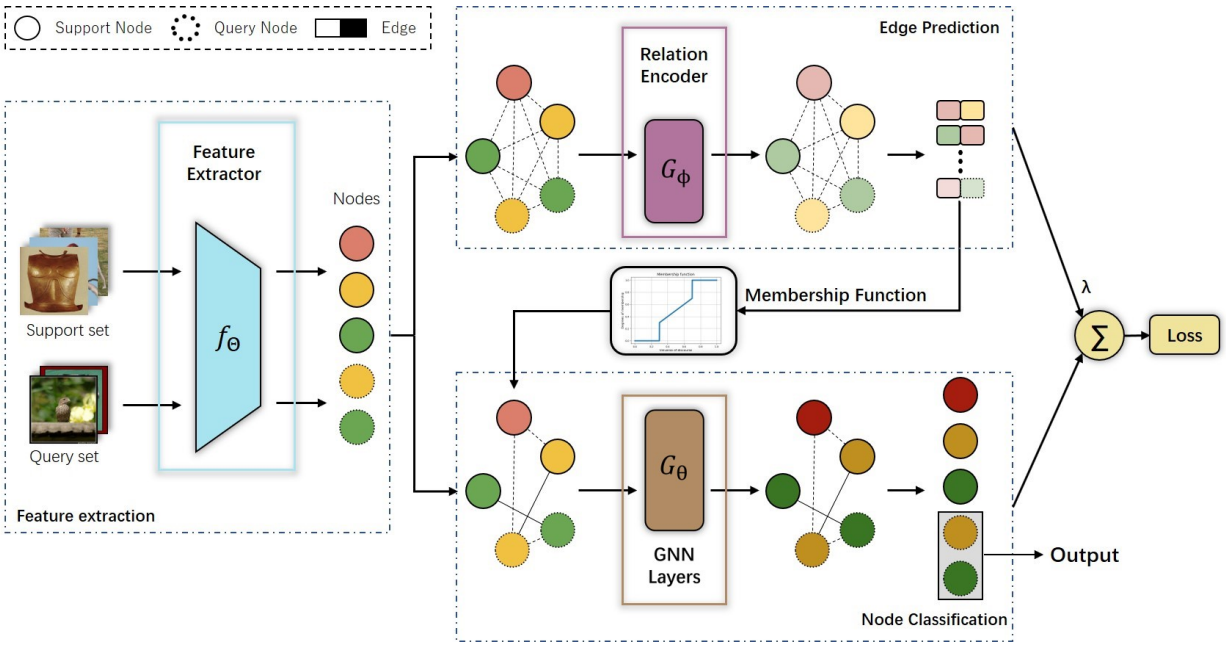

Fig. 1: The overall framework of the proposed FGNN model. As can be seen from the number of types of support nodes, it is obvious that this is a 3-way 1-shot problem. Different colors represent different categories. Nodes with solid lines represent labeled support samples, while nodes with dashed lines represent the unlabeled query sample. It is shown that some edges have broken during the graph update. The detailed process is described in Section III-B.

means dichotomous, that is, yes-or-no type rather than more-or-less type. These are called fuzzy sets, and each fuzzy set contains a universe of discourse and a membership function. The membership function measures the membership degree of element in the universe of discourse to the fuzzy set. In brief, fuzzy theory is a research approach that can deal with problems relating to ambiguous, subjective and imprecise judgments, and it can quantify the linguistic facet of available data and preferences for individual or group decision-making [25]. Actually, the fuzzy theory has been used in neural networks for many years [26]–[30]. In GNN methods for few-shot learning, the relationship between nodes is a fuzzy and subjective concept, which forces previous work to use a fully connected graph structure with little information. Inspired by the fuzzy theory, we design a membership function to map the relationship in the graph to the real unit interval [0,1] and generate a strong relational inductive bias by the membership degree.

III. Method

To make our paper more self-contained, we introduce the concept of few-shot classification task and meta-learning pattern involved in our FGNN, following related work [8], [12], [13], [31]. Then, we describe the modules and algorithm of the proposed FGNN in more detail.

A. Problem definition

The goal of few-shot classification is to classify unlabeled samples training on a few labeled samples from each class. Formally, in each classification task $\mathcal{T}$, there are two datasets: a support set S and a query set Q, which share the same label space. The support set S is a labeled set of input-label pairs and the query set Q is an unlabeled set to be predicted. N-way K-shot classification problem means that the number of classes is N and each class contains K labeled samples in the support set S.

Meta-learning has been demonstrated as an effective approach to tackle the problem of few-shot learning. The meta-learning model learns a base-learner and a meta-learner to adapt to new tasks quickly. During meta-training, the parameters of the base-learner are optimized by a training subset from a task, and then the parameters of the meta-learner are optimized by a test subset. In this paper, we adopt episodic training in MAML [13]. Since the few-shot learning here is an i.i.d. problem, the episode length of a task is set to 1.

B. Model

As shown in figure 1, our model consists of three parts: a pre-trained feature extractor, a relation encoder, and a GNN classifier. The relation encoder and the classifier are GNNs. In the first place, we train the feature extractor together with a temporary classifier on large-scale data, e.g. on the training set of miniImageNet [5] (Section III-B1). During a meta-learning phase, we designed a membership function to generate the membership degree which offers a strong inductive bias into the graph for the GNN classifier(Section III-B2).

1) Feature Extractor: As shallow neural networks will limit the effectiveness, we employ pre-trained ResNet-12 [2] to enhance the capabilities of the feature extractor. The loss $\mathcal{L}$

we use to optimize the feature extractor Θ and the classifier θ' is as follows:

$$\mathcal{L}([\Theta, \theta']) = \frac{1}{|\mathcal{D}_{train}|} \sum_{(x,y)\in\mathcal{D}_{train}} l_p(x, y; [\Theta, \theta']), \quad (1)$$

where l_p is defined as cross-entropy loss and $\mathcal{D}_{train}$ is the training split of the entire dataset $\mathcal{D}$.

After learning the feature extractor Θ, it will still be optimized with a small learning rate during the meta-learning phase.

2) Fuzzy GNN: The relation encoder outputs the relational representation by performing the edge prediction with node features. Then the membership function transforms the relational representation into a strong relational induction bias in the graph for node classification. In GNN methods for few-shot learning, only our FGNN provides a strong bias for the graph according to the output of the edge prediction.

In FGNN, the node represents each sample and the edge represents the relationship between the two connected nodes. We define $\mathcal{G} = (\mathcal{V}, \mathcal{E}; \mathcal{T})$ to be the graph on which the task $\mathcal{T}$ is learned. The $\mathcal{V} = \{V_i\}_{i=1:|\mathcal{T}|}$ is the set of nodes(of cardinality $|\mathcal{T}|$), where each v_i is the node's feature. The $\mathcal{E} = \{E_{ij}\}_{i,j=1:|\mathcal{T}|}$ is the set of edges, where each e_{ij} is the edge's feature. Let $X = \{x_i\}_{i=1:|\mathcal{T}|}$ be samples of each $\mathcal{T}$, and $Y = \{y_i\}_{i=1:|\mathcal{T}|}$ be category labels of samples. The ground truth of the edge prediction $\hat{Y} = \{\hat{y}_{ij}\}_{i,j=1:|\mathcal{T}|}$ is defined as:

$$\hat{y}_{ij} = \begin{cases} 0 & \text{if } y_i = y_j, \\ 1 & \text{if } y_i \neq y_j. \end{cases} \quad (2)$$

Relation encoder. Relation encoder is a metric network based on GNN that computes the similarity scores between samples. As the message propagates through the graph, nodes and edges can aggregate information from all nodes and edges in the graph, not just from their neighbors. As a result, the calculation of the similarity score does not depend solely on two nodes, but also on other pairs of nodes.

Different from EGNN [11], we improve their structure and transform the result of edge-label prediction to the bias for the node-focused GNNs by a designed membership function. In the edge prediction, features extracted from feature extractor are nodes' initial features. In l-th layer of relation encoder, edge features are updated firstly by the edge-update function, and the input is the feature of the nodes at both ends of the edge:

$$e_{ij}^{l+1} = f_e^l(v_i^l, v_j^l)_{i,j=1:|\mathcal{T}|}, \quad (3)$$

and then node features are updated by the node-update function,

$$v_i^{l+1} = f_v^l(\frac{\sum_j e_{ij}^{l-1} v_j^{l-1}}{\sum_j e_{ij}^{l-1}} || v_i^{l-1}), \quad (4)$$

where $||$ is the concatenation operation.

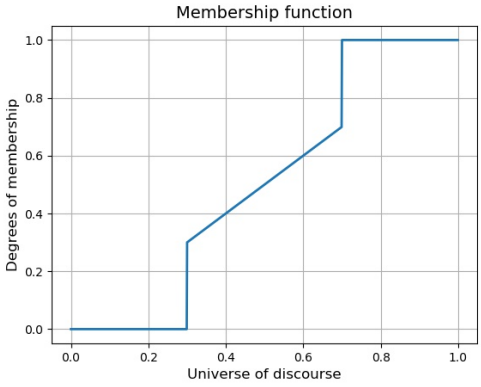

Fig. 2: The illustration for the membership function.

Relation encoder ϕ will be optimized by loss $\mathcal{L}'$ at the end of task $\mathcal{T}$,

$$\mathcal{L}'(\phi) = \frac{1}{|\mathcal{E}|} \sum_{i,j=1:|\mathcal{T}|} l_e(e_{ij}^{L_1}, \hat{y}_{ij}; \phi), \quad (5)$$

where L_1 represents the number of layers in the relation encoder. To avoid using a separate classifier, we set the output of the edge-update function of the relation encoder to an 1-dimensional vector for edge prediction.

Membership function. Theoretically, a fuzzy inference system can encompass any fuzzy aggregation strategy desired to be utilized. Herein, we focus on the membership function μ to infer the relationships predicted by the relation encoder. In general, a membership function is a mapping of data from the universe of discourse which is defined by the output of the edge-focused layers in our FGNN:

$$\mu(e_{ij}) = \begin{cases} 1 & \text{if } e_{ij} > \zeta, \\ f_\mu(e_{ij}) & \text{if } \zeta \geq e_{ij} \geq \eta, \\ 0 & \text{if } e_{ij} < \eta, \end{cases} \quad (6)$$

where ζ is larger than η.

Figure 2 shows the membership function we design for inferring the relational representations in our FGNN. Considering the outputs of the relation encoder is fuzzy, we artificially strengthen the relationship between samples that the model determines to be similar extremely and cut off the edges between two nodes with significant differences. The relationships with the similarity score between ζ and η are difficult for the relation encoder, and the score is unreliable. We employ a linear function f_μ to deal with these edges. The graph made up of these preserved edges will no longer be fully connected and have a strong relational inductive bias. Note that, the values of ζ and η affect the effect of the membership function for a long time.

GNN classifier. After transforming $\mathcal{E}$ by the membership function, it will be the adjacent matrix A of the graph for the GNN classifier θ. Different from it in the previous work [9], [11], [17], this adjacent matrix is sparse and offers a relational

inductive bias to the graph. The GNN classifier updates the node feature by a neighborhood aggregation procedure as Eq.(4). Note that node-update function in the relation encoder and GNN classifier have different parameters.

We optimize the GNN classifier θ for each task $\mathcal{T}$ before optimizing the entire model using the following empirical loss:

$$\mathcal{L}(\theta) = \frac{1}{N \times K} \sum_{i=1:|\mathcal{T}|} l_\theta(v_i^{L_2}, y_i; \theta), \tag{7}$$

where N is the number of classes, K is the number of samples for each class in N-way K-shot learning, l_θ represents cross-entropy loss and L_2 represents the number of layers in the GNN classifier. Actually, the samples in Eq.(7) are all from the support set S. The updated θ will be more suitable for present task $\mathcal{T}$.

We define $\mathcal{T}_q$ as the query set Q sampled from $\mathcal{T}$ and Y_q as the label of the query set Q. Then, we consider the Cross-entropy loss evaluated at node $*$ for all parameters:

$$\mathcal{L}([\Theta; \phi; \theta]) = -\sum_{k=1:N} y_k \log(P(Y_* = y_k|\mathcal{T}_q)). \tag{8}$$

Algorithm 1 outlines the training process of our method. In summary, FGNN has benefits in three aspects. 1) It offers a strong relational inductive bias for GNNs to obtain a more general representation. 2) It employs a membership function from the fuzzy theory to generate a more reasonable relational representation. 3) Mixed losses from edge prediction and node classification can achieve a better generalization ability to new tasks.

Algorithm 1: Fuzzy Graph Neural Network for Few-Shot Learning

Input: $\mathcal{G} = (\mathcal{V}, \mathcal{E}; \mathcal{T})$ where $\mathcal{T}$ is sampled from $p(\mathcal{T})$, dataset $\mathcal{D}$, learning rates α,β and γ, weight λ

Output: Feature extractor Θ, Relation encoder ϕ and GNN classifier θ

Randomly initialize Θ and a temporary classifier θ';
for *samples in* $\mathcal{D}$ **do**
  Evaluate $\mathcal{L}_\mathcal{D}([\Theta; \theta'])$ by Eq.(1);
  Optimize Θ and θ' by gradient descent;
  $[\Theta; \theta'] \leftarrow [\Theta; \theta'] - \alpha \nabla \mathcal{L}_\mathcal{D}([\Theta; \theta'])$;
end
Randomly initialize Relation Encoder ϕ and GNN classifier θ;
for *number of training iterations* **do**
  Sample batch of tasks $\mathcal{T}_i \sim p(\mathcal{T})$;
  while *not done* **do**
    Sample support set $S = \{(x_i, y_i)\}_{i=1:N\times K}$ and query set $Q = \{x_i\}_{i=N\times K+1:N\times(K+Q)}$ from $\mathcal{T}_i$;
    initialize graph node $v_i = f_\Theta(x_i), \forall i$;
    for $l = 1, ..., L_1$ **do**
      $\mathcal{E}^l \leftarrow EdgeUpdate(\mathcal{V}^{l-1})$;
      $\mathcal{V}^l \leftarrow NodeUpdate(\mathcal{E}^l, \mathcal{V}^{l-1})$;
    end
    Evaluate $\mathcal{L}'$ by Eq.(5);
    Transform $\mathcal{E}$ by Eq. (6);
    for *number of temporary iterations* **do**
      for $l = 1, ..., L_2$ **do**
        $\mathcal{V}^l \leftarrow NodeUpdate(\mathcal{E}, \mathcal{V}^{l-1})$;
      end
      Evaluate $\mathcal{L}(\theta)$ by Eq.(7);
      Optimize θ by gradient descent;
      $\theta \leftarrow \theta - \beta \nabla \mathcal{L}(\theta)$;
    end
    Evaluate $\mathcal{L}([\Theta; \phi; \theta])$ by Eq.(8);
    Optimize Θ,ϕ and θ by gradient descent;
    $[\Theta; \phi; \theta] \leftarrow [\Theta; \phi; \theta] - \gamma \nabla(\mathcal{L}([\Theta; \phi; \theta]) + \lambda \mathcal{L}')$;
  end
end

IV. Experiments

In this section, we firstly describe the miniImageNet dataset and the tieredImageNet dataset. Then we report experimental results to evaluate the efficacy of the proposed FGNN method in terms of the few-shot recognition accuracy and compare with other state-of-the-art methods. We also do an ablation research to explore the contribution of each module in our framework.

A. Datasets

miniImageNet. Proposed by Vinyals et al. [5], miniImagenet is the most popular few-shot learning benchmark. There are 100 classes with 600 samples of 84×84 color images per class from ImageNet ILSVRC-12 [32]. It is divided into training, validation, and test meta-sets, with 64, 16, and 20 classes respectively [12].

tieredImageNet. Similar to miniImageNet, tieredImageNet [33] is also a subset of ILSVRC-12 dataset [32], but it has a larger number of classes from ILSVRC-12. There are 608 classes with average 1281 samples of 84×84 color images per class. Note that, different from miniImageNet, tieredImageNet adopts a hierarchical category structure for broader categories corresponding to high-level nodes in ImageNet. The 34 categories belong to top hierarchy are divided into 20 training (351 classes), 6 validation (97 classes) and 8 test (160 classes) categories, which ensures that the training classes are distinct from the test classes semantically.

B. Experimental setup

Network architecture. We present the details for the feature extractor, relation encoder, and GNN classifier. Limited by the small-scale of datasets, the architecture of feature extractor Θ is ResNet-12, which is popular in recent works [8], [31], [34], [35]. ResNet-12 contains 4 residual blocks and each residual blocks consists of 3 CONV layers with 3×3 kernels. A 2×2 max-pooling layer is used to downsample feature map at the end of each residual block. As other works do, we set the number of filters to 64 at the first block and double

it every next block. Between the CNN residual block and the classifier of ResNet-12, a mean-pooling layer is employed. The relation encoder ϕ is an edge-focused graph neural network consists of edge-update function and node-update function. We employ FC layers, followed by batch normalization and a sigmoid activation function to update edges. The node-update function also consists of an FC layer, batch normalization and a LeakyReLU activation function. The GNN classifier θ is a node-focused graph neural network. Compared with the relation encoder, θ does not have a proprietary edge-update function because its edge features are provided by the membership function. Generally, a GNN model is quite shallow (2 or 3 layers) because if the model is deep with many layers, the output features will be over-smoothed and vertices from different clusters may become indistinguishable [36]. We set the the number of relation encoder layers $L_1 = 2$ and the number of the GNN classifier layers $L_2 = 1$.

Pre-train. We train ResNet-12 with all training data points and sampling task method as related works [8], [12], [13] with Adam optimizer with an initial learning rate of 10^{-1} and weight decay of 10^{-5}. We perform standard data augmentation techniques on training sets, e.g. random horizontal flips and random crops. The training process takes 100 epochs, and the learning rate is decreased in half every 30 epochs. In the ablation study, when we use 4CONV rather than ResNet-12, there is no pre-training operation because of its poor performance for large-scale data training [8].

Meta training. In order to compare with other works, we conducted 5-way 1-shot experiments and 5-way 5-shot experiments. We take 12k episodes to train the FGNN model, with each episode containing 1 or 5 supports and 15 queries from each of the 5 classes. Both the relation encoder and the GNN classifier are optimized with Adam, but the learning rate of the former is 10^{-2} and the latter is 10^{-3}. The learning rates of both ϕ and θ are reduced by 4/5 after 1k episodes. We also perform the same standard data augmentation techniques on training sets as in the pre-train step.

Ablation study. To prove the effectiveness of our approach, we designed four network structures and conducted both 5-way 1-shot experiments and 5-way 5-shot experiments. For the feature extractor, there are two options: 4CONV and pre-trained ResNet-12. In 4CONV, each convolutional block consists of 3×3 convolutions, a batch normalization [37], a LeakyReLU nonlinearity and a 2×2 max-pooling. For instance, $[\Theta 4; \theta]$ represents the model that used 4CONV and the GNN classifier but didn't use a relation encoder. To show the effectiveness of FGNN, the auxiliary task for edges become optional. To validate the importance of the fuzzy membership function, we test to make edge set $\mathcal{E}$ to a binary matrix by imposing threshold criteria for the GNN classifier and set the threshold to 0.5.

C. Results and analysis

Table I and Table II show the results of our experiment on the miniImageNet dataset and the tieredImageNet dataset. The tables are sorted according to the categories of the methods, and we compare our FGNN with other GNN methods for few-shot learning.

Models		1-shot	5-shot
Metric learning	Matching Nets [5]	43.44 ± 0.77	55.31 ± 0.73
	ProtoNets [6]	49.42 ± 0.78	68.20 ± 0.66
	Relation Net [7]	50.40 ± 0.80	65.30 ± 0.70
Memory network	SNAIL [34]	55.71 ± 0.99	68.88 ± 0.92
	TADAM [31]	58.50 ± 0.30	76.70 ± 0.30
Gradient descent	MAML [13]	48.70 ± 1.84	63.10 ± 0.92
	Qiao et al [38]	59.60 ± 0.41	73.74 ± 0.19
	LEO [39]†	61.76 ± 0.08	77.59 ± 0.12
	wDAE-MLP [17]	60.61 ± 0.15	76.56 ± 0.11
	MetaGAN [40]	52.71 ± 0.64	68.63 ± 0.67
	adaResNet [41]	56.88 ± 0.62	71.94 ± 0.57
	MTL [8]	61.20 ± 1.8	75.50 ± 0.8
GNN methods	GNN [9]	50.30	66.40
	TPN [18]	55.51	69.86
	EGNN [11]	58.98	76.37
	EGNN [11]∗	58.34	76.80
	wDAE-GNN [17]	61.07 ± 0.15	76.75 ± 0.11
	FGNN(Ours)	**64.15 ± 0.28**	**80.08 ± 0.35**

TABLE I: The 5-way, 1-shot and 5-shot Classification results on miniImageNet dataset on 5-way setting. The top results are highlighted. †: using also the validation classes for training. ∗: we implemented using pre-trained feature extractor.

Models		1-shot	5-shot
Metric learning	ProtoNets [6]	53.31 ± 0.89	72.69 ± 0.74
	Relation Net [7]	54.48 ± 0.93	71.32 ± 0.78
Gradient descent	MAML [13]	51.67 ± 1.81	70.30 ± 0.08
	Meta-SGD [42]	62.95 ± 0.03	79.34 ± 0.06
	Dynamic [43]	50.90 ± 0.46	66.69 ± 0.36
	LEO [39]†	66.33 ± 0.0	81.44 ± 0.09
Memory network	Incremental [44]‡	51.12 ± 0.45	66.40 ± 0.36
GNN methods	TPN [18]	59.91	73.30
	EGNN [11]	58.98	80.15
	EGNN [11]∗	59.30	80.22
	FGNN(Ours)	**69.09 ± 0.15**	**84.13 ± 0.18**

TABLE II: The 5-way, 1-shot and 5-shot Classification results on tieredImageNet dataset on 5-way setting. The top results are highlighted. †: using also the validation classes for training. ‡: replace all batch normalization layers with group normalization. ∗: we implemented using pre-trained feature extractor.

Result on miniImageNet. In Table I, it is shown that our FGNN achieves the best performance with 64.15% for 5-way 1-shot learning on miniImageNet. On the 5-way 5-shot experiment, our model also achieves 80.08% accuracy and ranked first. Among these graph network methods, our FGNN is also the best performer. On 5-way 1-shot and 5-way 5-shot experiments, the accuracy for FGNN is 3.08% and 3.33% more than wDAE-GNN, the best GNN method for few-shot learning previously. One thing to explain, wDAE-GNN designed another experiment that used validation classes for training and achieved accuracy similar to ours. But in order to be fair, we choose the results of training using the training set only.

Result on tieredImageNet. For tieredImageNet, as in Table II, we also achieve the best results on both 5-way 1-shot and 5-way 5-shot experiments. The results of EGNN on 1-shot experiments were not reported in their paper [11]. We get the result using their public code of EGNN on the Github website.

Model	Pre-train	Edge prediction	1-shot	5-shot
$[\Theta 4;\theta]$	No	No	55.73	72.30
$[\Theta 4;\phi;\theta]$	No	Yes	59.89	74.31
$[\Theta;\theta]$	Yes	No	59.50	74.07
$[\Theta;\phi;\theta]$(Threshold)	Yes	Yes	60.21	76.50
$[\Theta;\phi;\theta]$	Yes	Yes	64.15	80.08

TABLE III: The 5-way, 1-shot and 5-shot Classification results on miniImageNet dataset on 5-way setting using ablative models. $\Theta 4$ means that the model used 4CONV as the feature extractor.

Although the gap between the training set and test set in the tieredImageNet is deeper than in miniImageNet, the model achieved a better performance due to the more available data. We can see that FGNN consistently outperforms EGNN by large margins. Especially in the 1-shot experiment, the gap reached 10.06%. And for 5-shot, FGNN surpasses EGNN by around 3.91%.

The performance of FGNN with full components, membership function, edge prediction and pre-trained ResNet-12, is the best in all few-shot learning methods on both datasets, see Table I and Table II. We can conclude that our GNNs with a strong relational inductive bias significantly boost the few-shot learning performance. Note that our FGNN performed better than other GNN methods, which supports the effectiveness of our framework for few-shot learning. In [9], nodes communicate to each other only via their embedding feature similarities which proved to be meaningless in preliminary experiments. To generate a more reasonable graph structure, TPN [18] used a Laplacian matrix rather than feature similarities to propagate labels of the support set to the query set. In EGNN [11], they propagate to each other not only their node features but also edge-label information across to consider more complicated interactions between query samples. However, updating edge feature iteratively with the state at the previous time may make the model suffer from noise during the whole update process. In contrast, our FGNN allows us to consider a more reasonable graph structure by optimizing the relational representations with a membership function, and the relational representation that is, the membership degree will be frozen for node classification to exchange more relevant information between neighbors.

Result on ablation experiment. The difference between FGNN and the previous GNN method is mainly in two aspects: pre-trained feature extractor and the strong relational inductive bias from the membership function. Table III indicates the impact of these two parts on the results. Comparing the results of $[\Theta 4;\theta]$ and $[\Theta;\theta]$, we found that pre-trained ResNet-12 gives around 4% improvement to the model. FGNN with a strong relational inductive bias $[\Theta;\phi;\theta]$ surpasses the model updating in fully connected structure $[\Theta;\theta]$ by a relatively larger number of 4% for 5-shot and with 6% for 1-shot on miniImageNet. In particular, the performance of the model using binary matrix rather than a membership function is only a little better than that of $[\Theta;\theta]$. The effectiveness of model $[\Theta;\theta]$ and $[\Theta;\phi;\theta]$ is shown in Figure 3. As the

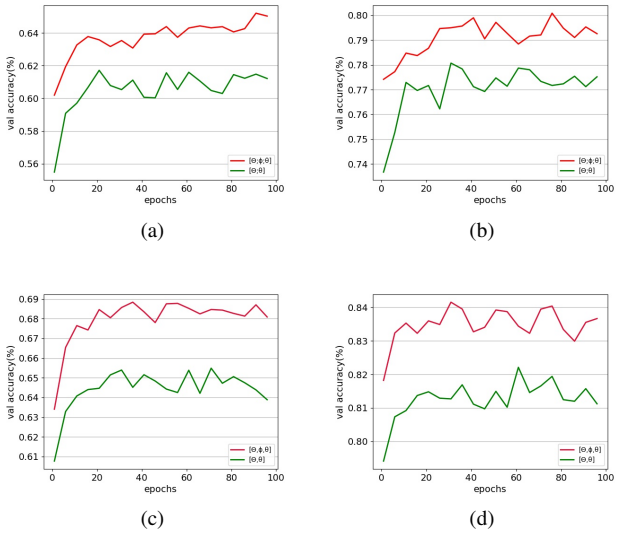

Fig. 3: (a)(b)show the results of 1-shot and 5-shot on miniImageNet; (c)(d) show the results of 1-shot and 5-shot on tieredImageNet. The only difference between the two models is in the relational inductive bias.

test is invisible to the model during training, the accuracy in Figure 3 is evaluated on the validation set. At the first epoch, $[\Theta;\phi;\theta]$ surpasses $[\Theta;\theta]$ by around 5%, which means that the membership function can successfully map relational representations to a more reasonable distribution, even if the representations are learned preliminarily. It is interesting to note that the gap between the results of these two models become larger on tieredImageNet. The possible reason is that there are more categories in tieredImageNet than in miniImageNet. The relationships between categories can be learned better, which makes membership function more effective.

V. Conclusion

In this paper, we propose a novel FGNN with a strong relational inductive bias to tackle few-shot classification problems. We incorporate the fuzzy theory into GNNs and design a membership function to generate a more reasonable graph structure from the samples with ambiguous relationships. On the task-specific graph structure, FGNN can learn more useful task-relevant features, which ensures the highly efficient for learning unseen tasks. The superiority was particularly achieved in both 1-shot and 5-shot tasks on two challenging benchmarks - miniImageNet and tieredImageNet. In addition, we believe that it also provides a new way to construct graphs from fuzzy relationships in spatial-based GNNs.

Acknowledgment

This work was supported by Military Key Research Foundation Project (No.AWS15J005), Shanghai Science and Technology Development Funds (19DZ2205700), National Natural Science Foundation of China (No.61672165 and No.61732004)

REFERENCES

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[3] L. Fei-Fei, R. Fergus, and P. Perona, "One-shot learning of object categories," *IEEE transactions on pattern analysis and machine intelligence*, vol. 28, no. 4, pp. 594–611, 2006.

[4] G. Koch, R. Zemel, and R. Salakhutdinov, "Siamese neural networks for one-shot image recognition," in *ICML deep learning workshop*, vol. 2, 2015.

[5] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra *et al.*, "Matching networks for one shot learning," in *Advances in neural information processing systems*, 2016, pp. 3630–3638.

[6] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Advances in Neural Information Processing Systems*, 2017, pp. 4077–4087.

[7] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 1199–1208.

[8] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 403–412.

[9] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," *arXiv preprint arXiv:1711.04043*, 2017.

[10] H. Li, D. Eigen, S. Dodge, M. Zeiler, and X. Wang, "Finding task-relevant features for few-shot learning by category traversal," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 1–10.

[11] J. Kim, T. Kim, S. Kim, and C. D. Yoo, "Edge-labeling graph neural network for few-shot learning," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 11–20.

[12] S. Ravi and H. Larochelle, "Optimization as a model for few-shot learning," 2016.

[13] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1126–1135.

[14] J. Schmidhuber, "Evolutionary principles in self-referential learning," *On learning how to learn: The meta-meta-... hook.) Diploma thesis, Institut f. Informatik, Tech. Univ. Munich*, vol. 1, p. 2, 1987.

[15] Q. Cai, Y. Pan, T. Yao, C. Yan, and T. Mei, "Memory matching networks for one-shot image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4080–4088.

[16] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2. IEEE, 2005, pp. 729–734.

[17] S. Gidaris and N. Komodakis, "Generating classification weights with gnn denoising autoencoders for few-shot learning," *arXiv preprint arXiv:1905.01102*, 2019.

[18] Y. Liu, J. Lee, M. Park, S. Kim, E. Yang, S. J. Hwang, and Y. Yang, "Learning to propagate labels: Transductive propagation network for few-shot learning," *arXiv preprint arXiv:1805.10002*, 2018.

[19] T. N. Kipf and M. Welling, "Semi-supervised classification with graph convolutional networks," *arXiv preprint arXiv:1609.02907*, 2016.

[20] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, "Neural message passing for quantum chemistry," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017, pp. 1263–1272.

[21] Y. Li, D. Tarlow, M. Brockschmidt, and R. Zemel, "Gated graph sequence neural networks," *arXiv preprint arXiv:1511.05493*, 2015.

[22] H. Dai, Z. Kozareva, B. Dai, A. Smola, and L. Song, "Learning steady-states of iterative algorithms over graphs," in *International Conference on Machine Learning*, 2018, pp. 1114–1122.

[23] J. Zhou, G. Cui, Z. Zhang, C. Yang, Z. Liu, L. Wang, C. Li, and M. Sun, "Graph neural networks: A review of methods and applications," *arXiv preprint arXiv:1812.08434*, 2018.

[24] L. A. Zadeh, "Fuzzy set theory," *Information and control 8th*, pp. 338–353, 1965.

[25] H.-C. Liu, J.-X. You, M.-M. Shan, and L.-N. Shao, "Failure mode and effects analysis using intuitionistic fuzzy hybrid topsis approach," *Soft Computing*, vol. 19, no. 4, pp. 1085–1098, 2015.

[26] J. J. Buckley and Y. Hayashi, "Fuzzy neural networks: A survey," *Fuzzy sets and systems*, vol. 66, no. 1, pp. 1–13, 1994.

[27] D. T. Anderson, G. J. Scott, M. A. Islam, B. Murray, and R. Marcum, "Fuzzy choquet integration of deep convolutional neural networks for remote sensing," in *Computational Intelligence for Pattern Recognition*. Springer, 2018, pp. 1–28.

[28] J. M. Keller, D. Liu, and D. B. Fogel, *Fundamentals of computational intelligence: neural networks, fuzzy systems, and evolutionary computation*. John Wiley & Sons, 2016.

[29] J. Fei and T. Wang, "Adaptive fuzzy-neural-network based on rbfnn control for active power filter," *International Journal of Machine Learning and Cybernetics*, vol. 10, no. 5, pp. 1139–1150, 2019.

[30] S. Hou, J. Fei, C. Chen, and Y. Chu, "Finite-time adaptive fuzzy-neural-network control of active power filter," *IEEE Transactions on Power Electronics*, vol. 34, no. 10, pp. 10 298–10 313, 2019.

[31] B. Oreshkin, P. R. López, and A. Lacoste, "Tadam: Task dependent adaptive metric for improved few-shot learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 721–731.

[32] O. Russakovsky, J. Deng, H. Su, J. Krause, S. Satheesh, S. Ma, Z. Huang, A. Karpathy, A. Khosla, M. Bernstein *et al.*, "Imagenet large scale visual recognition challenge," *International journal of computer vision*, vol. 115, no. 3, pp. 211–252, 2015.

[33] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J. B. Tenenbaum, H. Larochelle, and R. S. Zemel, "Meta-learning for semi-supervised few-shot classification," *arXiv preprint arXiv:1803.00676*, 2018.

[34] N. Mishra, M. Rohaninejad, X. Chen, and P. Abbeel, "A simple neural attentive meta-learner," *arXiv preprint arXiv:1707.03141*, 2017.

[35] L. Franceschi, P. Frasconi, S. Salzo, R. Grazzi, and M. Pontil, "Bilevel programming for hyperparameter optimization and meta-learning," *arXiv preprint arXiv:1806.04910*, 2018.

[36] Q. Li, Z. Han, and X.-M. Wu, "Deeper insights into graph convolutional networks for semi-supervised learning," in *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[37] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," *arXiv preprint arXiv:1502.03167*, 2015.

[38] S. Qiao, C. Liu, W. Shen, and A. L. Yuille, "Few-shot image recognition by predicting parameters from activations," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 7229–7238.

[39] A. A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, and R. Hadsell, "Meta-learning with latent embedding optimization," *arXiv preprint arXiv:1807.05960*, 2018.

[40] R. Zhang, T. Che, Z. Ghahramani, Y. Bengio, and Y. Song, "Metagan: An adversarial approach to few-shot learning," in *Advances in Neural Information Processing Systems*, 2018, pp. 2365–2374.

[41] T. Munkhdalai, X. Yuan, S. Mehri, and A. Trischler, "Rapid adaptation with conditionally shifted neurons," *arXiv preprint arXiv:1712.09926*, 2017.

[42] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-sgd: Learning to learn quickly for few-shot learning," *arXiv preprint arXiv:1707.09835*, 2017.

[43] S. Gidaris and N. Komodakis, "Dynamic few-shot visual learning without forgetting," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4367–4375.

[44] M. Ren, R. Liao, E. Fetaya, and R. S. Zemel, "Incremental few-shot learning with attention attractor networks," *arXiv preprint arXiv:1810.07218*, 2018.