Learn from Concepts: Towards the Purified Memory for Few-shot Learning

Xuncheng Liu[1*] , **Xudong Tian**[1*] , **Shaohui Lin**[1] , **Yanyun Qu**[2] ,
Lizhuang Ma[1] , **Wang Yuan**[1] , **Zhizhong Zhang**[1 †] and **Yuan Xie**[1†]
[1]School of Computer Science and Technology, East China Normal University, China
[2]School of Information Science and Engineering, Xiamen University, China
{51194501055, 51194501066, 51184501076}@stu.ecnu.edu.cn,
{shlin, lzma, zzzhang}@cs.ecnu.edu.cn,
yyqu@xmu.edu.cn, xieyuan8589@foxmail.com

Abstract

Human beings have a great generalization ability to recognize a novel category by only seeing a few number of samples. This is because humans possess the ability to learn from the concepts that already exist in our minds. However, many existing few-shot approaches fail in addressing such a fundamental problem, *i.e.,* how to utilize the knowledge learned in the past to improve the prediction for the new task. In this paper, we present a novel purified memory mechanism that simulates the recognition process of human beings. This new memory updating scheme enables the model to purify the information from semantic labels and progressively learn consistent, stable, and expressive concepts when episodes are trained one by one. On its basis, a Graph Augmentation Module (GAM) is introduced to aggregate these concepts and knowledge learned from new tasks via a graph neural network, making the prediction more accurate. Generally, our approach is model-agnostic and computing efficient with negligible memory cost. Extensive experiments performed on several benchmarks demonstrate the proposed method can consistently outperform a vast number of state-of-the-art few-shot learning methods.

1 Introduction

The success of deep learning stems from a large amount of labeled data [Noh *et al.*, 2017; Bertinetto *et al.*, 2016; Long *et al.*, 2015], while humans have a good generalization ability by only seeing a few number of samples. The gap between the two facts brings great attention to the research of few-shot learning [Vinyals *et al.*, 2016; Finn *et al.*, 2017; Sung *et al.*, 2018]. Unlike traditional deep learning scenario, few-shot learning is not to classify unseen samples, but fast adapts the meta-knowledge to new tasks, where only very few labeled data and knowledge gained from previous experience are given.

Recently, significant advantages [Vinyals *et al.*, 2016; Finn *et al.*, 2017; Snell *et al.*, 2017; Sung *et al.*, 2018] have been made to tackle this problem by using the idea of meta-learning coupled with episodic training [Vinyals *et al.*, 2016]. The intuition is to use a episodic sampling strategy, a promising trend to transfer knowledge from known categories (*i.e.,* seen categories with sufficient training examples) to new categories (*i.e.,* novel categories with few examples), simulating the human learning process. In this framework, metric-based approaches [Vinyals *et al.*, 2016; Snell *et al.*, 2017] and graph-based approaches [Garcia and Bruna, 2017; Liu *et al.*, 2018; Kim *et al.*, 2019; Yang *et al.*, 2020] are two representative methods by taking the primary advantage of the transferable meta-knowledge. Due to the ability to learn from graph data efficiently, graph-based approaches generally outperform metric-based method, which extends the pairwise query-support relationship to a graph structure.

In spite of the effectiveness of graph-based approaches [Kim *et al.*, 2019; Yang *et al.*, 2020], most of them ignore a critical issue that how the knowledge learned in the past could be useful for the new task when the episodes are trained one by one. As a kind of intuition, for an unseen task, human beings don't use the whole knowledge, but a few informative and relevant concepts, to improve the prediction ability for the new task. For example, if a man has learned the concepts of horses, tigers and pandas, it is easy to recognize zebras by finding zebras have the outline like horses, stripes like tigers, and black and white color like pandas. Motivated by this simple intuition, we propose an assumption that a few-shot learning model should explicitly establish the relationship between episodes and take full use of existing learned knowledge.

However, it raises two fundamental problems hindering existing graph-based approaches: 1) how to learn a stable and consistent concepts when episodes are rapidly coming; 2) how the learned concepts further help the prediction when adapting to new tasks. In this paper, we propose a purified memory framework to tackle these two problems. Our basic idea is simply that simulated the recognition process of human beings. To keep stable and consistent concepts, we hold a memory bank during episodic training, which learns an optimal prototype representation for each category from the perspective of information bottleneck principle [Tishby and Zaslavsky, 2015]. By progressively purifying the information from semantic label, the stored knowledge is supposed to be generally expressive, consistent and stable.

[*]Equal Contribution
[†]Contact Author

To make full use of the purified memory, we present a Graph Augmentation Module (GAM) as a way of mining meta-knowledge and establishing the correlation between different episodes. When dealing with a new task, GAM first retrieves the concepts of k-nearest neighbors by taking class center from the current task as query. Then retrieved concepts and episodic training samples are forwarded into a graph neural network (GNNs) with an adaptive weighting scheme. Thus the concepts learned in the past and knowledge learned from the new task are aggregated, which allow our model to make accurate prediction. It is worth noting our method is a model-agnostic approach and could be integrated into any advanced GNNs method flexibly with negligible computational cost.

Our major contributions are three-fold: (1) we present a new memory purifying mechanism with efficiency, consistency and vigorous expressive power; (2) the proposed GAM is able to mine the meta-knowledge and capture the correlation between different episodes; (3) our approach yields state-of-the-art few-shot results and our intriguing findings highlight the need to rethink the way we use meta-knowledge.

2 Method

This paper aims to address the problem of few-shot classification. The problem definition is fundamentally different from traditional classification, whose objective is not classify unseen samples but to fast adapt the meta-knowledge to new tasks. Specifically, a labeled dataset with sufficient training samples from base classes C^{base} is provided, and the goal is to learn the concepts with very limited data collected from a set of novel classes C^{novel}, where $C^{base} \cap C^{novel} = \emptyset$. An effective way to solve the few-shot problem is to use the episodic sampling strategy. In this framework, the samples in meta-training and meta-testing are not samples but episodes $\{\mathcal{T}\}$, each of which contains N classes (ways) and K shot per class. In particular, for a N-way K-shot task, a support set $S = \{(x_i, y_i)\}_{i=1}^{N\times K}$ and a query set $Q = \{(x_i, y_i)\}_{i=N\times K+1}^{N\times K+T}$ are sampled. Here, x_i and $y_i \in \{C_1, \cdots C_N\}$ are the i-th input data and is from C^{base}. In the meta-test, a test task is also sampled with the same sized episode from unseen categories C^{novel}. The aim is to classify T unlabeled samples in query set into N classes correctly.

2.1 Overview of Framework

The framework of the proposed method is illustrated in Fig. 1. It mainly consists of three components, *i.e.,* an encoder for discriminative feature extraction, a memory module for expressive meta-knowledge storage, a graph augmentation module for comprehensive inference. In general, our approach can be summarized into 3 stages (*i.e.,* Pre-Train, Meta-Train, Meta-Test).

Phase-I Pre-Train. We follow a simple baseline [Chen *et al.*, 2020]: learning a supervised representation on the meta-training set C^{base}, followed a linear classifier on top of this representation. It has been shown that this pre-train stage is beneficial for the downstream few-shot task [Tian *et al.*, 2020], and the trained feature extractor (*e.g.,* ResNet-12[He *et al.*, 2016]) and classifier are then used as the initialization of our encoder and memory bank, respectively.

Phase-II Meta-Train. We first extract the features of support and query samples as task-relevant embeddings V^t. Then to facilitate fast adaption, our approach holds a memory bank to store the expressive representations of the support set. This memory bank is optimized with a new updating scheme to progressively purify discriminative information (introduced in Sec. 2.2). Further, the purified memory is incorporated with a graph augmentation module for robust prediction (introduced in Sec. 2.3). In this module, we mine the relevant prototypes V^m, referred as meta-knowledge in this paper, to propagate the similarities between V^t and V^m via a graph neural network. Consequently, our model is able to generalize to new tasks conveniently with negligible memory cost.

Phase-III Meta-Test. The procedure of Meta-Test is similar with Meta-Train, where the episodic sampling strategy is also adopted. But unlike Phase-II, the memory bank and other modules will not be updated throughout the process. In other words, switch will be closed as shown in Figure 1.

2.2 Refined Memory Updating

Meta-knowledge plays an import role in learning new concepts from unseen samples, and recent FSL advances [Ramalho and Garnelo, 2019] often exploit a memory mechanism to store this meta-knowledge. In its typical setting, the memory tries to preserve as much information as possible (*e.g.,* store the whole features). However, we argue this strategy is both ineffective and inefficient. In the context of FSL, the episodic sampling makes the feature extractor rapidly learn new concept with very few samples, and this causes a problem that the feature in memory is updated when the feature extractor is under a very different task context. From this perspective, the representation learned from different tasks requires a purification process to be a stable concept.

To alleviate the above issues, we propose to refine the memory via learning an optimal prototype for each category. Specifically, considering a N-way K-shot task in FSL, we use $f_{sup}^{l} \in \mathbb{R}^{[N\times K,d]}$ to denote the feature representations of the support set in the l-th episode, and $\mathbb{M} \in \mathbb{R}^{[C,d]}$ to denote the memory bank, where C and d indicate the total number of categories and the dimension of the prototype, respectively.

To progressively purify semantic information from labels, we firstly conduct category-wise averaging to f_{sup}^{l} to obtain the centroids $f_{cen}^{l} \in \mathbb{R}^{[c,d]}$, each of which is then concatenated with the prototype $f_{p}^{l} \in \mathbb{R}^{[c,d]}$ (stored in the memory) that belongs to the same category. We forward the concatenation $f_{cat}^{l} \in \mathbb{R}^{[c,2\times d]}$ to a fully-connected layer to reduce the dimension, and utilize the output to refine the memory. Here we propose to use the information bottleneck principle to purify the concept. The following constraint is used to ensure the IB to be well working, *i.e.,* preserve semantic label information while avoiding task-irrelevant nuisances.

$$\max I(f_p^l; Y) - \beta I(f_{cat}^l; f_p^l), \tag{1}$$

where $I(.;.)$ denotes the mutual information, Y represents the label, and β is the Lagrange coefficient, respectively.

Specifically, Eq. (1) aims to learn prototype f_p^l that is maximally informative about the target Y while being maximally compressive about f_{cat}^l. However, Eq. (1) requires estimating

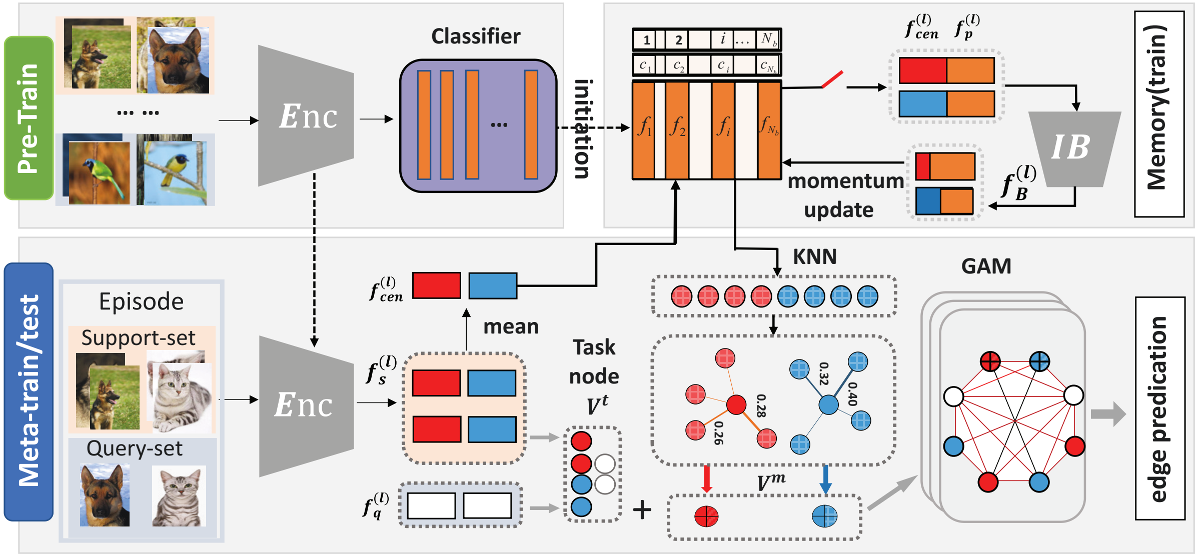

Figure 1: The flowchart of proposed method. We take 2-way 2-shot setting as an example.

mutual information in high dimension, which is intractable in such a high-dimension space. Fortunately, since our goal is to purify the concept, we show a self knowledge distillation loss could be strictly consistent with Eq. (1), where the mathematical deduction is shown in supplementary materials.

In practice, the following constraint is enforced to purify the discriminative information and further refine the memory:

$$\mathcal{L}_r = \mathbb{E}_{f_{cat}^l}\left[\mathbb{E}_{z\sim\phi(z|f_{cat}^l)}\left[D_{KL}[p(y|f_{cat}^l)||p(y|f_p^l)]\right]\right]. \quad (2)$$

Here, θ and ϕ denote the parameters of the encoder and the FC layer, $D_{KL}[.||.]$ represents the KL-divergence, y denotes the label. Note that both $p(y|f_{cat}^l)$ and $p(y|f_p^l)$ denote conditional distribution, and, in practice, are the output of an extra linear linear (detailed descriptions are shown in supplementary materials).

The refining of $\mathbb{M}$ is, in essence, iteratively aggregating discriminative information and diluting task-irrelevant nuisances. A naive solution is to append the output of the IB to $\mathbb{M}$ for every episode. But this solution leads to tremendous spatial and time cost and yields poor performance (see Sec. 3.4). In the view of above, we propose to refine the memory bank by momentum update. Formally, $\mathbb{M}$ is updated by:

$$f_p^l \leftarrow \lambda f_p^l + (1-\lambda) f_B^l, \quad (3)$$

where $\lambda \in [0,1)$ is a momentum coefficient and $f_B^l \in \mathbb{R}^d$ denotes the output of the IB at current episode.

In this way, the memory is supposed to be generally expressive, consistent and much more efficient. The refined prototype representations further incorporate and aggregate with meta-knowledge mining and is applied to facilitate the inference of FSL, as described next.

2.3 Graph Augmentation Module

For an unseen task, human beings don't use the whole knowledge, but a few informative and relevant concepts, to abstract the new task. Motivated by this, we propose a Meta-knowledge Mining approach to simulate this behavior. The core idea behind our approach is to aggregate similar features, rather than the entire memory bank, to help our model learn new concepts for an unseen task. In particular, we use a graph augmentation module (GAM) to capture the relationship between a specific task context and relevant concepts. Their similarities are then propagated through a graph neural network [Kim *et al.*, 2019], in which each layer performs node feature and edge feature update, to realize fast and comprehensive inference.

Meta-knowledge Mining. For each class centroid $f_{cen}^l[i]$ in l-th episode, we first compute the cosine similarities between $f_{cen}^l[i]$ and each prototype in the memory $\mathbb{M}$. Then we select k-nearest-neighbors of $f_{cen}^l[i]$, which are denoted as $MK = \{m_1, m_2, ..., m_k\}$. In order to perform the aggregation, we use an attention coefficient calculated by the centroid $f_{cen}^l[i]$ and selected embeddings m_j:

$$a_j = \frac{exp(\tau \cdot \langle f_{cen}^l[i], m_j \rangle)}{\sum_{\hat{j}} exp(\tau \cdot \left\langle f_{cen}^l[i], m_{\hat{j}} \right\rangle)}, \quad (4)$$

where $\langle \cdot, \cdot \rangle$ denotes the cosine similarity between two vectors and τ is a scalar parameter. Finally, the meta-knowledge node of each class is calculated as:

$$v_i^m = \sum\nolimits_{j=1}^{k} a_j f_{agg}\left(\left[m_j; f_{cen}^l[i]\right]; \theta_{agg}\right), \quad (5)$$

where $[\cdot;\cdot]$ is the concatenation operation and $f_{agg}(\cdot;\theta_{agg})$ performs a transformation : $\mathbb{R}^{2d} \rightarrow \mathbb{R}^d$ on the concatenated features which is composed of a fully-connected layer, with the parameter set θ_{agg}.

Augmented Graph Initialization. For a N-way K-Shot task, given the features extracted from the encoder and the mined meta-knowledge, a fully connected graph $G = (V, E)$ is constructed, where $V = \{v_i^t\}_{i=1}^{N\times K+T} \cup \{v_i^m\}_{i=1}^{N} =$

$\{v_i\}_{i=1}^{N\times(K+1)}$ and $E = \{e_{ij}\}_{i,j=1,\ldots,|V|}$ denote the set of nodes and edges, respectively. The node contains two types of points *i.e.,* task-relevant nodes V^t and meta-knowledge nodes V^m. The edge represents the similarity between two nodes and is initialized as:

$$\mathbf{e}_{ij}^0 = \begin{cases} 1, & if\ y_i = y_j\ and\ v_i, v_j \in \hat{S}, \\ 0, & if\ y_i \neq y_j\ and\ v_i, v_j \in \hat{S}, \\ 0.5, & otherwise, \end{cases} \tag{6}$$

where $\hat{S} = S \cup V^m$ denotes the union of support set and augmented meta-knowledge. Thus the meta-knowledge is augmented to existing inference task and allow the model to adapt to new task by taking advantage of learned concepts.

Node Feature Update. Given $v_i^{\ell-1}$ and $\mathbf{e}_{ij}^{\ell-1}$ from layer $\ell-1$, the feature node v_i^ℓ at layer ℓ is updated by a neighborhood aggregation procedure. This aggregation is weighted by the edge similarity between two neighbors. A feature transformation is also conducted to normalize the feature. Mathematically, the node feature update is defined as:

$$v_i^\ell = f_{node}([v_i^{\ell-1}, \sum_{j=1, j\neq i}^{|V|} v_i^{\ell-1} e_{ij}^{\ell-1}]; \theta_{node}), \tag{7}$$

where $[\cdot;\cdot]$ is the concatenation operation and $f_{node}(\cdot;\theta_{node})$ is a transformation block consisting of two convolutional layers[Glorot *et al.*, 2011; Ioffe and Szegedy, 2015], a LeakyReLU activation and a dropout layer.

Edge Feature Update. Edge feature update is done based on the newly updated node features v_i^ℓ. The similarities between every pair of nodes are re-calculated, and the feature of each edge e_{ij}^ℓ is updated by combining the previous edge feature value $\mathbf{e}_{ij}^{\ell-1}$ and the updated similarities as:

$$e_{ij}^\ell = \frac{f_{edge}(\|v_i - v_j\|; \theta_{edge}) e_{ij}^{\ell-1}}{\sum_k f_{edge}(\|v_i - v_j\|; \theta_{edge}) e_{ik}^{\ell-1} / \sum_k e_{ik}^{\ell-1}}, \tag{8}$$

where $f_{edge}(\cdot;\theta_{edge})$ is a metric network parameterized by θ_{edge}, which includes four convolutional blocks, a batch normalization layer, a LeakyReLu activation and a dropout layer. It is worth noting our GAM can be implemented with any other GNN, and substantially improve their performance.

2.4 Prediction and Optimization

When the optimization is complicated, the predicted probability of a node v_i belonging to C_k can be denoted as:

$$P_i^k = \sum_{j\neq i \wedge (x_i, y_i)\in\hat{S}} e_{ij}^L \delta(y_j = C_k), \tag{9}$$

where $\delta(y_j = C_k)$ is the Kronecker delta function that is equal to one when $y_j = C_k$ and zero otherwise, e_{ij} denotes the edge feature between two nodes v_i and v_j. A softmax layer is then used to normalize this probability.

During the meta-training stage, our model is optimized by minimizing the binary cross-entropy loss (BCE):

$$\mathcal{L}_q = \sum_{\ell=1}^{L} \lambda_\ell \sum_{i=1}^{T} BCE(e_i, \hat{y}_i^\ell), \tag{10}$$

where e_i and $\hat{y}_i^\ell$ are the ground-truth of query node edge-label and the query-edge predictions, respectively, and λ_ℓ is the coefficient for ℓ-th layer. In order to make the meta-knowledge nodes consistent with the predicted label, we also introduce another binary cross-entropy loss (BCE) $\mathcal{L}_m$ to estimate the discrepancy between the ground-truth and the predictions of meta-knowledge nodes edge-label.

Finally, the total loss $\mathcal{L}$ can be defined as:

$$\mathcal{L} = \mathcal{L}_q + \alpha\mathcal{L}_m + \beta\mathcal{L}_r, \tag{11}$$

where α is the coefficient to balance $\mathcal{L}_q$ and $\mathcal{L}_m$. In our experiments, we fix $\alpha = 0.2$ and $\beta = 0.01$.

3 Experiments

3.1 Experimental Setup

Datasets. We evaluate our approach on four few-shot learning benchmarks followed by [Yang *et al.*, 2020]: miniImageNet [Vinyals *et al.*, 2016], tieredImageNet [Ren *et al.*, 2018], CUB-200-2011[Wah *et al.*, 2011] and CIFAR-FS [Bertinetto *et al.*, 2018]. Among them, miniImageNet and tieredImageNet are collected from ImageNet, and CIFAR-FS is a subset from CIFAR-100. Unlike these datasets, CUB-200-2011 is a fine-grained bird classification dataset.

Evaluation. For evaluation, all the results are obtained under standard few-shot classification protocol: 5-way 1-shot and 5-shot task. No matter in 1 or 5-shot setting, only 1 query sample each class is used to test the accuracy. We report the mean accuracy (%) of 10K randomly generated episodes as well as the 95% intervals on test set. Notice that all the hyper-parameters are determined from the validation sets.

3.2 Implementation Details

Network Architecture. We utilize two networks as our encoder backbone,(*i.e.,* ConvNet and Resnet12 [Kim *et al.*, 2019; Lee *et al.*, 2019]). ConvNet contains four blocks, and each block includes a 3x3 convolutional layer, a batch normalization layer and a LeakyReLU activation. Similarly, ResNet12 consists of four residual blocks. Please refer to [He *et al.*, 2016] a comprehensive understanding. After the backbone network, there is a global average pooling layer and a fully-connected layer to produce 128-dimensional instance embeddings.

Training. In the pre-training stage, the baseline following prior work [Chen *et al.*, 2020] is trained from scratch with a batch size of 128 by minimizing the standard cross-entropy loss on base classes. After that, we randomly select 40 episodes per iteration for training the ConvNet in the meta-train stage. This sampling strategy is slightly different for ResNet12, where 5-way 5-shot task, we only sample 20 episodes per iteration due to the memory cost. The Adam optimizer is used in all experiments with the initial learning rate of 10^{-3}. We decay the learning rate by 0.1 per 8000 iterations and set the weight dacay to 10^{-5}. We train 50,000 epochs in total, and the encoder are frozen for the first 25000 iterations.

3.3 Main Results

In this section, we demonstrate the effectiveness of our approach against state-of-the-arts methods. For a fair compari-

			miniImageNet 5-way		tieredImageNet 5-way	
Model	**Backbone**	**Venue**	**1-shot**	**5-shot**	**1-shot**	**5-shot**
ProtoNet [Snell *et al.*, 2017]	ConvNet	NeurIPS'17	49.42 ± 0.78	68.20 ± 0.66	53.31 ± 0.89	72.69 ± 0.74
EGNN [Kim *et al.*, 2019]	ConvNet	CVPR'19	58.65 ± 0.55	76.34 ± 0.48	62.76 ± 0.52	80.24 ± 0.49
Ours (EGNN)	**ConvNet**	**Ours**	**63.82±0.53**	**78.58±0.43**	**66.67±0.51**	**82.27±0.43**
Meta-Baseline [Chen *et al.*, 2020]	ResNet12	ArXiv'20	63.17 ± 0.23	79.26 ± 0.17	68.62 ± 0.27	83.29 ± 0.18
Distill [Tian *et al.*, 2020]	ResNet12	ECCV'20	64.82 ± 0.60	82.14 ± 0.43	71.52 ± 0.69	86.03 ± 0.49
Neg-Cosine [Liu *et al.*, 2020]	ResNet12	ECCV'20	63.85 ± 0.81	81.57 ± 0.56	-	-
CBM [Wang *et al.*, 2020]	ResNet12	MM'20	64.77 ± 0.46	80.50 ± 0.33	71.27 ± 0.50	85.81 ± 0.34
DPGN [Yang *et al.*, 2020]	ResNet12	CVPR'20	67.77±0.32	84.60±0.43	72.45±0.51	87.24±0.39
EGNN$^+$[Kim *et al.*, 2019]	ResNet12	CVPR'19	60.27 ± 0.48	77.50 ± 0.44	64.58 ± 0.51	81.66 ± 0.48
Ours (EGNN)	**ResNet12**	**Ours**	**65.87±0.49**	**82.23±0.40**	**71.69±0.49**	**84.43±0.39**
DPGN$^+$ [Yang *et al.*, 2020]	ResNet12	CVPR'20	67.08 ± 0.48	84.28 ± 0.44	70.88 ± 0.49	85.87 ± 0.41
Ours (DPGN)	**ResNet12**	**Ours**	**69.19±0.53**	**85.87±0.41**	**74.81±0.51**	**88.14±0.39**
			CIFAR-FS 5-way		**CUB-200-2011 5-way**	
MAML [Finn *et al.*, 2017]	ConvNet	ICML'17	58.9 ± 1.9	71.5 ± 1.0	55.92 ± 0.95	72.09 ± 0.76
RelationNet [Sung *et al.*, 2018]	ConvNet	CVPR'18	55.0 ± 1.0	69.3 ± 0.8	62.45 ± 0.98	76.11 ± 0.69
Ours (EGNN)	**ConvNet**	**Ours**	**76.9±0.3**	**88.2±0.3**	**74.96±0.48**	**87.08±0.36**
Distill [Tian *et al.*, 2020]	ResNet12	ECCV'20	73.9 ± 0.8	86.9 ± 0.5	-	-
Neg-Cosine [Liu *et al.*, 2020]	ResNet12	ECCV'20	-	-	74.6 ± 0.4	89.9 ± 0.3
DPGN [Yang *et al.*, 2020]	ResNet12	CVPR'20	77.9 ± 0.5	90.2 ± 0.4	75.71±0.47	91.48±0.33
EGNN$^+$ [Kim *et al.*, 2019]	ResNet12	CVPR'19	66.5 ± 0.5	81.3 ± 0.5	70.24± 0.52	84.26 ± 0.42
Ours (EGNN)	**ResNet12**	**Ours**	**77.4±0.3**	**89.2±0.2**	**75.36±0.32**	**89.23 ± 0.28**
DPGN$^+$ [Yang *et al.*, 2020]	ResNet12	CVPR'20	76.6 ± 0.5	88.7 ± 0.4	72.66 ± 0.85	89.40 ± 0.36
Ours (DPGN)	**ResNet12**	**Ours**	**78.2±0.4**	**91.2±0.3**	**76.26±0.51**	**90.56±0.35**

Table 1: Few-shot classification accuracies on four few-shot learning benchmarks. "+" denotes that our re-implemented result with the official code. Red color indicates the best performance and blue color indicates the second best performance. Bold text means our results.

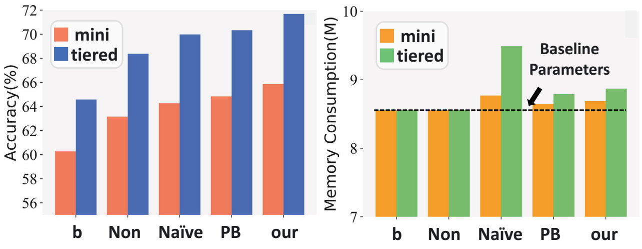

Figure 2: Ablation study when using different memory mechanism. "B": our baseline (EGNN); "Non-Mem": meta-knowledge nodes are implemented by the class center from the current episode; "Naive-Mem": a memory storing the entire features; "PB-Mem": prototype-based memory.

son, we adopt two representative few-shot graph neural networks *i.e.,* EGNN and DPGN as our GAM module. Further, by using two kinds of backbone ConvNet and ResNet12, we report the results under 5-way 1-shot and 5-way 1-shot setting on all benchmark datasets for a comprehensive evaluation.

Results on Generic Object Recognition. For the generic object classification, we evaluate our approach on miniImageNet, tieredImageNet and CIFAR-FS and report the results in Table 1. The main observations are as follows: 1) The proposed method outperforms all competitors demonstrating the effectiveness of our method. Further, the performance obtained by using ResNet12 is superior to that using ConvNet due to better representational capacity. 2) No matter which graph neural network are used, the proposed method evidently outperform the baseline with a clear margin. 3) In both 1-shot or 5-shot setting, our method is basically stable at the top performance. The improvement is more significant under 1-shot setting due to the purified memory. Hence it appears that our methods will be more effective when facing new tasks with fewer samples.

Result on Fine-grained Classification. For the fine-grained bird classification problem, the results of CUB-200-2011 is reported in Table 1. In particular, our method also outperforms other competitors by a large margin. Notice that on this dataset, different graph neural networks and backbones have less impact on the performance.

Discussion. Since the proposed method is based on the GNNs framework, Our approach could be integrated into any advanced GNNs method flexibly. Our results show that with purified memory and GAM module, the performance of GNNs would be remarkably promoted.

3.4 Ablation Study

We present experiments to confirm our main claims: 1) Purified memory can facilitate fast adaption. 2) Meta-knowledge with GAM is able to promote the existing GNNs models. All the experiments are conducted on tieredImageNet under 5-way 1-shot setting with ResNet12. Also the quantitative results of 5-shot are shown in supplementary materials.

Impact of Purified Memory. We compare four different memory banks and the results are illustrated in Fig. 2. Note that the baseline degenerates to EGNN when without memory. We can draw the following conclusions: 1) Without the help of memory, GAM can boost the performance of GNNs model, even incorporating with the class center in the current episode. 2) Three different memory banks are evidently superior to Non-Mem baseline, showing the importance of meta-

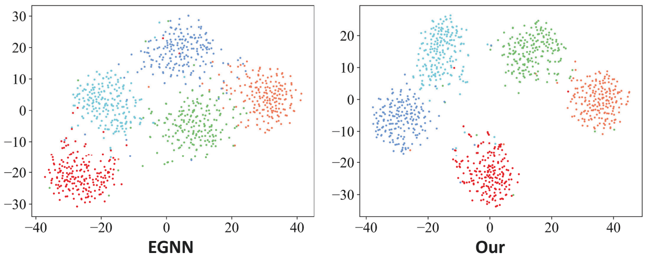

Figure 3: t-SNE visualization results obtained from our method and EGNN. Different colors represent different categories.

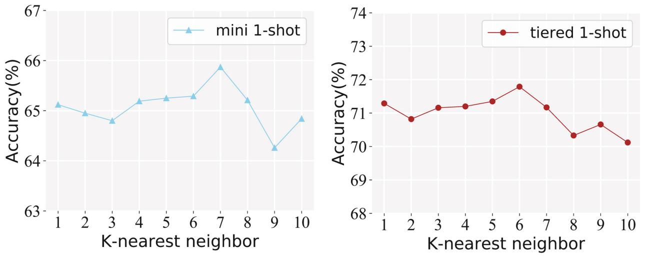

Figure 4: The performance impact of k-nearest neighbor.

knowledge. 3) The prototype-based memory is more efficient and effective, which confirms our assumption that storing the whole features is a sub-optimal solution. 4) The experimental results support our motivation of obtaining an optimal prototype representation for each category. Meanwhile, it seems that the memory cost of the proposed method is keeping the same level compared with the baseline.

Impact of GAM. To demonstrate the effectiveness of our graph augmented module, we first visualize the embedding space in Fig 3. In particular, we randomly select 5 classes, each of which contains 200 samples from tieredImageNet. We project the features trained by EGNN and our EGNN equipped with GAM into a 2D plane via t-SNE. The results show that the embedding space is mixed in EGNN, such that the discriminative ability of learned model is naturally limited. On the contrary, our model is able to distinguish different categories with a large inter-class margin, hence we get a substantial improvement. This indicates that with the help of purified meta-knowledge, the discriminative information could be further highlighted via the GAM.

Additionally, to visualize how the meta-knowledge help the prediction process, we choose a test scenario where the ground truth classes of five query images are non-overlapping (*i.e.,* 5-way 1-shot) and visualize instance-level similarities as shown in Figure 5. Specifically, we select two kinds of instance-level similarities to demonstrate the effectiveness of our approach. Notably, the heatmap shows GAM refines the instance-level similarity matrix after several layers and makes the right predictions for five query samples in the final layer compared with EGNN. We can also find this refinement is owing to the augmentation of meta-knowledge node. Due to the purified concept, the heatmap is essentially clean and hence the meta-knowledge provides auxiliary strong supervision. These similarities are then further propagated through the graph neural network, allowing the model to take the advantages of memorized concept and knowledge learned from new tasks. This experimental result convincingly supports our hypothesis.

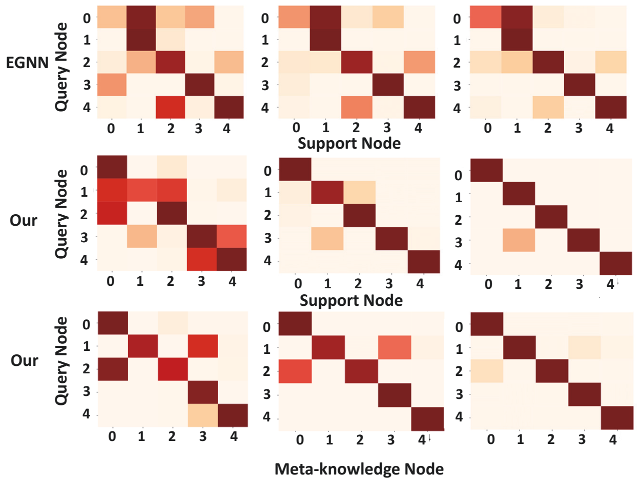

Figure 5: The visualization of edge prediction in each layer of our method. The sub-figure from left to right denotes the prediction from layer 1 to layer 3 of graph neural network. The dark denotes higher score and the shallow denotes lower confidence. The left axis stands for the index of 5 query images and the bottom axis stands for 5 support classes or our meta-knowledge node.

Impact of k-nearest neighbor. In the Meta-knowledge Mining stage, we retrieve the most similar k samples from memory to augment the graph. Here we discuss its impact when the k varies. As shown in Figure 4, few-shot recognition performance keeps improving when k increases, and when k increases to a certain value, the accuracy begins to decline on both datasets. Hence it s recommended to set this value as 6, empirically.

4 Conclusion

In this work, we have presented a new memory updating scheme for few-shot learning, which progressively purifies the semantic label information from the perspective of information theory. Purified memory is generally expressive, consistent, efficient, and then naturally cooperated with a graph augmented module. GAM further exploits the meta-knowledge and the knowledge learned from the new task to make a precise prediction. This scheme is a model-agnostic module and could be integrated into any advanced GNNs method flexibly.

Acknowledgements

This work is sponsored by National Natural Science Foundation of China (61972157, 61772524, 61876161, 61902129, 61976212), Natural Science Foundation of Shanghai(20ZR1417700), CAAI-Huawei MindSpore Open Fund, Shanghai Sailing Program (21YF1411200), National Key Research and Development Program of China (2019YFC1521104), Zhejiang Lab (No.2019KD0AC02, 2020NB0AB01), the Fundamental Research Funds for the Central Universities.

References

[Bertinetto *et al.*, 2016] Luca Bertinetto, Jack Valmadre, Joao F Henriques, Andrea Vedaldi, and Philip HS Torr. Fully-convolutional siamese networks for object tracking. In *European conference on computer vision*, pages 850–865. Springer, 2016.

[Bertinetto *et al.*, 2018] Luca Bertinetto, Joao F Henriques, Philip HS Torr, and Andrea Vedaldi. Meta-learning with differentiable closed-form solvers. *arXiv preprint arXiv:1805.08136*, 2018.

[Chen *et al.*, 2020] Yinbo Chen, Xiaolong Wang, Zhuang Liu, Huijuan Xu, and Trevor Darrell. A new meta-baseline for few-shot learning. *arXiv preprint arXiv:2003.04390*, 2020.

[Finn *et al.*, 2017] Chelsea Finn, Pieter Abbeel, and Sergey Levine. Model-agnostic meta-learning for fast adaptation of deep networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pages 1126–1135. JMLR. org, 2017.

[Garcia and Bruna, 2017] Victor Garcia and Joan Bruna. Few-shot learning with graph neural networks. *arXiv preprint arXiv:1711.04043*, 2017.

[Glorot *et al.*, 2011] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 315–323, 2011.

[He *et al.*, 2016] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

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

[Kim *et al.*, 2019] Jongmin Kim, Taesup Kim, Sungwoong Kim, and Chang D Yoo. Edge-labeling graph neural network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11–20, 2019.

[Lee *et al.*, 2019] Kwonjoon Lee, Subhransu Maji, Avinash Ravichandran, and Stefano Soatto. Meta-learning with differentiable convex optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 10657–10665, 2019.

[Liu *et al.*, 2018] Yanbin Liu, Juho Lee, Minseop Park, Saehoon Kim, Eunho Yang, Sung Ju Hwang, and Yi Yang. Learning to propagate labels: Transductive propagation network for few-shot learning. *arXiv preprint arXiv:1805.10002*, 2018.

[Liu *et al.*, 2020] Bin Liu, Yue Cao, Yutong Lin, Qi Li, Zheng Zhang, Mingsheng Long, and Han Hu. Negative margin matters: Understanding margin in few-shot classification. *arXiv preprint arXiv:2003.12060*, 2020.

[Long *et al.*, 2015] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[Noh *et al.*, 2017] Hyeonwoo Noh, Andre Araujo, Jack Sim, Tobias Weyand, and Bohyung Han. Large-scale image retrieval with attentive deep local features. In *Proceedings of the IEEE international conference on computer vision*, pages 3456–3465, 2017.

[Ramalho and Garnelo, 2019] Tiago Ramalho and Marta Garnelo. Adaptive posterior learning: few-shot learning with a surprise-based memory module. *arXiv preprint arXiv:1902.02527*, 2019.

[Ren *et al.*, 2018] Mengye Ren, Eleni Triantafillou, Sachin Ravi, Jake Snell, Kevin Swersky, Joshua B Tenenbaum, Hugo Larochelle, and Richard S Zemel. Meta-learning for semi-supervised few-shot classification. *arXiv preprint arXiv:1803.00676*, 2018.

[Snell *et al.*, 2017] Jake Snell, Kevin Swersky, and Richard Zemel. Prototypical networks for few-shot learning. In *Advances in neural information processing systems*, pages 4077–4087, 2017.

[Sung *et al.*, 2018] Flood Sung, Yongxin Yang, Li Zhang, Tao Xiang, Philip HS Torr, and Timothy M Hospedales. Learning to compare: Relation network for few-shot learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1199–1208, 2018.

[Tian *et al.*, 2020] Yonglong Tian, Yue Wang, Dilip Krishnan, Joshua B Tenenbaum, and Phillip Isola. Rethinking few-shot image classification: a good embedding is all you need? *arXiv preprint arXiv:2003.11539*, 2020.

[Tishby and Zaslavsky, 2015] Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 IEEE Information Theory Workshop (ITW)*, pages 1–5. IEEE, 2015.

[Vinyals *et al.*, 2016] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. In *Advances in neural information processing systems*, pages 3630–3638, 2016.

[Wah *et al.*, 2011] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical report, 2011.

[Wang *et al.*, 2020] Zeyuan Wang, Yifan Zhao, Jia Li, and Yonghong Tian. Cooperative bi-path metric for few-shot learning. In *Proceedings of the 28th ACM International Conference on Multimedia*, pages 1524–1532, 2020.

[Yang *et al.*, 2020] Ling Yang, Liangliang Li, Zilun Zhang, Xinyu Zhou, Erjin Zhou, and Yu Liu. Dpgn: Distribution propagation graph network for few-shot learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 13390–13399, 2020.