Highlights

Hierarchical Knowledge Propagation and Distillation for Few-Shot Learning

Chunpeng Zhou,Haishuai Wang*,Sheng Zhou,Zhi Yu,Danushka Bandara,Jiajun Bu*

- We highlight the significance of the inductive Few-Shot Learning in the real-world settings
- The existing inductive Few-Shot Learning methods usually ignore the relations between sample-level and class-level representations
- The proposed HKPD framework can leverage these relations, which is designed for the inductive setting
- Deploy a self-distillation module on the hierarchical architecture to improve the performance further

Hierarchical Knowledge Propagation and Distillation for Few-Shot Learning

Chunpeng Zhou[a], Haishuai Wang*[a], Sheng Zhou[a], Zhi Yu[a], Danushka Bandara[b] and Jiajun Bu*[a]

[a]College of Computer Science, Zhejiang University, Hangzhou, 310000, China
[b]Department of Computer Science and Engineering, Fairfield University, Fairfield, 06824, USA

ARTICLE INFO

Keywords:
Few-Shot Learning
Knowledge Distillation
Inductive Learning
Feature Representation
Classification

ABSTRACT

Recent research efforts on Few-Shot Learning (FSL) have achieved extensive progress. However, the existing efforts primarily focus on the transductive setting of FSL, which is heavily challenged by the limited quantity of the unlabeled query set. Although a few inductive-based FSL methods have been studied, most of them emphasize learning superb feature extraction networks. As a result, they may ignore the relations between sample-level and class-level representations, which are particularly crucial when labeled samples are scarce. This paper proposes an inductive FSL framework that leverages the Hierarchical Knowledge Propagation and Distillation, named HKPD. To learn more discriminative sample-level representations, HKPD first constructs a sample-level information propagation module that explores pairwise sample relations. Subsequently, a class-level information propagation module is designed to obtain and update the class-level information. Moreover, a self-distillation module is adopted to further improve the learned representations by propagating the obtained knowledge across this hierarchical architecture. Extensive experiments conducted on the commonly used few-shot benchmark datasets demonstrate the superiority of the proposed HKPD method, which outperforms the current state-of-the-art methods.

1. Introduction

Recently, deep learning has achieved remarkable success in various tasks, e.g., Image classification (He, Zhang, Ren and Sun, 2016), Object detection (Lin, Goyal, Girshick, He and Dollár, 2017), Instance segmentation (He, Gkioxari, Dollar and Girshick, 2017), and Image reconstruction (He, Chen, Xie, Li, Dollár and Girshick, 2022b). However, deep neural networks usually suffer from being data hungry (Aggarwal et al., 2018), which means that the effectiveness of deep neural networks is heavily reliant on large-scale labeled training data, such as the ImageNet dataset that contains more than 14 million annotations (Russakovsky, Deng, Su, Krause, Satheesh, Ma, Huang, Karpathy, Khosla, Bernstein et al., 2015). Data labeling is typically time-consuming and costly and may require abundant domain knowledge. Therefore, to reduce the cost of data annotation, many researchers aim to develop deep learning models that can learn from very few samples (i.e., one available labeled sample per class). These models can even perform comparably to humans (Fei-Fei, Fergus and Perona, 2006; Lake, Salakhutdinov and Tenenbaum, 2015). To achieve these objectives, Few-Shot Learning (FSL for short) has garnered significant attention in recent years (Wang, Yao, Kwok and Ni, 2020; Li, Sun, Xue and Ma, 2021).

Although several methods have been proposed to improve the performance of FSL tasks, most of the current research mainly focuses on the transductive setting of FSL (Liu, Lee, Park, Kim, Yang, Hwang and Yang, 2019; Kim, Kim, Kim and Yoo, 2019; Ma, Bai, An, Liu, Liu, Zhen and Liu, 2020; Yang, Li, Zhang, Zhou, Zhou and Liu, 2020; Liu, Song and Qin, 2020b; Chen, Yang, Xu, Huang and Ma, 2021; Zhu and Koniusz, 2022; Hu, Gripon and Pateux, 2021). However, the proposed transductive FSL methods necessitate the use of both labeled samples (a.k.a. support set) and unlabeled samples (a.k.a. query set) simultaneously for the inference (Liu et al., 2019; Ma et al., 2020) as we have shown in Fig. 1(a). Consequently, the transductive FSL methods rely heavily on the quantity of the unlabeled query set. Previous studies have demonstrated that the performance of transductive FSL methods will deteriorate dramatically when the quantity of unlabeled samples is limited (Hu et al., 2021; Zhu and Koniusz, 2022). Given that the quantity of a query set cannot always be guaranteed in a realistic

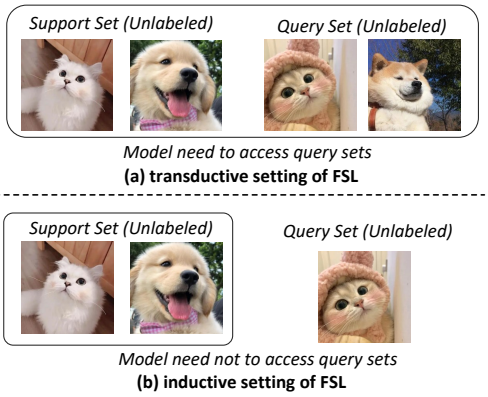


Figure 1: An illustration of the difference between the transductive and inductive setting of few-shot learning. (a) shows the setting of the transductive of FSL, where models need to access the support set and query set simultaneously. (b) shows the setting of the inductive of FSL, where models only need to access the support set and are not reliant on the query set.

*Corresponding authors
ORCID(s):

environment, we argue that the inductive-based FSL is more generally applicable in real-world settings than the transductive FSL. As shown in Fig. 1(b), the inductive FSL methods need not access the query set during the training stage and are not reliant on the quantity of the unlabeled query set. Consequently, we mainly focus on the inductive settings for FSL in this paper.

Some classical FSL works, e.g., prototypical network (Snell, Swersky and Zemel, 2017), MAML (Finn, Abbeel and Levine, 2017), are designed for the inductive setting of FSL. However, their performance is currently lagging. To improve the FSL performance under the inductive setting, researchers have attempted to pre-train a feature extraction network (a.k.a. backbone) to obtain a good representation (Chen, Liu, Kira, Wang and Huang, 2019; Dhillon, Chaudhari, Ravichandran and Soatto, 2020; Tian, Wang, Krishnan, Tenenbaum and Isola, 2020). Despite those methods achieving promising results under the inductive FSL, they seldom consider the relations between sample-level and class-level representations, which can potentially improve the quality of the representation, especially when the number of available samples is minimal (Liu et al., 2019; Kim et al., 2019; Yang et al., 2020; Chen et al., 2021; Hu et al., 2021). For instance, given an FSL task, we aim to obtain a good representation of the cat when only a single cat instance is available. In this situation, we may observe that the dog is more similar to a cat than a bird by exploring the relations among samples. Consequently, we can aggregate the dog's representation to improve the cat's representation ability. Some recent methods have partially attempted to address this problem. For example, the FEAT (Ye, Hu, Zhan and Sha, 2020) constructed the relations among different classes, though it ignores the relations among samples. The HGNN (Yu, He, Song and Xiang, 2022) constructed a dual-graph structure to utilize the relations between samples and classes, respectively. Since these relations are learned individually in this dual-graph structure, they can not be optimized jointly. So they may lead to inferior performance under the inductive setting of FSL.

To address the above issues, we propose a novel framework designed for the inductive FSL setting, called HKPD, based on Hierarchical Knowledge Propagation and Distillation. Firstly, we obtain the vanilla representations of samples in an FSL task by a deep feature extraction module (e.g., ResNet (He et al., 2016)), and then construct the sample-level information propagation module to explore the pairwise sample relations. In this way, the HKPD can obtain more sample information from similar samples to update the representations, which can help to learn more discriminative sample-level representations. During this processing, only the samples with labels in a support set will be involved and updated for the inductive setting of FSL. Thus, our method does not rely on the number of samples in the query set. Secondly, the refined sample-level representations are used to obtain the class-level representations by computing the class prototype (Snell et al., 2017). Then the class-level representations are updated by the class-level information propagation module by exploring the pairwise class relations. After that, we can obtain the refined class-level representations, which leverage similar information from other classes. The refined class-level representations can then be used for FSL directly. Finally, to further improve the representations, a self-distillation module (Zhang, Song, Gao, Chen, Bao and Ma, 2019) is employed to propagate the knowledge across this hierarchical architecture, including the deep feature extraction module, sample-level information propagation module, and class-level information propagation module. The information learned by different modules can serve as the inner supervision signals and regularizations (Zhang et al., 2019; Cho and Hariharan, 2019) that jointly improve each other further through the proposed self-distillation module. In this proposed hierarchical end-to-end framework, the earlier modules feed the more informative representations to the later modules to learn more accurate relations. On the other hand, the refined higher-level relations obtained by the later modules provide more accurate supervision signals to guide the earlier modules to optimize better. When we want to infer the labels of query samples, we can just use the vanilla representations of query examples.

Additionally, compared with other inductive-based FSL methods (Finn et al., 2017; Chen et al., 2019; Dhillon et al., 2020; Tian et al., 2020; Liu, Cao, Lin, Li, Zhang, Long and Hu, 2020a; Afrasiyabi, Lalonde and Gagné, 2021), our method does not require fine-tuning during the reference period. Therefore this method can easily be expanded to other applications, especially when computing resources are limited and the backpropagation algorithm is unavailable.

To summarize, our main contributions are as follows:

i We highlight the significance of the inductive setting on FSL, and develop a novel framework based on Hierarchical Knowledge Propagation and Distillation designed for the inductive setting. This proposed framework enables the exploration of the relationships between sample-level and class-level information simultaneously to obtain distinctive representations.

ii To the best of our knowledge, in the field of FSL, this is the first effort to introduce a self-distillation module to allow information propagation across the hierarchical architecture and improve the quality of learned representations further.

iii Our experimental results on commonly used FSL benchmarks demonstrate that the proposed HKPD outperforms the state-of-the-art methods under the inductive setting of FSL.

2. Related Work

2.1. Transductive few-shot learning

Transductive few-shot learning methods simultaneously model the support and query sets in an FSL task. The TPN (Liu et al., 2019) was the first attempt to introduce the transductive setting of FSL, which learns to propagate labels

from the support set to the query set via a constructed graph structure. Then, EGNN (Kim et al., 2019) extended the TPN by adopting a more powerful deep graph neural network instead of label propagation on the edge-labeling graph. The TRPN method extended the TPN and explicitly modeled relations about support-query pairs via graph neural networks. The DPGN (Yang et al., 2020) modeled the distribution-level inference explicitly and attempted to use deeper neural networks. The BD-CSPN (Liu et al., 2020b) proposed prototype rectification based on Prototypical Networks under the transductive setting of FSL. It utilized label propagation and feature shifting to improve the prototype by incorporating the representations of query sets. The ECKPN (Chen et al., 2021), also designed for transductive settings, explored the class-level knowledge by incorporating external multi-modal knowledge to guide the inference of query samples. The external multi-modal knowledge contains word embeddings obtained by a GloVe model (Pennington, Socher and Manning, 2014), which was pre-trained on a large text dataset. The Graph Interpolation (Hu et al., 2021) built a new three-stage method for the transductive setting, which contains pretraining, graph-based feature interpolation, and logistic regression. The EASE (Zhu and Koniusz, 2022) attempted to improve transductive performance by training a simple linear projection onto a subspace built from representations of both the support set and the query set.

As discussed above, although these proposed methods can achieve good performance under the transductive setting of FSL, some experimental results (Hu et al., 2021; Zhu and Koniusz, 2022) show that the number of samples in the query set significantly influences accuracy up to a few dozen. Moreover, the performance dramatically deteriorates when the quantity of unlabeled query samples decreases.

2.2. Inductive few-shot learning

As discussed above, inductive few-shot learning does not access or utilize the information from the query sets during the training stage, which is more applicable in real-world settings compared to the transductive FSL. Some earlier few-shot learning works, e.g., prototypical network (Snell et al., 2017), MAML (Finn et al., 2017), are designed under the inductive setting. The prototypical network obtains the prototype of each class by averaging the representations of samples in each class. After that, the unlabeled query samples find the nearest prototype for inference. MAML introduced the two-order gradient-based meta-learning into the field of FSL. It updates the model parameters by utilizing given support examples. Although these methods can work under the inductive setting of FSL, their performance is low compared to the state-of-the-art. In order to improve the performance further, some works focus on pretraining a good feature extraction network by utilizing the base dataset and then freezing the parameters of the feature extraction network. After that, they replace the old classifier with a simple linear classifier and only fine-tune the new classifier (Chen et al., 2019; Dhillon et al., 2020; Tian et al., 2020) using the support set. Subsequently, the pre-trained backbone with a new classifier can be used for inference. For example, baseline++ (Chen et al., 2019) used a cosine similarity classifier to reduce intra-class variation among features during training. Fine-tuning (Dhillon et al., 2020) extended the baseline++ with a Shannon Entropy regulation, and it can also work under the transductive setting. RFS (Tian et al., 2020) utilized the Born-again strategy (Furlanello, Lipton, Tschannen, Itti and Anandkumar, 2018) to boost the pre-trained feature extraction network. At the same time, some works attempt to explore the relations between sample-level and class-level representations. The FEAT (Ye et al., 2020) constructed the relations among each class to learn more robust prototype representations and then used these prototypes of each class for inference. However, FEAT ignored the relations among samples. The HGNN (Yu et al., 2022) constructed a dual-graph neural network structure to exploit the relations of samples and classes, respectively. While these relations are learned individually in this proposed structure, they are not optimized jointly.

Additionally, we want to highlight the difference with the most recent baseline HGNN (Yu et al., 2022): our method can learn the relations between sample-level and class-level representations simultaneously with the propagation of hierarchical information. Moreover, these obtained representations will further benefit from the self-distillation module. Furthermore, HGNN can model and predict only one query sample each time under the inductive setting (Yu et al., 2022), while HKPD can predict multiple query samples each time.

3. Preliminary

3.1. Problem Formulation

As previously discussed, FSL is intended to learn from a limited number of samples and recognize unknown samples from unseen or novel classes. The dataset contains novel classes, denoted as $\mathbf{D}_{\mathbf{novel}}$, where only the limited labeled samples per class are available. This paper follows the standard few-shot setting (Vinyals, Blundell, Lillicrap, Wierstra et al., 2016; Snell et al., 2017; Finn et al., 2017). In detail, an N-way K-shot few-shot learning task includes N classes with K labeled samples for each class, and these labeled samples are named as the support set in novel classes, denoted as $\mathbf{S}_{\mathbf{novel}} = \{(x_1, y_1), \ldots, (x_{N\times K}, y_{N\times K})\} \in \mathbf{D}_{\mathbf{novel}}$. The unknown or unlabeled samples that need to be classified are denoted as the query set in novel classes, such that $\mathbf{Q}_{\mathbf{novel}} = \{(x_{N\times K+1}, y_{N\times K+1}), \ldots, (x_{N\times K+T}, y_{N\times K+T})\} \in \mathbf{D}_{\mathbf{novel}}$, where T is the number of query samples in each FSL task. Specifically, we focus on the inductive setting of FSL (Kim et al., 2019; Yu et al., 2022) in this paper, meaning that we do not use the information from the query samples, and inference can be performed one-by-one (Tian et al., 2020; Yu et al., 2022), without relying on the size of a query set. On the contrary, the transductive FSL methods must access enough query samples to guarantee their performance (Hu et al., 2021; Zhu and Koniusz, 2022).

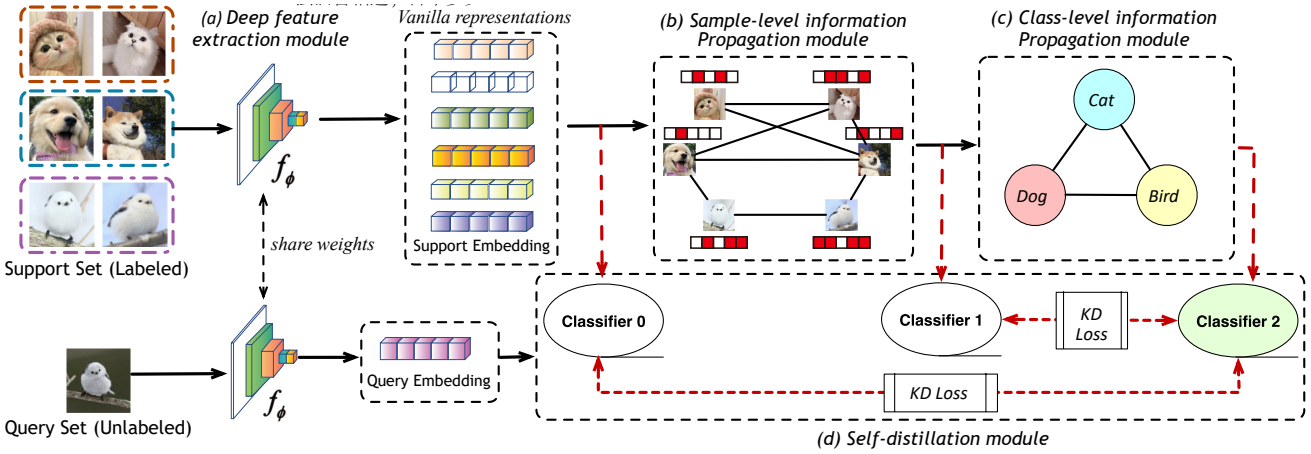


Figure 2: The overall architecture of our proposed HKPD method with an example of the 3-way 2-shot learning task. (a) The vanilla sample-level representations are obtained via a deep feature extraction module. (b) The sample-level information propagation module updates the vanilla representations. (c) The class-level representations are obtained and updated by the class-level information propagation module. (d) A self-distillation module further improves the representations and propagates the knowledge across this hierarchical architecture by leveraging the KD loss among the deep feature extraction, sample-level information propagation, and class-level information propagation modules.

3.2. Meta-training

Although we have the support set $\mathbf{S}_{\mathbf{novel}}$ with annotations in $\mathbf{D}_{\mathbf{novel}}$, training the FSL model just using these very few labeled samples (i.e., one labeled sample per class) will cause a high risk of overfitting. To obtain a FSL model with good generalization and satisfactory performance for a given few-shot task, the standard few-shot settings usually require to include an additional base dataset in which samples are annotated (Vinyals et al., 2016; Snell et al., 2017; Chen et al., 2019), denoted as $\mathbf{D}_{\mathbf{base}}$, which can be used to assist the FSL model to alleviate overfitting during the training stage. Also, it is worth noting that the labels of the base dataset and novel dataset are entirely disjoint, i.e., $\mathbf{C}_{\mathbf{base}} \cap \mathbf{C}_{\mathbf{base}} = \phi$. Thus, there is no overlap in the class label spaces or samples between $\mathbf{D}_{\mathbf{base}}$ and $\mathbf{D}_{\mathbf{base}}$. In order to reduce the gap between these two datasets, the meta-training strategy, a.k.a. episodic training, is introduced (Vinyals et al., 2016; Snell et al., 2017). The meta-training strategy aims to simulate the N-way K-shot few-shot testing task in the novel dataset by sampling a series of N-way K-shot few-shot tasks in the base dataset during the training stage, i.e., $\mathbf{S}_{\mathbf{base}} = \{(x_1, y_1), (x_2, y_2), \ldots, (x_{N\times K}, y_{N\times K})\} \in \mathbf{D}_{\mathbf{base}}$, and $\mathbf{Q}_{\mathbf{base}} = \{(x_{N\times K+1}, y_{N\times K+1}), \ldots, (x_{N\times K+T}, y_{N\times K+T})\} \in \mathbf{D}_{\mathbf{base}}$, which is similar with $\mathbf{S}_{\mathbf{novel}}$ and $\mathbf{Q}_{\mathbf{novel}}$ in novel classes. Consequently, we first will train parameters of the FSL model by utilizing series support set $\mathbf{S}_{\mathbf{base}}$ to predict the query set $\mathbf{Q}_{\mathbf{base}}$, which are all sampled from the base data $\mathbf{D}_{\mathbf{base}}$. Then we will repeat this process until the FSL model converges. By this method, the FSL model can learn the transferable knowledge from $\mathbf{D}_{\mathbf{base}}$ via meta-training. Then, we evaluate the trained FSL model during the testing stage, where we attempt to classify the query set $\mathbf{Q}_{\mathbf{novel}}$ by utilizing the support set $\mathbf{S}_{\mathbf{novel}}$ in the novel dataset $\mathbf{D}_{\mathbf{novel}}$.

3.3. Prototypical network

For the convenience of understanding the subsequent content, we will introduce the details about the Prototypical network (Snell et al., 2017) in this section before introducing our proposed method. As discussed above, the prototypical network obtains the prototype of each class for inference. Formally, we have:

$$X(i,j) = f_\Theta(x(i,j)) \tag{1}$$

$$\mathbf{P}(i) = \frac{1}{K}\sum_{j=1}^{K} X(i,j) \tag{2}$$

where $x(i,j)$ denotes the j-th sample of i-th class from a support set, $\mathbf{P}(i)$ denotes the obtained prototype of i-th class. The f_Θ denotes a feature extraction module (e.g., ResNet (He et al., 2016)), the aim of which is to extract a low-dimensional representation of the original image. Θ means the learnable parameter set in f_Θ. When an unlabeled sample x_q from the query set needs to be recognized, the prototypical network predicts it by a non-parametric distance-based classifier with a softmax function:

$$\hat{y}_q = \frac{\exp(-d(f_\Theta(x_q)), \mathbf{P}(i))}{\sum_{i'} \exp(-d(f_\Theta(x_q)), \mathbf{P}(i'))} \tag{3}$$

where $\hat{y}_q$ denotes the predicted label of the query sample x_q and $d(\ )$ is a distance metric function, e.g., the Euclidean distance.

During the meta-training stage, the prototypical network is trained by utilizing $\mathbf{S}_{\mathbf{base}}$ and $\mathbf{Q}_{\mathbf{base}}$, which are all sampled from $\mathbf{D}_{\mathbf{base}}$. The parameters of the prototypical network are updated by minimizing the cross entropy loss: $L_{proto} = CE(\hat{y}_q, y_q)$, where y_q is the ground-truth label. After that, we use this trained prototypical network to predict the label of samples in $\mathbf{Q}_{\mathbf{novel}}$ by utilizing the $\mathbf{S}_{\mathbf{novel}}$, which are all from $\mathbf{D}_{\mathbf{novel}}$.

4. Methods

4.1. Overview

In this section, we introduce our proposed HKPD method and provide details of its components. Unlike previous inductive FSL methods that usually ignore learning the significant relations between the sample-level and class-level representations, HKPD exploits these relations simultaneously and further improves them through a novel hierarchical architecture. As illustrated in Fig.2, besides the commonly used Deep feature extraction module, the HKPD architecture contains three other novel modules: Sample-level Information Propagation Module, Class-level Information Propagation Module, and Self-Distillation Module.

4.2. Sample-level Information Propagation

In the Sample-level Information Propagation Module, we exploit the relations of sample-level information to obtain more discriminative representations. Given an N-way K-shot learning task, we have the $N \times K$ labeled support samples, denoted as $\mathbf{S} = \{x_i, y_i\}_{i=1}^{N\times K}$, and T unlabeled query samples denoted as $\mathbf{Q} = \{x_i, y_i\}_{N\times K+1}^{N\times K+T}$. Then, We use $\mathbf{X_0^S}, \mathbf{X_0^Q}$ to denote their vanilla representation matrix respectively, obtained by a deep feature extraction module f_Θ. Formally, we have:

$$\mathbf{X_0^S} = f_\Theta(S) \in \mathbb{R}^{NK\times d_0} \tag{4}$$

$$\mathbf{X_0^Q} = f_\Theta(Q) \in \mathbb{R}^{T\times d_0} \tag{5}$$

where d_0 is the output dimension of the f_Θ.

Due to the inductive setting of FSL, we only consider the information propagation in the support set. The Sample-level Information Propagation explores the pairwise sample relations among support samples, which helps learn more discriminative and robust sample-level representations, denoted as $\mathbf{X_{sam}}$. Initially, we obtain the sample-level similarity matrix $\mathbf{A_{sam}} \in \mathbb{R}^{NK\times NK}$ through the vanilla support representation $\mathbf{X_0^S}$, based on the principle that more information will be propagated if two samples are similar. Each element in $\mathbf{A_{sam}}$ represents the similarity relation between two samples. To fully explore the relations among samples and reduce the inductive bias, we choose the Transformer-based architecture as our module because many previous works (Vaswani, Shazeer, Parmar, Uszkoreit, Jones, Gomez, Kaiser and Polosukhin, 2017; Ye et al., 2020; Liu, Hamilton, Long, Jiang and Larochelle, 2021) have demonstrated the outstanding performance of Transformers in various downstream tasks. Formally, we have

$$\mathbf{X_{sam}} = \mathbf{A_{sam}}\mathbf{V_{sam}} \tag{6}$$

$$\mathbf{A_{sam}} = \text{softmax}(\frac{\mathbf{Q_{sam}}\mathbf{K_{sam}^\top}}{\sqrt{d_1}} \odot \mathbf{M}) \tag{7}$$

where $\mathbf{Q_{sam}}, \mathbf{K_{sam}}, \mathbf{V_{sam}} \in \mathbb{R}^{NK\times d_1}$ are the different learned linear projections about the input sample representations $\mathbf{X_0}$. Formally, this process can described as follows:

$$\begin{aligned}\mathbf{Q_{sam}} &= \mathbf{X_0^S}\mathbf{W_0^q},\ \mathbf{W_0^q} \in \mathbb{R}^{d_0\times d_1}\\ \mathbf{K_{sam}} &= \mathbf{X_0^S}\mathbf{W_0^k},\ \mathbf{W_0^k} \in \mathbb{R}^{d_0\times d_1}\\ \mathbf{V_{sam}} &= \mathbf{X_0^S}\mathbf{W_0^v},\ \mathbf{W_0^v} \in \mathbb{R}^{d_0\times d_1}\end{aligned} \tag{8}$$

where $\mathbf{W_0^q}$, $\mathbf{W_0^k}$, and $\mathbf{W_0^v}$ are all learnable parameters. d_1 denotes the dimension of sample-level representations here.

As a result, the matrix $\mathbf{A_{sam}}$ describes the pairwise similarity among samples. The representation of a sample will be updated by obtaining knowledge from similar samples in an FSL task. Moreover, the refined sample-level representation matrix $\mathbf{X_{sam}}$ is obtained through this information propagation.

It is worth noting that the function of mask matrix $\mathbf{M} \in \mathbb{R}^{NK\times NK}$ in Eq.(7) is to select the most informative samples adaptively, which performs similarly with the graph pooling operator (Ying, You, Morris, Ren, Hamilton and Leskovec, 2018; Zhang, Bu, Ester, Zhang, Li, Yao, Huifen, Yu and Wang, 2021b). Every element in the mask matrix $\mathbf{M}$ is 1 or 0, and $\odot$ denotes Hadamard product. To compute it, we first obtain the indices of the top-ranked value in each row of the matrix ($\mathbf{Q_{sam}}\mathbf{K_{sam}^\top}$) in Eq.(7). Then, the elements in $\mathbf{M}$ corresponding to chosen indices will be set to 1. Moreover, the other elements in $\mathbf{M}$ will be set to 0. The pseudo-code for this adaptive choosing process is as follows:

```
Index = QK.topk(m,dim=-1,largest=True)
M_mask = torch.zeros_like((NK,NK))
M_mask = M_mask.scatter(index=Index, value=1)
```

Thus, only top-ranked similar samples in an FSL task will be selected for propagation for each sample, which aims to select the most useful information further and filter some uncorrelated noise.

Additionally, we add the commonly used layer normalization (Ba, Kiros and Hinton, 2016) and residual connection (He et al., 2016) components in this module to alleviate the exploding or vanishing gradient and stabilize learning towards convergence. Furthermore, to mitigate overfitting, this module applies the Dropout technique (Srivastava, Hinton, Krizhevsky, Sutskever and Salakhutdinov, 2014). Formally, we have:

$$\mathbf{X_{sam}} = Layernorm(FC_{Dropout}(\mathbf{A_{sam}}\mathbf{V_{sam}}) + \mathbf{V_{sam}}) \tag{9}$$

where $FC_{Dropout}$ stands for a fully-connected neural network with Dropout.

Using the Sample-level Information Propagation Module, the output representations $\mathbf{X_{sam}}$ will be more general and discriminative.

4.3. Class-level Information Propagation

In the Class-level Information Propagation Module, we exploit the relations of sample-level information to obtain more discriminative representations. This module differs from the sample-level Information Propagation module in that higher-level information is considered. Firstly, we compute the representation of each class (a.k.a. prototype (Snell

et al., 2017; Liu et al., 2020b)) by averaging all the representations of samples in each class:

$$\mathbf{P}_{\mathbf{sam}}(i) = \frac{1}{K}\sum_{j=1}^{K}\mathbf{X}_{\mathbf{sam}}(i,j) \tag{10}$$

where $\mathbf{P}_{\mathbf{sam}}(i)$ means the i-th prototype corresponding the i-th class in a FSL task, and $\mathbf{X}_{\mathbf{sam}}(i,j)$ means the the refined representation of j-th samples in i-th class obtained by the previous sample-level information propagation module. Here, we use $\mathbf{X}_{\mathbf{sam}}$ to compute prototypes, rather than using $\mathbf{X}_{\mathbf{0}}$ obtained directly from the feature extraction networks, which can make the representation of prototypes more informative.

Then, we update the prototypes by the Class-level Information Propagation, which can explore the pairwise class relations between prototypes. We also use a transformer-based architecture as the Class-level Information Propagation Module. Similarly, we have:

$$\mathbf{P}_{\mathbf{cla}} = Layernorm(FC_{Dropout}(\mathbf{A}_{\mathbf{cls}}\mathbf{V}_{\mathbf{sam}}) + \mathbf{V}_{\mathbf{sam}}) \tag{11}$$

$$\mathbf{A}_{\mathbf{cla}} = \mathrm{softmax}(\frac{\mathbf{Q}_{\mathbf{cla}}\mathbf{K}_{\mathbf{cla}}^{\top}}{\sqrt{d}}) \tag{12}$$

where $\mathbf{Q}_{\mathbf{cla}}, \mathbf{K}_{\mathbf{cla}}, \mathbf{V}_{\mathbf{cla}}$ are the different learned linear projections about the input prototypical representations $\mathbf{P}_{\mathbf{sam}}$:

$$\begin{aligned}\mathbf{Q}_{\mathbf{cla}} &= \mathbf{P}_{\mathbf{sam}}\mathbf{W}_{\mathbf{1}}^{\mathbf{q}},\ \mathbf{W}_{\mathbf{1}}^{\mathbf{q}} \in \mathbb{R}^{d_1\times d_2}\\ \mathbf{K}_{\mathbf{cla}} &= \mathbf{P}_{\mathbf{sam}}\mathbf{W}_{\mathbf{1}}^{\mathbf{k}},\ \mathbf{W}_{\mathbf{1}}^{\mathbf{k}} \in \mathbb{R}^{d_1\times d_2}\\ \mathbf{V}_{\mathbf{cla}} &= \mathbf{P}_{\mathbf{sam}}\mathbf{W}_{\mathbf{1}}^{\mathbf{v}},\ \mathbf{W}_{\mathbf{1}}^{\mathbf{v}} \in \mathbb{R}^{d_1\times d_2}\end{aligned} \tag{13}$$

where $\mathbf{W}_{\mathbf{1}}^{\mathbf{q}}$, $\mathbf{W}_{\mathbf{1}}^{\mathbf{k}}$, and $\mathbf{W}_{\mathbf{1}}^{\mathbf{v}}$ are all the learnable parameters. d_2 denotes the dimension of class-level representations here. $\mathbf{P}_{\mathbf{cla}}$ is the refined prototypical representation, obtained by aggregating similar information based on the class-level similarity matrix $\mathbf{A}_{\mathbf{cla}}$. Similarly, the layer normalization, residual connection, and Dropout technique are also applied in this module. We do not use masking here because the number of classes is usually less than the number of total samples empirically. Consequently, we choose the $\mathbf{P}_{\mathbf{cla}}$ as the final representation of each class, which is more informative and discriminative when obtained by this Class-level Information Propagation Module.

4.4. Hierarchical Propagation and Distillation

In this section, we will introduce the proposed self-distillation module. This module helps the knowledge propagation further and improves the representation obtained by previous modules. To date, we have obtained the refined prototypical representations $\mathbf{P}_{\mathbf{cla}}$, which can be applied to the classification tasks directly. Formally, given a query sample $x_i \in \mathbf{X}^{\mathbf{Q}}$, we obtain the predicted label directly by a prototypical classifier with a softmax function (Snell et al., 2017) :

$$p_{i,j}^{cla} = \frac{\exp(-d(f_{\Theta}(x_i)), \mathbf{P}_{\mathbf{cla}}(j))}{\sum_{j'}\exp(-d(f_{\Theta}(x_i)), \mathbf{P}_{\mathbf{cla}}(j'))} \tag{14}$$

where $p_{i,j}^{cla}$ denotes the probability of the i-th query belonging to the class j, and $d()$ denotes the Euclidean distance.

In addition to the refined prototypical representations $\mathbf{P}_{\mathbf{cla}}$, we also have obtained the middle prototypical representations $\mathbf{P}_{\mathbf{sam}}$ in previous Class-level Information Propagation Module. Further, the vanilla prototypical representations $\mathbf{P}_{\mathbf{0}}$ can be computed by leveraging the vanilla representations $\mathbf{X}_{\mathbf{0}}^{\mathbf{S}}$ per class, which are the outputs of the feature extraction module $f_{\Theta}()$ used in Prototypical Network. So we have $\mathbf{P}_0(i) = \frac{1}{K}\sum_{j=1}^{K}\mathbf{X}_{\mathbf{0}}^{\mathbf{S}}(i,j)$. Thus, the proposed HKPD produce three different hierarchical prototypical representations $\mathbf{P}_{\mathbf{cla}}$, $\mathbf{P}_{\mathbf{sam}}$ and $\mathbf{P}_{\mathbf{0}}$, all of which can be applied to classification tasks solely. As described in Fig.2, we apply three prototypical classifiers for classification tasks. Formally, given a query sample $x_i \in \mathbf{X}^{\mathbf{Q}}$, besides the Eq(14), we can obtain the two extra predictions of it as follows:

$$p_{i,j}^{0} = \frac{\exp(-d(f_{\Theta}(x_i)), \mathbf{P}_{\mathbf{0}}(j))}{\sum_{j'}\exp(-d(f_{\Theta}(x_i)), \mathbf{P}_{\mathbf{0}}(j'))} \tag{15}$$

$$p_{i,j}^{sam} = \frac{\exp(-d(f_{\Theta}(x_i)), \mathbf{P}_{\mathbf{sam}}(j))}{\sum_{j'}\exp(-d(f_{\Theta}(x_i)), \mathbf{P}_{\mathbf{sam}}(j'))} \tag{16}$$

where $p_{i,j}^{sam}$ and $p_{i,j}^{0}$ represent the probabilities obtained by prototypical classifiers respectively. Consequently, we have three classification losses totally: $L_2 = \sum_j^N CE(p_{i,j}^{cla}, y_i)$, $L_1 = \sum_j^N CE(p_{i,j}^{sam}, y_i)$ and $L_0 = \sum_j^N CE(p_{i,j}^{0}, y_i)$, corresponding to three classifiers respectively.

Although three predictions have been obtained from the above three classifiers, we want to further improve their performance by utilizing these predictions. Consequently, we employ the self-distillation module (Zhang et al., 2019) among these three classifiers to improve the knowledge propagation across the hierarchical architecture. Unlike the vanilla knowledge distillation (Hinton, Vinyals, Dean et al., 2015), the proposed self-Distillation module yields supervision signals from the network itself rather than another giant pre-trained teacher network, which can further improve the performances of a network. In other words, we let the three modules learn from each other, including the deep feature extraction module, sample-level information propagation module, and class-level information propagation module. In detail, the self-distillation loss employed can be described as:

$$L_{self} = KD(p_{i,j}^{cla}, p_{i,j}^{0}) + KD(p_{i,j}^{cla}, p_{i,j}^{sam}) \tag{17}$$

where $KD()$ denotes the knowledge distillation loss. In this paper, we use Jensen–Shannon divergence.

In summary, the final loss of the proposed HKPD mainly consists of two terms, including the classification loss and the self-distillation loss, which can be described as:

$$L_{total} = \sum_{i=0}^{2}\alpha_i L_i + \beta L_{self} \tag{18}$$

where a_i and β are weighting factors. The whole process to compute the final loss L_{total} during the meta-training stage

Algorithm 1 Meta-training stage of the proposed HKPD

Input: θ: the parameters of HKPD; $\mathbf{D_{base}}$: a base dataset for meta-training; L: the max training iteration; $learning\_rate$: the learning rate of BP algorithm;
Output: the trained parameters θ of HKPD;

while $l < L$ **do**
  Sample $\mathbf{S_{base}}$ and $\mathbf{Q_{base}}$ from $\mathbf{D_{base}}$;
  Compute vanilla sample representation $\mathbf{X_0^S}$ by Eq(4) and vanilla prototype $\mathbf{P_0}$ by Eq(2);
  Compute first prediction $p_{i,j}^0$ by Eq(15);
  Compute first classification loss: $L_0 = \sum_j^N CE(p_{i,j}^0, y_i)$;
  Compute refined sample representation $\mathbf{X_{sam}}$ by Eq(9);
  Compute second prediction $p_{i,j}^{sam}$ by Eq(16);
  Compute second classification loss: $L_1 = \sum_j^N CE(p_{i,j}^{sam}, y_i)$;
  Compute refined class representation $\mathbf{P_{cla}}$ by Eq(11);
  Compute third prediction $p_{i,j}^{cls}$ by Eq(14);
  Compute third classification loss: $L_2 = \sum_j^N CE(p_{i,j}^{cla}, y_i)$;
  Compute the self-distillation loss L_{self} by Eq(17);
  Compute the final loss: $L_{total} = \sum_{i=0}^{2} \alpha_i L_i + \beta L_{self}$;
  Update parameters by BP algorithm: $\theta \leftarrow \theta - learning\_rate * \frac{\partial L_{total}}{\partial \theta}$;
end while

is provided in Algorithm 1. Finally, it is worth noting that our model does not rely on the number of query samples. The HKPD does not require fine-tuning during the inference period and can directly predict the query sets through the trained model.

5. Experiments

In this section, we evaluate our proposed HKPD framework on widely used few-shot learning benchmarks and compare it with current state-of-the-art methods. Then, we analyze the functions of each of the modules in this framework..

5.1. Datasets and Setups

We conduct experiments on three datasets, including MiniImageNet (Vinyals et al., 2016), TieredImageNet (Ren, Triantafillou, Ravi, Snell, Swersky, Tenenbaum, Larochelle and Zemel, 2018), and CIFAR-FS (Dhillon et al., 2020; Krizhevsky, Hinton et al., 2009).

MiniImageNet is a subset of the ILSVRC-12 dataset (Russakovsky et al., 2015), and we strictly follow the standard process to split the datasets (Snell et al., 2017; Yu et al., 2022) for a fair comparison. The MiniImageNet contains 100 classes and 60,000 images, in which 64 classes are assigned to the base dataset, 16 classes to the validation dataset, and 20 classes to the novel dataset, respectively.

TieredImageNet is another larger subset of ILSVRC-12 dataset compared with the MiniImageNet, containing 608 classes and 779,165 images, and its categories belong to one of 34 higher-level categories sampled from the ILSVRC-12 dataset to make the base and novel datasets more semantically different. We also followed the previous split proposed by (Ren et al., 2018), in which 351, 97, and 160 classes are used for the base, validation, and novel datasets, respectively.

CIFAR-FS is derived from the CIFAR-100 dataset (Krizhevsky et al., 2009), containing 100 classes and 600 images per class. Following previous works (Dhillon et al., 2020), we use 64 classes to construct the base dataset, 16 classes for the validation dataset, and the remaining 20 classes for the novel dataset.

Experimental setup. In this section, we provide a detailed description of our implementation and experimental settings. For each dataset, we train our model using the base dataset and evaluate its performance on the novel dataset. Furthermore, we employ the commonly-used ResNet-12 (He et al., 2016; Ye et al., 2020; Yu et al., 2022) with four residual blocks as our feature extraction module in the HKPD. The obtained representations, which will be fed into the next module, are extracted from the penultimate layer of this feature extraction module. Unless otherwise specified, the dimension of obtained representation we used is 512. Additionally, following previous work (Ye et al., 2020; Yu et al., 2022), we pre-train our feature extraction module with the base dataset to accelerate the convergence. Then, the meta-training strategy is taken to train the model.

We employ the Adam optimizer (Kingma and Ba, 2015) in all experiments with the initial learning rate set to 1e-3, and the weight decay set to 5e-4. And the learning rate will be decreased with a factor of 0.1 at every 5,000 meta-training iterations. We empirically set the hyper-parameters as following: $\alpha_0 = 0$, $\alpha_1 = 0.5$, $\alpha_2 = 1$ and $\beta = 1e3$. For evaluation, we evaluate our method in 5 way-1 shot/5 shot settings on the novel dataset and randomly sample 10,000 few-shot tasks from it. Then, we report the top-1 mean accuracy (%) with the 95% confidence interval.

Besides the aforementioned baselines, we also assessed various other recent SOTA few-shot learning methods under the inductive setting, including MetaOptNet (Lee et al., 2019), Robust (Dvornik et al., 2019), Self-Supervise (Gidaris et al., 2019), AFHN(Li et al., 2020), MTL(Wang et al., 2021), MCRNet-RR (Zhong et al., 2021), ArL(Zhang et al., 2021a), FRN(Wertheimer et al., 2021), and DCAP (He et al., 2022a). All experiments we conducted are implemented with Pytorch on a single NVIDIA RTX3090 GPU.

5.2. Main Results

As detailed in Tables 1, 2, and 3, we compare the performance of our proposed HKPD with current state-of-the-art FSL methods under the inductive setting. From these experimental results, we can observe that: (1) Our proposed method obtained satisfactory accuracy on the 5-way 1-shot learning and 5-way 5-shot learning and outperforms previous approaches on the MiniImageNet, TieredImageNet, and CIFAR-FS datasets. For instance, the accuracy of HKPD is 74.14% on the TieredIMageNet for the 5-way 1-shot

Methods	Fine-tuning	Backbone	MiniImageNet	
			5-way 1-shot	5-way 5-shot
ProtoNet (Snell et al., 2017)		ResNet-12	60.34 ± 1.20	80.54 ± 1.13
MAML (Finn et al., 2017)	✓	ResNet-12	58.05 ± 0.10	72.41 ± 0.20
MetaOptNet (Lee, Maji, Ravichandran and Soatto, 2019)		ResNet-12	64.09 ± 0.62	80.00 ± 0.45
Robust (Dvornik, Schmid and Mairal, 2019)	✓	ResNet-18	63.73 ± 0.62	81.19 ± 0.43
Fine-tuning (Dhillon et al., 2020)	✓	Wide-ResNet	57.73 ± 0.62	78.17 ± 0.49
Self-Supervised (Gidaris, Bursuc, Komodakis, Pérez and Cord, 2019)	✓	Wide-ResNet	62.93 ± 0.45	79.87 ± 0.33
FEAT (Ye et al., 2020)		ResNet-12	66.78 ± 0.20	82.05 ± 0.14
Neg-Margin (Liu et al., 2020a)	✓	ResNet-12	63.85 ± 0.76	81.57 ± 0.56
RFS (Tian et al., 2020)	✓	ResNet-12	62.02 ± 0.63	79.64 ± 0.44
AFHN (Li, Zhang, Li and Fu, 2020)	✓	ResNet-18	62.38 ± 0.72	78.16 ± 0.53
MTL (Wang, Zhao and Li, 2021)	✓	ResNet-12	59.84 ± 0.22	77.72 ± 0.09
MCRNet-RR (Zhong, Gu, Huang, Li, Chen and Lin, 2021)	✓	ResNet-12	61.32 ± 0.64	78.16 ± 0.49
MCRNet-SVM (Zhong et al., 2021)	✓	ResNet-12	62.53 ± 0.64	80.34 ± 0.47
ArL (Zhang, Koniusz, Jian, Li and Torr, 2021a)	✓	ResNet-12	65.21 ± 0.58	80.41 ± 0.49
FRN (Wertheimer, Tang and Hariharan, 2021)		ResNet-12	66.45 ± 0.19	82.83 ± 0.13
MixtFSL (Afrasiyabi et al., 2021)	✓	ResNet-12	63.98 ± 0.79	82.04 ± 0.49
MixtFSL (Afrasiyabi et al., 2021)	✓	Wide-ResNet	64.31 ± 0.79	81.66 ± 0.60
DCAP (He, Hong, Liu, Xu and Sun, 2022a)		ResNet-12	65.20 ± 0.67	80.93 ± 0.53
HGNN (Yu et al., 2022)		ResNet-12	67.02 ± 0.20	83.00 ± 0.13
HKPD (Ours)		ResNet-12	$\mathbf{68.80 \pm 1.15}$	$\mathbf{84.18 \pm 0.93}$

Table 1
: 5-way 1-shot and 5-way 5-shot classification accuracy (%) and 95% confidence interval on MiniImageNet Dataset. The best results are reported in bold font.

Methods	Fine-tuning	Backbone	TieredImageNet	
			5-way 1-shot	5-way 5-shot
ProtoNet (Snell et al., 2017)		ResNet-12	69.63 ± 0.53	84.82 ± 0.36
MAML (Finn et al., 2017)	✓	ResNet-12	63.85 ± 0.76	81.57 ± 0.56
MetaOptNet (Lee et al., 2019)		ResNet-12	65.81 ± 0.74	81.75 ± 0.53
Robust (Dvornik et al., 2019)	✓	ResNet-18	70.44 ± 0.32	85.43 ± 0.21
Fine-tuning (Dhillon et al., 2020)	✓	Wide-ResNet	66.58 ± 0.70	85.55 ± 0.48
Self-Supervised (Gidaris et al., 2019)	✓	Wide-ResNet	70.53 ± 0.51	84.98 ± 0.36
FEAT (Ye et al., 2020)		ResNet-12	66.78 ± 0.20	82.05 ± 0.14
RFS (Tian et al., 2020)	✓	ResNet-12	71.52 ± 0.69	86.03 ± 0.49
MTL (Wang et al., 2021)	✓	ResNet-12	67.11 ± 0.12	83.69 ± 0.02
FRN (Wertheimer et al., 2021)		ResNet-12	71.16 ± 0.22	86.01 ± 0.15
MixtFSL (Afrasiyabi et al., 2021)	✓	ResNet-12	70.97 ± 1.03	86.16 ± 0.67
DCAP (He et al., 2022a)		ResNet-12	70.15 ± 0.74	85.33 ± 0.55
HGNN (Yu et al., 2022)		ResNet-12	72.05 ± 0.23	86.49 ± 0.15
HKPD (Ours)		ResNet-12	$\mathbf{74.14 \pm 1.03}$	$\mathbf{87.12 \pm 0.89}$

Table 2
:5-way 1-shot and 5-way 5-shot classification accuracy (%) and 95% confidence interval on TieredImageNet Dataset. The best results are reported in bold font.

learning tasks, which presents a relative improvement of 2.9%, compared to the HGNN method proposed in 2022. (2) Even though we use a ResNet-12 as the feature extraction module, the proposed HKPD can still surpass some methods, such as Fine-tuning (Dhillon et al., 2020) and MixtFSL (Afrasiyabi et al., 2021), which utilize more powerful Wide-ResNet network (Zagoruyko and Komodakis, 2016). (3) It's worth noting that although our HKPD does not need to fine-tune the parameters during the reference period, it can still perform satisfactorily. However, as summarized in Tables 1, 2, and 3, most inductive-based FSL methods rely on fine-tuning to transfer knowledge from the base dataset, which may limit their application within source-limited environments. Furthermore, compared to the recent HGNN method (Yu et al., 2022), which could only infer one query sample at a time, our HKPD can efficiently infer all query samples every time, making it more applicable in real-world scenarios.

Tables 1, 2, and 3 have demonstrated the satisfactory experimental results of our proposed HKPD, which applied the standard few-shot training strategy, which means the number of shots and ways are exactly the same in both the

Methods	Fine-tuning	Backbone	CIFAR-FS	
			5-way 1-shot	5-way 5-shot
ProtoNet (Snell et al., 2017)		ResNet-12	72.20 ± 0.70	83.50 ± 0.50
MetaOptNet (Lee et al., 2019)		ResNet-12	68.72 ± 0.67	86.11 ± 0.47
Fine-tuning (Dhillon et al., 2020)	✓	Wide-ResNet	72.00 ± 0.70	84.20 ± 0.50
Self-Supervised (Gidaris et al., 2019)	✓	Wide-ResNet	73.62 ± 0.31	86.05 ± 0.22
Neg-Margin (Liu et al., 2020a)	✓	ResNet-12	63.85 ± 0.76	81.57 ± 0.56
RFS (Tian et al., 2020)	✓	ResNet-12	71.50 ± 0.80	86.00 ± 0.50
AFHN (Li et al., 2020)	✓	ResNet-18	68.32 ± 0.93	81.45 ± 0.87
MTL (Wang et al., 2021)	✓	ResNet-12	69.50 ± 0.30	84.10 ± 0.20
MCRNet-RR (Zhong et al., 2021)	✓	ResNet-12	73.80 ± 0.70	85.20 ± 0.50
MCRNet-SVM (Zhong et al., 2021)	✓	ResNet-12	74.70 ± 0.70	86.80 ± 0.50
Ours (HKPD)		ResNet-12	$\mathbf{76.16 \pm 1.26}$	$\mathbf{88.22 \pm 0.84}$

Table 3
5-way 1-shot and 5-way 5-shot classification accuracy (%) and 95% confidence interval on CIFAR-FS Dataset. The best results are reported in bold font.

Training Setting	MiniImageNet		TieredImageNet		CIFAR-FS	
	1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
Standard Setting	68.80	84.18	74.14	87.12	76.16	88.22
Higher-shot	**69.68**	**84.28**	**74.66**	**87.38**	**76.76**	**88.56**
Relative Improvement (%)	1.3	0.1	0.7	0.3	0.8	0.4

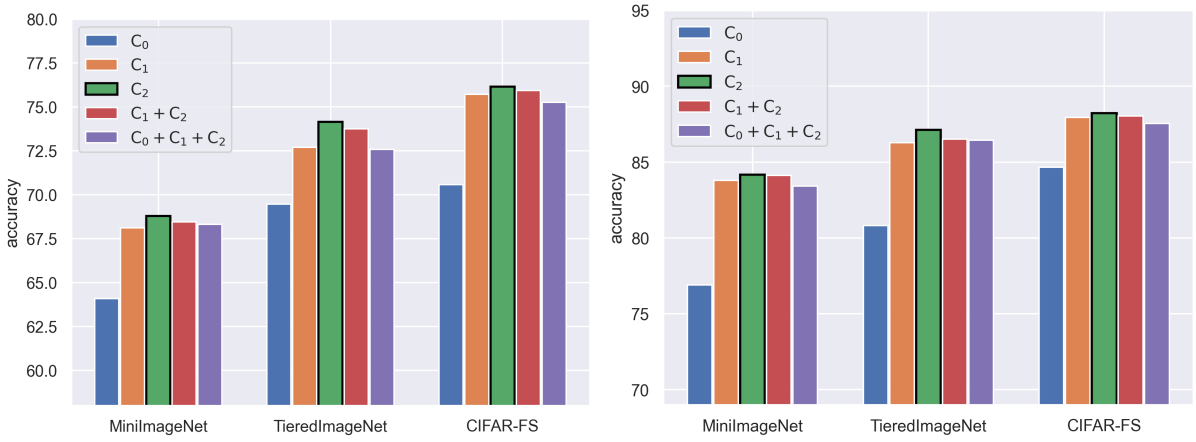
Table 4
The performance comparisons between the standard few-shot learning and the higher shot few-shot learning

meta-training and meta-testing period (Snell et al., 2017; Finn et al., 2017). Here we aim to further explore the performance of our proposed method on these benchmarks. Therefore, we adopt the "Higher Shot" setting proposed in (Liu et al., 2019) to re-evaluate our HKPD, which means we adopt Higher Shot during the meta-training period while keeping the number of shots and ways the same during the meta-testing period. In more detail, we adopt the 5-shot and 10-shot meta-training for 1-shot and 5-shot meta-testing, respectively. We emphasize that the Higher Shot setting is adopted only for the base dataset, and the evaluation settings are the same as before without affecting the fairness of comparisons. We use the same three FSL datasets here, and the experimental results in the Higher Shot settings are presented in Table 4. We can observe that the performances of our HKPD continue to improve to varying degrees across all three FSL benchmarks when the Higher-shot is applied in the meta-training stage. Notably, for the 1-shot case in the MiniImageNet dataset, the accuracy increases from 68.80% to 69.68%, with a relative improvement of 1.3%. These improvements show that HKPD can effectively acquire more transferable knowledge to help predict unknown testing samples when more labeled samples are available.

5.3. Ablation Studies

As explained above, our HKPD method includes a deep feature extraction module, sample-level information propagation module, and class-level propagation module, each with a different classifier (denoted by C_0, C_1, C_2 for short). We aim to show the effectiveness of different classifiers in the proposed modules, and report the accuracy of the outputs obtained by these three classifiers in Figures 3(a) and 3(b). Notably, we observe that the predictions of C_2, produced by the class-level information propagation module, can achieve a better performance directly, both in 5-way 1-shot learning and 5-way 5-shot learning on all three datasets, compared with the C_0 and C_1. Therefore, due to its outstanding performance, we choose the output of the classifier C_2 as our final prediction in the proposed HKPD. Our results also show that the proposed hierarchical framework is valid by simultaneously learning the relations between sample-level and class-level representations. Furthermore, we also observe that the output of C_1 could also obtain competitive results in these benchmarks. Both Figure 3(a) and 3(b) also shows that the accuracy of the ensemble voting of C_2 with other classifiers (such as, combine C_1, C_2 denoted as C_1+C_2; combine C_0, C_1, C_2 denoted as $C_0 + C_1 + C_2$), which do not to improve the performance. We suspect that this may be caused by the self-distillation module, which makes the outputs similar and may weaken the effectiveness of the voting.

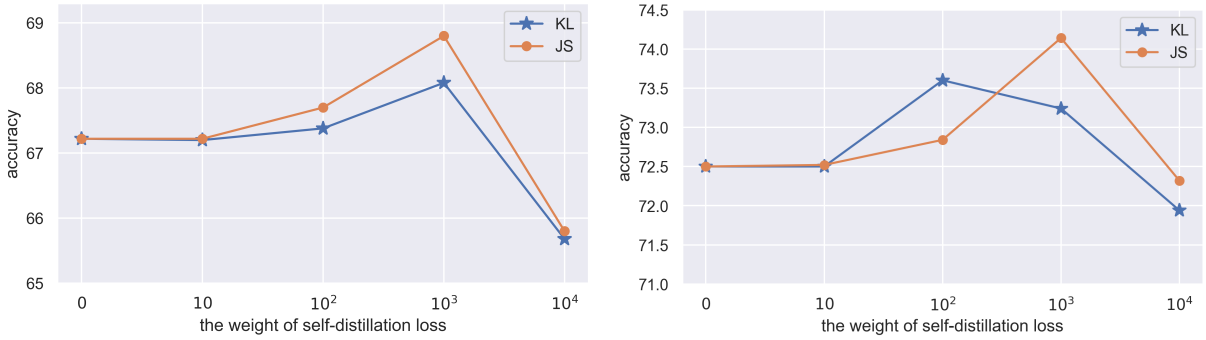
Next, we conduct a detailed test to evaluate the impact of our proposed self-distillation module on the performance of our model. Fig 4(a) and 4(b) show the results of the 5-way 1-shot learning on the MiniImageNet and TieredImageNet. Recall that the self-distillation module contains a hyperparameter β, serving as a weighting factor of the self-distillation loss. When we set $\beta = 0$, the self-distillation module has no effect. Notably, we can observe similar phenomena from these two subfigures that choosing an appropriate weighting factor β of self-distillation loss improves the performance significantly both on MiniImageNet and



(a) The accuracy of 5-way 1-shot learning

(b) The accuracy of 5-way 5-shot learning

Figure 3: The performance comparisons of the different classifiers, from the feature extraction module, sample-level propagation module and class-level propagation module respectively, denoted by C_0, C_1, and C_2; $C_1 + C_2$ means the voting of C_1 and C_2; $C_0 + C_1 + C_2$ means the voting of C_0, C_1 and C_2.



(a)

(b)

Figure 4: The impacts of the self-distillation loss in 5-way 1-shot learning on the MiniImageNet and TieredImageNet

TieredImageNet, compared to the results without using the self-distillation module. For instance, the accuracy of the predictions increases from 67.22% to 68.80% in 5-way 1-shot learning on the MiniImageNet dataset when we set $\beta = 1e3$. These results show the effectiveness of our self-distillation module in HKPD. They are also consistent with the conclusions in (Zhang et al., 2019) and (Cho and Hariharan, 2019), that models can benefit from their inner supervision signals and even a classifier with weak performance can serve as a teacher to improve the performance further. It can be easily found that too-large a value of β may deteriorate the performance because a large self-distillation loss will make all outputs of modules similar, causing less discrimination. Additionally, we compare different distillation losses. These two subgraphs show that the JS divergence achieves better accuracy than KL divergence in most cases. Therefore, we choose JS divergence as our self-distillation loss.

Further, we explore the impact of the proposed masking matrix $\mathbf{M}$ on the performance. Therefore, Figure 5 shows the prediction accuracy of our proposed HKPD with the different number of chosen samples for propagation in an FSL task, which also equals the number of non-zero elements in every column in the masking matrix $\mathbf{M}$. As shown, the number of chosen samples will influence the prediction accuracy. When set equal to 15, Our model can achieve the best performance in 5-way 5-shot learning on both MiniImageNet and TieredImageNet. When the number of chosen samples equals 25, the performance will decrease, meaning that all the samples in an FSL task will be considered in the propagation process, which may learn from several uncorrelated samples and cause noise. Consequently, we can validate the effectiveness of the proposed masking matrix $\mathbf{M}$, which can help to filter some uncorrelated noise.

6. Conclusion

In this paper, we emphasized the significance of inductive Few-Shot Learning and introduced a novel and effective FSL method, HKPD, designed for the inductive setting. HKPD could simultaneously leverage sample-level and

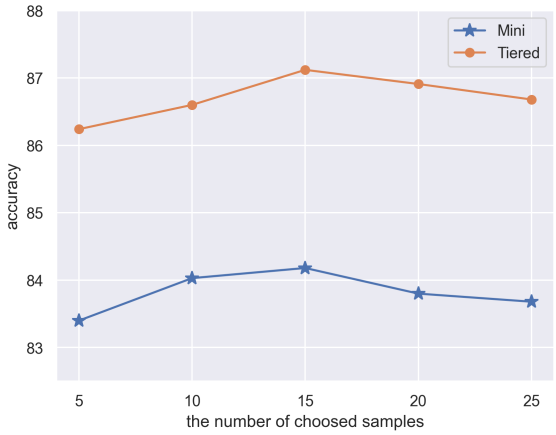


Figure 5: The impact of the masking matrix **M** on classification results. The horizontal axis denotes the number of samples for propagation in few-shot learning, correlation the number of non-zero elements in every column in the masking matrix **M**.

class-level information to capture more useful information in FSL tasks. We added a self-distillation module to allow the hierarchical information to propagate across different modules in this hierarchical structure, further improving the learned representations. Furthermore, HKPD is not dependent on finetuning, making it applicable to a broader range of application scenarios compared to most inductive-based FSL methods. We conducted extensive experiments and demonstrated that the proposed HKPD achieved superior performance on the FSL benchmarks. Based on our results, we argue that low-level information (e.g., fine-grained features) in the field of FSL is worth more careful consideration.

Acknowledgments

This research was supported by Zhejiang Provincial Key Research and Development Program of China under Grant No. 2021C01106.

References

Afrasiyabi, A., Lalonde, J.F., Gagné, C., 2021. Mixture-based feature space learning for few-shot image classification, in: Proc. of ICCV.

Aggarwal, C.C., et al., 2018. Neural networks and deep learning. Springer 10, 3.

Ba, J.L., Kiros, J.R., Hinton, G.E., 2016. Layer normalization. ArXiv preprint abs/1607.06450.

Chen, C., Yang, X., Xu, C., Huang, X., Ma, Z., 2021. Eckpn: Explicit class knowledge propagation network for transductive few-shot learning, in: Proc. of CVPR.

Chen, W., Liu, Y., Kira, Z., Wang, Y.F., Huang, J., 2019. A closer look at few-shot classification, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net.

Cho, J.H., Hariharan, B., 2019. On the efficacy of knowledge distillation, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, IEEE. pp. 4793–4801.

Dhillon, G.S., Chaudhari, P., Ravichandran, A., Soatto, S., 2020. A baseline for few-shot image classification, in: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020, OpenReview.net.

Dvornik, N., Schmid, C., Mairal, J., 2019. Diversity with cooperation: Ensemble methods for few-shot classification, in: Proceedings of the IEEE/CVF international conference on computer vision, pp. 3723–3731.

Fei-Fei, L., Fergus, R., Perona, P., 2006. One-shot learning of object categories. IEEE transactions on pattern analysis and machine intelligence 28, 594–611.

Finn, C., Abbeel, P., Levine, S., 2017. Model-agnostic meta-learning for fast adaptation of deep networks, in: Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017, PMLR. pp. 1126–1135.

Furlanello, T., Lipton, Z., Tschannen, M., Itti, L., Anandkumar, A., 2018. Born again neural networks, in: International Conference on Machine Learning, PMLR. pp. 1607–1616.

Gidaris, S., Bursuc, A., Komodakis, N., Pérez, P., Cord, M., 2019. Boosting few-shot visual learning with self-supervision, in: Proceedings of the IEEE/CVF international conference on computer vision, pp. 8059–8068.

He, J., Hong, R., Liu, X., Xu, M., Sun, Q., 2022a. Revisiting local descriptor for improved few-shot classification. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM) 18, 1–23.

He, K., Chen, X., Xie, S., Li, Y., Dollár, P., Girshick, R., 2022b. Masked autoencoders are scalable vision learners, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pp. 16000–16009.

He, K., Gkioxari, G., Dollar, P., Girshick, R., 2017. Mask r-cnn, in: Proceedings of the IEEE International Conference on Computer Vision (ICCV).

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition, in: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, IEEE Computer Society. pp. 770–778.

Hinton, G., Vinyals, O., Dean, J., et al., 2015. Distilling the knowledge in a neural network. ArXiv preprint abs/1503.02531.

Hu, Y., Gripon, V., Pateux, S., 2021. Graph-based interpolation of feature vectors for accurate few-shot classification, in: Proc. of ICPR.

Kim, J., Kim, T., Kim, S., Yoo, C.D., 2019. Edge-labeling graph neural network for few-shot learning, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, Computer Vision Foundation / IEEE. pp. 11–20.

Kingma, D.P., Ba, J., 2015. Adam: A method for stochastic optimization, in: 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings.

Krizhevsky, A., Hinton, G., et al., 2009. Learning multiple layers of features from tiny images .

Lake, B.M., Salakhutdinov, R., Tenenbaum, J.B., 2015. Human-level concept learning through probabilistic program induction. Science 350, 1332–1338.

Lee, K., Maji, S., Ravichandran, A., Soatto, S., 2019. Meta-learning with differentiable convex optimization, in: IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019, Computer Vision Foundation / IEEE. pp. 10657–10665.

Li, K., Zhang, Y., Li, K., Fu, Y., 2020. Adversarial feature hallucination networks for few-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 13470–13479.

Li, X., Sun, Z., Xue, J.H., Ma, Z., 2021. A concise review of recent few-shot meta-learning methods. Neurocomputing 456, 463–468.

Lin, T., Goyal, P., Girshick, R.B., He, K., Dollár, P., 2017. Focal loss for dense object detection, in: IEEE International Conference on Computer Vision, ICCV 2017, Venice, Italy, October 22-29, 2017, IEEE Computer Society. pp. 2999–3007.

Liu, B., Cao, Y., Lin, Y., Li, Q., Zhang, Z., Long, M., Hu, H., 2020a. Negative margin matters: Understanding margin in few-shot classification, in: Proc. of ECCV.

Liu, J., Song, L., Qin, Y., 2020b. Prototype rectification for few-shot learning, in: Proc. of ECCV.

Liu, L., Hamilton, W.L., Long, G., Jiang, J., Larochelle, H., 2021. A universal representation transformer layer for few-shot image classification, in: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021, OpenReview.net.

Liu, Y., Lee, J., Park, M., Kim, S., Yang, E., Hwang, S.J., Yang, Y., 2019. Learning to propagate labels: Transductive propagation network for few-shot learning, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net.

Ma, Y., Bai, S., An, S., Liu, W., Liu, A., Zhen, X., Liu, X., 2020. Transductive relation-propagation network for few-shot learning., in: IJCAI, pp. 804–810.

Pennington, J., Socher, R., Manning, C.D., 2014. Glove: Global vectors for word representation, in: Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP), pp. 1532–1543.

Ren, M., Triantafillou, E., Ravi, S., Snell, J., Swersky, K., Tenenbaum, J.B., Larochelle, H., Zemel, R.S., 2018. Meta-learning for semi-supervised few-shot classification, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., et al., 2015. Imagenet large scale visual recognition challenge. International journal of computer vision .

Snell, J., Swersky, K., Zemel, R.S., 2017. Prototypical networks for few-shot learning, in: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp. 4077–4087.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R., 2014. Dropout: a simple way to prevent neural networks from overfitting. The journal of machine learning research 15, 1929–1958.

Tian, Y., Wang, Y., Krishnan, D., Tenenbaum, J.B., Isola, P., 2020. Rethinking few-shot image classification: a good embedding is all you need?, in: Proc. of ECCV.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I., 2017. Attention is all you need, in: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA, pp. 5998–6008.

Vinyals, O., Blundell, C., Lillicrap, T., Wierstra, D., et al., 2016. Matching networks for one shot learning. Advances in neural information processing systems 29.

Wang, H., Zhao, H., Li, B., 2021. Bridging multi-task learning and meta-learning: Towards efficient training and effective adaptation, in: International Conference on Machine Learning, PMLR. pp. 10991–11002.

Wang, Y., Yao, Q., Kwok, J.T., Ni, L.M., 2020. Generalizing from a few examples: A survey on few-shot learning. ACM computing surveys (csur) .

Wertheimer, D., Tang, L., Hariharan, B., 2021. Few-shot classification with feature map reconstruction networks, in: Proc. of CVPR.

Yang, L., Li, L., Zhang, Z., Zhou, X., Zhou, E., Liu, Y., 2020. DPGN: distribution propagation graph network for few-shot learning, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, IEEE. pp. 13387–13396.

Ye, H., Hu, H., Zhan, D., Sha, F., 2020. Few-shot learning via embedding adaptation with set-to-set functions, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020, IEEE. pp. 8805–8814.

Ying, Z., You, J., Morris, C., Ren, X., Hamilton, W.L., Leskovec, J., 2018. Hierarchical graph representation learning with differentiable pooling, in: Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada, pp. 4805–4815.

Yu, T., He, S., Song, Y.Z., Xiang, T., 2022. Hybrid graph neural networks for few-shot learning, in: Proc. of AAAI.

Zagoruyko, S., Komodakis, N., 2016. Wide residual networks, in: Proceedings of the British Machine Vision Conference 2016, BMVC 2016, York, UK, September 19-22, 2016, BMVA Press.

Zhang, H., Koniusz, P., Jian, S., Li, H., Torr, P.H., 2021a. Rethinking class relations: Absolute-relative supervised and unsupervised few-shot learning, in: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition, pp. 9432–9441.

Zhang, L., Song, J., Gao, A., Chen, J., Bao, C., Ma, K., 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation, in: 2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019, Seoul, Korea (South), October 27 - November 2, 2019, IEEE. pp. 3712–3721.

Zhang, Z., Bu, J., Ester, M., Zhang, J., Li, Z., Yao, C., Huifen, D., Yu, Z., Wang, C., 2021b. Hierarchical multi-view graph pooling with structure learning. IEEE Transactions on Knowledge and Data Engineering .

Zhong, X., Gu, C., Huang, W., Li, L., Chen, S., Lin, C.W., 2021. Complementing representation deficiency in few-shot image classification: A meta-learning approach, in: 2020 25th International Conference on Pattern Recognition (ICPR), IEEE. pp. 2677–2684.

Zhu, H., Koniusz, P., 2022. Ease: Unsupervised discriminant subspace learning for transductive few-shot learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pp. 9078–9088.