Information Sciences 639 (2023) 119012

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

# Hierarchical few-shot learning with feature fusion driven by data and knowledge

Zhiping Wu [a,b], Hong Zhao [a,b,*]

[a] School of Computer Science, Minnan Normal University, Zhangzhou, Fujian, 363000, China
[b] Key Laboratory of Data Science and Intelligence Application, Fujian Province University, Zhangzhou, Fujian, 363000, China

ARTICLE INFO

*Keywords:*
Few-shot learning
Hierarchical classification
Granular computing
Feature fusion
Data- and knowledge-driven

ABSTRACT

Few-shot learning (FSL) aims to use only a few samples to learn a model and utilizes that model to identify unseen classes. Recent, metric-based feature fusion methods mainly focus on the fusion of inter-layer features and show superior performance in solving FSL problems. However, due to the data scarcity in FSL, existing methods still face severe challenges in obtaining high-quality sample features for the improvement of classification performance. In this paper, we propose a hierarchical metric FSL model with comprehensive feature fusion driven by data and knowledge (HFFDK), which is based on intra-layer channel-feature and hierarchical class structure perspectives. First, we utilize the network hierarchy to construct an intra-layer channel feature fusion, which transfers the intra-layer fused features of the higher layer to the lower layer. The model can extract high-quality sample features in a data-driven manner. Moreover, we focus on different levels of granularity to obtain various levels of information, while hierarchical class structures can provide both coarse- and fine-grained information in a knowledge-driven manner. Then, we utilize the coarse-grained information to assist fine-grained recognition. Finally, we optimize hierarchical FSL with coarse- and fine-grained relational constraints and similarity measures among samples. Experiments on four benchmark datasets show that HFFDK achieves state-of-the-art performance.

## 1. Introduction

Few-shot learning (FSL) aims to learn a model that can be applied to new tasks where the known label class information for the related tasks is limited [9]. This process mimics how the human brain associates prior knowledge so that it can make inferences when judging unknown things. The main practice of FSL in image classification is to use one or a few samples to learn to recognize invisible classes [29]. Although the images from common classes are easily accessible, there may be many visual data-scarce classes, such as endangered animal and plant species. FSL usually needs to overcome the problems of labeled data being rarely available and data collection being expensive. In summary, the main challenge of FSL is that few training samples are available, while it needs to adapt to a large number of unknown classes.

An FSL method based on metric learning has been widely studied and has achieved excellent results. The metric-based learning method aims to design a measurement model of support and query samples to make similar samples closely related and heterogeneous samples separate [11]. For example, Snell et al. [34] proposed a method for constructing the prototype of each class and then

* Corresponding author at: School of Computer Science, Minnan Normal University, Zhangzhou, Fujian, 363000, China.
*E-mail address:* hongzhaocn@163.com (H. Zhao).

https://doi.org/10.1016/j.ins.2023.119012
Received 13 June 2022; Received in revised form 19 April 2023; Accepted 22 April 2023

Available online 28 April 2023

employed the Euclidean distance metric to query the distance between the sample and the prototype. Sung et al. [37] calculated similarity scores via a relation network to measure the distances between support and query set samples. Similarly, Vinyals et al. [39] introduced the idea of nearest neighbor classification and used the cosine distance to measure the distance between sample features. In addition, Koch et al. [11] employed $L_1$ distances to measure image similarity. Although these metric-based few-shot learning methods have made impressive progress, they mainly focus on the metric among samples while ignoring the significance of sample features for the metric.

Recently, feature extraction has been widely used in metric-based FSL. High-quality sample features help a model to classify when using a limited number of class samples for learning, so extracting sample features is particularly important. For instance, Li et al. [15] proposed an adaptive super-pixel feature extraction network, which can extract representative prototypes via the fusion of similar feature vectors. Ding et al. [4] designed a relational network to extract multi-scale image features, which are extracted by fusing the features of the different convolutional inter-layers of the network. Similarly, Shermin et al. [33] proposed a two-step dense attention mechanism that optimizes inter-layer features with mutually beneficial information to discover attribute-guided local visual features. Besides, Zhang et al. [49] employed a saliency network to segment the foreground and background of image samples, obtain the corresponding sample features, and use feature fusion to generate new training data.

The aforementioned methods show that the features extracted from samples play an important role in measuring the distance between samples in metric-based learning. Feature extraction plays a crucial role in model construction that is based on measuring the similarity among samples. Most existing methods use feature fusion to fuse features between inter-layers so that the output features have a better representation. However, there are often multiple channels in each network layer, and the information among channels may be associate. These methods ignore feature fusion among channels in the intra-layer, which is critical and challenging in FSL due to the scarcity of samples. Unlike the method proposed by Zhang et al. [46], we relieve the learning pressure of the network (ResNet12) by mapping fully connected layers and exploiting both coarse- and fine-grained relations to improve the representation of visual features.

In this paper, we propose a hierarchical metric-based FSL model with intra-layer feature fusion driven by data and knowledge (HFFDK). Specifically, the HFFDK model can provide high-quality features and facilitate better intra-layer channel fusion features than the inter-layer through hierarchical knowledge. Firstly, we embedded an intra-layer channel feature fusion module into each network layer of a ResNet12 backbone and performed iterative fusion in the inter-layer. This fusion process is data-driven that transfers the image detail features fused in higher intra-layers to lower layers to obtain high-quality features. There is a relationship between coarse- and fine-grained classes in the hierarchical class structure. Then, we use the coarse-grained information to assist fine-grained recognition according to the coarse-grained close to the fine-grained center characteristic, which is driven by prior knowledge of the hierarchical class structure. Finally, we utilize two hyper-parameters to fit the hierarchical FSL model. One is used to constrain the relationship between the coarse and fine granularities, and the other is utilized to optimize the metric relationship between the support and query sets.

To verify the effectiveness of our proposed HFFDK model, we compared its performance with those of existing FSL methods on four benchmark datasets. The experimental results demonstrate that HFFDK is effective in hierarchical classification. Especially compared with the suboptimal method, HFFDK obtains a 1.420% improvement in F-measure and a 2.953% improvement of accuracy in an experiment of 5-way 1-shot on *CIFAR-FS*. We have uploaded the HFFDK code to GitHub (https://github.com/fhqxa/HFFDK). The major contributions of this paper are summarized as follows:

- From a data perspective, we construct an intra-layer channel feature fusion module to transfer the intra-layer feature and obtain a high-quality image feature. Moreover, unlike traditional inter-layer feature fusion, this can preserve the main information of image features in the intra-layers.
- From a knowledge perspective, we fully mine the hierarchical structure knowledge among classes. On the one hand, this is utilized as auxiliary information for classification. On the other hand, it drives the model with data to learn better class features.
- Experimental results on four datasets demonstrate that the proposed HFFDK model can improve the classification accuracy by 1∼4%.

The remainder of this paper is organized as follows: Section 2 presents a brief review of the work related to the HFFDK. Section 3 elaborates on the details of the proposed method. Section 4 introduces the experimental settings of the datasets, implementation details, comparison methods, and evaluation metrics. Then, the experimental results and an analysis are given in Section 5. Finally, Section 6 concludes the paper and suggests ideas for further work.

## 2. Related work

In this section, we briefly summarize related work into two categories: (1) feature fusion and (2) few-shot learning.

### 2.1. Feature fusion

Feature fusion combines features from different layers or branches to transfer the feature information from higher layers to lower layers. This makes the features more representative of the original class information [43]. Traditional machine learning uses feature selection to handle the dimensionality of the data and represents all the features with a small number of representative features [22]. In deep learning, multi-layer networks are often designed to enable a model to extract an effective feature representation [23]. Feature

fusion between network layers is a ubiquitous part of modern network architectures [24]. The attention fusion mechanism used in deep learning is a part of the feature fusion method. It mimics the human visual attention mechanism and top-down observation habits. It can prompt the model to retain high-layer semantic features as well as low-layer visual information. For example, Jiang et al. [9] fused the rich semantic features of the high-layer with the high-resolution visual information of the low-layer to obtain distinctive features. Li et al. [19] explored the potential associations between images and labels and mined the unions among image contexts. Moreover, Piao et al. [28] designed an effective depth thinning block using residual connections for complete extraction and fused multi-layer paired complementary cues from depth streams and RGB. Inspired by the combination of visual and semantic knowledge mentioned in the above methods, we focus on the fusion of global and local features within the network layer to fully mine the upper layer information and transfer it to the lower layer.

### 2.2. Few-shot learning

Few-shot learning purpose is to use only a few labeled samples to recognize unlabeled samples in invisible classes [35]. Currently, FSL can be mainly grouped into: transfer-based learning [25], meta-based learning [41], and metric-based learning [47].

The essence of transfer-based learning is to extract the knowledge learned on task $A$ and apply it to a new task $B$, which can make the model converge faster and decrease the workload required to re-collecting training data for task $B$ [27]. The primary idea of few-shot transfer learning is to adjust the pre-trained model parameters leveraging only a few labeled samples. This is done by changing the training tasks with new SoftMax layers after freezing the parameters of the first few layers of the pretrained network. For example, Sun et al. [36] proposed a meta-transfer learning method that migrates a pretrained deep neural network through the scaling and shifting of weights. It improves the knowledge transfer ability of the model and enables the model to conveniently convert from tasks other to few-shot classification tasks. Besides, Li et al. [14] designed a transferable visual feature model that learns transferable visual features by exploiting class hierarchies that encode semantic relationships between source and target classes. To obtain predictive prototypes at different granularity levels in few-shot classification tasks, Liu et al. [21] proposed a Prototype Propagation Network (PPN) with concept relations. Li et al. [18] proposed a transfer network combining vision and knowledge classifiers to use semantic information between related classes fully. Peng et al. [27] and Xing et al. [42] suggested combining visual and semantic features to improve FSL classification performance. Similarly, Schwartz et al. [32] aimed to embed multiple pieces of semantic information into an FSL framework to improve the few-shot classification performance. Inspired by the above methods, we aimed to utilize the semantic information of the data itself instead of clustering and adhere to the idea of lightweight design. We construct a coarse- and fine-grained classification-loss process by mapping the fully connected layer of the network. The purpose is to guide feature fusion learning utilizing coarse-grained knowledge.

Meta-based learning tries to transfer some “meta-knowledge” from previously learned pre-trained network tasks, so it is also called learning to learn. This can facilitate the rapid learning of new tasks. Meta-learning is popular among FSL approaches [30]. It aims to learn a model that can be directly adapted to multi-task learning scenarios. Based on a meta-concept inference network, Zhang et al. [46] cleverly leveraged the prior knowledge of a coarse class set to construct a classifier with different layers of categories to further improve few-shot classification performance. For instance, Rusu et al. [31] proposed a latent embedding optimization learning model that learns a data-dependent low-dimensional latent generation representation of the model parameters by transfer and performs meta-learning in the low-dimensional latent space. Zhang et al. [47] introduced semantic knowledge to extract representative features as a meta-learning prototype representative, which effectively solves the few-shot fine-tuning feature improvement problem. Additionally, Zhang et al. [45] investigated an interpretable decision tree scheme for FSL, which effectively solves the black-box problem of FSL classification in deep networks. Moreover, Qin et al. [29] integrated multi-instance learning to propose a meta-learning method that divides the original image into small pieces, focusing on the location of the category object and greatly reducing the model parameters.

Metric-based learning aims to make the intra-class samples close and the inter-class samples separate. It measures the distances between samples to learn a shared feature representation space with different tasks [34,37,39]. For example, Wang et al. [41] employed a statistical method of case reliability reasoning, which measures the reliability of pseudo-labels and improves the spatial distribution of pseudo-labels for FSL. Similarly, Zhang et al. [48] designed a crossover mechanism to measure the excellent matching angle between image regions, which can effectively reduce the influence of background and intra-class appearances. In addition, Kang et al. [10] proposed an attentional observation-based metric that learns reliable co-attention between images by measuring their transferable structural patterns. In our work, we consider two methods for measuring the simplicity and flexibility of learning classes, namely RENet [10] and DeepEMD [48]. Moreover, the proposed intra-layer channels feature fusion and hierarchical loss processes are compatible with several other few-shot classification solutions.

## 3. Hierarchical few-shot learning with feature fusion driven by data and knowledge

In the cognition and processing of real-world problems, humans adopt a strategy of observing and analyzing problems from at different levels. This section proposes a hierarchical FSL method from a data- and knowledge-driven perspective (HFFDK).

### 3.1. Basic framework

In few-shot classification, the data is divided into two parts: training data $D_{train}$ and test data $D_{test}$. The training data $D_{train}$ is from the training classes $c_{train}$ and $D_{test}$ is from the test classes $c_{test}$, where $c_{train} \cap c_{test} = \phi$ and $D_{train} \cap D_{test} = \phi$. According to the

**Table 1**
A summary of the main symbols.

| Symbols | Definition | Symbols | Definition |
|---|---|---|---|
| $D_{train}$ | training data | $\sigma(\cdot)$ | sigmoid function |
| $D_{test}$ | test data | $B(\cdot)$ | batch normalization |
| $c_{train}$ | coarse-grained classes in training set | $L(\cdot)$ | compute local channels attention function |
| $c_{test}$ | coarse-grained classes in test set | $G(\cdot)$ | compute global channels attention function |
| $f_{train}$ | fine-grained classes in training set | $\mathbf{X}$ | base feature map |
| $f_{test}$ | fine-grained classes in test set | $\mathbf{F}$ | fusion feature |
| $(x, y)$ | samples | $\mathbf{W}$ | weight |
| $N$ | number of classes | $\mathbf{I}_s$ and $\mathbf{I}_q$ | interested map |
| $K$ | number of samples | $\lambda$ and $\beta$ | hyper-parameter |

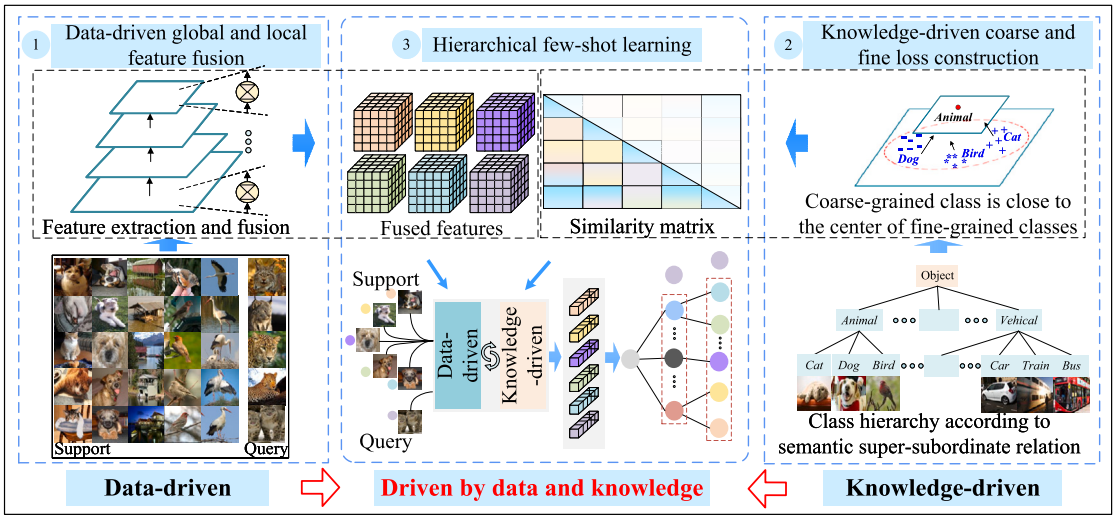



**Fig. 1.** Framework of the HFFDK model. The data-driven module fuses the global and local features of the intra-layer and transmits them to the next layer. The knowledge-driven module is guided by semantic knowledge to construct a coarse- and fine-grained loss structure that aims to assist fine-grained recognition. Data and knowledge jointly drive the hierarchical few-shot learning module to generate more discriminative features.

semantic knowledge of the benchmark datasets, the coarse-grained classes include the fine-grained classes. There is no intersection between training $f_{train}$ and test classes $f_{test}$, where $f_{train} \cap f_{test} = \phi$. Both training and test processes contain multiple episodes. Each episode is divided into a support set $\mathcal{Q} = \{(x_q^{(l)}, y_q^{(l)})\}_{l=1}^{n}$ and a query set $\mathcal{S} = \{(x_s^{(l)}, y_s^{(l)})\}_{l=1}^{m}$, where $n = NK$, $m = NK'$, $N$ is the number of classes, $K$ is the number of query set samples for each class, and $K'$ is the number of support set samples for each class. This is defined as $N$-way $K$-shot tasks (e.g., 5-way 1-shot tasks or 5-way 5-shot tasks), which is an episode setting in FSL studies. During the training process, we randomly extract class samples from $D_{train}$ and aim to train the model that can correctly predict from $(\mathcal{S}, x_q^{(n)})$ to $y_q^{(n)}$. In summary, the main formula symbols mentioned in this paper are listed in Table 1.

Human cognition is a coarse- to fine-grained transformation process. The process can be simulated by considering class granularity in the process of hierarchical transformation from coarse- to fine-grained. The basic flowchart of HFFDK is shown in Fig. 1. This model is mainly grouped into three parts:

(1) Data-driven global and local feature fusion. Image samples perform feature fusion in global and local channels intra-layers through forward computation between the network inter-layers. This allows the features extracted from the image retain detailed information to obtain high-quality features.

(2) Knowledge-driven coarse and fine loss construction. HFFDK utilizes coarse-grained information to assist fine-grained recognition according to semantic knowledge super-subordinate relations.

(3) Hierarchical FSL. On the one hand, we employ the parameter $\lambda$ to trade-off between the contributions of the coarse and fine granularities in hierarchical classification. On the other hand, we optimize the matching relationship between the support and query samples.

### 3.2. Data-driven global and local feature fusion

In this section, we introduce a process of global and local feature fusion of intra-layer channels (IGLF) to obtain better image data features. Inspired by SA-Net [50], we design an IGLF module from the perspective of the data, as shown in Fig. 2. First, we embed the IGLF module into each layer of the backbone network of a ResNet12 network. Specifically, given an input image $x$, ResNet12 first extracts feature map $f^l(x) = \mathbf{X}_l \in \mathbb{R}^{C \times H \times W}$, where parameter $l$ represents the $l^{th}$ layer, and $C$, $H$, and $W$ indicate the channel

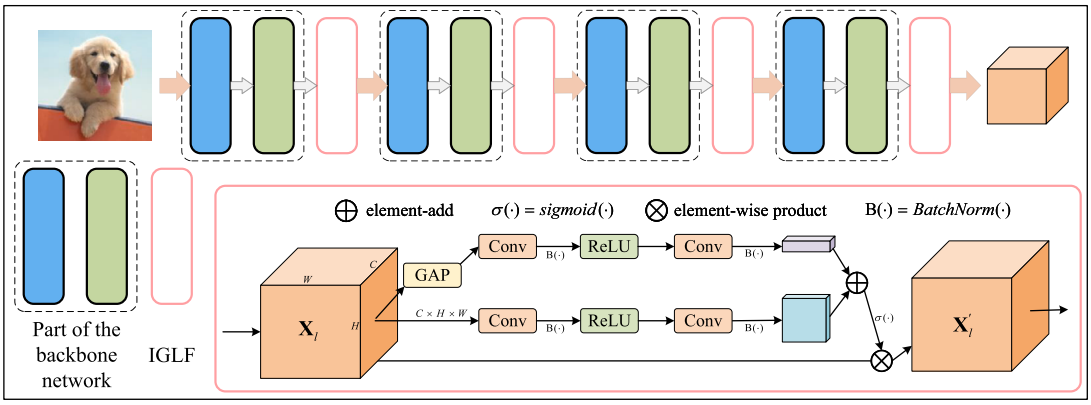



**Fig. 2.** Intra-layer channels global and local feature fusion. Top: The IGLF module is embedded into the backbone network of ResNet12 to extract image features. Bottom: Detailed design details of the IGLF module. The parameters $C$, $H$ and $W$ represents the image channel number, spatial height, and width, respectively. The GAP means global average pool.

number, spatial height, and width, respectively. Then, we put $\mathbf{X}_l$ into the fusion module to obtain the global and local fusion features of the layer, which is formulated as

$$\begin{aligned} f_\alpha^l(\mathbf{X}_l) &= f_\alpha^l(f_\alpha^{(l-1)}(\mathbf{X}_l)) \\ &= \mathbf{X}'_l, \end{aligned} \tag{1}$$

where $f_\alpha^0(\mathbf{X}_l) == \mathbf{X}_l$ when $l = 1$, $\mathbf{X}_l$ represents the feature map of the $l^{th}$ layer of ResNet12, and parameter $\alpha$ is the optimization parameter of the IGLF module. Deep network forward calculation will cause the characteristics of the fusion of the upper network affect the calculation of the features of the lower network. We use the following formula for forward convolution calculation:

$$\begin{aligned} & f^l(\mathbf{X}'_{l-1}) = \mathbf{X}_l \\ & \quad s.t. \quad l > 1, \end{aligned} \tag{2}$$

where $\mathbf{X}'_{l-1}$ is the feature map formed by fusion of the $l-1$ layer feature's global and local channels, $\mathbf{X}_l$ is the $l$ layer feature's map in ResNet12. To better describe the IGLF module, we use the following steps:

First, we use point-wise convolution $W_p^T$ as a local channel attention calculation, which only exploits the point-wise channel interactions for each spatial position. We use the convolution block $W_{p_d}^T$ of kernel size $C \times \frac{C}{r} \times 1 \times 1$ (We set the parameter $r = 4$) first to reduce the dimensionality of the image features obtained by the $l$ layer $W_{p_d}^T(\mathbf{X}_l)$, where $d$ represents the reduced dimension convolution block parameter, which is automatically learned by the network. We utilize the activation function of the Rectified Linear Unit (ReLU) to enhance the nonlinear relationship between the network layers $\delta(B(W_{p_d}^T(\mathbf{X}_l)))$, where $\delta(\cdot)$ denotes ReLU, and $B(\cdot)$ denotes representative batch normalization (BN).

Second, we use the convolution block $W_{p_u}^T$ of the kernel size $\frac{C}{r} \times C \times 1 \times 1$ (We set the parameter $r = 4$) to upgrade the dimensionality of the image features obtained by the $l$ layer $W_{p_u}^T(\delta(B(W_{p_d}^T(\mathbf{X}_l))))$, where $u$ represents the upgraded dimension convolution block parameter, which is automatically learned by the network. We continuously optimize the IGLF module parameters through forward calculation.

Finally, we define the local channel attention computation via a bottleneck structure as follows:

$$L(\mathbf{X}_l) = B(W_{p_u}^T(\delta(B(W_{p_d}^T(\mathbf{X}_l))))), \tag{3}$$

where $W_{p_u}^T$ and $W_{p_d}^T$ have kernel sizes $\frac{C}{r} \times C \times 1 \times 1$ and $C \times \frac{C}{r} \times 1 \times 1$, respectively, $r$ is a positive constant, $p_u$ and $p_d$ represent different convolutions, parameter $\delta$ is the ReLU, and $B(\cdot)$ is the BN. We add a global average pool (GAP) to the global channel attention module to preserve some details in each layer. The GAP is computed as follows:

$$g(\mathbf{X}_l) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} \mathbf{X}_l[:, i, j], \tag{4}$$

where $\mathbf{X}_l$ is the feature map extracted by the $l^{th}$ layer of ResNet12, $\mathbf{X}_l[:, i, j]$ represents the position of each pixel in the feature map except the channel, and $i, j$ indicate each pixel's position. There are other ways to transfer the feature information of the upper layer of the network to the next layer, such as the residual network design idea proposed by He et al. [8]. However, the main purpose of our GAP design is to keep the module lightweight while providing auxiliary information to the lower layer. We use the GAP to enable each layer of the network to retain the detailed feature information of the image part during the forward convolution process.

Similarly, we perform point-wise convolution $\tilde{W}_p^T$, $B(\cdot)$ and ReLU $\delta(\cdot)$ after the GAP feature. We define the global channel attention computation through the bottleneck structure as follows:

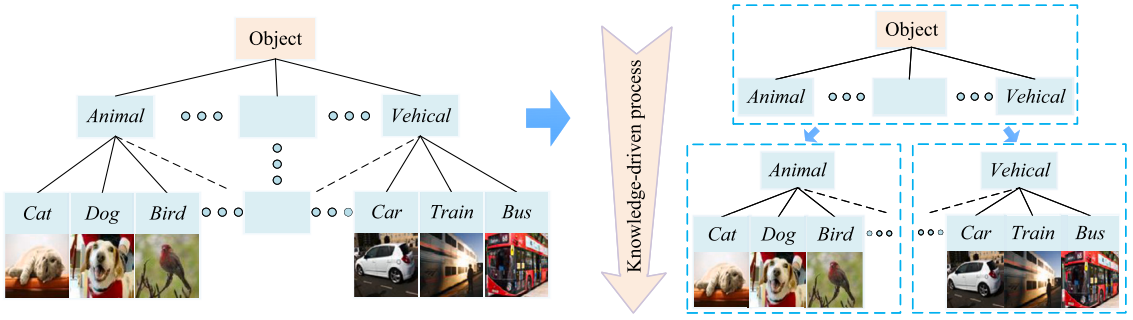



**Fig. 3.** A knowledge-driven process of coarse-grained assistance to fine-grained recognition.

$$G(\mathbf{X}_l) = B(\tilde{W}_{p_u}^T(\delta(B(\tilde{W}_{p_d}^T(g(\mathbf{X}_l)))))), \tag{5}$$

where $\tilde{W}_{p_u}^T$ and $\tilde{W}_{p_d}^T$ have kernel sizes $\frac{C}{r} \times C \times 1 \times 1$ and $C \times \frac{C}{r} \times 1 \times 1$, respectively. Their attention positions on the image are different when calculating the global and local channel attention features. The function $L(\mathbf{X}_l)$ preserves and optimizes the fine details of the low-layer features of the network. The function $G(\mathbf{X}_l)$ enriches the high-layer semantics information through forward calculation of the network. The output feature of function $L(\mathbf{X}_l)$ has the same shape as the input feature $\mathbf{X}_l$. We fuse the global and local attention features of the intra-layer channels as follows:

$$\mathbf{X}_l' = \mathbf{X}_l \otimes \sigma(L(\mathbf{X}_l) \oplus G(\mathbf{X}_l)), \tag{6}$$

where $\otimes$ is a representative element-wise product, $\oplus$ is an element addition and $\sigma(\cdot)$ is a sigmoid function. We use the Sigmoid function to calculate the fused eigenvalues and perform a weighted operation with $\mathbf{X}_l$. This is done to transfer the fusion information of the local and global channel features of this layer to the next layer. Then, we put the obtained features $\mathbf{X}_l'$ into the network for forward calculation. Finally, we obtain the global and local attention fusion feature of the final intra-layer channels $\mathbf{F}$ via forward calculation in ResNet12.

### 3.3. Knowledge-driven coarse and fine loss construction

From a philosophical point of view, when humans perceive, measure, conceptualize, or reason about anything, the idea of granular thinking is applicable. A fundamental task of granular computing is to observe views according to different levels of granularity. We need to shift the perspective of the problem accordingly when moving from one level of detail to another. We map the features of the different granularities that occur at fully connected layers [40]. Different representations of the same problem at different granularity levels can be linked by mapping. In general, different levels of mapping can be utilized to classify and explore different types of granulations. Through the fully connected layer, we map the feature $\mathbf{F}$ fused by the network into coarse-grained $\mathbf{F}_c$ and fine-grained $\mathbf{F}_f$ features. According to the hierarchical tree structure of classes, we transform the fine-grained classification task into a sub-classification task via coarse-grained auxiliary classification [5]. In the classification tasks, we can explore such properties by eliminating the study of subtasks in coarse-grained spaces to improve problem-solving efficiency. Fig. 3 shows an example of a knowledge-driven process of coarse-grained assistance to fine-grained recognition.

According to this hierarchical semantic structure, the fine-grained class *Dog* belongs to the coarse-grained class *Animal*, and the fine-grained class *Car* belongs to the coarse-grained class *Vehical*. We employ coarse-grained knowledge to assist fine-grained recognition. The coarse-grained similarity probability of the samples $x \in D_{train}$ is calculated as follows:

$$P_c(x) = \frac{\exp(\mathbf{W}_c^T \mathbf{F}_c)}{\sum_{c'=1}^{n_c} \exp(\mathbf{W}_{c'}^T \mathbf{F}_{c'})}, \tag{7}$$

where $n_c = |c_{train}|$ represents the number of coarse granularity $c_{train}$, $[\mathbf{W}_1^T, \cdots, \mathbf{W}_{n_c}^T]$ are the coarse-grained fully-connected layer weights, and feature $\mathbf{F}_c \in \mathbb{R}^{C \times H \times W}$ represents the coarse-grained feature of the backbone network mapping query images. We define the coarse-grained classification loss calculated by cross-entropy $\mathcal{L}_{CE}(\cdot)$, which guides the correct classification between query samples and coarse-grained $c \in c_{train}$ representations:

$$\mathcal{L}_c = \mathcal{L}_{CE}(P_c(x), y_c). \tag{8}$$

The goal of classification is to correctly identify fine-grained classes. Fine-grained classification can be made more accurate with the help of coarse-grained discrimination results. The fine-grained similarity probability of samples $x$ is calculated as follows:

$$P_f(x) = \frac{\exp(\mathbf{W}_f^T \mathbf{F}_f)}{\sum_{f'=1}^{n_f} \exp(\mathbf{W}_{f'}^T \mathbf{F}_f)}, \tag{9}$$

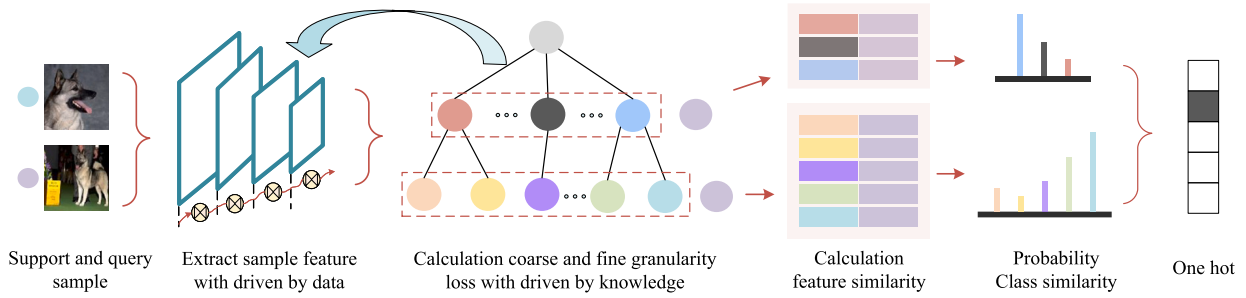



**Fig. 4.** An example of hierarchical few-shot learning driven by data and knowledge.

where $n_f = |f_{train}|$ represents the number of fine granularity $f_{train}$, feature $\mathbf{F}_f \in \mathbb{R}^{C\times H\times W}$ represents the fine-grained feature of the backbone network mapping query images, and $[\mathbf{W}_1^T, \cdots, \mathbf{W}_{n_f}^T]$ indicate the weights of fine-grained fully-connected layers. The model guided by loss correctly distinguishes the fine-grained $f \in f_{train}$ of the query sample, which is defined as follows:

$$\mathcal{L}_f = \mathcal{L}_{CE}(P_f(x), y_f). \tag{10}$$

The classification loss constructed at coarse and fine granularities is guided by the semantic knowledge of the hierarchy. It is part of the whole HFFDK model, while data-driven global and local feature fusion is another part. We aim to jointly drive data and knowledge so that the HFFDK can learn more discriminative features.

### 3.4. Hierarchical few-shot learning driven by data and knowledge

The process of information granulation is the main inspiration for the design of our hierarchical FSL model. Information granulation is a natural process of human thinking and reasoning, which applies to almost all human activities. Granular construction and decomposition operations are crucial issues in granular computing. We construct coarse- and fine-grained hierarchies with semantic knowledge inherent in the data. Our feature extraction module fuses features intra-layers and enables the fused features to transfer information to the next layer through forward computation. We design a data-driven module that uses the channel information in the fusion intra-layer to promote the features extracted by the model to have a high-quality representation, while the knowledge-driven module can use the hierarchical class semantic knowledge of the data itself to assist model classification. The sample features extracted by the model facilitate the granularity discrimination of the model, and the result of granularity discrimination facilitates the model to extract distinguishing sample features, as shown in Fig. 4.

We transfer the informational deviation in the intra-layer to the next layer so that the final output features can facilitate coarse-grained recognition and correct fine-grained classification. This process is completed under the guidance of two aspects: 1) The data-driven method promotes the feature fusion of intra-layer channels (Section 3.2) and 2) the knowledge-driven approach uses the semantic knowledge of the class hierarchy to guide model learning (Section 3.3). According to the sample features of the support set and query set extracted by the model, we define the loss calculation of the similarity feature of the support and query sets as follows:

$$\mathcal{L}_k = -\log \frac{\exp(\mathrm{sim}(\bar{\mathbf{s}}^{(k)}, \bar{\mathbf{q}}^{(k)})/\tau)}{\sum_{k'=1}^{n_k} \exp(\mathrm{sim}(\bar{\mathbf{s}}^{(k')}, \bar{\mathbf{q}}^{(k')})/\tau)}, \tag{11}$$

where $\mathrm{sim}(\cdot)$ is cosine similarity, $n_k$ represents the number of test query samples, $\bar{\mathbf{s}}^{(k)}$ represents a set of prototype embedding, which is the average of $K$ support vectors, $\bar{\mathbf{q}}^{(k)}$ represents the average of the $K$ query embedding vectors, participating in the $k^{th}$ support context of the $N^{th}$ class, and $\tau$ is a scalar temperature factor.

We control coarse- and fine-grained auxiliary knowledge participation in the hierarchical tree through the hyper-parameter $\lambda$. The hyper-parameter $\beta$ is used to alter the global and local fusion features between channels. The hierarchical FSL loss function is defined as follows:

$$\mathcal{L} = \sum_{f=1}^{n_f} \mathcal{L}_f + \lambda \sum_{c=1}^{n_c} \mathcal{L}_c + \beta \sum_{k=1}^{n_k} \mathcal{L}_k, \tag{12}$$

where $n_f$ is the number of fine-grained classes, $n_c$ is the number of coarse-grained classes, and $n_k$ is the number of test query samples.

Finally, a detailed procedure for hierarchical FSL with data- and knowledge-driven feature fusion is provided as Algorithm 1. It provides the training model and pseudo-code for a training episode. First, we randomly select training samples from the training set and feed them into ResNet12 to extract the fusion features listed at lines 3∼7. We calculate the coarse- and fine-grained similarity probabilities and losses, which are listed at lines 8∼9. Then, we compute the similarity metric loss between the support and query samples listed at line 10. Finally, we compute the hierarchical FSL loss listed in line 11 and update the parameters and final loss listed at lines 12∼13.

**Algorithm 1** Hierarchical few-shot learning with feature fusion driven by data and knowledge (HFFDK).

**Input**: Training $D_{train}$ and test data $D_{test}$; learning rates of inner loop, momentum, and weight decay; hyper-parameters $\lambda$ and $\beta$.
**Initialization**: Randomly initialize model parameters $\mathbf{W}_c^T$, and $\mathbf{W}_f^T$.
**Iteration**:

**for** each episode **do**
  $(S_l, Q_l)$ =RandomSample($D_{train}$);
  Put the $(S_l, Q_l)$ into ResNet12;
  **for** each network layer **do**
    Fuse the local $L(\mathbf{X})$ and global $G(\mathbf{X})$ attention features of the intra-layer channels according to Eq. (6);
  **end for**
  Obtain the fusion features $\mathbf{F}$ to map $\mathbf{F}_c$ and $\mathbf{F}_f$;
  Compute coarse-grained $P_c(x)$ and fine-grained $P_f(x)$ similarity probability;
  Compute a query sample coarse and fine granularity loss $\mathcal{L}_c$ and $\mathcal{L}_f$ according to Eqs. (8) and (10);
  Measure the similarity between the query set and the support set prototype, according to Eq. (11);
  Compute the hierarchical few-shot learning loss $\mathcal{L}$ according to Eq. (12);
  Update parameters $\mathbf{W}_c^T$, and $\mathbf{W}_f^T$;
  Return backward loss $\mathcal{L}$;
**end for**

**Table 2**
Benchmark datasets description. Symbol "-" means that there is no data for this item.

| Item | tieredImageNet | miniImageNet | CIFAR-FS | FC100 |
|---|---|---|---|---|
| structure | Y | N | Y | Y |
| depth | 3 | 1 | 3 | 3 |
| coarse training/val/test | 20/6/8 | - | 20/11/13 | 12/4/4 |
| fine training/val/test | 351/97/160 | 64/16/20 | 60/16/20 | 60/20/20 |
| all fine-grained classes | 608 | 100 | 100 | 100 |
| sample | 779,165 | 60,000 | 60,000 | 60,000 |
| size | $84 \times 84 \times 3$ | $84 \times 84 \times 3$ | $32 \times 32 \times 3$ | $32 \times 32 \times 3$ |

**Table 3**
tieredImageNet dataset description.

| Item | Training | Validation | Test | Total |
|---|---|---|---|---|
| Coarse | 20 | 6 | 8 | 34 |
| Fine | 351 | 97 | 160 | 608 |
| Images | 448695 | 124261 | 206209 | 779165 |

## 4. Experimental settings

In this section, we describe the datasets and implementation details used in our experiment. Then, we introduce several state-of-the-art methods and evaluation measures.

### *4.1. Datasets*

We selected four benchmark datasets that have been widely used in previous work: tieredImageNet, miniImageNet, CIFAR-FS, and Few-shot CIFAR100 (FC100). Their details are listed in Table 2.

(1) **tieredImageNet:** This dataset was originally proposed in [30] and is a large subset of the ILSVRC-12 dataset. It consists of 779,165 color images in 608 classes. Following [30], we divide its classes into 20/6/8 coarse-grained classes for training/validation/testing, respectively. We split 351/97/160 fine-grained classes based on these super-sets. The number of coarse- and fine-grained classes in the tieredImageNet dataset used in the training, validation, and test sets are listed in Table 3. The hierarchical tree structure of the tieredImageNet dataset is shown in Fig. 5.

(2) **miniImageNet** [39] is a subset of the ImageNet dataset. It consists of 60,000 color images in 100 classes. Following [39], we grouped the dataset into 64, 16, and 20 classes for training/validation/test fine-grained classes, respectively.

(3) **CIFAR-FS:** This dataset [1] was generated from the CIFAR100 dataset with the same criteria as used for miniImageNet. Following [30], we used the same training/validation/test splits consisting of 20/11/13 coarse-grained classes, respectively. These coarse-grained classes were grouped into 64, 16, and 20 fine-grained classes used for the training, validation, and test sets, respectively. The hierarchical tree structure of the CIFAR-FS dataset is shown in Fig. 6.

(4) **FC100:** This dataset [26] is built from the CIFAR100 dataset leveraging similar criteria as tieredImageNet. Separation based on coarse-grained object categories is a challenging scenario due to the lower image resolution and more challenging meta-training/test segmentation. It consists of 100 classes, each containing 600 samples. We split it into training/validation/test sets following [26], which consisted of 12/4/4 coarse-grained classes and 60/20/20 fine-grained classes, respectively.

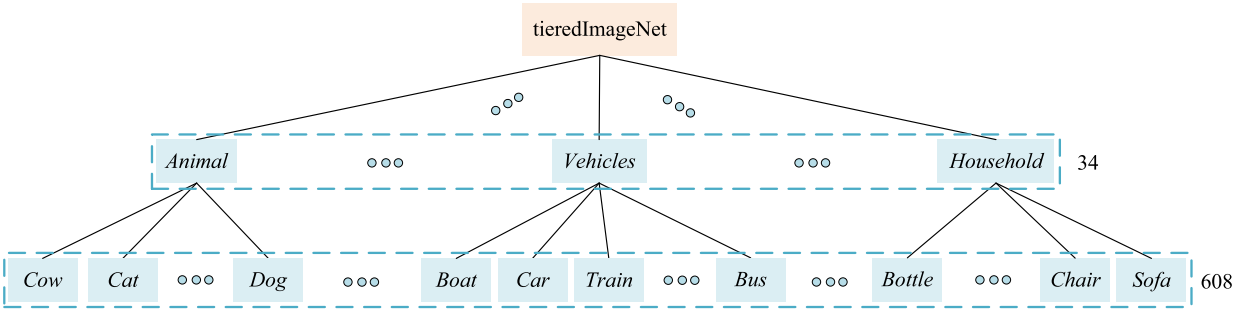



**Fig. 5.** Tree structure of the tieredImageNet dataset.

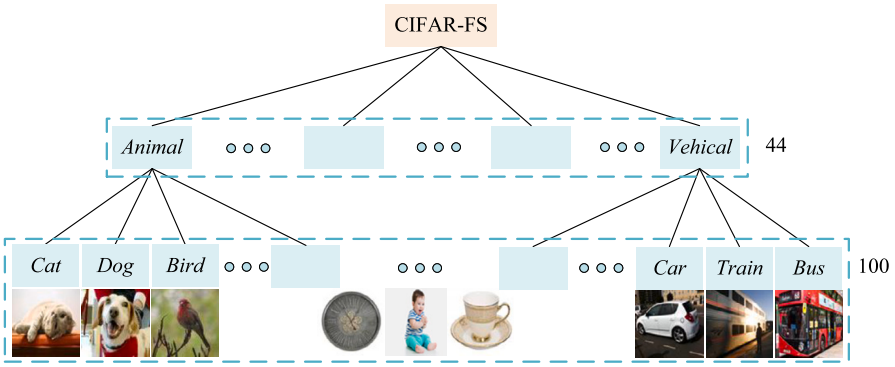



**Fig. 6.** Tree structure of the CIFAR-FS dataset.

### 4.2. Implementation details

We adopt ResNet12 as the backbone network following recent few-shot classification work [44,48]. We input images of size $84 \times 84$ pixels into the HFFDK network feature extraction module, which obtains a base representation $\mathbf{F} \in \mathbb{R}^{5\times5\times640}$. We add IGLF modules after each layer of the backbone network to ensure the base representation is of high-quality. We adopt the SGD optimizer with an initial learning rate of $10^{-1}$, a momentum of 0.9, and a weight decay of 0.005 to train the proposed model [10].

We set the parameter milestones to 60 and 70 in the 5-way 1-shot experiments. Similarly, we set the parameter milestones to 40 and 50 in the 5-way 5-shot experiments. Therefore, we utilize recent and reliable evaluation settings for the $N$-way $K$-shot evaluations. In the test, we use 15 query samples from each class for test in each episode. We utilize random sampling of 2,000 samples per episode to evaluate the mean test accuracy and 95% confidence. The hyper-parameter $\lambda$ is set to 0.65 and 0.4 for the 5-way 1-shot and 5-way 5-shot datasets, respectively. The hyper-parameter $\beta$ is set to 0.25, 0.5, and 0.5 for tieredImageNet, FC100, and CIFAR-FS, respectively. We set $\rho = 2$, $\tau = 0.2$, and $U, V = 5$. All our experimental environments were run on desktop computers under the Ubuntu 20.04 operating system, using an NVIDIA GTX3090 graphics card with 24.0 GB video memory and a 2.40 GHz × 24 Intel Xeon Silver 4214R CPU.

### 4.3. Comparison methods

In this subsection, we compare HFFDK with several few-shot learning methods. The details of them are introduced as follows:

(1) Boosting [6]: This approach combines the complementarity of few-shot learning and self-supervision to improve feature representations.

(2) Cosine classifier [2]: This method proposes a standard fine-tuning baseline approach to make a fair comparison between different methods.

(3) wDAE-GNN [7]: This method proposes a denoised automatic encoder network that changes feature distribution to Gaussian distribution and reconstructs the target recognition classification weight.

(4) MetaOptNet [12]: This method exploits two properties of linear classifiers that use high-dimensional embedding to generalize well.

(5) ProtoNet [34]: This approach constructs a prototype for each class and then measures the distance between the prototype and the query samples using the Euclidean distance.

(6) RFS-simple [38]: This method proposes an extremely simple baseline that consists of a linear model learned on top of a pre-trained embedding.

(7) CTM [16]: This approach proposes a class traversal module that could find task-relevant features.

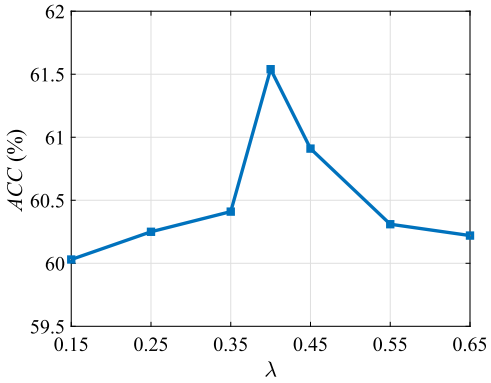



**Fig. 7.** Effectiveness of different values of the parameter $\lambda$ in the 5-way 5-shot on the FC100 dataset.

(8) FEAT [44]: This method embeds the instance into the target classification task through a set-to-set function to make it specific and discriminative.

(9) RENet [10]: This method computes the correlation between two image feature representations using self-correlational and cross-correlation.

### 4.4. Evaluation metrics

**Tree-induced error (*TIE*)**: We measure the distance between the predicted and the correct label in the hierarchical tree structure through *TIE* [20], which is defined as follows:

$$TIE(y, \hat{y}) = |E_h(y, \hat{y})|, \tag{13}$$

where $y$ is the true label of the sample $x$ and $\hat{y}$ is the predicted label of the sample $x$.

**Hierarchical F-measure evaluation ($F_H$)**: We evaluation of the hierarchical structure by $F_H$ [20], which is defined as follows:

$$\begin{cases} F_H(y, \hat{y}) = \dfrac{2 \cdot P_H R_H}{P_H + R_H}, \\ P_H(y, \hat{y}) = \dfrac{|Y_{aug} \bigcap \hat{Y}_{aug}|}{|Y_{aug}|}, \\ R_H(y, \hat{y}) = \dfrac{|Y_{aug} \bigcap \hat{Y}_{aug}|}{|\hat{Y}_{aug}|}, \end{cases} \tag{14}$$

where $P_H$ means hierarchical precision, $R_H$ represents recall rate, $Y_{aug}$ represents the set of the true class, and $\hat{Y}_{aug}$ indicates the set of the predicted class, including its coarse nodes.

## 5. Experimental results and analysis

In this section, we present the experimental results and a discussion from five perspectives. In Section 5.1, we compare the impacts of different values of parameters $\lambda$ and $\beta$ on HFFDK. In Sections 5.2∼5.4, we describe an ablation study conducted to analyze the role of each module. In Section 5.5, we compare several methods to demonstrate the effectiveness of the proposed HFFDK method. In Section 5.6, we perform t-SNE feature visualization of the proposed HFFDK method.

### 5.1. Performance comparison with different values of parameters $\lambda$ and $\beta$

In this section, we explore the effects of using different values of parameters $\lambda$ and $\beta$ on the effectiveness of HFFDK. First, we fix parameter $\beta$ to 0.5 to study the effect of various values of parameter $\lambda = \{0.15, 0.25, 0.35, 0.40, 0.45, 0.55, 0.65\}$. We discuss the impacts on HFFDK classification via a series of 5-way 5-shot experiments on the FC100 dataset. Then, we fix parameter $\lambda$ to 0.65 to study the effect of using various values of parameter $\beta = \{0.2, 0.3, 0.4, 0.4, 0.5, 0.6, 0.7, 0.8\}$. For this, we utilize a series of 5-way 1-shot experiments on the CIFAR-FS dataset.

Fig. 7 shows the effect of different values of $\lambda$ on the effectiveness of HFFDK, while Table 4 lists the *TIE* and $F_H$ values. We can make the following observations:

(1) A series of 5-way 5-shot experiments was carried out on the FC100 dataset with a fixed value of parameter $\beta$. The *ACC* increases from 60.03% to 61.54% as $\lambda$ increases from 0.15 to 0.40. This means that coarse-grained participation plays an auxiliary role in fine-grained recognition. Besides, we also observe that too much coarse-grained participation will hinder fine-grained recognition. It is obvious that the *ACC* decreases from 61.54% to 60.22% as $\lambda$ increases from 0.40 to 0.65.

**Table 4**
Result of different values of the parameter $\lambda$ in the 5-way 5-shot on the FC100 dataset (%). The best results of each set are highlighted in bold.

| $\lambda$ | | 0.15 | 0.25 | 0.35 | 0.4 | 0.45 | 0.55 | 0.65 |
|---|---|---|---|---|---|---|---|---|
| *TIE* | ↓ | 0.4162 | 0.4099 | 0.4041 | **0.3869** | 0.3994 | 0.4011 | 0.4131 |
| $F_H$ | ↑ | 89.05 | 89.63 | 89.87 | **90.33** | 90.02 | 89.94 | 89.61 |

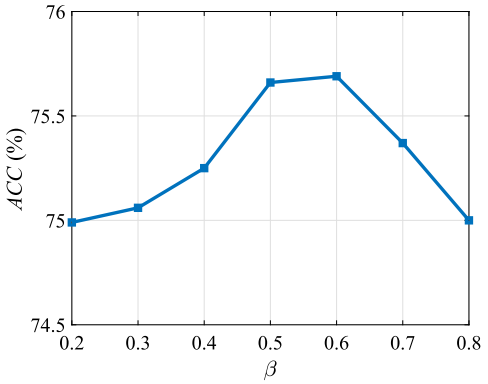



**Fig. 8.** Effectiveness of different values of the parameter $\beta$ in the 5-way 1-shot on the CIFAR-FS dataset.

**Table 5**
Result of different values of the parameter $\beta$ in the 5-way 1-shot on the CIFAR-FS dataset (%). The best results of each set are highlighted in bold.

| $\beta$ | | 0.2 | 0.3 | 0.4 | 0.5 | 0.6 | 0.7 | 0.8 |
|---|---|---|---|---|---|---|---|---|
| *TIE* | ↓ | 0.3752 | 0.3724 | 0.3697 | **0.3592** | 0.3595 | 0.3659 | 0.3689 |
| $F_H$ | ↑ | 90.61 | 90.91 | 90.98 | 91.22 | **91.23** | 91.06 | 90.95 |

(2) Similarly, the *TIE* and $F_H$ indicate the best results of 0.3769% and 90.33%, respectively, when $\lambda = 0.4$. The fine-grained recognition is optimal when coarse-grained participation accounts for 40% of the coarse- and fine-grained classes. Moreover, the experimental results of $F_H$ also show this trend.

Fig. 8 shows how the different values of parameter $\beta$ influence the effectiveness of HFFDK, while Table 5 lists the *TIE* and $F_H$ values. We can observe the followings:

(1) A series of 5-way 1-shot experiments was carried out on the CIFAR-FS dataset with a fixed value of parameter $\lambda$. The *ACC* increases from 74.99% to 75.69% when $\beta$ increases from 0.2 to 0.6. The ratio of the sample similarity measure and overall loss between the support and query sets impact HFFDK classification. Moreover, it can also be observed that the sample similarity measure between the support and query accounts for an excessive proportion of the total loss. When $\beta$ increases from 0.6 to 0.8, the *ACC* decreases from 75.69% to 75.00%, which verifies that excessive $\beta$ participation negatively impacts the final classification result of HFFDK.

(2) Especially, the *TIE* is best when $\beta = 0.5$, reaching 0.3592%. However, parameter $F_H$ is best when $\beta = 0.6$ (91.23%). The HFFDK classification performance is little different in terms of the evaluation metrics *ACC*, *TIE*, and $F_H$ when $\beta = 0.5$ and $\beta = 0.6$. In particular, the difference in the $F_H$ evaluation metric is only 0.01% at $\beta = 0.5$ and 0.6. According to the experimental results, we take *TIE* as the primary evaluation metric for datasets with a hierarchical tree structure.

### *5.2. Effectiveness of intra-layer channels global and local feature fusion*

In this section, we explore the effectiveness of intra-layer channels global and local feature fusion (IGLF) on the HFFDK. We conduct 5-way 1-shot and 5-way 5-shot experiments on four FSL benchmark datasets. We utilize the latest FSL model RENet as the baseline (BS) model for our comparison. We set the parameter milestone to [40,50], meanings that the learning rate will halve when Episode = 40 or Episode = 50.

Figs. 9 and 10 show the effectiveness of IGLF on the 5-way 5-shot experimental training loss values on the CIFAR-FS, FC100, and tieredImageNet datasets. We can make the following observations:

(1) The experiments on the FSL benchmark datasets FC100, CIFAR-FS, and tieredImageNet show that adding the IGLF to BS reduces the training loss of BS+IGLF compared with that of BS alone. This demonstrates the IGLF module's effectiveness, which can retain the inherent detailed information between network layers and simultaneously enrich semantic information.

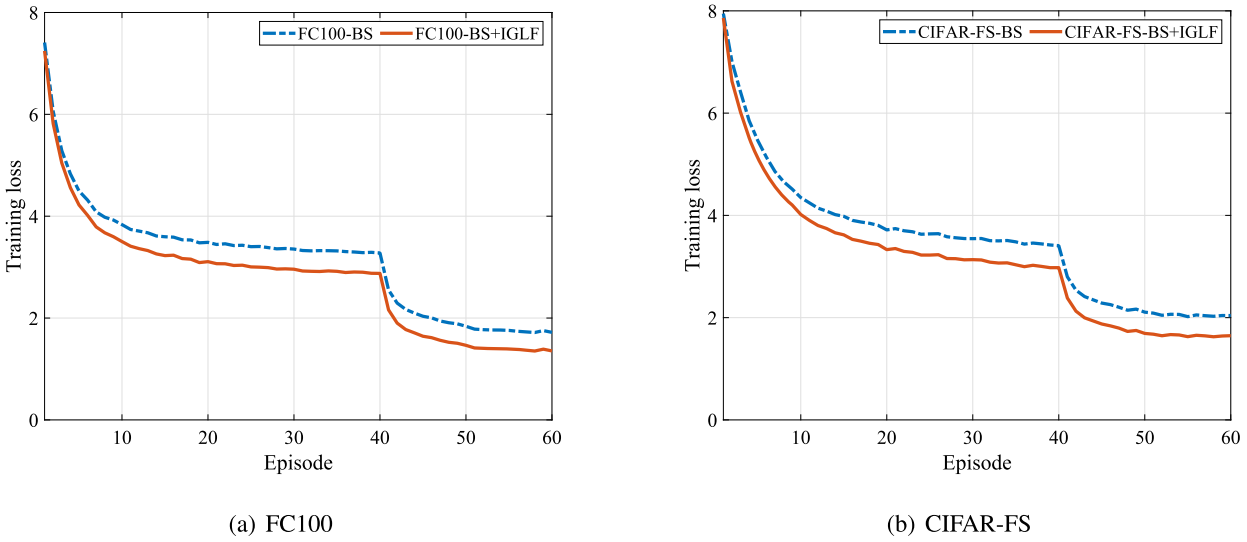



**Fig. 9.** Effectiveness of IGLF on the (a) FC100 and (b) CIFAR-FS datasets.

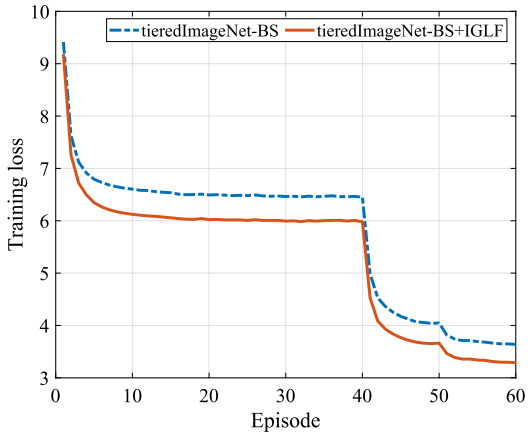



**Fig. 10.** Effectiveness of the IGLF on the tieredImageNet dataset.

(2) Furthermore, it is worth noting that the training losses of BS and BS+IGLF are reduced in the experiment with FSL benchmark datasets when the learning rate is halved. The results indicate that the IGLF can reliably improve the classification performance of hierarchical tree datasets.

Table 6 shows the accuracy of BS and BS+IGLF on the different benchmark datasets. We can obtain the following observations:

(1) Compared with BS, the classification accuracy of BS+IGLF in 5-way 1-shot and 5-way 5-shot experiments is improved by nearly 1∼3%. Especially compared with BS, the *ACC* of BS+IGLF was 3.249% higher in the 5-way 1-shot experiment with the CIFAR-FS dataset. Similarly, the *ACC* of BS+IGLF was improved by 1.389% in the 5-way 1-shot experiment with the FC100 dataset. This indicates that BS+IGLF provides good performance in 5-way 1-shot experiments with benchmark datasets.

(2) The experimental results with different benchmark datasets demonstrate that the performance of BS+IGLF is less impressive in the 5-way 5-shot experiment compared with the 5-way 1-shot one. The main reason is that the support set uses the average feature mapping of five samples for comparison with the query set when measuring the similarity between the support and query set samples.

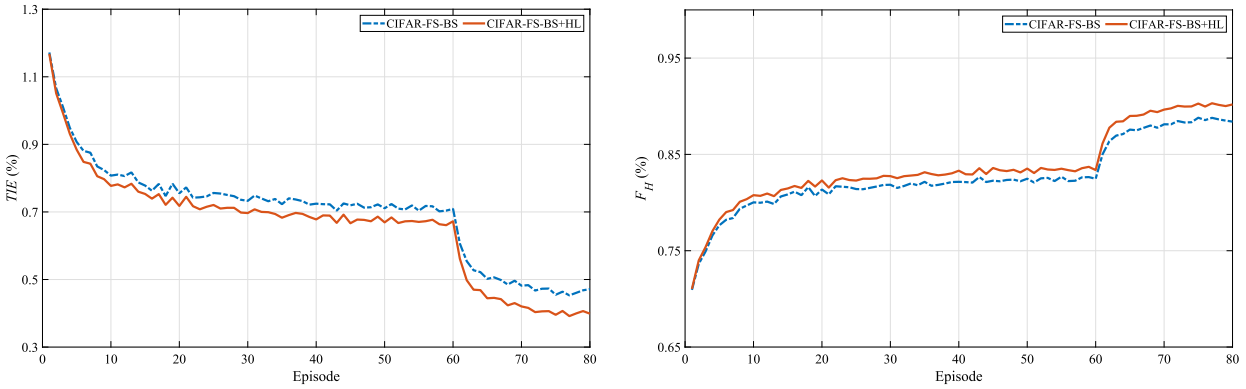
### *5.3. Effectiveness of hierarchical loss*

In this section, we explore the effectiveness of hierarchical loss (HL) on the HFFDK. We conduct 5-way 1-shot experiments on three FSL benchmark datasets. Similarly, we use the latest FSL model RENet as the baseline (BS) model for comparison. We set the parameter milestone to 60 and 70, which means that the learning rate will halve when Episode = 60 and Episode = 70.

Fig. 11 shows the effectiveness of HL in the 5-way 1-shot experiment training *TIE* and $F_H$ on the CIFAR-FS dataset, and we can obtain the following observations:

**Table 6**
Accuracy of BS and BS+IGLF on different datasets (%).

| Dataset | 5-way 1-shot | | 5-way 5-shot | |
|---|---|---|---|---|
| | BS | BS+IGLF | BS | BS+IGLF |
| FC100 | 43.625 ± 0.41 | 45.014 ± 0.41 | 59.891 ± 0.40 | 60.897 ± 0.39 |
| tieredImageNet | 70.786 ± 0.50 | 71.342 ± 0.51 | 84.933 ± 0.35 | 85.703 ± 0.35 |
| CIFAR-FS | 71.698 ± 0.47 | 74.947 ± 0.47 | 86.315 ± 0.32 | 86.537 ± 0.33 |




(a) $TIE$ (b) $F_H$

**Fig. 11.** Effectiveness of HL on the CIFAR-FS dataset.

**Table 7**
Accuracy comparison of BS and BS+HL on different datasets (%).

| Dataset | 5-way 1-shot | |
|---|---|---|
| | BS | BS+HL |
| FC100 | 43.625 ± 0.41 | 45.545 ± 0.41 |
| tieredImageNet | 70.786 ± 0.50 | 71.147 ± 0.51 |
| CIFAR-FS | 71.698 ± 0.47 | 73.322 ± 0.47 |

(1) The experimental results verify that BS+HL performs better in terms of *TIE* and $F_H$ during training compared with BS. In particular, it is worth noting that the training *TIE* value decreases, while the training $F_H$ value increases the BS on the CIFAR-FS dataset when the learning rate is halved.

(2) *TIE* and $F_H$ decrease and increase accordingly when the HL module is added to the BS. This result indicates that the HL module plays an auxiliary role in fine-grained recognition. The reduction in learning rate does not affect the robustness of the model.

Table 7 compares the accuracy of BS and BS+HL on different benchmark datasets. We can make the following observations:

(1) BS+HL carries out hierarchical learning on datasets with a hierarchical tree and achieves better classification results than BS, which does not use a hierarchical tree. BS+HL uses coarse-grained information to assist fine-grained recognition according to the inherent semantic information of the dataset. Compared with BS, the classification accuracy of BS+HL in the 5-way 1-shot experiment was improved by nearly 1∼2%. Adding the HL module to BS can improve the model's classification performance.

(2) Especially compared with BS, the *ACC* of BS+HL is 1.624% higher in the 5-way 1-shot experiment with the CIFAR-FS dataset. Similarly, the *ACC* of BS+HL is improved by 1.920% in the 5-way 1-shot experiment with the FC100 dataset. These experimental results demonstrate that the semantic knowledge of the hierarchical tree guides the HL module's construction, which plays an auxiliary role in class recognition.

### *5.4. Effectiveness of the combined action of IGLF and HL*

In this section, we explore the effectiveness of the combined action of IGLF and HL on the HFFDK. We conduct 5-way 1-shot and 5-way 5-shot experiments using three FSL benchmark datasets. To highlight the impact of different vital modules on the performance of the proposed method, we perform an ablative analysis by adding IGLF and HL modules to the BS model. Table 8 compares the performance of BS, BS+IGLF, BS+HL, and HFFDK on different benchmark datasets. The best results of each set are highlighted in bold. We can obtain the following observations:

(1) The HFFDK conducts in the 5-way 5-shot experiments on different benchmark datasets. Compared with BS, BS+IGLF, and BS+HL, the HFFDK experimental obtained the best results. In particular, the *TIE* decreased from 0.4055% to 0.3869% in the 5-way 5-shot experiment with the FC100 dataset. Similarly, HFFDK has an excellent performance in the 5-way 1-shot excitation experiment.

**Table 8**
Performance comparison of different strategies on different benchmark datasets (%).

| Dataset | Strategy | 5-way 1-shot | | 5-way 5-shot | |
|---|---|---|---|---|---|
| | | *TIE* ↓ | $F_H$ ↑ | *TIE* ↓ | $F_H$ ↑ |
| FC100 | BS | 0.6538 | 83.75 | 0.4055 | 89.82 |
| | BS+IGLF | 0.6329 | 84.26 | 0.3935 | 90.12 |
| | BS+HL | **0.6207** | **84.54** | 0.4097 | 89.71 |
| | BS+IGLF+HL (HFFDK) | 0.6393 | 84.12 | **0.3869** | **90.33** |
| CIFAR-FS | BS | 0.4477 | 89.12 | 0.1813 | 95.56 |
| | BS+IGLF | 0.3754 | 90.81 | 0.1728 | 95.75 |
| | BS+HL | 0.4174 | 89.89 | 0.1812 | 95.56 |
| | BS+IGLF+HL (HFFDK) | **0.3592** | **91.22** | **0.1704** | **95.82** |
| tieredImageNet | BS | 0.3385 | 91.38 | 0.1534 | 96.08 |
| | BS+IGLF | 0.3165 | 91.90 | 0.1418 | 96.37 |
| | BS+HL | 0.3254 | 91.69 | 0.1503 | 96.16 |
| | BS+IGLF+HL (HFFDK) | **0.3132** | **91.97** | **0.1415** | **96.39** |

However, its performance is not as good as that of BS+HL on the FC100 dataset. The reason may be that there are few coarse-grained classes, and the knowledge provided by the hierarchical structure is not sufficient for the current classification task.

(2) Furthermore, it is worth noting that compared with BS, BS+IGLF, and BS+HL, HFFDK achieves greater improvements in terms of *TIE* and $F_H$ in the 5-way 1-shot experiments on the CIFAR-FS and tieredImageNet datasets. For example, the *TIE* decreased from 0.4477% to 0.3592%, while $F_H$ increased from 89.12% to 91.22% on CIFAR-FS. There are more coarse-grained classes in CIFAR-FS than in FC100, which can provide more hierarchical tree knowledge.

(3) We also observe that *TIE* decreases and $F_H$ increases when using the HL module with the three datasets. The HL module provides a hierarchical tree structure to guide coarse-grained recognition and uses coarse-grained information to assist fine-grained recognition. The experimental results with the IGLF module indicate that the feature quality of dataset sample extraction plays a crucial role in measuring the similarity between samples. The IGLF module preserves the details of the image and enriches its semantic information. Driven by the IGLF and HL modules, the best classification results is obtained.

### 5.5. Comparison with other state-of-the-art methods

In this section, we compare our HFFDK method with several state-of-the-art FSL methods on three benchmark datasets. We utilize the same experimental setup for the 5-way 1-shot and 5-way 5-shot experiments and obtain a set of baseline experimental results. Tables 9 and 10 compare the performance of HFFDK with several state-of-the-art FSL methods on different benchmark datasets. The best results of each set are highlighted in bold.

Note that we only leverage the training set for training. The results from reported in Kang et al. [10] are denoted with "$*$", the best results from the original author represented by "$\star$", while "$\dagger$" denotes larger backbones than ResNet12. We can obtain the following observations:

(1) Compared with the WRN-28-10$^\dagger$ backbone network method, the classification performance of HFFDK is slightly improved in the 5-way 1-shot and 5-way 5-shot experiments on the tieredImageNet dataset. Meanwhile, HFFDK shows better performance in fine-grained recognition than the ResNet12 backbone network method. In particular, HFFDK is 2% better than the suboptimal method (Boosting) in the 5-way 1-shot experiment on the CIFAR-FS dataset.

(2) HFFDK performs better in the 5-way 1-shot and 5-way 5-shot experiments on four benchmark datasets compared with the ConvNet128, ResNet18, and ResNet34$^\dagger$ backbone network methods. The experimental performance improvement of HFFDK in the 5-way 1-shot experiment is better than that in the 5-way 5-shot experiment. This is because the features obtained from the support set samples in the 5-shot are not the best. However, HFFDK needs to rely on these features to calculate the similarity between the support and query set samples.

(3) Compared with KSTNET on the miniImageNet dataset, HFFDK performs better without using semantic information. HFFDK performs better than KSTNET in 5-shot experiments when KSTNET uses semantic information and HFFDK does not. KSTNET performs better than HFFDK experimentally with miniImageNet 1-shot using semantic or attribute information. The results of the 5-way 1-shot and 5-way 5-shot experiments using HFFDK on four FSL benchmark datasets are slightly better than when using RENet. This demonstrates the effectiveness of HFFDK, which effectively uses the knowledge of the hierarchical tree structure and the features of inter-channel fusion.

### 5.6. Feature visualization

We randomly extract sample features with 50 query samples and 250 query samples per class and visualize them under 5-way 1-shot experiments in the CIFAR-FS dataset and 5-way 5-shot experiments in the tieredImageNet dataset, respectively. As shown in Figs. 12 and 13, the inter-class features are far from each other and the intra-class features are close to each other, proving the effectiveness of HFFDK.

**Table 9**
Accuracy comparison of the state-of-the-art with 95% confidence intervals on the tieredImageNet and miniImageNet datasets (%).

(a) tieredImageNet

| Method | Semantic | Backbone | 5-way 1-shot | 5-way 5-shot |
|---|---|---|---|---|
| wDAE-GNN* [7] | N | WRN-28-10† | 68.18 ± 0.16 | 83.09 ± 0.12 |
| ProtoNet* [34] | N | ResNet12 | 68.23 ± 0.23 | 84.03 ± 0.16 |
| MatchNet* [39] | N | ResNet12 | 68.50 ± 0.92 | 80.60 ± 0.71 |
| CTM* [16] | N | ResNet18 | 68.41 ± 0.39 | 84.28 ± 1.73 |
| FEAT* [44] | N | ResNet12 | 70.80 ± 0.23 | 84.79 ± 0.16 |
| RENet [10] | N | ResNet12 | 70.79 ± 0.50 | 84.93 ± 0.35 |
| MNE★ [17] | Y | ResNet12 | 60.04 ± 0.28 | 73.63 ± 0.21 |
| HMRN★ [35] | Y | ResNet12 | 57.98 ± 0.26 | 74.70 ± 0.24 |
| AM3-PNet★ [42] | Y | ResNet12 | 67.23 ± 0.34 | 78.95 ± 0.22 |
| **HFFDK** | Y | ResNet12 | **71.59 ± 0.53** | **85.68 ± 0.36** |

(b) miniImageNet

| Method | Semantic | Backbone | 5-way 1-shot | 5-way 5-shot |
|---|---|---|---|---|
| ProtoNet* [34] | N | ResNet12 | 62.39 ± 0.21 | 78.63 ± 0.46 |
| MatchNet* [39] | N | ResNet12 | 63.08 ± 0.80 | 75.99 ± 0.60 |
| CTM* [16] | N | ResNet18 | 64.12 ± 0.82 | 80.51 ± 0.13 |
| FEAT* [44] | N | ResNet12 | 66.78 ± 0.20 | 82.05 ± 0.14 |
| RENet [10] | N | ResNet12 | 67.20 ± 0.50 | 82.31 ± 0.35 |
| KSTNET★ [18] | N | ResNet12 | 64.81 ± 0.81 | 81.54 ± 0.50 |
| FSLKT★ [27] | Y | ConvNet128 | 64.42 ± 0.72 | 74.16 ± 0.24 |
| AM3-PNet★ [42] | Y | ResNet12 | 65.21 ± 0.30 | 75.20 ± 0.27 |
| AM3-TRAML★ [13] | Y | ResNet12 | 67.10 ± 0.52 | 79.54 ± 0.60 |
| KSTNET★ [18] | Y | ResNet12 | **71.51 ± 0.73** | 82.61 ± 0.48 |
| **HFFDK** (only IGLF) | N | ResNet12 | 67.31 ± 0.53 | **83.18 ± 0.36** |

**Table 10**
Accuracy comparison of the state-of-the-art with 95% confidence intervals on the CIFAR-FS and FC100 datasets (%).

(a) CIFAR-FS

| Method | Backbone | 5-way 1-shot | 5-way 5-shot |
|---|---|---|---|
| Cosine classifier* [2] | ResNet34† | 61.49 ± 0.91 | 82.37 ± 0.67 |
| S2M2* [25] | ResNet34† | 72.92 ± 0.83 | 86.55 ± 0.51 |
| RFS-simple* [38] | ResNet12 | 71.50 ± 0.80 | 86.00 ± 0.50 |
| ProtoNet* [34] | ResNet12 | 72.20 ± 0.70 | 83.50 ± 0.50 |
| MetaOptNet* [12] | ResNet12 | 72.60 ± 0.70 | 84.30 ± 0.50 |
| Boosting* [6] | WRN-28-10† | 73.60 ± 0.30 | 86.00 ± 0.20 |
| RENet [10] | ResNet12 | 71.70 ± 0.47 | 86.32 ± 0.32 |
| **HFFDK** | ResNet12 | **75.66 ± 0.48** | **86.77 ± 0.33** |

(b) FC100

| Method | Backbone | 5-way 1-shot | 5-way 5-shot |
|---|---|---|---|
| TADAM* [26] | ResNet12 | 40.10 ± 0.40 | 56.10 ± 0.40 |
| MetaOptNet* [12] | ResNet12 | 41.10 ± 0.60 | 55.50 ± 0.60 |
| Baseline2020★ [3] | ResNet12 | 36.82 ± 0.51 | 49.72 ± 0.55 |
| RENet [10] | ResNet12 | 43.63 ± 0.41 | 59.89 ± 0.40 |
| **HFFDK** | ResNet12 | **44.78 ± 0.40** | **61.54 ± 0.41** |

## 6. Conclusions and future work

In this paper, we proposed a method of hierarchical few-shot learning with feature fusion driven by data and knowledge (HFFDK). It effectively uses the semantic knowledge contained in a hierarchical tree and the fuses the feature of global and local intra-layer channels. First, the fusion of global and local intra-layer channels is leveraged to preserve image detail information and enrich the semantic information of the features. Then, the coarse-grained information is utilized to assist fine-grained identification according to the membership relationship between coarse- and fine-grained classes. HFFDK also uses the hierarchical tree knowledge of the datasets to improve fine-grained recognition. Finally, we constructed a hierarchical few-shot learning model that is both data- and knowledge-driven and is based on intra-layer channels fusion and the hierarchical tree structure. Moreover, HFFDK uses a semantic hierarchical tree structure as a knowledge guide for classification. Other class structures can be used as auxiliary knowledge to guide

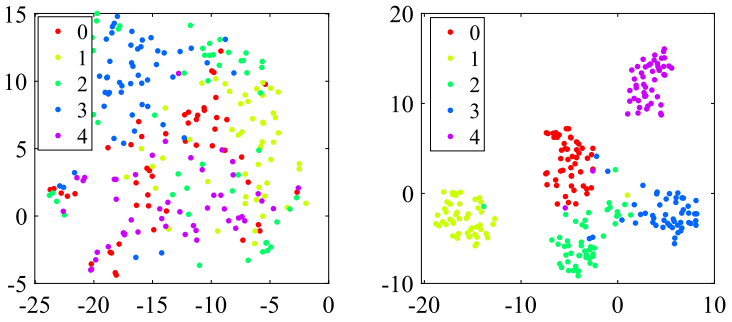



**Fig. 12.** Feature visualization from BS, HFFDK (our). Different colors represent different classes. (a) show the baseline visualization results before applying our method, (b) show the baseline visualization results after applying HFFDK. Experiments are conducted on the Cifar-FS dataset by using ResNet12.

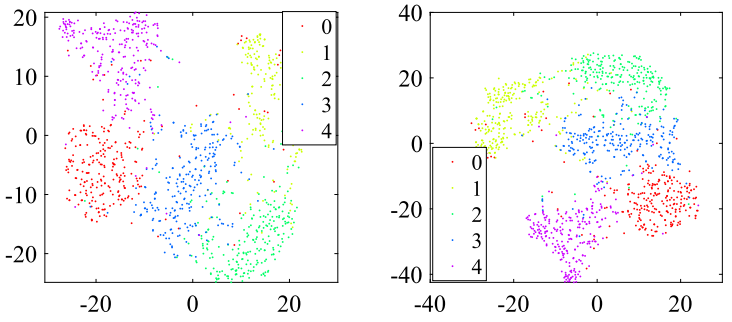



**Fig. 13.** Feature visualization from BS, HFFDK (our). Different colors represent different classes. (a) show the baseline visualization results before applying our method, (b) show the baseline visualization results after applying HFFDK. Experiments are conducted on the tieredImageNet dataset by using ResNet12.

classification, such as semantic knowledge graphs and graph structures. In future, we will explore few-shot learning models for use in classification tasks involving other types of knowledge.

## CRediT authorship contribution statement

**Zhiping Wu:** Conceptualization, Methodology, Software, Writing – original draft. **Hong Zhao:** Conceptualization, Methodology, Supervision, Validation, Writing – review & editing.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

No data was used for the research described in the article.

## Acknowledgements

This work was supported by the National Natural Science Foundation of China under Grant No. 62141602 and the Natural Science Foundation of Fujian Province under Grant No. 2021J011003.

## References

[1] L. Bertinetto, J.F. Henriques, P. Torr, A. Vedaldi, Meta-learning with differentiable closed-form solvers, in: International Conference on Learning Representations, 2018.
[2] W. Chen, Y. Liu, Z. Kira, Y.F. Wang, J. Huang, A closer look at few-shot classification, in: International Conference on Learning Representations, 2019.
[3] G.S. Dhillon, P. Chaudhari, A. Ravichandran, S. Soatto, A baseline for few-shot image classification, in: International Conference on Learning Representations, 2020.
[4] Y. Ding, X. Tian, L. Yin, X. Chen, S. Liu, B. Yang, W. Zheng, Multi-scale relation network for few-shot learning based on meta-learning, in: International Conference on Computer Vision Systems, 2019, pp. 343–352.
[5] J. Duan, G. Wang, X. Hu, H. Bao, Hierarchical quotient space-based concept cognition for knowledge graphs, Inf. Sci. 597 (2022) 300–317.

[6] S. Gidaris, A. Bursuc, N. Komodakis, P. Pérez, M. Cord, Boosting few-shot visual learning with self-supervision, in: IEEE/CVF International Conference on Computer Vision, 2019, pp. 8059–8068.
[7] S. Gidaris, N. Komodakis, Generating classification weights with GNN denoising autoencoders for few-shot learning, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 21–30.
[8] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.
[9] W. Jiang, K. Huang, J. Geng, X. Deng, Multi-scale metric learning for few-shot learning, IEEE Trans. Circuits Syst. Video Technol. 31 (3) (2020) 1091–1102.
[10] D. Kang, H. Kwon, J. Min, M. Cho, Relational embedding for few-shot classification, in: IEEE/CVF International Conference on Computer Vision, 2021.
[11] G. Koch, R. Zemel, R.a. Salakhutdinov, Siamese neural networks for one-shot image recognition, in: ICML Deep Learning Workshop, vol. 2, 2015.
[12] K. Lee, S. Maji, A. Ravichandran, S. Soatto, Meta-learning with differentiable convex optimization, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 10657–10665.
[13] A. Li, W. Huang, X. Lan, J. Feng, Z. Li, L. Wang, Boosting few-shot learning with adaptive margin loss, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12576–12584.
[14] A. Li, T. Luo, Z. Lu, T. Xiang, L. Wang, Large-scale few-shot learning: knowledge transfer with class hierarchy, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 7212–7220.
[15] G. Li, V. Jampani, L. Sevillalara, D. Sun, J. Kim, J. Kim, Adaptive prototype learning and allocation for few-shot segmentation, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 8334–8343.
[16] H. Li, D. Eigen, S. Dodge, M. Zeiler, X. Wang, Finding task-relevant features for few-shot learning by category traversal, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 1–10.
[17] S. Li, D. Chen, B. Liu, N. Yu, R. Zhao, Memory-based neighbourhood embedding for visual recognition, in: IEEE/CVF International Conference on Computer Vision, 2019, pp. 6102–6111.
[18] Z. Li, H. Tang, Z. Peng, G.-J. Qi, J. Tang, Knowledge-guided semantic transfer network for few-shot image recognition, IEEE Trans. Neural Netw. Learn. Syst. (2023).
[19] Z. Li, J. Tang, T. Mei, Deep collaborative embedding for social image understanding, IEEE Trans. Pattern Anal. Mach. Intell. 41 (9) (2018) 2070–2083.
[20] Z. Li, H. Zhao, Y. Lin, Multi-task convolutional neural network with coarse-to-fine knowledge transfer for long-tailed classification, Inf. Sci. 608 (2022) 900–916.
[21] L. Liu, T. Zhou, G. Long, J. Jiang, L. Yao, C. Zhang, Prototype propagation networks (PPN) for weakly-supervised few-shot learning on category graph, in: International Joint Conference on Artificial Intelligence, 2019, pp. 3015–3022.
[22] X. Liu, Y. Zhou, H. Zhao, Robust hierarchical feature selection driven by data and knowledge, Inf. Sci. 551 (2021) 341–357.
[23] X. Liu, L. Jiao, L. Li, X. Tang, Y. Guo, Deep multi-level fusion network for multi-source image pixel-wise classification, Knowl.-Based Syst. 221 (2021) 106921.
[24] X. Liu, L. Li, F. Liu, B. Hou, S. Yang, L. Jiao Gafnet, Group attention fusion network for PAN and MS image high-resolution classification, IEEE Trans. Cybern. 52 (10) (2022) 10556–10569.
[25] P. Mangla, N. Kumari, A. Sinha, M. Singh, B. Krishnamurthy, V. Balasubramanian, Charting the right manifold: manifold mixup for few-shot learning, in: IEEE/CVF Winter Conference on Applications of Computer Vision, 2020, pp. 2218–2227.
[26] B. Oreshkin, P. Rodriguez, A. Lacoste TADAM, Task dependent adaptive metric for improved few-shot learning, in: International Conference on Neural Information Processing Systems, 2020, pp. 721–731.
[27] Z. Peng, Z. Li, J. Zhang, Y. Li, G. Qi, J. Tang, Few-shot image recognition with knowledge transfer, in: IEEE/CVF International Conference on Computer Vision, 2019, pp. 441–449.
[28] Y. Piao, W. Ji, J. Li, M. Zhang, H. Lu, Depth-induced multi-scale recurrent attention network for saliency detection, in: IEEE/CVF International Conference on Computer Vision, 2019, pp. 7253–7262.
[29] Z. Qin, H. Wang, C. Mawuli, W. Han, R. Zhang, Q. Yang, J. Shao, Multi-instance attention network for few-shot learning, Inf. Sci. 611 (2022) 464–475.
[30] M. Ren, E. Triantafillou, S. Ravi, J. Snell, K. Swersky, J.B. Tenenbaum, H. Larochelle, R.S. Zemel, Meta-learning for semi-supervised few-shot classification, in: International Conference on Learning Representations, 2018.
[31] A.A. Rusu, D. Rao, J. Sygnowski, O. Vinyals, R. Pascanu, S. Osindero, R. Hadsell, Meta-learning with latent embedding optimization, in: International Conference on Learning Representations, 2019.
[32] E. Schwartz, L. Karlinsky, R. Feris, R. Giryes, A. Bronstein, Baby steps towards few-shot learning with multiple semantics, Pattern Recognit. Lett. 160 (2022) 142–147.
[33] T. Shermin, S.W. Teng, F. Sohel, M. Murshed, G. Lu, Integrated generalized zero-shot learning for fine-grained classification, Pattern Recognit. 122 (2022) 108246.
[34] J. Snell, K. Swersky, R. Zemel, Prototypical networks for few-shot learning, in: International Conference on Neural Information Processing Systems, 2017, pp. 4080–4090.
[35] Y. Su, H. Zhao, Y. Lin, Few-shot learning based on hierarchical classification via multi-granularity relation networks, Int. J. Approx. Reason. 142 (2022) 417–429.
[36] Q. Sun, Y. Liu, T. Chua, B. Schiele, Meta-transfer learning for few-shot learning, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 403–412.
[37] F. Sung, Y. Yang, L. Zhang, T. Xiang, P.H. Torr, T.M. Hospedales, Learning to compare: relation network for few-shot learning, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, pp. 1199–1208.
[38] Y. Tian, Y. Wang, D. Krishnan, J.B. Tenenbaum, P. Isola, Rethinking few-shot image classification: a good embedding is all you need?, in: European Conference on Computer Vision, 2020, pp. 266–282.
[39] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, D. Wierstra, Matching networks for one shot learning, in: International Conference on Neural Information Processing Systems, 2016, pp. 3637–3645.
[40] Y. Wang, Q. Hu, H. Chen, Y. Qian, Uncertainty instructed multi-granularity decision for large-scale hierarchical classification, Inf. Sci. 586 (2022) 644–661.
[41] Y. Wang, C. Xu, C. Liu, L. Zhang, Y. Fu, Instance credibility inference for few-shot learning, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12833–12842.
[42] C. Xing, N. Rostamzadeh, B. Oreshkin, P.O.O. Pinheiro, Adaptive cross-modal few-shot learning, in: International Conference on Neural Information Processing Systems, vol. 32, 2019.
[43] S. Yang, M. Wang, Y. Lu, W. Qi, L. Jiao, Fusion of multiparametric SAR images based on SW-nonsubsampled contourlet and PCNN, Signal Process. 89 (12) (2009) 2596–2608.
[44] H. Ye, H. Hu, D. Zhan, F. Sha, Few-shot learning via embedding adaptation with set-to-set functions, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 8808–8817.
[45] B. Zhang, H. Jiang, X. Li, S. Feng, Y. Ye, R.Ye. MetaDT, Meta decision tree for interpretable few-shot learning, arXiv preprint, arXiv:2203.01482, 2022.
[46] B. Zhang, C. Leung Ka, X. Li, Y. Ye, Learn to abstract via concept graph for weakly-supervised few-shot learning, Pattern Recognit. 117 (2021) 107946.
[47] B. Zhang, X. Li, Y. Ye, Z. Huang, L. Zhang, Prototype completion with primitive knowledge for few-shot learning, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 3754–3762.
[48] C. Zhang, Y. Cai, G. Lin, C. Shen DeepEMD, Few-shot image classification with differentiable Earth mover's distance and structured classifiers, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2020, pp. 12200–12210.

[49] H. Zhang, J. Zhang, P. Koniusz, Few-shot learning via saliency-guided hallucination of samples, in: IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, pp. 2765–2774.
[50] Q. Zhang, Y. Yang, SA-Net: shuffle attention for deep convolutional neural networks, in: IEEE International Conference on Acoustics, Speech and Signal Processing, 2021, pp. 2235–2239.