International Journal of Machine Learning and Cybernetics
https://doi.org/10.1007/s13042-022-01640-5

ORIGINAL ARTICLE

Knowledge transfer based hierarchical few-shot learning via tree-structured knowledge graph

Zhong Zhang[1,2] · Zhiping Wu[1,2] · Hong Zhao[1,2] · Minjie Hu[1,2]

Received: 16 March 2022 / Accepted: 21 August 2022

Abstract
Few-shot learning poses a great challenge for obtaining a classifier that recognizes new classes from a few labeled examples. Existing solutions perform well by leveraging meta-learning models driven by data information. However, these models only utilize the flat data information and ignore the existing hierarchical knowledge structure among classes. In this paper, we propose a knowledge transfer based hierarchical few-shot learning model, which takes advantage of a tree-structured knowledge graph to facilitate the classification results. First, we consider a tree-structured class hierarchy according to the semantic information among classes as a knowledge graph to alleviate the low-data problem. Second, we divide the tree structure into class structure and data, and build a multi-layer classifier to obtain classification results in the two parts. Finally, we consider the tradeoff between structure loss and data loss for hierarchical few-shot learning, which takes class structure information to assist learning. Experimental results on benchmark datasets show that our model outperforms several state-of-the-art models.

Keywords Few-shot learning · Hierarchical classification · Knowledge transfer · Tree-structured knowledge graph

1 Introduction

Few-shot learning (FSL) aims to tackle the problem of quickly adapting a deep learner to understanding new classes with few samples and has attracted much attention in machine learning [45, 55]. A key challenge in FSL is to make the best use of the limited data to find the right class for each classification task [28]. To overcome this challenge, the flat meta-learning based FSL methods have become topical [5]. They leverage previous knowledge and experience to guide the learning of new classes, which makes the network can classify new classes with insufficient samples [27]. The meta-learning approaches are categorized into three branches: (1) model-based, (2) optimization-based, and (3) metric-based.

Both model-based and optimization-based methods hope to get a better meta learner to achieve few-shot classification [3, 30]. The model-based approach aims to explore the commonality among tasks. Based on this idea, Finn et al. [10] presented a model-agnostic meta-learning method, which can be applied to many network models and task types. Moreover, Jamal et al. [17] changed the objective function and developed a task-agnostic meta-learning. The optimization-based method converges by training learners for millions or tens of millions of iterations. Therefore, Ravi et al. [34] suggested a meta-learner to learn the optimization parameters so that the model can use only a small number of samples and converge quickly simultaneously. Similarly, Andrychowicz et al. [2] used LSTM instead of a traditional optimizer to make gradient descent optimize the optimizer itself. Meanwhile, Jiang et al. [18] achieved semblable results on various timescales and provided a multi-time scale optimization.

Metric learning is also one of the main few-shot learning methods, which means expressing the correlation between two samples [21, 31]. It can be considered that in projection space, the closer the distance is, the more similar the sample is. There are various metric methods, among which the Cosine distance and Euclidean distance are widely used [4]. For instance, Vinyas et al. [43] adopted the idea of nearest neighbor classification and leveraged Cosine distance to measure the similarity among sample features. Similarly, Snell et al. [39] creatively proposed a prototypical network

✉ Hong Zhao
hongzhaocn@163.com

[1] School of Computer Science, Minnan Normal University, Zhangzhou 363000, Fujian, China

[2] Key Laboratory of Data Science and Intelligence Application, Fujian Province University, Zhangzhou 363000, Fujian, China

Published online: 09 September 2022

to obtain the prototypes of each class and calculated the Euclidean distance between the query samples and the prototypes. It differs from the way of calculating the distance above, Sung et al. [41] established a relation network for calculating relationship score for the first time. Its main idea is to measure the similarity between the sample features of query and support sets and classify new classes by calculating their relationship score. Furthermore, a memory-augmented and a graph embedding relation networks were presented by Hui et al. [16] and He et al. [13], respectively.

Features extracted from the data play an important role in label classification. The schemes mentioned above have achieved effective results in flat classification via data features. However, these models assume that classes are independent of each other, ignoring the hierarchical information of class structure within a dataset. Fortunately, a knowledge graph is ubiquitous in most datasets and has strong expression ability and modeling flexibility as a semantic network [12, 54]. One of the most popular knowledge graphs is a tree-structured class hierarchy, which is widely used as a simple and effective auxiliary knowledge [24]. We use the tree structure as external auxiliary information to assist model classification rather than data features.

In this paper, we propose a hierarchical few-shot learning model based on knowledge transfer (HFKT) using a tree-structured knowledge graph to improve the lack of samples. First, we consider the hierarchical tree structure as auxiliary information among classes, alleviating the problem of insufficient information with few-shot learning. HFKT divides the hierarchical tree structure into two parts, the data, and the class structure, representing the data and tree structure information. Second, we design the fully-connected layers corresponding to these two parts to classify samples. At the same time, through the propagation of the network, the guided extracted features can achieve satisfactory performance at different layers of the tree structure. Finally, we build a joint loss function to transfer the effective knowledge in the class structure to the data layer, which promotes the classification performance of different tree structure layers to improve the classification ability of the model.

The main contributions of our manuscript can be summarized as follows:

(1) We propose a knowledge transfer based model by handling the coarse- and fine-grained classes within a unified framework to tackle the few-shot problem. The traditional few-shot learning methods assume that classes are independent of each other, ignoring the hierarchical knowledge structure that we make full use of as auxiliary classification knowledge.
(2) A multi-layer hierarchical classifier is used for further classification when the traditional classifiers focus on fine-grained classes. We optimize the model in different granularity class classifications by combining structure and data loss. Unlike traditional few-shot learning methods, the proposed model takes advantage of the characteristics of coarse- and fine-grained classes instead of using only fine-grained classes.

The experiments are mainly carried out on four benchmark datasets with three evaluation metrics to compare the proposed HFKT with several existing FSL approaches. On the one hand, experimental results show that HFKT achieves a satisfactory performance with 70.63% and 83.90% in 5-way 1-shot and 5-shot episodes on the tieredImageNet dataset [36], which outperforms the most advanced FSL models. On the other hand, HFKT gets an F_1-measure improvement of 1.40% and an accuracy improvement of 2.37% over the next best under the 5-way 1-shot setting on the CIFAR-FS dataset [25].

The remainder of this paper is organized as follows. In Sect. 2, we introduce the details of our proposed approach. The experimental settings of our proposed model are given in Sect. 3. The experimental results and analysis are summarized in Sect. 4 and the paper concludes in Sect. 5.

2 HFKT model

This section introduces the main framework of the proposed hierarchical few-shot learning model based on knowledge transfer (HFKT), and then we elaborate on its components.

2.1 Basic framework

Knowledge graphs are graph-structured knowledge bases, which consists of numerous nodes [7]. These nodes are not independent of each other but are the expression of different granularities of knowledge. For the tree-structured knowledge graph, the nodes close to the root node express more abstract coarse-grained knowledge. On the contrary, the nodes near the leaf node represent specific fine-grained knowledge. The basic flowchart of the HFKT is shown in Fig. 1.

This model is composed of two parts:

(1) The first part is a tree-structured knowledge graph based on underlying semantic relationships between object classes. We use the semantic tree as external auxiliary knowledge to help classification. This tree structure is divided into two parts, representing the bottom layer of data and other layers of class structure.
(2) The second part is hierarchical few-shot learning based on knowledge transfer. We transfer useful knowledge from the class structure to the data layer by a joint loss function.

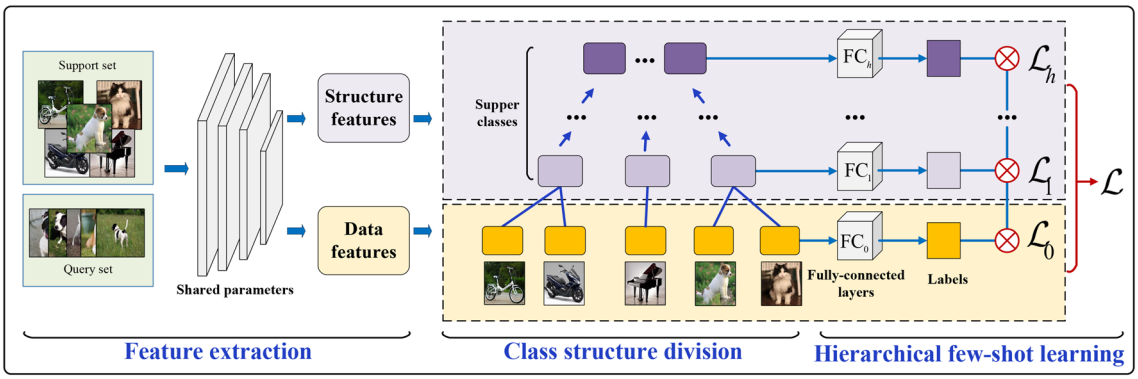

Fig. 1 Framework of the HFKT model in a 5-way 1-shot setting. FC_ℓ means the ℓ-th fully-connected layer corresponding to the hierarchical tree structure, where $\ell = 0, 1, ..., h$. $\mathcal{L}_0, ..., \mathcal{L}_h$ are the losses of classification results at different layers

2.2 Class structure division

Knowledge graph exists in most real-world and classification tasks [37]. A tree structure and a directed acyclic graph structure are two common representations of the knowledge graph [1]. In this work, we focus on the tree-structured knowledge graph, which contains rich hierarchical knowledge among classes.

Let $\mathbf{Y}$ be the class label set. The class labels at different layers obtained from hierarchical tree structure are:

$$\mathbf{Y} = \mathbf{Y}^{(0)} \cup \cdots \cup \mathbf{Y}^{(\ell)} \cup \cdots \cup \mathbf{Y}^{(h)}, \tag{1}$$

where $\mathbf{Y}^{(\ell)}$ represents the ℓ-th layer labels, and $\mathbf{Y}^{(0)}$ and $\mathbf{Y}^{(h)}$ represent the bottom and top layer classes, respectively.

The hierarchical structure can be visualized as a tree structure, and each superclass can cover multiple subclasses. The hierarchical tree structure of classes is defined as a pair $(\mathbf{Y}, \prec)$, where $\prec$ represents an *is-a* relationship, which is a *subclass-of* relationship with the following properties [20, 54]:

(1) Asymmetry: if $y_i^{(\ell)} \prec y_j^{(\ell+1)}$ then $y_j^{(\ell+1)} \not\prec y_i^{(\ell)}$ for every $y_i^{(\ell)} \in \mathbf{Y}^{(\ell)}$ and $y_j^{(\ell+1)} \in \mathbf{Y}^{(\ell+1)}$.
(2) Anti-reflexivity: $y_i^{(\ell)} \not\prec y_i^{(\ell)}$ for every $y_i^{(\ell)} \in \mathbf{Y}^{(\ell)}$.
(3) Transitivity: if $y_i^{(\ell-1)} \prec y_j^{(\ell)}$ and $y_j^{(\ell)} \prec y_t^{(\ell+1)}$, then $y_i^{(\ell-1)} \prec y_t^{(\ell+1)}$ for every $y_i^{(\ell-1)} \in \mathbf{Y}^{(\ell-1)}, y_j^{(\ell)} \in \mathbf{Y}^{(\ell)}$, and $y_t^{(\ell+1)} \in \mathbf{Y}^{(\ell+1)}$.

The classes are organized in taxonomies in hierarchical classification compared with flat classification, which is considered unrelated. Each node has a single parent node except the root node for tree structure, which imposes a *parent–child* relationship among the classes. In other words, if a node belongs to a class, it also belongs to all ancestors of this class.

Semantic and visual class hierarchies are two types of hierarchical class structures. In this work, we concentrate on the semantic class hierarchy, constructed by the semantic subordination and relation among the classes in WordNet [32]. Many researchers use the semantic hierarchical class structure to design their models [11, 48, 49]. We leverage Example 1 to give the semantic explanation of the hierarchical tree structure.

Example 1 We give an example of the tieredImageNet dataset to describe the hierarchical tree structure as shown in Fig. 2. The root class Object contains all the classes and has no parent node. From the vertical perspective, the nodes at different layers constitute the parent–child relationship of the tree structure. Class nodes without a child node are termed "leaf class nodes". In contrast, we call all their ancestors "non-leaf class node". The class closer to the leaf node is more specific. On the contrary, the class farther away from the leaf node is more abstract. We call the bottom layer, where $\ell = 0$, the data layer, while the other layers are called the class structure layer. For example, "Lion", at layer 0, is the sample's label, representing the information from the data. "Mammal" and "Animal", at layer 1 and layer h respectively, are the ancestors of "Lion", displaying knowledge via the hierarchical tree structure. This can be summarized that the hierarchical class structure consists of two parts, the data part at layer 0 and the class structure part from layer 1 to layer h.

We simulate the few-shot learning setting through the episode-based training and revisit the hierarchical tree structure for every few-shot learning task. In contrast to traditional hierarchical classification tasks, e.g., the classifier directly uses the

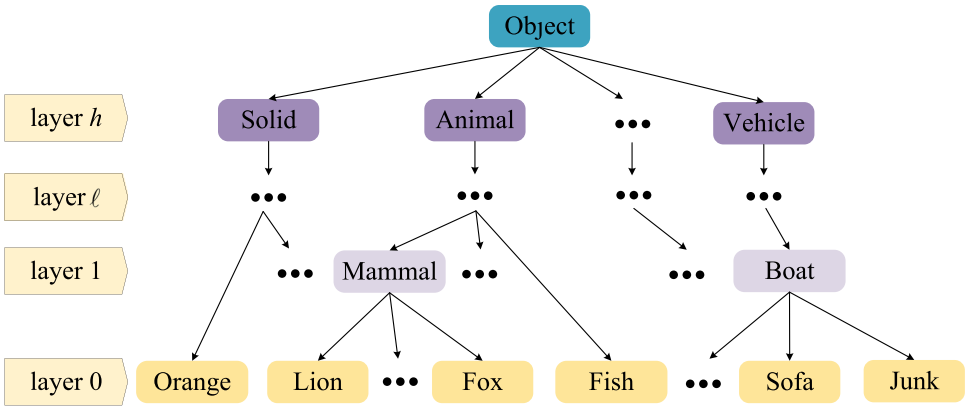

Fig. 2 Hierarchical tree knowledge graph of the tieredImageNet dataset

whole tree structure for auxiliary classification, we attempt to utilize the hierarchical tree structure differently. Corresponding to the training task, each episode corresponds to a subtree in the hierarchical tree structure.

We consider the episode of C-way K-shot for few-shot classification task. For every task, the C classes make up a subtree which belongs to the entire hierarchical tree structure. In each training task, we select C classes from training set with K' labelled samples of each class randomly to make up the query set $\mathcal{Q} = \{(\tilde{\mathbf{x}}_1, \tilde{\mathbf{y}}_1), ..., (\tilde{\mathbf{x}}_i, \tilde{\mathbf{y}}_i), ..., (\tilde{\mathbf{x}}_n, \tilde{\mathbf{y}}_n)\}_{i=1}^{n}$, where $\tilde{\mathbf{x}}_i$ is the features of the i-th query sample, $\tilde{\mathbf{y}}_i = \{\tilde{y}_i^{(0)}, ..., \tilde{y}_i^{(\ell)}, ..., \tilde{y}_i^{(h)}\}$ is the multi-layer labels of $\tilde{\mathbf{x}}_i$, and n is the number of the samples in query set. Then, we select K samples from the remainder of the C classes served as support set $\mathcal{S} = \{(\mathbf{x}_1, \mathbf{y}_1), ..., (\mathbf{x}_i, \mathbf{y}_i), ..., (\mathbf{x}_m, \mathbf{y}_m)\}_{i=1}^{m}$, where m is the sample number of the support set and $\mathbf{y}_i = \{y_i^{(0)}, ..., y_i^{(\ell)}, ..., y_i^{(h)}\}$ is the multi-layer labels of $\mathbf{x}_i$. Our experiments are based on 5-way 1-shot and 5-shot settings, which means each training task contains five classes. For each sample, we obtain its labels at different layers. At layer 0, there are five different labels representing the true class labels, while the label of their ancestors might be the same at other layers.

2.3 Hierarchical few-shot learning with tree-structured knowledge graph

In this section, we adopt ResNet12 as the backbone network of our network and develop a hierarchical classification task for few-shot learning by a loss function combining structure loss and data loss, which is formulated as:

$$\mathcal{L} = \mathcal{L}_{data} + \lambda \mathcal{L}_{structure} + \beta \mathcal{L}_{metric}, \tag{2}$$

where $\mathcal{L}_{data}$ denotes a loss for data depart, $\mathcal{L}_{structure}$ denotes a loss for tree structure part, and $\mathcal{L}_{metric}$ denotes a loss for metric learning. The hyper-parameters λ and β are used to balance the weight of tree structure loss and metric loss, respectively.

Granular computing model divides a problem space into many subspaces that form a hierarchical structure before solving the problem [50]. Inspired by the idea of granular computing, the coarse- and fine-grained classes are distributed at each layer according to the hierarchical tree structure [26]. There are fully-connected layers to classify samples at different layers of the hierarchical tree structure. The loss of query sample $\tilde{\mathbf{x}}_i$ at the ℓ-th layer is as follows:

$$\mathcal{L}_{\ell} = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{|\mathbf{Y}^{(\ell)}|} y_{ij}^{(\ell)} log \frac{exp(p_{ij}^{(\ell)})}{\sum_{k=1}^{|\mathbf{Y}^{(\ell)}|} exp(p_{ik}^{(\ell)})}, \tag{3}$$

where $y_{ij}^{(\ell)} = 1$ if the real class of query sample $\tilde{\mathbf{x}}_i$ is the j-th class at layer ℓ and $y_{ij}^{(\ell)} = 0$ otherwise; $p_{ij}^{(\ell)}$ is the probability which is computed with an ℓ-th fully-connected classification layer on query sample $\tilde{\mathbf{x}}_i$; n means the number of all query samples and $|\mathbf{Y}^{(\ell)}|$ denotes the number of classes at layer ℓ, which ranges from 0 to h.

For computer cognition, effective information is extracted through analyzing the input data and used to solve the problems [44]. Data is knowledge in the lowest granularity layer [46]. According to the representation of granularity at different layers of the hierarchical tree structure, the fully-connected layer is designed for final classification at layer 0. In contrast, the output of other fully-connected layers generates knowledge of the class structure to assist the final classification task. We consider the loss at layer 0 as the data loss, which is as follows:

$$\mathcal{L}_{data} = \mathcal{L}_0 = -\frac{1}{n}\sum_{i=1}^{n}\sum_{j=1}^{|\mathbf{Y}^{(0)}|} y_{ij}^{(0)} log \frac{exp(p_{ij}^{(0)})}{\sum_{k=1}^{|\mathbf{Y}^{(0)}|} exp(p_{ik}^{(0)})}. \tag{4}$$

In the field of machine learning, granular computing provides an effective method for simulating human cognitive mechanisms, including learning, thinking, reasoning, and solving complicated problem [47]. Human cognition is from coarser granularity layers to finer layers. Then, we employ the knowledge of layers from 1 to h according to

the hierarchical tree structure. This loss guides the model at different layers to correctly classify a query sample, which is as follows:

$$\begin{aligned}\mathcal{L}_{structure} &= \sum_{\ell=1}^{h} \mathcal{L}_{\ell} \\ &= -\frac{1}{n}\sum_{\ell=1}^{h}\sum_{i=1}^{n}\sum_{j=1}^{|\mathbf{Y}^{(\ell)}|} y_{ij}^{(\ell)} log \frac{exp(p_{ij}^{(\ell)})}{\sum_{k=1}^{|\mathbf{Y}^{(\ell)}|} exp(p_{ik}^{(\ell)})},\end{aligned} \tag{5}$$

where h denotes the number of layers except layer 0.

According to the parent–child relationship in every few-shot learning task, we gather the classes with the same ancestors to form a new subtree. Continuing with Example 1, we use Example 2 to explain how to get the subtree for each training task.

Example 2 Fig. 3 gives an example to describe the process of subtree construction in detail. The tree above is the tree structure of the entire dataset which contains all classes. The tree below is a subtree of the tree above. In a training task, five classes are selected randomly: Goldfinch, Bulbul, Agama, Iguana, and Plow. Based on the tree above, Goldfinch and Bulbul belong to the same parent, Passerine. In the same way, we obtain the membership of the remaining three classes and generate a new subtree for this training task. In the subtree, the multi-layer labels of the class Goldfinch are Goldfinch, Passerine.

Before computing the metric loss, we obtain prototype embedding following [39] to get a better feature representation of support and query samples. We use $\bar{\mathbf{q}}^{(c)}$ to represent a set of prototype embeddings which averages the K' query embedding vectors attended in the context of the support from c-th class of a C-way K-shot task and $\bar{\mathbf{s}}^{(c)}$ represents the average of K support embedding vectors for each class similarly. Figure 4 shows an example of the prototype embedding of support set in a 5-way 5-shot task.

After previous step, we use the metric-based loss to guide the model map a query embedding close to the prototype embedding of the same class in every few-shot learning task, which is as follows:

$$\mathcal{L}_{metric} = -\sum_{c=1}^{C} \log \frac{\exp(\text{sim}(\bar{\mathbf{s}}^{(c)}, \bar{\mathbf{q}}^{(c)})/\tau)}{\sum_{c'=1}^{C} \exp(\text{sim}(\bar{\mathbf{s}}^{(c')}, \bar{\mathbf{q}}^{(c')})/\tau)}, \tag{6}$$

where $\text{sim}(\cdot)$ is cosine similarity, and τ is a scalar temperature factor.

Knowledge at different granularity layers is the abstract representation of classes, and plays an auxiliary role in the classification ability of the model in data [44]. So we integrate human and computer cognition as multi-granularity cognition. The hierarchical few-shot learning loss function is constructed as follows:

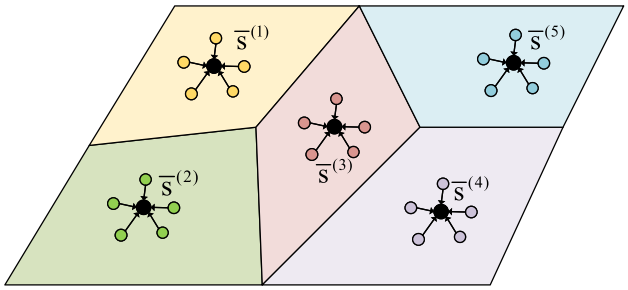

Fig. 4 The prototype embedding of support set in a 5-way 5-shot task

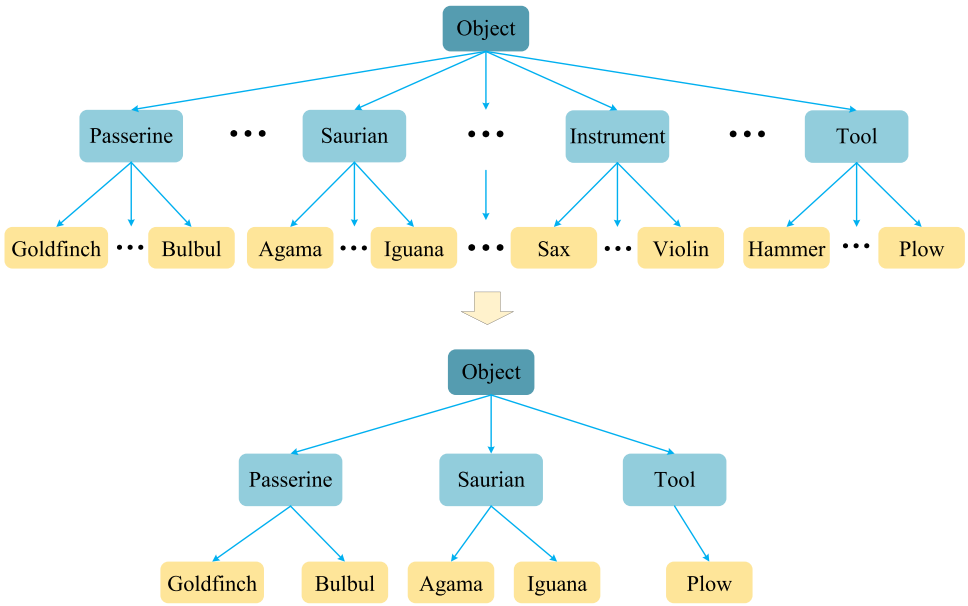

Fig. 3 Subtree of a training task establishment

$$\begin{aligned}\mathcal{L} &= \mathcal{L}_{data} + \lambda\mathcal{L}_{structure} + \beta\mathcal{L}_{metric} \\ &= \mathcal{L}_0 + \lambda\sum_{\ell=1}^{h}\mathcal{L}_\ell + \beta\mathcal{L}_{metric}.\end{aligned} \tag{7}$$

We balance the tradeoff of the participation of auxiliary knowledge in the hierarchical tree structure through the hyper-parameter λ. The hyper-parameter β is used to affect metric loss.

Algorithm 1 provides the training model and pseudocode for a training episode. We first select C classes from the training set and K, K' samples of each class to construct support set $\mathcal{S} = \{(x_1, y_1), ..., (x_m, y_m)\}$ and query set $\mathcal{Q} = \{(x_1, y_1), ..., (x_n, y_n)\}$ in line 2. Second, we construct a subtree and feed samples into *ResNet*12 to extract features

Algorithm 1 Knowledge Transfer Based Hierarchical Few-shot Learning via Tree-structured Knowledge Graph (HFKT).

Input: Training set $\mathcal{D} = \{(x_1, y_1), ..., (x_M, y_M)\}$ $(y \in [1, 2, ..., N])$, where M, N mean the number of samples and classes in the training set.

Initialization: Initialize model parameters learning rates of inner loop 10^{-1}, a momentum of 0.9, and weight decay of 0.005; hyper-parameters λ and β.

Iteration:

1: **for** each episode **do**
2: Randomly select C from N classes and K, K' samples of each class to construct support set $\mathcal{S} = \{(x_1, y_1), ..., (x_m, y_m)\}$ and query set $\mathcal{Q} = \{(x_1, y_1), ..., (x_n, y_n)\}$ separately;
3: Construct a subtree for the C classes, and the label of class layers in the subtree are $\{y^0, ..., y^\ell, ..., y^h\}$ $(\ell = 0, 1, ..., h)$.
4: Put the $\mathcal{Q}$ and $\mathcal{S}$ into $ResNet12$ to extract features;
5: Initialize loss $\mathcal{L} = 0$;
6: Compute data loss $\mathcal{L}_{data}$ according to Eq. (4);
7: **for** $\ell = 1 : h$ **do**
8: Compute structure loss $\mathcal{L}_{struture}$ by Eq. (5);
9: **end for**
10: Compute cosine similarity between a query and support prototype embedding of the same class and obtain $\mathcal{L}_{metric}$ according to Eq. (6);
11: Compute the hierarchical classification loss $\mathcal{L}$ according to Eq. (7);
12: Update loss $\mathcal{L}$;
13: **end for**

in lines 3 and 4. Then, we compute the data, structure, and metric loss listed from lines 5–10 and calculate the hierarchical classification loss in line 11. Finally, we use an SGD optimizer to minimize the loss and update the model parameters.

2.4 Discussion

The proposed model mainly consists of a tree-structured knowledge graph based on underlying semantic relationships and a knowledge transfer strategy for hierarchical few-shot learning.

(1) From the perspective of knowledge division, we consider a tree-structured knowledge graph, which is constructed by the semantic subordination and relation among the classes in WordNet. Then, we process the coarse- and fine-grained knowledge of this hierarchical tree structure within a unified framework instead of assuming that classes are independent of each other.
(2) From the knowledge transfer perspective, we propose a loss function to optimize the model by combining coarse- and fine-grained tasks in different granularities. The coarse-grained knowledge as external auxiliary knowledge is transferred into the fine-grained task to assist the classification of the proposed model.

3 Experimental settings

This section first introduces three datasets and implementation details used in our experiments. We then introduce nine comparison methods and three evaluation measures.

3.1 Datasets

We evaluate our model on four few-shot datasets, including tieredImageNet [36], CIFAR100 Few-Shots (CIFAR-FS) [25], Few-shot CIFAR100 (FC100) [9], and miniImagenet [43]. Table 1 lists the description of these datasets. They have a hierarchical class structure composed of class and superclass layers.

tieredImageNet [36]: This dataset is selected from the Imagenet dataset. The hierarchical class structure of tieredImageNet is shown in Fig. 5. It contains 34 superclasses, each of which contains 10–30 classes. Each class has several image samples, a total of 608 classes and 779,165 images (an average of 1281 images per class). The 34 superclasses can be divided into a training set (20 superclasses), a val set (6 superclasses), and a test set (8 superclasses).

CIFAR-FS [25]: The CIFAR-FS dataset is fully called the CIFAR100 Few-shot dataset, which is from the CIFAR100 dataset, including 100 classes, 600 images in each class, and 60,000 images in total. It is usually divided into a training set (64 classes), val set (16 classes), and test set (20 classes), and the image size is 32×32.

FC100 [9]: The full name of the FC100 dataset is the Few-shot CIFAR100 dataset, which is like the CIFAR-FS dataset above. It is also from the CIFAR100 dataset, including 100 classes, 600 images in each class, and 60,000 images in total. However, the difference is that FC100 is not divided by class but by superclass. It contains 20 superclasses (60 classes), including 12 superclasses in the training set, four superclasses (20 classes) in the val set, and four superclasses (20 classes) in the test set.

miniImagenet [43]: Deepmind team used the miniImagenet dataset for few-shot learning research for the first time. Since then, miniImagenet has become a benchmark dataset in meta-learning and few-shot learning. The miniImagenet dataset is taken from Imagenet and contains 100 classes, each containing 600 samples. Among them, 64 classes are used as a training set, 16 classes are used as a val set, and 20 classes are used as a test set.

3.2 Comparison methods

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

(1) Shot-Free [35]: This approach uses shot-free with an unlimited number of samples and embedded class models to enhance the ability of class representation.
(2) TPN [29]: This method proposes a novel meta-learning framework to alleviate the low-data problem via learning to propagate labels.
(3) MetaOptNet [23]: This method discriminatively trains linear classifiers that use high-dimensional embeddings to obtain better generalization.
(4) ProtoNet [39]: This approach learns a metric space in which classification can be performed by computing distances among the prototypes of each class.
(5) MatchNet [43]: This method leverages deep neural features to measure the distance of samples and augments neural networks with external memories.
(6) RFS-simple [42]: This method shows a simple baseline that trains a linear classifier on top of a pre-trained representation and achieves an additional boost through the use of self-distillation.
(7) DSN-MR [38]: This method proposes a framework for few-shot learning by introducing dynamic classifiers, which exploit a subspace method as the central block.

(8) Meta-Baseline [5]: This method explores a new meta-learning process over a whole-classification pre-trained model and analyses the tradeoffs between the meta-learning and the whole-classification objective.
(9) RENet [19]: This method proposes to tackle the problem of few-shot classification from a relational perspective by using self-correlational representation and cross-correlational attention modules.
(10) NCA [22]: This method uses nonparametric approaches, such as nearest neighbors to investigate the usefulness of episodic learning in methods for few-shot learning.
(11) UDS [15]: This method leverages a descriptor selection module and a task-related aggregation module to enhance internal representations.

3.3 Implementation details

We choose a residual network [14] with 12 layers (i.e., ResNet12) as the backbone network following the recent few-shot classification work [19, 51, 52]. For all experiments, we take the training episode C-way K-shot strategy by testing 15 query samples for each class in an episode. In test settings, we conduct a few-shot classification on randomly sampled 2000 episodes from the test set and compute the mean accuracy (%) and the 95% confidence interval. The optimizer is SGD with a momentum of 0.9, an initial learning rate of 10^{-1}, and a weight decay of 0.005. The hyper-parameter β is set to 0.25, 0.25, 0.5, 0.5 for tieredImageNet, miniImagenet, CIFAR-FS, FC100, respectively. We set scalar temperature factor $\tau = 0.2$. The above parameters are consistent with [19]. We determine hyper-parameter λ by experiment after determining parameter β. The hyper-parameter λ is set to 0.6, 0.6, 0.6, 0.5 for tieredImageNet, miniImagenet, CIFAR-FS, FC100 in 5-way 1-shot episode and 0.6, 0.4, 0.4, 0.4 in 5-way 5-shot episode, respectively. Also, the batch size is set to 64 for the three datasets. Our experiments are implemented in Pytorch with a GeForce RTX 2080 Ti Nvidia GPU card and a 3.60 GHz × 12 Intel Xeon W-2133 CPU. The basic data and code for this study have been uploaded to GitHub and can be accessed via the following link: https://github.com/fhqxa/HFKT.

3.4 Evaluation metrics

In this section, we introduce three hierarchical evaluation measures proposed in [20].

Tree induced error: We should punish different types of classification errors in hierarchical classification tasks. In the proposed model, the penalty is defined by the distance between the predicted class label and the true class label, which is termed the Tree Induced Error (*TIE*) [6], and its detailed formula is:

$$TIE(y_p, y_t) = |E_h(y_p, y_t)|, \tag{8}$$

where $E_h(y_p, y_t)$ is the edge set along the path from predicted class label y_p to true class label y_t in the class hierarchy and $|\cdot|$ is expressed as the number of elements in the set.

Hierarchical F_1-measure evaluation: The evaluation criteria for hierarchical classification models called hierarchical F_1-measure (F_H), which is defined as follows:

$$F_H = \frac{2 \cdot R_H \cdot P_H}{R_H + P_H}, \tag{9}$$

where R_H denotes hierarchical rate and P_H is hierarchical precision. They are written as follows:

Table 1 Benchmark datasets description

	Depth	Size	Training/val/test		Total
			Superclasses	Classes	
tieredImageNet	3	84 × 84 × 3	20/6/8	351/97/160	608
FC100	3	32 × 32 × 3	12/4/4	60/20/20	100
CIFAR-FS	3	32 × 32 × 3	20/11/13	64/16/20	100
miniImagenet	3	84 × 84 × 3	10/5/5	64/16/20	100

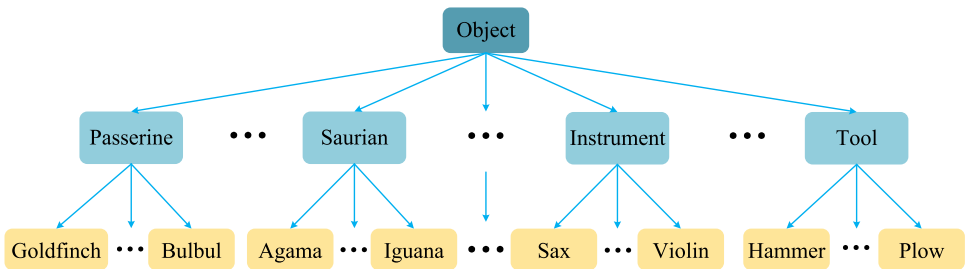

Fig. 5 The class hierarchical structure of the tieredImageNet dataset

$$R_H = \frac{|y_{aug} \cap \hat{y}_{aug}|}{|y_{aug}|}, \quad P_H = \frac{|y_{aug} \cap \hat{y}_{aug}|}{|\hat{y}_{aug}|}, \tag{10}$$

where y_{aug} is the real label of the test sample including its ancestor nodes, $\hat{y}_{aug}$ represents the augmented set of the predicted class, and $|\cdot|$ is expressed as the number of elements in the set.

Prediction accuracy (*ACC*): Classification accuracy is the most popular and simplest evaluation metric for flat and hierarchical classification. A label with the highest probability denotes a predicted label. The *ACC* is calculated as the ratio of the number of correctly predicted labels to the total number of test samples.

4 Experimental results and analysis

In this section, we first discuss the parameter effects of different datasets in the proposed model. Then the classification effect of tree-structured loss is discussed to verify the effectiveness of the HFKT. Furthermore, the effect and efficiency are compared with other models.

4.1 Performance comparison of different parameter λ values

In this section, we exploit the effects of the different parameter values λ on the effectiveness of HFKT. Firstly, we fix the parameter β to 0.5 and parameter λ is selected from the candidates $\{0.1, 0.2, 0.3, 0.4, 0.5, 0.6\}$. We discuss the impact of different λ values on HFKT classification by a series of 5-way 5-shot experiments on the CIFAR-FS dataset.

The *TIE* values of different λ on the CIFAR-FS dataset are shown in Fig. 6, and we can observe the followings:

(1) A series of experiments of a 5-way 5-shot is carried out on the CIFAR-FS dataset when the parameter β is fixed. The *TIE* evaluation reflects the degree of sample prediction error in the hierarchical structure. The *TIE* decreases from 0.1968% to 0.1837% when the value of the parameter λ increases from 0.1 to 0.4. This indicates that the knowledge from class structure assists in classification, and the improvement of the classification effect is more available with the increase of proportion.
(2) In addition, not all the knowledge from the class structure is useful for the classification tasks. Too much external knowledge will affect the final classification results. We find that the *TIE* increases from 0.1837% to 0.1882% when the parameter λ increases from 0.4 to 0.6. This is because the loss of class structure is too prominent in the model, ignoring the role of data feature loss, which reduces the classification effect of the model.

Table 2 lists *ACC* and F_H, and we can obtain the following observations:

(1) The ratio of *ACC* increases from 0.1 to 0.4, starting from 85.41% at 0.1, undergoing 85.97% at 0.2 and 86.08% at 0.3, and eventually ending up with 86.33% at 0.4. This shows that the classification accuracy can reach the highest when the loss of tree structure accounts for 40%.
(2) The accuracy and recall rate of the hierarchy can evaluate the affiliation of classes in the hierarchy. Moreover, F_H combines the results of accuracy and recall rate. The higher the value of F_H, the better the experimental effect. Similarly, the value of F_H achieves the best result when $\lambda = 0.4$, which presents proof from the perspective of the hierarchical tree structure.

4.2 Effectiveness of tree-structured loss

In this section, we exploit the effects of tree-structured loss (TSL) on the effectiveness of HFKT. We use the latest FSL model RENet [19] as the baseline (BS) model for our comparison. We set the parameter milestone to 60 or 70, which means that the learning rate will be halved when episode=60 or episode=70.

Figure 7 shows the effect of TSL on the 5-way 1-shot experiments training *TIE* and F_H on the CIFAR-FS dataset, and we can obtain the following observations:

(1) With the model starting training, F_H and *TIE* of BS+TSL begin to differentiate from the BS after several episodes, and then they are always above and below BS, respectively. This proves that TSL plays a positive role in the model using the hierarchical tree structure.
(2) The value of F_H and *TIE* with TSL maintains this trend when the learning rate is halved at episode 60. Then, they get the lowest and highest values of F_H and *TIE* respectively, at episode 80. This indicates that TSL remains stable after the learning rate changes.

Table 3 shows the performance of BS and BS + TSL accuracy on three benchmark datasets, and we can obtain the following observations. Compared with BS, TSL obtains the best results, and the hierarchical tree structure improves the

classification ability. Especially, TSL gets improved 2.37% on the CIFAR-FS dataset 5-way 1-shot experiments. Moreover, it outperforms BS by about 0.13% and 0.93% on the other two datasets on the 5-way 1-shot, respectively. This demonstrates that the semantic knowledge of the hierarchical tree structure plays an auxiliary role in class recognition.

4.3 Ablation study

This section shows the contribution of three loss functions to classification accuracy. We consider four cases: (1) using the data loss and metric loss; (2) using the structure loss and metric loss; and (3) using the three loss functions (HFKT). We only changed the loss function in the experiment, and the other settings remained the same. The main results are listed in Table 4, and we can obtain the following conclusions:

(1) Compared with metric loss only, the loss function combining data loss obtained accuracies of 71.32% and 85.86%, which are about 6.05% and 5.79% higher than that of the strategy of combining structure loss in 1-shot and 5-shot episodes, respectively. Structure loss can assist with fine-grained classification but cannot replace it because the goal of classification is to get fine-grained results,
(2) Combining the three losses can improve the accuracies by 2.37% and 0.47% in 1-shot and 5-shot episodes, respectively, because structure loss as external auxiliary knowledge can help fine-grained classification. In particular, the improvements of the three loss functions in the 1-shot episode are greater than that in the 5-shot episode, which indicates that the structure loss plays a greater role when training data is extremely scarce.

4.4 Comparison with other few-shot models

In this section, we compare HFKT with several state-of-the-art few-shot learning models on four datasets: tieredImageNet, CIFAR-FS, FC100, and miniImagenet, all of which have a hierarchical tree structure. The accuracies of the comparison models are copied from their original papers except for the RENet. Note that the results reported in Kang et al. [19] are denoted with "*". We use ResNet12 as the backbone network and follow the same experiment settings for the 5-way 1-shot and 5-shot experiments. Experimental results of the proposed HFKT and state-of-the-art models on it are listed in Tables 5, 6, 7, and 8.

From Table 5, we can obtain the following observations:

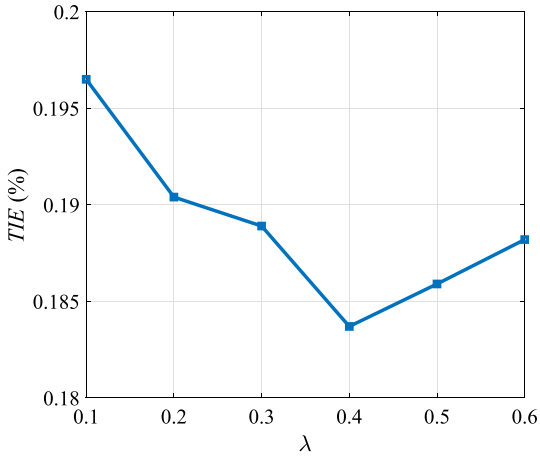

Fig. 6 Performance of parameter λ in joint loss function on CIFAR-FS (5-way 5-shot)

(1) The accuracy of HFKT is higher than other classification methods on the tieredImageNet dataset under 5-way 1-shot episodes. In the experiments of 5-way 1-shot, there is only one sample for each class, and models are difficult to recognize new classes via insufficient information. Therefore, as external auxiliary knowledge, a hierarchical tree structure can improve this problem.
(2) On the 5-way 5-shot, the proposed model is slightly higher than other comparison methods and almost the same as the ProtoNet. The prototype representation method has a satisfactory effect on the 5-way 5-shot experiments, which is almost similar to our model using class structure information.

Table 6 reports the performance of various models on the CIFAR-FS dataset, and we can obtain the following observations. Compared with the various models, HFKT classification performance is improved on the CIFAR-FS dataset on the 5-way 1-shot and 5-way 5-shot experiments. Especially, it outperforms the RENet model by about 2.38% under 5-way 1-shot episodes, which confirms that the knowledge from class structure benefits few-shot learning. Thus, the effectiveness of our HFKT is proved.

As listed in Table 7, under both 5-way 1-shot and 5-shot episodes, HFKT achieves accuracies of 45.10% and 60.42%, which are slightly higher than the best model RENet and outperform most other classification models. Because of its different data set partition methods, the effect of the semantic tree is affected.

Table 8 shows the accuracies of miniImagenet, which are 63.23% and 78.81%, under both 5-way 1-shot and 5-shot

Table 2 Performance of different values of the parameter λ on CIFAR-FS 5-way 5-shot experiments (%). The best results of each set are highlighted in bold

	0.1	0.2	0.3	0.4	0.5	0.6
ACC	85.54	85.97	86.08	**86.33**	86.03	85.96
F_H	95.18	95.34	95.37	**95.56**	95.41	95.37

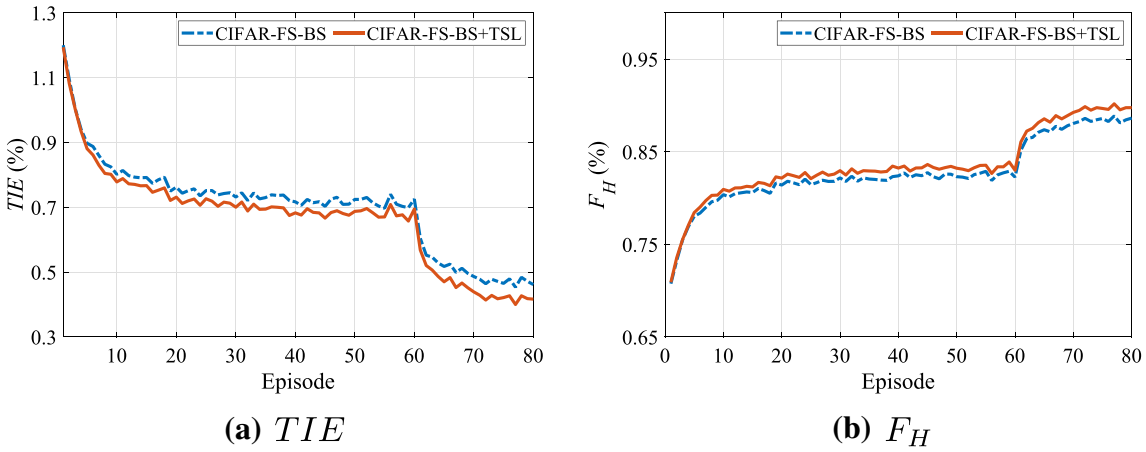

Fig. 7 Effectiveness of tree-structured loss on CIFAR-FS: (**a**) *TIE*; (**b**) F_H

Table 3 Performance of BS and BS+TSL accuracy on different datasets (%)

Dataset	BS	BS+TSL
FC100	44.97 ± 0.42	**45.10 ± 0.43**
tieredImageNet	69.68 ± 0.51	**70.63 ± 0.51**
CIFAR-FS	71.32 ± 0.47	**73.69 ± 0.48**

Table 4 Contributions of different loss functions on CIFAR-FS. The best performance is highlighted

$\mathcal{L}_{data}$	$\mathcal{L}_{structure}$	$\mathcal{L}_{metric}$	5-way	
			1-shot	5-shot
	✓	✓	65.27 ± 0.51	80.07 ± 0.37
✓		✓	71.32 ± 0.47	85.86 ± 0.34
✓	✓	✓	**73.69 ± 0.48**	**86.33 ± 0.33**

episodes. The performance of HFKT outperforms the RENet model and other methods in the flat dataset, which proves the universality of HFKT.

5 Conclusions and future work

This paper proposed a hierarchical classification method for few-shot learning based on a knowledge transfer strategy, which fully uses the knowledge from the semantic tree to relieve the problem of insufficient samples. The model employed the class structure knowledge to assist the classification tasks according to the hierarchical tree structure. First, we leverage a tree-structured class hierarchy among classes divided into two parts: the underlying leaf nodes are the data information, and their ancestors represent the additional class structure information. Second, this tree structure guides features to tend to two different forms and correspond to two different fully-connected layers, respectively. This strategy can get classification results in both class structure and data features. Finally, we transfer the useful class structure knowledge to the data layer as auxiliary information for a better classification result. Experimental results show that HFKT is comparable with several state-of-the-art few-shot learning models. We make full use of the structural information of the hierarchical label tree to construct the model. However, we only consider the fine-grained result during the label prediction. In the future, we will construct the hierarchical classification combining coarse- and fine-grained label prediction for few-shot learning.

Table 5 Comparison with the state-of-the-art 5-way 1-shot and 5-way 5-shot accuracy (%) with 95% confidence intervals on tieredImageNet. All these methods use ResNet-12 as embedding network. The best results of each set are highlighted in bold. Note that the results from the results reported in kang et al. [19] are denoted with "∗"

Method	5-way 1-shot	5-way 5-shot
TPN* [29]	59.91 ± 0.94	73.30 ± 0.75
MetaOptNet* [23]	65.99 ± 0.72	81.56 ± 0.53
ProtoNet* [39]	68.23 ± 0.23	84.03 ± 0.16
MatchNet* [43]	68.50 ± 0.92	80.60 ± 0.71
DSN-MR [38]	67.39 ± 0.82	82.85 ± 0.56
Meta-Baseline [5]	68.62 ± 0.27	83.74 ± 0.18
NCA [22]	68.35 ± 0.12	83.20 ± 0.09
RENet [19]	69.68 ± 0.51	83.86 ± 0.36
HFKT	**70.63 ± 0.51**	**84.17 ± 0.36**

Table 6 Comparison with the state-of-the-art 5-way 1-shot and 5-way 5-shot accuracy (%) with 95% confidence intervals on CIFAR-FS. All these methods use ResNet-12 as embedding network. The best results of each set are highlighted in bold. Note that the results from the results reported in Kang et al. [19] are denoted with "∗"

Method	5-way 1-shot	5-way 5-shot
Shot-Free* [35]	69.20	84.70
RFS-simple* [42]	71.50 ± 0.80	86.00 ± 0.50
ProtoNet* [39]	72.20 ± 0.70	83.50 ± 0.50
NCA [22]	72.49 ± 0.12	85.15 ± 0.09
MetaOptNet* [23]	72.60 ± 0.70	84.30 ± 0.50
RENet [19]	71.31 ± 0.47	85.86 ± 0.34
HFKT	**73.69 ± 0.48**	**86.33 ± 0.33**

Table 7 Comparison with the state-of-the-art 5-way 1-shot and 5-way 5-shot accuracy (%) with 95% confidence intervals on FC100. All these methods use ResNet-12 as embedding network. The best results of each set are highlighted in bold. Note that the results from the results reported in Kang et al. [19] are denoted with "∗"

Method	5-way 1-shot	5-way 5-shot
TADAM* [33]	40.10 ± 0.40	56.10 ± 0.40
MetaOptNet* [23]	41.10 ± 0.60	55.50 ± 0.60
Baseline2020 [8]	36.82 ± 0.51	49.72 ± 0.55
Meta-UAFS [53]	41.99 ± 0.58	57.43 ± 0.38
ConstellationNet [53]	43.80 ± 0.20	59.70 ± 0.20
RENet [19]	44.97 ± 0.42	59.89 ± 0.40
HFKT	**45.10 ± 0.43**	**60.42 ± 0.40**

Table 8 Comparison with the 5-way 1-shot and 5-way 5-shot accuracy (%) with 95% confidence intervals on miniImagenet. The best results of each set are highlighted in bold. Note that the results from the results reported in Kang et al. [19] are denoted with "∗"

Method	5-way 1-shot	5-way 5-shot
UDS [15]	56.00 ± 0.80	75.67 ± 0.61
TADAM* [33]	58.50 ± 0.30	76.70 ± 0.30
MetaOptNet* [23]	62.64 ± 0.82	78.63 ± 0.46
MatchNet* [43]	63.08 ± 0.80	75.99 ± 0.60
NCA [22]	62.55 ± 0.12	78.27 ± 0.09
MTL* [40]	61.20 ± 1.80	75.50 ± 0.80
RENet [19]	62.15 ± 0.45	78.11 ± 0.33
HFKT	**63.23 ± 0.46**	**78.81 ± 0.33**

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 Nos. 2021J011003 and 2021J011006.

Declarations

Conflict of 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.

References

1. AbuSalih B, AlTawil M, Aljarah I, Faris H, Wongthongtham P, Chan KY, Beheshti A (2021) Relational learning analysis of social politics using knowledge graph embedding. Data Min Knowl Disc 1–40
2. Andrychowicz M, Denil M, Gomez S, Hoffman MW, Pfau D, Schaul T, De Freitas N (2016) Learning to learn by gradient descent by gradient descent. In: International conference on neural information processing systems
3. Ayoub A, Jia Z, Szepesvari C, Wang M, Yang L (2020) Model-based reinforcement learning with value-targeted regression. In: International conference on machine learning
4. Bellet A, Habrard A, Sebban M (2015) Metric learning. Synthesis Lectures on Artificial Intelligence and Machine Learning 9(1):1–151
5. Chen Y, Liu Z, Xu H, Darrell T, Wang X (2021) Meta-baseline: exploring simple meta-learning for few-shot learning. In: IEEE/CVF International Conference on Computer Vision
6. Dekel O, Keshet J, Singer Y (2004) Large margin hierarchical classification. In: International Conference on Machine Learning
7. Dettmers T, Minervini P, Stenetorp P, Riedel S (2018) Convolutional 2D knowledge graph embeddings. In: Proceedings of the AAAI Conference on Artificial Intelligence
8. Dhillon GS, Chaudhari P, Ravichandran A, Soatto S (2020) A baseline for few-shot image classification. In: International Conference on Learning Representations

9. Everingham M, Van Gool L, Williams CKI, Winn J, Zisserman A (2010) The pascal visual object classes (VOC) challenge. Int J Comput Vision 88(2):303–338
10. Finn C, Abbeel P, Levine S (2017) Model-agnostic meta-learning for fast adaptation of deep networks. In: International Conference on Machine Learning
11. Ge Y, Li S, Li X, Fan F, Xie W, You J, Liu X (2021) Embedding semantic hierarchy in discrete optimal transport for risk minimization. In: IEEE international conference on acoustics, speech and signal processing
12. Guo S, Zhao H, Yang W (2021) Hierarchical feature selection with multi-granularity clustering structure. Inf Sci 568:448–462
13. He J, Hong R, Liu X, Xu M, Zha Z, Wang M (2020) Memory-augmented relation network for few-shot learning. In: International Conference on Multimedia
14. He K, Zhang X, Ren S, Sun J (2016) Deep residual learning for image recognition. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition
15. Hu Z, Li Z, Wang X, Zheng S (2022) Unsupervised descriptor selection based meta-learning networks for few-shot classification. Pattern Recogn 122:108304
16. Hui B, Zhu P, Hu Q, Wang Q (2019) Self-attention relation network for few-shot learning. In: IEEE International Conference on Multimedia and Expo Workshops
17. Jamal MA, Qi G (2019) Task agnostic meta-learning for few-shot learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition
18. Jiang R, Zhang J, Yan R, Tang H (2021) Few-shot learning in spiking neural networks by multi-timescale optimization. Neural Comput 33(9):2439–2472
19. Kang D, Kwon H, Min J, Cho M (2021) Relational embedding for few-shot classification. In: IEEE/CVF International Conference on Computer Vision
20. Kosmopoulos A, Partalas I, Gaussier E, Paliouras G, Androutsopoulos I (2015) Evaluation measures for hierarchical classification: a unified view and novel approaches. Data Min Knowl Disc 29(3):820–865
21. Kulis B et al (2012) Metric learning: a survey. Foundations and Trends in Machine. Learning 5(4):287–364
22. Laenen S, Bertinetto L (2021) On episodes, prototypical networks, and few-shot learning. Adv Neural Inf Process Syst 34:24581–24592
23. Lee K, Maji S, Ravichandran A, Soatto S (2019) Meta-learning with differentiable convex optimization. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition
24. Li A, Luo T, Lu Z, Xiang T, Wang L (2019) Large-scale few-shot learning: knowledge transfer with class hierarchy. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition
25. Li D, Ju Y, Zou Q (2016) Protein folds prediction with hierarchical structured svm. Curr Proteomics 13(2):79–85
26. Li J, Li Y, Mi Y, Wu W (2020) Meso-granularity labeled method for multi-granularity formal concept analysis. Journal of Computer Research and Development 57(2):447–458
27. Li X, Sun Z, Xue JH, Ma Z (2021) A concise review of recent few-shot meta-learning methods. Neurocomputing 456:463–468
28. Lim JY, Lim KM, Ooi SY, Lee CP (2021) Efficient-prototypicalnet with self knowledge distillation for few-shot learning. Neurocomputing 459:327–337
29. Liu Y, Lee J, Park M, Kim S, Yang E, Hwang S, Yang Y (2019) Learning to propagate labels: transductive propagation network for few-shot learning. In: International Conference on Learning Representations
30. Liu Z, Winata GI, Xu P, Fung P (2020) Coach: a coarse-to-fine approach for cross-domain slot filling. In: Annual Meeting of the Association for Computational Linguistics
31. Löffler C, Reeb L, Dzibela D, Marzilger R, Witt N, Eskofier BM, Mutschler C (2021) Deep siamese metric learning: a highly scalable approach to searching unordered sets of trajectories. ACM Trans Intell Syst Technol 13(1):1–23
32. Miller GA (1995) WordNet: a lexical database for english. Commun ACM 38(11):39–41
33. Oreshkin B, Rodriguez P, Lacoste A (2020) TADAM: task dependent adaptive metric for improved few-shot learning. In: International Conference on Neural Information Processing Systems
34. Ravi S, Larochelle H (2017) Optimization as a model for few-shot learning. In: International Conference on Learning Representations
35. Ravichandran A, Bhotika R, Soatto S (2019) Few-shot learning with embedded class models and shot-free meta training. In: IEEE/CVF International Conference on Computer Vision
36. Ren M, Triantafillou E, Ravi S, Snell J, Swersky K, Tenenbaum JB, Larochelle H, Zemel RS (2018) Meta-learning for semi-supervised few-shot classification. In: International Conference on Learning Representations
37. Rossi A, Barbosa D, Firmani D, Matinata A, Merialdo P (2021) Knowledge graph embedding for link prediction: a comparative analysis. ACM Trans Knowl Discov Data 15(2):1–49
38. Simon C, Koniusz P, Nock R, Harandi M (2020) Adaptive subspaces for few-shot learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition
39. Snell J, Swersky K, Zemel R (2017) Prototypical networks for few-shot learning. In: International Conference on Neural Information Processing Systems
40. Sun Q, Liu Y, Chua TS, Schiele B (2019) Meta-transfer learning for few-shot learning. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition
41. Sung F, Yang Y, Zhang L, Xiang T, Torr PH, Hospedales TM (2018) Learning to compare: relation network for few-shot learning. In: IEEE Conference on Computer Vision and Pattern Recognition
42. Tian Y, Wang Y, Krishnan D, Tenenbaum JB, Isola P (2020) Rethinking few-shot image classification: a good embedding is all you need? In: European Conference on Computer Vision
43. Vinyals O, Blundell C, Lillicrap T, Kavukcuoglu K, Wierstra D (2016) Matching networks for one shot learning. In: International Conference on Neural Information Processing Systems
44. Wang G (2017) DGCC: data-driven granular cognitive computing. Granular Computing 2(4):343–355
45. Wang S, Zhu W (2018) Sparse graph embedding unsupervised feature selection. IEEE Transactions on Systems Man and Cybernetics Systems 48(3):329–341
46. Wang X, Li J (2020) New advances in three-way decision, granular computing and concept lattice. Int J Mach Learn Cybern 11(5):945–946
47. Wang X, Wang P, Yang X, Yao Y (2021) Attribution reduction based on sequential three-way search of granularity. International Journal of Machine Learning and Cybernetics 12(5):1439–1458
48. Wang Y, Wang Z, Hu Q, Zhou Y, Su H (2021) Hierarchical semantic risk minimization for large-scale classification, IEEE Transactions on Cybernetics, 1–13
49. Yang F, Wang R, Chen X (2022) SEGA: semantic guided attention on visual prototype for few-shot learning. In: IEEE/CVF Winter Conference on Applications of Computer Vision
50. Yang J, Wang G, Zhang Q (2018) Knowledge distance measure in multigranulation spaces of fuzzy equivalence relations. Inf Sci 448:18–35
51. Ye H, Hu H, Zhan D, Sha F (2020) Few-shot learning via embedding adaptation with set-to-set functions. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition
52. Zhang C, Cai Y, Lin G, Shen C (2020) DeepEMD: few-shot image classification with differentiable earth mover's distance and structured classifiers. In: IEEE/CVF Conference on Computer Vision and Pattern Recognition
53. Zhang Z, Lan C, Zeng W, Chen Z, Chang SF Uncertainty-aware few-shot image classification, arXiv preprint arXiv:2010.04525

54. Zhao H, Hu Q, Zhu P, Wang Y, Wang P (2021) A recursive regularization based feature selection framework for hierarchical classification. IEEE Trans Knowl Data Eng 33(7):2833–2846
55. Zhu Q, Mao Q, Jia H, Noi OEN, Tu J (2022) Convolutional relation network for facial expression recognition in the wild with few-shot learning. Expert Syst Appl 189:116046