CSTS: Exploring Class-Specific and Task-Shared Embedding Representation for Few-Shot Learning

Hong Zhao, *Member, IEEE*, Yuling Su, Zhiping Wu, and Weiping Ding, *Senior Member, IEEE*

***Abstract*— Few-shot learning (FSL) is a challenging yet promising technique that aims to discriminate objects based on a few labeled examples. Learning a high-quality feature representation is key with few-shot data, and many existing models attempt to extract general information from the sample or task levels. However, the common sample-level means of feature representation limits the models generalizability to different tasks, while task-level representation may lose class characteristics due to excessive information aggregation. In this article, we synchronize the class-specific and task-shared information from the class and task levels to obtain a better representation. Structure-based contrastive learning is introduced to obtain class-specific representations by increasing the interclass distance. A hierarchical class structure is constructed by clustering semantically similar classes using the idea of granular computing. When guided by a class structure, it is more difficult to distinguish samples in different classes that have similar characteristics than those with large interclass differences. To this end, structure-guided contrastive learning is introduced to study class-specific information. A hierarchical graph neural network is established to transfer task-shared information from coarse to fine. It hierarchically infers the target sample based on all samples in the task and yields a more general representation for FSL classification. Experiments on four benchmark datasets demonstrate the advantages of our model over several state-of-the-art models.**

***Index Terms*— Few-shot learning (FSL), granular computing, hierarchical graph (HG) neural network, structure-guided contrastive learning (SCL).**

I. Introduction

FEW-SHOT learning (FSL) is a significant research hot spot in machine learning. Despite deep learning models having made remarkable achievements in this area, FSL remains challenging [1]. Deep learning models mainly benefit from having many labeled data and abundant model iterations [2], [3]. However, this seriously limits the generalization ability of deep learning models under the scene with little labeled data. In contrast, humans can learn a new object based on several samples by exploring previously accumulated knowledge. Inspired by this ability, FSL models have been developed to recognize new objects from a few labeled instances, and have been applied to various fields.

Meta-learning is the mainstream approach to dealing with the problem of data-scarcity. It advocates across-task learning and then adapts to new tasks [9]. The learning paradigm employed in meta-learning involves training a model on a base class with an ample number of samples to acquire prior knowledge, and subsequently transferring the knowledge to new tasks with a limited number of examples. Existing models based on meta-learning can be divided into three categories: optimization-based, data augmentation, and metric learning. The optimization-based approach mainly fine-tunings the model parameters to adapt to new tasks without retraining the model. The data augmentation method strives to augment training samples or enhance feature representation to compensate for the data scarcity problem [5], [20]. The optimization-based approach suffers from local optimization problems due to using few-shot data, while augmenting the data incurs additional costs.

Metric learning is another popular method, which focuses on learning a feature metric space to measure similarities for classification [8], [9], [10]. It usually adopts an average pooling approach to obtain support class representations, then a query sample is classified by matching with these class representations, where the class with the highest similarity is the predicted class. However, it commonly obtains the feature representations without considering the connections among classes at the sample level, resulting in limited generalization for different task scenes. Furthermore, graph learning is applied to FSL to explore the general information according to connections between the target sample and other samples at the task level [11], [12], [13], [14]. Additionally, some researchers have applied the class structure knowledge of data as a priori knowledge to FSL graph learning, demonstrating strong class semantic relations [15], [16]. Overall, the general information is primarily obtained through multiple iterations of information propagation within a task. However, the learned feature representations of different samples may tend to assimilate and lose their characteristics, i.e., intraclass variances or interclass commonness, due to excessive information aggregation.

In this article, an effective few-shot model is proposed to overcome these challenges by synchronously mining

Manuscript received 11 April 2023; revised 28 October 2023; accepted 30 December 2023. This work was supported in part by the National Natural Science Foundation of China under Grant 62141602 and Grant 61976120, in part by the Natural Science Foundation of Fujian Province under Grant 2021J011003 and Grant 2021J02049, in part by the Natural Science Foundation of Jiangsu Province under Grant BK20231337, and in part by the Natural Science Key Foundation of Jiangsu Education Department under Grant 21KJA510004. *(Corresponding author: Hong Zhao.)*

Hong Zhao, Yuling Su, and Zhiping Wu are with the School of Computer Science, Minnan Normal University, Zhangzhou, Fujian 363000, China, and also with the Key Laboratory of Data Science and Intelligence Application, Fujian Province University, Zhangzhou, Fujian 363000, China (e-mail: hongzhaocn@163.com; syuling@126.com; zhipingwucn@163.com).

Weiping Ding is with the School of Information Science and Technology, Nantong University, Nantong 226019, China (e-mail: dwp9988@163.com).

Digital Object Identifier 10.1109/TNNLS.2024.3380833

the class-specific and task-shared (CSTS) embedding representations. The CSTS model is divided into two parts: structure-guided contrastive learning (SCL) and hierarchical few-shot graph (HFG) learning. Specifically, a new contrastive learning-guided hierarchical class structure is introduced to effectively mine label information for better class-level representation. The hierarchical structure among classes provides more class variance and commonness, which compensates exactly for the data scarcity problem. In this way, samples with the same class are clustered in the embedding space, pushing away clusters of samples from different classes. Second, we establish a hierarchical graph (HG) neural network to infer the target sample with all the samples in the task and yield a more general representation from coarse to fine. In many practical applications, one regularly departs from specific fine-grained entity data and discovers commonness rules from coarse-grained data via abstraction based on granular computing. Simulating this human recognition mechanism, we select important near-node information from large to small scopes to enhance the target sample's features for FSL classification. In addition, HFG learning also provides a reasonable interpretation in the decision-making process. In this way, we obtain a better representation of learning from the class- and task-level information and improve the model generalizability for different FSL tasks. Experiments on several benchmark datasets under different scenarios are conducted to validate the competitive performance of CSTS against the existing advanced FSL models. The contributions of this article are summarized as follows.

1) We design a new FSL framework to endue feature representation with more latent information from the class and task level under conditions of data scarcity.
2) We introduce a structure-guided contrastive loss function to explore class-specific feature representation by defining negative samples from different aspects.
3) A HFG neural network is proposed to learn the task-shared characteristics among the samples inside the task. It transfers class commonness to enhance target sample features from coarse to fine while eliminating some useless features.

The following sections are organized as follows. We introduce research related to this study in Section II. Then, Section III presents the details of the proposed model. Next, Section IV introduces the experimental settings and presents and analyses the experimental results. Finally, we summarize the main conclusions of this article and ideas for further study in Section V.

II. Related Work

We review related literature on FSL based on meta-learning and contrastive learning, and discuss the differences between our model and existing ones.

A. FSL Based on Meta-Learning

Recently, many meta-learning-based FSL models have been proposed. They can be categorized into three main categories: data augmentation, optimization-based, and metric learning.

Data augmentation models attempt to solve the problem of insufficient data by augmenting samples or enhancing the data features [5]. This requires training an additional model, resulting in more cost. Optimization-based models aim to fine-tune model parameters adapted for new tasks under the few-shot regime, such as MAML [6] and MetaOptNet [19]. They learn model parameters with a few labeled examples to obtain a model that predicts new classes based on the initialized model [20]. Third, metric learning models aim to learn a pairwise similarity metric between query and support samples, where more similar samples have higher similarity, and less similar ones have lower similarity [2]. FSL models based on metric learning can be categorized into relation and distance metrics. The former focus on studying a learnable relation module through a neural network based on similarity metrics [8], [21]. The latter utilizes distance formulas to measure the similarity between two samples. For instance, Vinyals et al. [2] adopted the cosine distance, while the prototypical network [22] used the Euclidean distance. In addition, DeepEMD [9] embedded the Earth Mover's distance to compute the structural distance between dense samples to determine image relevance. However, most models obtain the feature of each sample dependently and ignore its relations with other samples.

Several studies have applied graph neural networks to FSL problems, exploring the relation and interdependency among samples to enhance feature representation [23], [24]. They learn the general information between the target sample and other samples in the FSL task by information transmission [11], [14], [25]. Furthermore, some researchers embed class structure knowledge to guide FSL graph learning to mine more label information [15]. The hierarchical class structure demonstrates strong semantic intraclass and interclass relations, which provides additional information under data-scarce scenes [26], [27]. For example, an explicit class knowledge propagation network [28] was designed to obtain more class knowledge representations for guiding the inference of query samples. In contrast, Chen et al. [16] proposed a HG neural network enabling efficient learning of multilevel relationships, which imitates the ideal hierarchical class structure. They use the class structure knowledge to propagate task-level information for representation learning.

B. FSL Based on Contrastive Learning

Contrastive learning computes contrastive losses acting on the low-dimensional feature representations, which measure the similarities of different samples in the feature space [17], [29]. Recently, contrastive learning has been widely used in FSL problems to learn distinguishable class features, which benefits from the success of contrastive learning in representation learning [30]. For instance, Ouali et al. [31] designed a spatial contrastive learning approach to obtain locally discriminative and class-agnostic features. Similarly, based on local features, Liu et al. [32] proposed a noise contrastive estimation approach to establish local feature contrastive learning and eliminate the inductive bias of known classes. In contrast to single-branch learning, Kwon et al. [33] introduced a

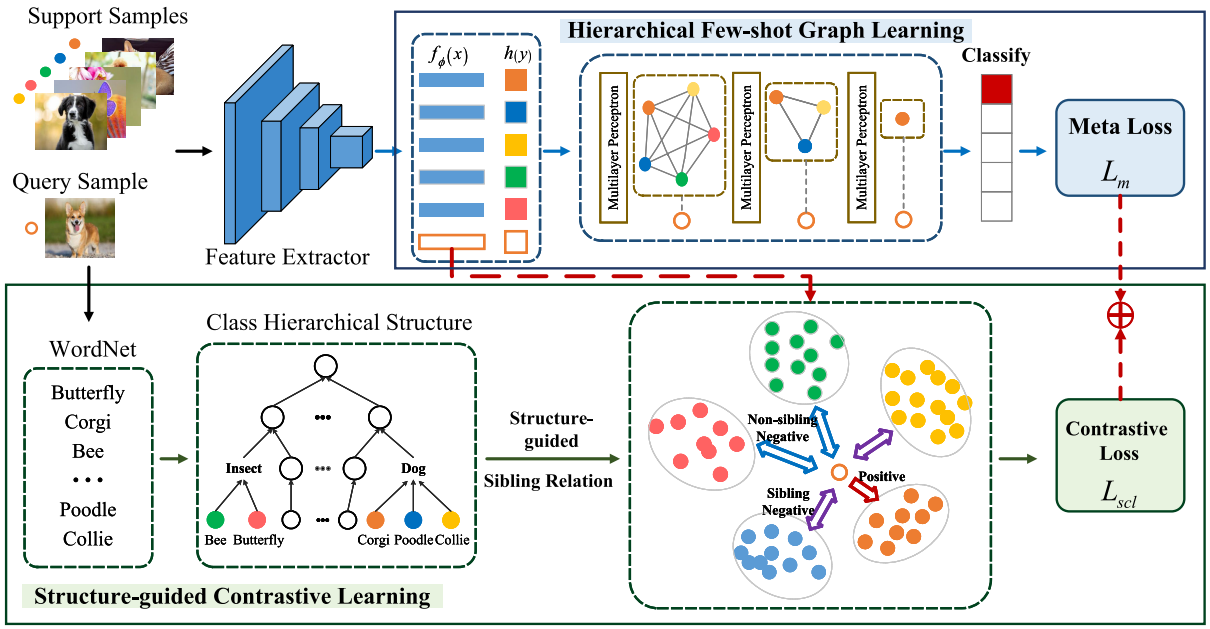

Fig. 1. Framework of CSTS. CSTS mainly consists of SCL and HFG learning. f_θ represents the feature encoder, $h(\cdot)$ is a label encoding function, and L_m and L_{scl} are the meta and semantic-guided supervised contrastive losses, respectively.

dual prototypical contrastive learning approach. They defined samples in the same class as positive samples and treated all samples of different classes equally, defining them as negative samples.

C. Discussion

Although great efforts have been dedicated to FSL, the noted models suffer from the following limitations.

1) Common sample-level means of feature representation limit the model's generalizability to different tasks, while using information transmission technology to learn task-level information may lead to missing class characteristics.
2) Contrastive-learning-based few-shot models focus on the differences in interclass features, which limits their generalizability to different tasks.

In this article, we explore CSTS embedding representations synchronously. The SCL approach is proposed to study class-specific features. It works by defining negative samples from different aspects to obtain further information on interclass variances in FSL scenarios. Compared with supervised contrastive learning, the proposed approach puts greater importance on distinguishing classes with similar characteristics rather than all classes. This achieves the goal of reducing intraclass distances and expanding interclass distances. Meanwhile, inspired by [16], we introduce a HG neural network to mine general information that is task-shared between the target sample and all samples in a task. Unlike [16], we mimic granular computing and hierarchically infer the important near samples of the target sample from coarse to fine while eliminating some useless samples. This approach can provide a reasonable interpretations of the decision-making process to detect errors in classification or transmission.

III. Proposed Approach

In this section, we present the framework and specific classification steps of CSTS (Fig. 1). First, SCL is introduced to learn distinctive feature representations by exploring the intraclass variances in the class hierarchical structure. Second, we establish a HFG neural network to propagate the task-shared information to enhance query samples used for FSL classification from coarse to fine. HFG includes four main parts: class prototype acquisition, intralevel and interlevel propagation, and multilevel fusion prediction (MFP). To summarize, CSTS aims to simultaneously explore class-specific variances and task-shared commonness to obtain useful query features that promote the model's classification abilities. For the test phase, we utilize a hierarchical few-shot model trained to classify query samples after obtaining a feature representation from a trained feature extractor.

A. Problem Formulation

The FSL approach aims to train a model that can generalize to novel classes with only a few training examples per class. We adopt episodic training for meta-learning-based FSL problems, which is commonly utilized in many studies [8], [16], [22]. Each FSL task consists of a support set $\mathcal{S} = \{(x_1, y_1), \ldots, (x_{n_s}, y_{n_s})\}$ and a query set $\mathcal{Q} = \{(x_1, y_1), \ldots, (x_{n_q}, y_{n_q})\}$, where x represents a sample, y is its corresponding label, and n_s, n_q are the numbers of their samples. If the support set contains K labeled samples for each of N classes, the problem is called an N-way K-shot problem [14], [28].

In episodic learning, the training and test tasks of the N-way K-shot problem are formed: for each FSL task, $\mathcal{T} = \mathcal{S} \cup \mathcal{Q}$. Note that the sample labels in both the support and query sets are known in the training phase. While the sample labels

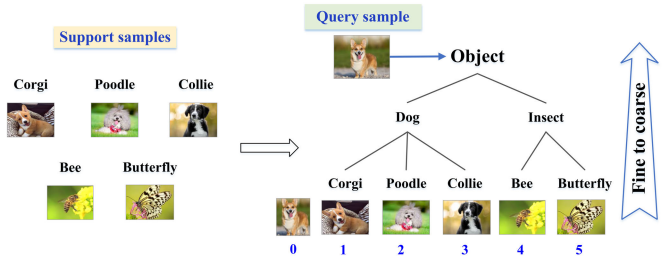

Fig. 2. Example of hierarchical class structure. Sample 0 is the data augmentation (cropping) of the query sample.

in the support set are known in the test phase, those in the query set are unknown. The purpose is to train a model based on training set samples and enable the learned classifier to classify the query set into the available classes of the support set in the test phase.

B. Structure-Guided Contrastive Learning

Representative features provide good adaptability in the test phase, which plays a crucial role in FSL classification. We explore contrastive learning as an auxiliary objective to capture class-specific features for learning general-purpose feature representations. This facilitates recognition of new, unseen classes in the test phase. Furthermore, we embed the class hierarchical structure to guide contrastive learning by the sibling strategy, which aims to learn more class-specific information.

1) Class Hierarchical Structure Construction: The hierarchical class structure is required to obtain the multigranularity class relationships, which indicate the strong semantic relations between classes. This uses granular computing to describe the specific classes of fine-grained data at a low level, and formulates the coarse-grained data knowledge pieces at a higher, abstract level. A coarse-grained class contains common features of multiple fine-grained classes, while fine-grained classes with sibling relationships reveal the class variances [35]. In addition to the multigranularity relationships of the dataset itself, the semantic clustering approach is an effective way to construct the hierarchical structure. First, we obtain each class name from Wikipedia documents, then the classes are exploited as fine-grained classes to form the bottom class layer of the hierarchical class structure. Then, from WordNet [36], we use path similarity to compute the semantic similarities of every two classes. Starting from the leaves, we obtain the coarse-grained nodes in the upper layer by clustering over the class semantic similarities of the nodes in the lower layer. Based on this, a hierarchical class with two granularity classes is constructed. Fig. 2 intuitively shows the semantic dependency and visual similarity of the hierarchy structure in the support samples. The fine-grained classes *Corgi*, *Poodle*, and *Collie* belong to the same coarse-grained class *Dog*, which shows the strong semantic relations between coarse-grained and fine-grained classes. These three fine-grained classes are siblings and have similar visual features.

2) Structure-Guided Contrastive Learning: Sibling relations indicate that classes have a close relationship, while two classes with nonsibling relations belong to different coarse-grained classes and have a further interclass distance. Classifying sibling classes with similar visual concepts is more difficult than classifying nonsibling classes with different appearances. The cost of distinguishing classes with sibling relations is greater than that of recognizing classes without them. For example, the difficulty in distinguishing between classes *Corgi* and *Poodle* is greater than that between classes *Corgi* and *Bee*, since the former are sibling classes. Thus, we introduce SCL to learn general-purpose feature representations by the sibling strategy. The structure-guided contrastive loss $\mathcal{L}_{\text{scl}}$ is followed as:

$$\mathcal{L}_{\text{scl}} = \alpha \mathcal{L}_s + (1-\alpha)\mathcal{L}_{\bar{s}} \tag{1}$$

where $\mathcal{L}_s$ and $\mathcal{L}_{\bar{s}}$ represent the sibling and nonsibling contrastive losses, respectively, and α is a weight factor used to trade off the sibling and nonsibling contrastive losses, and ranges from 0 to 1. The model pays greater attention to the sibling contrastive loss than the nonsibling contrastive loss when $\alpha > 0.5$. Therefore, in a task $\mathcal{T}$, given a sample x_i, the sibling and nonsibling contrastive losses are calculated as

$$\begin{cases} \mathcal{L}_s = \displaystyle\sum_{i\in\mathcal{T}} \frac{-1}{|P(i)|} \sum_{p\in P(i)} \log \frac{\exp\left(\mathbf{x}_i \cdot \mathbf{x}_p\right)/\tau}{\sum_{s\in S(i)} \exp(\mathbf{x}_i \cdot \mathbf{x}_s)/\tau} \\ \mathcal{L}_{\bar{s}} = \displaystyle\sum_{i\in\mathcal{T}} \frac{-1}{|P(i)|} \sum_{p\in P(i)} \log \frac{\exp\left(\mathbf{x}_i \cdot \mathbf{x}_p\right)/\tau}{\sum_{\bar{s}\in \bar{S}(i)} \exp(\mathbf{x}_i \cdot \mathbf{x}_{\bar{s}})/\tau} \end{cases} \tag{2}$$

where $\mathbf{x}_i$ represents the feature of the sample x_i, $P(i) = \{p \in A(i) : y_p = y_i\}$ is the set of indices of all positives of x_i and $|P(i)|$ is its cardinality, $A(i)$ is the set of indices excluding i, $S(i)$ is the set of indices of all sibling samples of x_i and $\bar{S}(i)$ represents the set of indices of all nonsibling samples of x_i in a task, and τ is a scalar temperature parameter. Also, we adopt a progressive iteration to update the temperature parameter τ, as follows:

$$\tau = \frac{\tau_0}{1+\exp(T)} \tag{3}$$

where τ_0 represents the initial value of the temperature parameter and T is the number of the current episode. As the iterative training continues, the temperature parameter gradually decreases, accelerating convergence of the model. In this way, SCL makes the sample features of the same class close and distinct from those of different classes.

According to the definitions of positive samples, contrastive learning can be categorized into two methods: self-supervised contrastive methods [29], [31], [37] and supervised contrastive methods [17], [32], [33]. We take Example 1 to further explain the differences between SCL and the two methods listed in Table I.

Example 1: Here, we interpret the different positive and negative samples of different strategies. In Fig. 2, given a query sample x_i in class *Corgi*, we have the following results [positive set $P(i)$ and negative set $N(i)$].

1) *Self-Supervised Contrastive:* The positive samples are only the data augmentation of the query sample, i.e., $P(i) = \{0\}$ and $N(i) = \{1, 2, 3, 4, 5\}$.

TABLE I

COMPARISON OF DIFFERENT CONTRASTIVE LEARNING STRATEGIES. GIVEN A SAMPLE x_i, Aug(i): THE DATA AUGMENTATION OF x_i, Sim(i): THE SAMPLES IN THE SAME CLASS AS x_i, AND Dif(i): THE SAMPLES IN DIFFERENT CLASSES FROM x_i

Contrastive learning	Positive samples	Negative samples
Self-supervised contrastive	$Aug(i)$	$Sim(i)+Dif(i)$
Supervised contrastive	$Sim(i)$	$Dif(i)$
SCL	$Sim(i)$	$S(i)+\bar{S}(i)$

2) *Supervised Contrastive:* The positive samples include all samples in the same class as x_i, i.e., $P(i)=\{0,1\}$ and $N(i)=\{2,3,4,5\}$.
3) *SCL:* The definition of positive samples is the same as for the supervised contrastive method, but the negative samples are divided into sibling and nonsibling samples, i.e., $P(i)=\{0,1\}$, $S(i)=\{2,3\}$, and $\bar{S}(i)=\{4,5\}$.

Compared with these two methods, we focus on distinguishing relevant and similar classes instead of exploring individual classes. Distinguishing the similar classes is given more cost by the sibling strategy in effectively learning the intraclass characteristics. This aims to increase the interclass distance and reduce the intraclass distance in the feature space. SCL can degenerate into supervised contrastive learning when the weights of sibling and nonsibling loss are equal. This indicates that our method is more general than the other two methods.

C. HFG Learning

We establish a HFG neural network to learn interclass commonness for FSL classification from the task level. As shown in Fig. 3, HFG can be divided into four parts: class prototype acquisition, intralevel propagation, interlevel propagation, and MFP. We introduce a weighted fusion strategy to obtain the class prototype representation, which aims to obtain a great class information representation. Then, we reuse the features in the previous level to enhance the node features in the current level because there may be useful information remaining in it. Finally, a MFP strategy is designed to conduct multilevel prediction instead of single-level prediction, which can alleviate classification bias and increase the diversity of classification. We transfer the task-shared information to enhance the query feature representation from coarse to fine, which selects important node information while eliminating some useless features. This simulates the human recognition mechanism and intends to provide reasonable interpretation in the decision-making process. In the fusion prediction process, CSTS can provide a clear prediction sequence at different graph levels and help to analyze and locate the root causes of erroneous classification.

Specifically, we first generate hierarchical multilevel graphs, apply neighbor node aggregation to each level graph (intralevel propagation), use the ℓth level graph to update the nodes of the $(\ell+1)$th level graph by choosing the k-nearest nodes (interlevel propagation), and fuse the multilevel predictions to classify query samples. Before interlevel propagation, we leverage a weighted averaging strategy to obtain the class prototype. By increasing the levels of the graph, we can transfer more task-shared information to enhance the feature representation of the target sample based on having more samples in a task. However, a deep graph level degrades the classification performance. This is because when the graph level reaches a certain depth, all images tend to be at the same point in the feature space and lose their characteristics, making classification difficult.

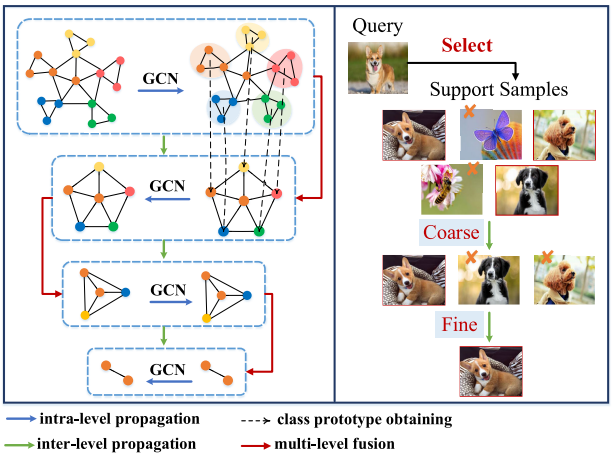

Fig. 3. HFG progress. We adopt weighted aggregation to obtain the prototype representation of each class, then aggregate effective information through intralevel propagation and select more important nodes for downward interlevel propagation. Finally, a multilevel fusion strategy is introduced for prediction.

Specifically, in a task, let $G^{(\ell)}=(\mathbf{V}^{(\ell)},\mathbf{E}^{(\ell)})$ be hierarchical multilevel graphs, where $0\le\ell\le l$ and l represents the number of graph levels. In a graph $G^{(\ell)}$, each node $\mathbf{v}_i^{(\ell)}\in\mathbf{V}^{(\ell)}$ denotes the feature vector of every sample, and the edge set $\mathbf{E}^{(\ell)}$ is represented as an adjacency matrix $\mathbf{A}^{(\ell)}\in R^{n_\ell\times n_\ell}$, where $a_{i,j}^{(\ell)}\in\mathbf{A}^{(\ell)}$ denotes similarity between the two connected nodes $\mathbf{v}_i^{(\ell)}$ and $\mathbf{v}_j^{(\ell)}$, and n_ℓ means the number of nodes in the ℓth level graph.

Initially, for a N-way K-shot task $\mathcal{T}=\mathcal{S}\cup\mathcal{Q}$, all samples are put into the feature extractor f_ϕ to obtain the sample features. The node $\mathbf{v}_i^{(0)}$ is represented by concatenating the label encoding with the feature of a sample x_i from $\mathcal{T}$, as follows:

$$\mathbf{v}_i^{(0)}=\left(f_\phi(x_i),h(y_i)\right) \tag{4}$$

where y_i is the label of sample x_i and $h(\cdot)$ is a label encoding function. For a support sample, the label encoding is its one-hot encoding vector while using a uniform distribution vector $[(1/N),\ldots,(1/N)]$ containing N values to represent the query sample, where N is the number of classes. In this way, the node set is represented by $\mathbf{V}^{(0)}\in R^{(n_s+n_q)\times d}$, where d is the dimension of the node feature vector. The adjacency matrix $\mathbf{A}^{(0)}\in R^{(n_s+n_q)\times(n_s+n_q)}$ are initialed as: for the support set, $a_{i,j}^{(0)}=1$ if the node $\mathbf{v}_i^{(0)}$ is associated with the node $\mathbf{v}_j^{(0)}$, otherwise $a_{i,j}^{(0)}=0$; for the query set, $a_{i,j}^{(0)}=0.5$ whether the two nodes are connected or not.

1) Intralevel Propagation: We propose an intralevel propagation strategy to explore the class commonness among the nodes at the same level. It aims to propagate neighbor information to enhance features. First, we use a nonlinear

multilayer perceptron (MLP_θ) to update the adjacent matrix of the relations between node representations. The $\tilde{a}_{i,j}^{(\ell)} \in \tilde{\mathbf{A}}^{(\ell)}$ is the new adjacent value of the relation between nodes $\mathbf{v}_i^{(\ell)}$ and $\mathbf{v}_j^{(\ell)}$, which is calculated by the absolute difference between the two node features, as follows:

$$\tilde{a}_{i,j}^{(\ell)} = \text{MLP}_\theta\left(\text{abs}\left(\mathbf{v}_i^{(\ell)} - \mathbf{v}_j^{(\ell)}\right)\right), \quad i, j = 1, \ldots, n^\ell \tag{5}$$

where $\tilde{\mathbf{A}}^{(\ell)}$ means the updated adjacent matrix in the ℓth level, and n^ℓ is the number of nodes in the ℓth level. Then, we take the updated adjacent matrix $\tilde{\mathbf{A}}^{(\ell)}$ to learn node features $\tilde{\mathbf{X}}^{(\ell)}$ from the current node representations by information passing

$$\tilde{\mathbf{V}}^{(\ell)} = \sigma\left(\tilde{\mathbf{D}}^{(-\frac{1}{2})}\tilde{\mathbf{A}}^{(\ell)}\tilde{\mathbf{D}}^{(-\frac{1}{2})}\mathbf{V}^{(\ell)}\mathbf{W}^{(\ell)}\right) \tag{6}$$

where $\mathbf{V}^{(\ell)}$ is the input node features, $\sigma(\cdot)$ is an activation function, $\tilde{\mathbf{D}}^{(-(1/2))}$ represents the degree matrix of $\tilde{\mathbf{A}}^{(\ell)}$, and $\mathbf{W}^{(\ell)}$ a learnable weight matrix of a linear transformation changing the dimensionality of the feature space. In this way, we achieve the interclass commonness propagation between levels.

2) Class Prototype Acquisition: Using all samples to train the model for the K-shot ($K > 1$) will consume time and increase computational complexity. To tackle this, HGNN [16] adopted a K-Nearest Graph Pooling strategy to select the nearest node to the centroid of a class, as shown in Fig. 4(a). However, this can easily filter out useful information if only one sample is selected. In this article, we design a class prototype strategy that fuses all of the sample information to obtain a good class representation [Fig. 4(b)]. In addition, we adopt a weighted fusion strategy using node importance scores to aggregate information. A node close to a class centroid is indicative of great importance and a high score. We utilize graph convolutions and MLP to project all node features to importance scores. The class prototype is obtained in level 1 ($\ell = 1$) after the intralevel propagation of level 0, and the progress is as follows:

$$\begin{cases} \mathbf{v}_i^{(1)} = \displaystyle\sum_{k \in Y(i)} s_k^{(0)} \tilde{\mathbf{v}}_k^{(0)} \\ a_{i,j}^{(1)} = \displaystyle\sum_{k \in Y(i)} s_k^{(0)} \tilde{a}_{k,j}^{(0)} \end{cases} \tag{7}$$

where $Y(i)$ is the index set of all nodes of support set in class i, $i = 1, \ldots, N$. The $s_k^{(\ell)}$ is the importance score of node k in level ℓ ($\ell = 0$ in this case), which is obtained by a lightweight fully connected layer. The adjacent matrix is correspondingly changed with acquisition of the class prototype features of the nodes. For a one-shot task, there is no such process in obtaining the class prototype. We only adopt the class prototype obtained for the support set while reserving all nodes of the query set.

3) Interlevel Propagation: We introduce an interlevel propagation strategy to downsample the support nodes, which chooses the K-Nearest nodes and removes other unrelated nodes. Interlevel propagation aims to reduce the size of the classes in the support. We only adopt interlevel propagation for the support set while reserving all the nodes of the query set. It explores the relations between the support and query nodes to learn meaningful query features by passing the class commonness of the support set. We take the node importance scores to select the K-Nearest nodes to the centroid. The process is given as follows:

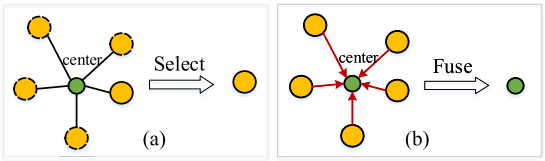

Fig. 4. Differences of the class prototype obtained between HGNN and ours. (a) HGNN: select the nearest node to the centroid. (b) Ours: fuse all sample information to obtain a class prototype representation.

$$c^{(\ell)} = \frac{1}{K^{(\ell)}} \sum_{i=1}^{K^{(\ell)}} s_i^{(\ell)} \tag{8}$$

where $c^{(\ell)}$ is the value of the centroid in the ℓ ($\ell \geq 1$) level, $K^{(\ell)}$ means the number of nodes in ℓ level, and $s_i^{(\ell)}$ is the score of node i in ℓ level.

After obtaining the value of the level centroid, we select the K-Nearest nodes to the centroid according to the importance scores

$$\text{idx}_s^{(\ell+1)} = \text{Nearest}\left(c^{(\ell)}, S^{(\ell)}, K^{(\ell+1)}\right) \tag{9}$$

where Nearest function obtains the indices of the top $K^{(\ell+1)}$ nearest nodes to the centroid of the $(\ell+1)$ level, $\text{idx}_s^{(\ell+1)}$ is the indices of the selected support nodes, and $S^{(\ell)}$ represents the score set of support nodes in ℓ level. Then, we merge the selected support and unchanged query indices to obtain the indices $\text{idx}^{(\ell+1)}$ of all nodes in the next level: $\text{idx}^{(\ell+1)} = \text{idx}_s^{(\ell+1)} \cup \text{idx}_q$, where idx_q is the unchanged indices of the query nodes.

Next, we leverage the obtained indices $\text{idx}^{(\ell+1)}$ to obtained next adjacent matrix $\mathbf{A}^{(\ell+1)}$ and node features $\mathbf{V}^{(\ell+1)}$ of the $(\ell+1)$ level. In addition, we also utilize the importance scores $S_{\text{idx}^{(\ell+1)}}^{(\ell)}$ to control the information aggregation. The node features $\tilde{\mathbf{V}}^{(\ell+1)}$ is obtained by conducting the element-wise product of $S_{\text{idx}^{(\ell+1)}}^{(\ell)}$ and $\tilde{\mathbf{V}}^{(\ell)}$. The progress can be formulated as follows:

$$\begin{cases} \mathbf{A}^{(\ell+1)} = \tilde{\mathbf{A}}_{\text{idx}^{(\ell+1)}}^{(\ell)} \\ \mathbf{V}^{(\ell+1)} = \tilde{\mathbf{V}}_{\text{idx}^{(\ell+1)}}^{(\ell)} \odot S_{\text{idx}^{(\ell+1)}}^{(\ell)} \end{cases} \tag{10}$$

where $\tilde{\mathbf{A}}^{(\ell)}$ and $\tilde{\mathbf{V}}^{(\ell)}$ are the adjacent matrix and node features obtained by the intralevel propagation, and $\odot$ means the element-wise product. Furthermore, we believe that the node information of the previous level is still useful and should not be completely discarded. We combine the query node features of the previous level to generate the query node features in the current level

$$\mathbf{V}^{(\ell+1)} = \eta \mathbf{V}_{\text{idx}_q}^{(\ell+1)} + (1-\eta)\tilde{\mathbf{V}}_{\text{idx}_q}^{(\ell)} \tag{11}$$

where η is a trade-off factor. The purpose is to merge the useful information of different levels to enhance query features.

4) Multilevel Fusion Prediction: Interlevel propagation provides the multilevel query features. We then use graph convolutions and a Softmax layer to map the query features of each level to the classification probability of each support class. Then, we fuse the multilevel probabilities to obtain the final classification probabilities of the query samples. This aims to utilize the information of the previous levels to assist in learning the last level because they may contain useful information. The final classification probability p_{ij} of the ith query sample that is classified to jth class, which is calculated as

$$p_{ij} = \sum_{\ell=0}^{l} \mu^{(\ell)} p_{ij}^{(\ell)} \tag{12}$$

where $p_{ij}^{(\ell)}$ is the classification probability of the ith query sample that is classified to jth class in (ℓ) level, and $\mu^{(\ell)}$ means the weight of different levels. Parameter l is the number of levels, where $l = 2$ is in a one-shot task while $l = 3$ is in a five-shot task. We fuse multilevel probabilities to interpret the process of classification decisions reasonably. Next, we adopt the cross-entropy loss to compute the meta-learning loss $\mathcal{L}_m$ to optimize CSTS, which is denoted as

$$\mathcal{L}_m = -\sum_{i=1}^{n_q} \sum_{j=1}^{N} y_{ij} p_{ij} \tag{13}$$

where n_q is the number of query samples and N is the number of classes in a task, and $y_{ij} = 1$ means the truth label of ith query sample is jth class, otherwise $y_{ij} = 0$.

Finally, we combine the structure-guided contrastive loss $\mathcal{L}_{\text{scl}}$ and the classification loss $\mathcal{L}_m$ to obtain the final loss of our model. According to (1) and (13), the total loss $\mathcal{L}$ is calculated as

$$\mathcal{L} = \mathcal{L}_m + \lambda \mathcal{L}_{\text{scl}} \tag{14}$$

where λ is a weight factor to trade off the structure-guided supervised contrastive and classification losses, and ranges from 0 to 1. In this way, we can simultaneously learn the intraclass variances and interclass commonness to enhance the classification ability of the model.

IV. Experiments

In this section, we introduce the experimental datasets and implementation details. Five experiments were conducted to demonstrate and analyze the effectiveness of CSTS.

A. Datasets

We conducted few-shot classification experiments on three popular benchmark datasets with hierarchical structures: *MiniImageNet* [2], *tieredImageNet* [38], and *CIFAR-FS* [39]. *tieredImageNet* and *CIFAR-FS* both have their own hierarchical structures, while we utilize semantic clustering to construct a class hierarchy for *MiniImageNet*. In addition, we add a fine-grained dataset of 200 bird species, *CUB200* [40], to verify the generalization ability of our model. Table II provides the basic statistics of the four datasets and lists their numbers of coarse- and fine-grained classes.

TABLE II
Dataset Description

Datasets	Train		Valid		Test	
	Coarse	Fine	Coarse	Fine	Coarse	Fine
tieredImageNet	20	351	6	97	8	160
MiniImageNet	10	64	5	16	5	20
CIFAR-FS	20	64	11	16	13	20
CUB200	-	100	-	50	-	50

B. Implementation Details

Our experiments focus on demonstrating the effectiveness of each component of CSTS and evaluating it in comparison with several state-of-the-art FSL models. For a fair comparison, we adopted 4CONV as the feature extractor f_ϕ, following in [16], [23], [25], and [47]. The last feature is input into a fully connected layer to obtain a 128-D feature representation. We utilize the Adam optimizer to train our model with an initial learning rate of 10^{-3} and a weight decay of 10^{-6}. Our experiments were implemented in Pytorch with a GeForce RTX 2080 Ti Nvidia GPU card.

For the experimental settings, we used 3/4 graph levels to transfer the information in a one-shot/five-shot setting, with a hierarchical inference strategy of $5\to3\to1/15\to5\to3\to1$ on all datasets. We initially set the temperature parameter $\tau = 25.5$ °C. The other parameter values used on the different datasets are shown in Section IV-D. For the test phase, we conducted a few-shot classification on 10 000 random episode samples from the test set and computed the mean accuracy with 95% confidence interval.

C. Comparisons

In this section, some accuracy comparisons are made between CSTS and several state-of-the-art FSL models on the *MiniImageNet*, *tieredImageNet*, *CIFAR-FS*, and *CUB200* datasets. The main results are listed in Tables III and IV. In these tables, the FSL models are divided into two categories according to whether they use a class structure; * denotes that the model is based on graph learning, and the best results are marked in bold. Compared with other FSL models, we can obtain the following conclusions.

1) CSTS achieves very competitive and effective performance compared to most FSL models without using the semantic knowledge of the four datasets. Taking *MiniImageNet* as an example, CSTS obtains about 2.40% and 3.40% better performance than the baseline EGNN. This confirms that exploring class structure greatly benefits FLS.
2) We observe a notable gain in performance when adopting the class structure, which highlights the benefits of using the semantic relation when conducting SCL. In the semantic structure application, CSTS is comparable to most other models. Models like SKD map the semantic relation as internal semantic embedding to the feature space. We utilize the semantic structure as external auxiliary information to guide feature contrastive learning,

TABLE III
ACCURACY COMPARISON WITH DIFFERENT MODELS ON *MiniImageNet* AND *tieredImageNet*

Model	Class Structure	*MiniImageNet*		*tieredImageNet*	
		1-shot	5-shot	1-shot	5-shot
MatchNet [2]	N	43.56 ± 0.84	55.31 ± 0.73	54.02 ± 0.00	70.11 ± 0.00
ProtoNet [22]	N	53.31 ± 0.89	72.69 ± 0.74	53.31 ± 0.89	72.69 ± 0.74
RelationNet [8]	N	50.40 ± 0.80	65.30 ± 0.70	54.48 ± 0.93	71.31 ± 0.70
FEAT [41]	N	55.15 ± 0.20	71.61 ± 0.16	-	-
EPNet [42]*	N	59.32 ± 0.88	72.95 ± 0.64	59.97 ± 0.95	73.91 ± 0.75
CPLAE [30]	N	56.83 ± 0.44	74.31 ± 0.34	58.23 ± 0.49	75.12 ± 0.40
LDGP [43]	N	56.32 ± 0.28	72.64 ± 0.26	58.43 ± 0.38	76.17 ± 0.34
HyperShot [44]	N	53.18 ± 0.45	69.62 ± 0.20	-	-
GNN [23]*	N	54.72 ± 0.70	75.41 ± 0.47	64.56 ± 0.87	81.89 ± 0.47
TPN [11]*	N	57.53 ± 0.96	72.85 ± 0.74	57.53 ± 0.96	72.85 ± 0.74
EGNN [25]*	N	58.98 ± 0.36	76.37 ± 0.30	63.25 ± 0.42	80.15 ± 0.30
SPN [45]*	N	55.36 ± 0.70	71.23 ± 0.60	-	-
HMRN [18]	Y	-	-	57.98 ± 0.26	74.70 ± 0.24
VSM [46]*	Y	54.73 ± 1.60	68.01 ± 0.90	56.88 ± 1.71	74.65 ± 0.81
SKD [47]	Y	58.07 ± 0.27	68.03 ± 0.44	-	-
HGNN [16]*	Y	60.84 ± 0.51	78.78 ± 0.47	63.39 ± 0.51	**83.17 ± 0.44**
HTS [48]	Y	58.96 ± 0.18	75.17 ± 0.14	53.20 ± 0.22	72.38 ± 0.19
CSTS (ours)*	Y	**62.38 ± 0.48**	**79.77 ± 0.44**	**64.84 ± 0.26**	82.95 ± 0.44

TABLE IV
ACCURACY COMPARISON WITH DIFFERENT MODELS ON *CIFAR-FS* AND *CUB200*

Model	Class Structure	*CIFAR-FS*		*CUB200*		*MiniImageNet→CUB200*	
		1-shot	5-shot	1-shot	5-shot	1-shot	5-shot
MAML [6]	N	55.50 ± 0.70	72.00 ± 0.60	56.11 ± 0.69	74.84 ± 0.62	34.01 ± 1.25	48.83 ± 0.62
ProtoNet [22]	N	55.50 ± 0.70	72.00 ± 0.60	52.52 ± 1.90	75.93 ± 0.46	33.27 ± 1.09	52.16 ± 0.17
RelationNet [8]	N	55.00 ± 1.00	69.30 ± 0.80	-	-	37.13 ± 0.20	51.76 ± 1.48
R2D2 [39]	N	60.20 ± 1.80	70.91 ± 0.91	-	-	-	-
MetaQDA [49]	N	60.52 ± 0.88	77.33 ± 0.73	-	-	47.25 ± 0.58	64.40 ± 0.65
Feature Transfer [50]	N	-	-	46.19 ± 0.64	68.40 ± 0.79	32.77 ± 0.35	50.34 ± 0.27
Meta-Baseline [51]	N	-	-	54.29 ± 0.96	72.65 ± 0.72	-	-
HGNN [16]*	Y	60.44 ± 0.51	81.05 ± 0.44	-	-	47.13 ± 0.50	**73.21 ± 0.52**
CSTS (ours)*	Y	**62.47 ± 0.47**	**81.12 ± 0.42**	**60.83 ± 0.45**	**77.12 ± 0.44**	**48.94 ± 0.47**	71.80 ± 0.51

which can obtain class-specific features and expand the interclass distance in the feature space.

3) Compared with most graph-learning-based models, CSTS performs better. For one-shot performance, CSTS outperforms HGNN and EGHH, which leverage the same GCN operations. It demonstrates the guiding effect of the semantic relation on few-shot classification. While CSTS is slightly inferior to HGNN in the five-shot scene on *tieredImageNet*, we conjecture that this is because HGNN adopts a skip connections strategy despite CSTS utilizing similar hierarchical reasoning.
4) The improvements obtained by CSTS for the one-shot task are more significant than those for a five-shot task on the four datasets. That proves that CSTS is effective when training data are extremely scarce. There is little available and useful information for classification of a scene with one or very few samples.

In addition, we used the cross-domain experiments to test the generalization ability of our model, in which the model was evaluated on tasks coming from different datasets than the one it had been trained on. Specifically, the results in Tables IV and V show that CSTS has some generalization ability and can be applied to the classification of new datasets.

TABLE V
CROSS-DOMAIN TEST OF FEW-SHOT CLASSIFICATION IN ONE-SHOT SETTING

Training→Test	HGNN	CSTS
tieredImageNet→CUB200	39.87 ± 0.46	**40.68 ± 0.45**
CIFAR-FS→CUB200	42.66 ± 0.47	**46.24 ± 0.48**
MiniImageNet→tieredImageNet	43.66 ± 0.46	**46.48 ± 0.48**
CIFAR-FS→tieredImageNet	37.46 ± 0.45	**40.30 ± 0.46**
tieredImageNet→MiniImageNet	**54.75 ± 0.53**	54.50 ± 0.50
CIFAR-FS→MiniImageNet	40.35 ± 0.48	**44.55 ± 0.48**
tieredImageNet→CIFAR-FS	39.47 ± 0.44	**45.04 ± 0.48**
MiniImageNet→CIFAR-FS	39.93 ± 0.43	**42.58 ± 0.46**

D. Parameter Analysis

We conducted experiments on *MiniImageNet* and *tieredImageNet* to illustrate the impacts of using different parameter settings in our proposed model. First, three parameters

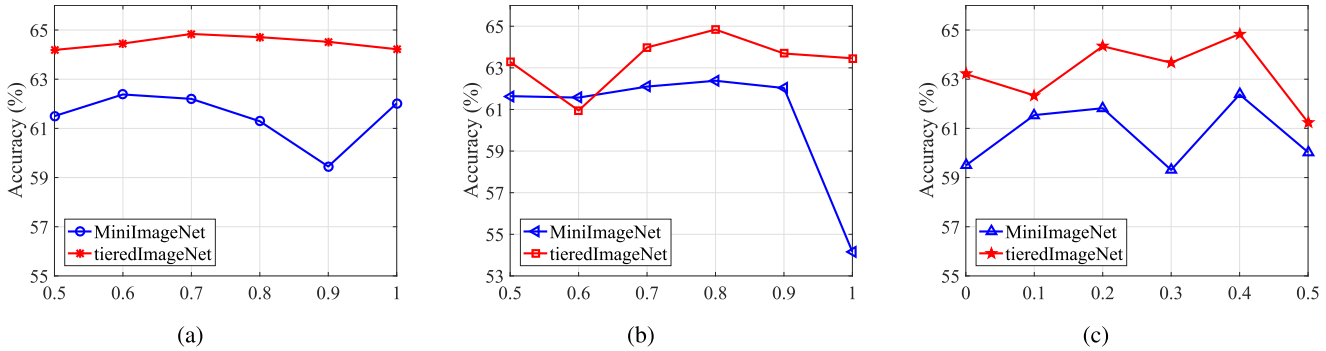

Fig. 5. Sensitivity analysis for the parameters of CSTS in the one-shot task. (a) Parameter α. (b) Parameter η. (c) Parameter λ.

TABLE VI
ACCURACY OF DIFFERENT MULTILEVEL FUSION STRATEGIES IN ONE-SHOT SETTING

Multi-level fusion	*MiniImageNet*	*tieredImageNet*
0.3-0.3-0.4	61.76 ± 0.48	63.83 ± 0.49
0.2-0.4-0.4	61.97 ± 0.49	64.19 ± 0.48
0.1-0.4-0.5	**62.38 ± 0.48**	64.52 ± 0.50
0.2-0.3-0.5	62.02 ± 0.48	**64.84 ± 0.26**
0.1-0.3-0.6	61.71 ± 0.49	63.88 ± 0.48

TABLE VII
ACCURACY OF DIFFERENT MULTILEVEL FUSION STRATEGIES IN FIVE-SHOT SETTING

Multi-level fusion	*MiniImageNet*	*tieredImageNet*
0.1-0.1-0.3-0.5	79.55 ± 0.44	**82.95 ± 0.44**
0.1-0.1-0.2-0.6	**79.77 ± 0.44**	82.66 ± 0.41
0.0-0.1-0.4-0.5	78.83 ± 0.44	82.55 ± 0.43
0.1-0.2-0.3-0.4	79.09 ± 0.44	82.14 ± 0.42

(i.e., α, η, and λ) were analyzed in the five-way one-shot experimental setting, with the main results shown in Fig. 5. We can obtain the following.

1) Parameter α represents the attention of the model of sibling contrastive loss, which aims to balance the learning of the sibling and nonsibling contrastive losses. As shown in Fig. 5(a), our model achieves the best performance when $\alpha = 0.6$ on *MiniImageNet* and $\alpha = 0.7$ on *tieredImageNet*.
2) The value of η controls the influence on the feature of the following level. Fig. 5(b) shows that the user-selected value of η greatly affects the performance of our proposed model. We set $\eta = 0.8$ on all datasets.
3) Parameter λ trades off the structure-guided contrastive and classification studies, which focus on simultaneously learning the intraclass variances and the interclass commonness by exploiting the class-specific and class-shared embedding. Fig. 5(c) shows that the choice of λ greatly influences our model; we set $\lambda = 0.4$ on all datasets in our experiments.

Next, we explore the effects of different prediction strategies on our model. We fuse multilevel instead of single-level prediction probabilities, weighting the fusion (μ_ℓ) between different levels. The values of μ_ℓ are the weights of the ℓth level, where $\ell = 0, \ldots, 2$ in a one-shot task, while $\ell = 0, \ldots, 3$ in a five-shot task. We analyze the influence of μ_ℓ on the performance of our models, with the main results listed in Tables VI and VII. We can see that the weight of the higher level is greater than that of the lower level. The information learned at the high level has a greater impact on model performance than that at the low level.

TABLE VIII
ACCURACY COMPARISON WITH DIFFERENT GRAPH LEVELS AND INFERENCE STRATEGIES. NOTE THAT ONE-CSTS MEANS CSTS WITH ONE GRAPH LEVEL; $n \to m$ MEANS SELECTING THE MOST IMPORTANT m FROM n SAMPLES, AND $n = m$ REPRESENTS NO SELECTION

Model	*Inference*	*MiniImageNet*		*CUB200*	
		1-shot	5-shot	1-shot	5-shot
1-CSTS	**5**	54.72±0.70	75.41±0.47	59.63±0.47	73.34±0.42
2-CSTS	**5→5**	60.96±0.47	68.84±0.40	60.41±0.46	69.03±0.47
	5→1	**61.67±0.49**	**79.45±0.45**	**60.88±0.46**	**74.28±0.42**
3-CSTS	**5→5→5**	61.79±0.47	79.15±0.45	59.48±0.48	76.12±0.45
	5→3→1	**62.38±0.48**	**79.77±0.44**	**60.83±0.45**	**77.12±0.44**

E. Analysis of HFG Learning

This section verifies the performance of HFG. In HFG, we use a hierarchical inference strategy to infer the task-shared information between the target sample and all the samples in the task, from coarse to fine, to yield a more general representation. Therefore, we tested the effectiveness of CSTS with different graph levels and inference strategies. The results are reported in Table VIII. We can found that:

1) Having more graph levels gives CSTS a more obvious effect. By increasing the number of graph levels, we can transfer more task-shared information to enhance the feature representation of the target sample based on more samples in a task.
2) Compared with horizontal inference, hierarchical inference conducts more information transmission, which retains important features while removing useless information. This is conducive to merging task-shared information, which enhances the feature representation

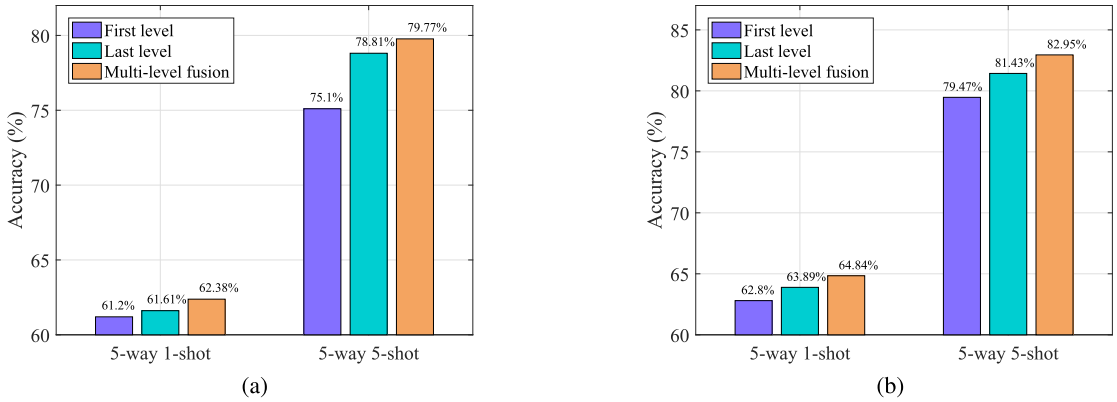

Fig. 6. Accuracy comparison between multilevel and single-level strategies. The first level (baseline) means the model classification without a hierarchical inference strategy; the last level represents that model adopts the hierarchical inference but does not fuse multilevel probabilities; and the multilevel means that the model uses the hierarchical inference with multilevel fusion. (a) MiniImageNet. (b) tieredImageNet.

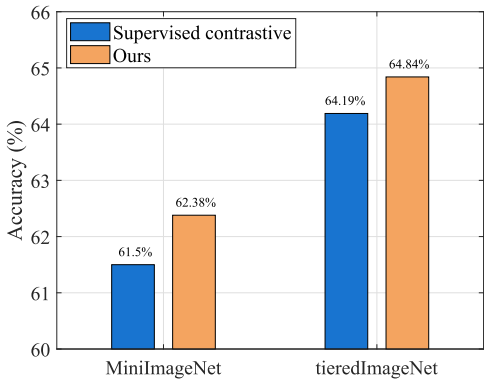

Fig. 7. Accuracy comparison with supervised contrastive learning.

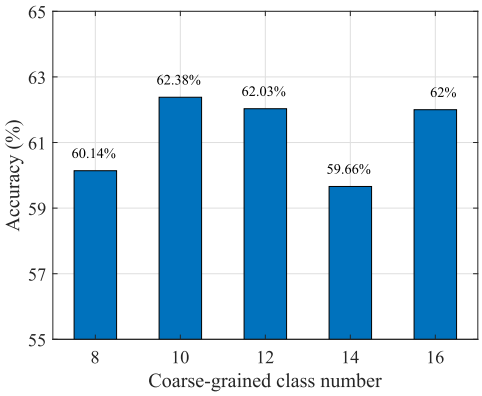

Fig. 8. Performance of different class semantic hierarchy structures of the training set on *MiniImageNet*.

of query samples while boosting the generalizability of the model.

Then, we compared multilevel fusion and single-level strategies to demonstrate the effectiveness of multilevel fusion. Fig. 6 compares the accuracy obtained using multilevel and single-level strategies on *MiniImageNet* and *tieredImageNet*. We can conclude that:

1) The accuracy of the last level is superior to that of the first level (baseline), demonstrating the effectiveness of adopting a hierarchical inference strategy. For instance, the accuracy of the last level was by 1.40% and 3.70% better on *MiniImageNet* in one-shot and five-shot tasks, respectively.
2) Based on the hierarchical inference, multilevel fusion performs better than using a single-level, since it can take advantage of the more abundant multilevel information. We leverage hierarchical learning to exploit the task-shared commonness to enhance the query features by reducing less important nodes level by level. Additionally, we take a multilevel fusion strategy to combine multilevel probabilities to assist classification of the last level, since we consider that the reduced nodes in low levels may contain useful information for further exploitation. The experimental results also highlight the benefit of low-level information for high-level classification.

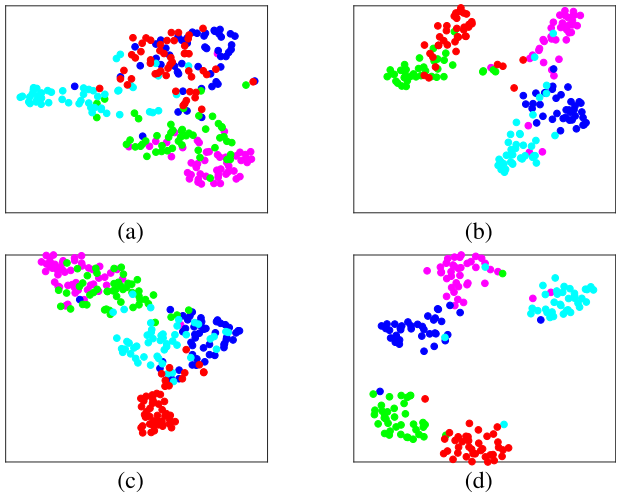

Fig. 9. t-SNE visualization under five-way one-shot and five-shot episodes on *tiredImageNet*. Different colors represent different classes. (a) Baseline-one-shot. (b) Baseline-five-shot. (c) CSTS-one-shot. (d) CSTS-five-shot.

F. Analysis of SCL

In this section, we validate the performance of the proposed structure-guided supervised contrastive learning approach. Based on supervised contrastive learning, we embed semantic knowledge to guide feature contrastive learning by the sibling strategy. This aims to increase the discrimination of similar

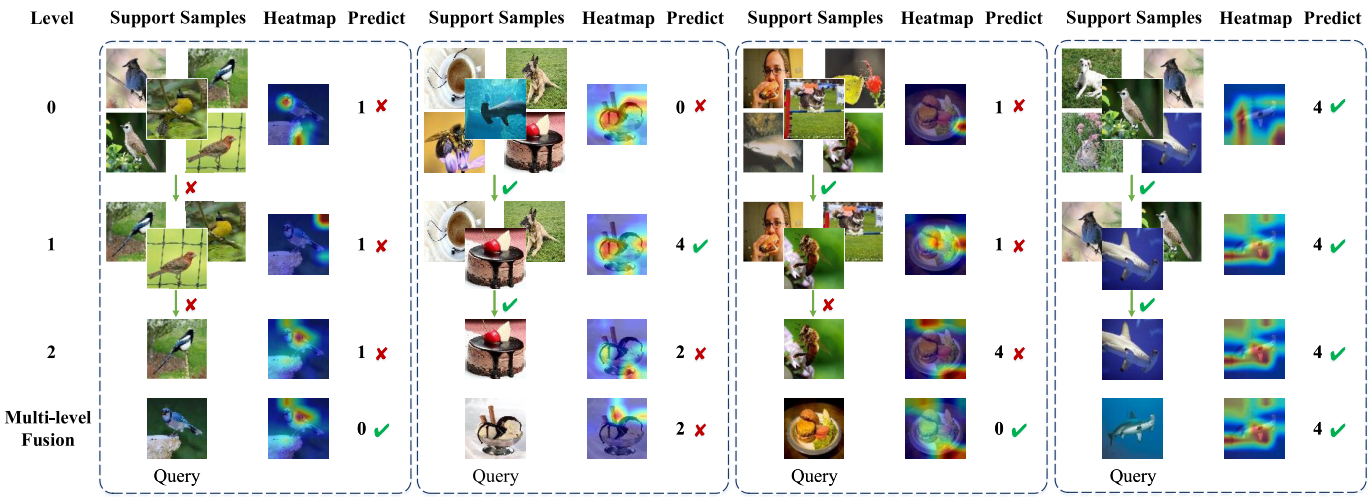

Fig. 10. Prediction interpretability and heatmap analysis of CSTS. Numbers 0–4 represent five classes, and Levels 0–2 mean different graph levels.

TABLE IX
CONTRIBUTIONS OF DIFFERENT COMPONENTS IN CSTS ON DIFFERENT DATASETS. THE BEST RESULTS ARE MARKED IN BOLD

SCL	HG	MFP	*MiniImageNet*		*tieredImageNet*	
			1-shot	5-shot	1-shot	5-shot
			57.75±0.48	74.96±0.44	61.98±0.48	76.74±0.42
✔			61.14±0.49	75.10±0.42	62.80±0.47	79.47±0.42
	✔		57.81±0.50	76.74±0.46	62.14±0.50	78.35±0.43
	✔	✔	59.49±0.50	78.42± 0.42	63.65±0.49	82.16±0.42
✔	✔		61.61±0.50	78.81±0.45	63.89±0.51	81.43±0.42
✔	✔	✔	**62.38±0.48**	**79.77±0.44**	**64.84±0.26**	**82.95±0.44**

classes because the cost of distinguishing similar sibling classes is greater than that of nonsibling classes. Fig. 7 shows that the performance of our model is better than with supervised contrastive learning. This indicates that the sibling strategy is beneficial for guiding feature contrastive learning.

As the class semantic structure is the key to structure-guided supervised contrastive learning, we conducted experiments to evaluate the influence of the class semantic structure on our model. Fig. 8 shows the accuracy obtained using different class hierarchy structures on a *MiniImageNet* training set constructed with different coarse-grained class numbers. We can see that the constructed semantic hierarchy structure greatly affects the performance of the model. Our model performs best when the training set has ten coarse-grained classes.

G. Ablation Study and Visualization

In this section, we describe ablation studies conducted to validate each component of CSTS, including SCL and HFG learning. HFG learning consists of HG learning and MFP. Table IX lists the experimental results of the different components in our model on different datasets of five-way one-shot and five-shot settings.

From Table IX, we can conclude that:

1) Each component of CSTS plays a role in improving model performance. HG is essential for CSTS, as it conducts HG learning to transform the intraclass commonness, which lays the foundation for our model. MFP merges the multilevel rich information for prediction. In contrast, SCL aims to leverage class structure knowledge to guide the learning of intraclass variance.
2) With the three components together, CSTS achieves 62.38 ± 0.48/79.77 ± 0.44 and 64.84 ± 0.26/82.95 ± 0.44 under the one-shot/five-shot episodes on the two datasets, outperforming the baseline by about 4.60/4.80 and 2.85/6.20. By integrating all components into the FSL framework, CSTS obtains greater improvements than those in any other case. The consistent improvements across the two datasets empirically validate our claim that joint learning of class-specific variance and task-shared commonness benefits FSL tasks from the class- and task level.

Furthermore, we used visualization technology to specifically and intuitively demonstrate the effectiveness of CSTS. First, we visualized the feature representations of several samples to explain the ability to capture intraclass variance and interclass commonness in the feature space. We sampled 50/30 query samples per class under the five-way one-shot/five-shot settings and used the t-SNE to visualize the feature embedding of CSTS. Fig. 9 shows that the features extracted by baseline do not exhibit clear class boundaries, and the intraclass distribution is sparse. Meanwhile, our model makes intraclass features more tightly aggregated and interclass features more sparsely distributed, thus achieving better classification performance. This indicates that CSTS is conducive to reducing the intraclass distance and expanding the interclass spacing by simultaneously learning the class-specific variance and task-shared commonness.

In addition, to show the prediction interpretability of our CSTS model, we conducted several five-way one-shot learning tasks on *tieredImageNet* to illustrate the interpretable classification and hierarchical inference processes, including right and wrong prediction cases, which are shown in Fig. 10. We can find that:

1) CSTS can provide a clear prediction sequence at different graph levels and a detailed visual analysis for sample inference at each graph level.

2) The interpretable prediction process can help us analyze and locate the root causes of this erroneous classification.

Although the hierarchical inference is wrong in the first and third cases, the final classification, which benefits from the multilevel fusion strategy, is correct. The heatmap can accurately capture the characteristics of query samples. On the contrary, in the second case, the occurrence of classification errors in the final output are attributed to inference errors at multiple layers, which further hinders the accurate localization of key characteristics by the heatmap.

V. Conclusion and Future Work

In this article, we propose a hierarchical few-shot model to settle the problem of limited model generalizability due to the difficulty in extracting general information from only one or a very few training examples (CSTS). It simultaneously exploits CSTS embedding from the class and task levels. From the perspective of the class level, a structure-guided contrastive loss was introduced to obtain class-specific representations, which is guided by the rich semantic knowledge of the hierarchical class structure. From the task-level perspective, we proposed a HFG neural network to hierarchically infer the sample relations in tasks and enhance query features by passing the task-shared information. To sum up, to obtain a better representation, we explore and take advantage of the class-level and task-level information and improve the generalizability of the model. Experimental results indicate the advantages of CSTS over several state-of-the-art models on four benchmark datasets.

Several limitations in CSTS remain, which will be the focus of future study. Adjusting the parameter values for few-shot tasks in different datasets is time-consuming. Accordingly, we aim to design an adaptive strategy for choosing optimal parameters. In addition, CSTS utilizes the semantic knowledge of class structure to guide feature representation learning. However, an semantic gap problem exists between the images and their semantic labels as excessively depending on the semantic information. For example, *Shark* and *Dolphin* have discriminative semantic label descriptions but represent similar visual concepts. Thus, the semantic structure is detrimental to learning a distinctive feature representation of *Shark* and *Dolphin*. In the future, we will focus on solving the semantic gap problem to promote the models generalization ability.

References

[1] D. Kang, H. Kwon, J. Min, and M. Cho, "Relational embedding for few-shot classification," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 8822–8833.

[2] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, and D. Wierstra, "Matching networks for one shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3637–3645.

[3] S. Baik, M. Choi, J. Choi, H. Kim, and K. M. Lee, "Learning to learn task-adaptive hyperparameters for few-shot learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 3, pp. 1441–1454, Mar. 2024.

[4] A. Zhao, G. Balakrishnan, F. Durand, J. Guttag, and A. Dalca, "Data augmentation using learned transformations for one-shot medical image segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 8543–8553.

[5] B. Yan et al., "Augmented bi-path network for few-shot learning," in *Proc. 25th Int. Conf. Pattern Recognit.*, 2021, pp. 8461–8468.

[6] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. Int. Conf. Mach. Learn.*, 2017, pp. 1126–1135.

[7] Q. Sun, Y. Liu, T.-S. Chua, and B. Schiele, "Meta-transfer learning for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 403–412.

[8] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. S. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 1199–1208.

[9] C. Zhang, Y. Cai, G. Lin, and C. Shen, "DeepEMD: Few-shot image classification with differentiable earth mover's distance and structured classifiers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2020, pp. 12203–12213.

[10] Z. Li, Z. Hu, W. Luo, and X. Hu, "SaberNet: Self-attention based effective relation network for few-shot learning," *Pattern Recognit.*, vol. 133, Jan. 2023, Art. no. 109024.

[11] Y. Liu et al., "Learning to propagate labels: Transductive propagation network for few-shot learning," in *Proc. Int. Conf. Learn. Represent.*, 2019, pp. 1–14.

[12] S. Li, D. Chen, B. Liu, N. Yu, and R. Zhao, "Memory-based neighbourhood embedding for visual recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2019, pp. 6101–6110.

[13] J. He, R. Hong, X. Liu, M. Xu, Z. Zha, and M. Wang, "Memory-augmented relation network for few-shot learning," in *Proc. ACM Int. Conf. Multimedia*, 2020, pp. 1236–1244.

[14] L. Yang, L. Li, Z. Zhang, X. Zhou, E. Zhou, and Y. Liu, "DPGN: Distribution propagation graph network for few-shot learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 13390–13399.

[15] R. Chen, T. Chen, X. Hui, H. Wu, G. Li, and L. Lin, "Knowledge graph transfer network for few-shot recognition," in *Proc. AAAI Conf. Artif. Intell.*, 2020, vol. 34, no. 7, pp. 10575–10582.

[16] C. Chen, K. Li, W. Wei, J. T. Zhou, and Z. Zeng, "Hierarchical graph neural networks for few-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 1, pp. 240–252, Jan. 2021.

[17] P. Khosla et al., "Supervised contrastive learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 18661–18673.

[18] Y. Su, H. Zhao, and Y. Lin, "Few-shot learning based on hierarchical classification via multi-granularity relation networks," *Int. J. Approx. Reasoning*, vol. 142, pp. 417–429, Mar. 2022.

[19] K. Lee, S. Maji, A. Ravichandran, and S. Soatto, "Meta-learning with differentiable convex optimization," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 10657–10665.

[20] M. Rolinek, P. Swoboda, D. Zietlow, A. Paulus, V. Musil, and G. Martius, "Deep graph matching via blackbox differentiation of combinatorial solvers," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 407–424.

[21] B. Hui, P. Zhu, Q. Hu, and Q. Wang, "Self-attention relation network for few-shot learning," in *Proc. IEEE Int. Conf. Multimedia Expo Workshops (ICMEW)*, Jul. 2019, pp. 198–203.

[22] J. Snell, K. Swersky, and R. Zemel, "Prototypical networks for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 4080–4090.

[23] V. Garcia and J. Bruna, "Few-shot learning with graph neural networks," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–13.

[24] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and S. Y. Philip, "A comprehensive survey on graph neural networks," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 32, no. 1, pp. 4–24, Mar. 2020.

[25] J. Kim, T. Kim, S. Kim, and C. Yoo, "Edge-labeling graph neural network for few-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 11–20.

[26] A. Li, T. Luo, Z. Lu, T. Xiang, and L. Wang, "Large-scale few-shot learning: Knowledge transfer with class hierarchy," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 7205–7213.

[27] A. Li, Z. Lu, J. Guan, T. Xiang, L. Wang, and J.-R. Wen, "Transferrable feature and projection learning with class hierarchy for zero-shot learning," *Int. J. Comput. Vis.*, vol. 128, no. 12, pp. 2810–2827, Dec. 2020.

[28] C. Chen, X. Yang, C. Xu, X. Huang, and Z. Ma, "ECKPN: Explicit class knowledge propagation network for transductive few-shot learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6596–6605.

[29] Y. Tian, D. Krishnan, and P. Isola, "Contrastive multiview coding," in *Proc. Eur. Conf. Comput. Vis.*, 2020, pp. 776–794.

[30] Y. Gao, N. Fei, G. Liu, Z. Lu, and T. Xiang, "Contrastive prototype learning with augmented embeddings for few-shot learning," in *Proc. 37th Conf. Uncertainty Artif. Intell.*, 2021, pp. 140–150.

[31] Y. Ouali, C. Hudelot, and M. Tami, "Spatial contrastive learning for few-shot classification," in *Proc. Joint Eur. Conf. Mach. Learn. Knowl. Discovery Databases*, 2021, pp. 671–686.
[32] C. Liu et al., "Learning a few-shot embedding model with contrastive learning," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 10, pp. 8635–8643.
[33] H. Kwon, S. Jeong, S. Kim, and K. Sohn, "Dual prototypical contrastive learning for few-shot semantic segmentation," 2021, *arXiv:2111.04982*.
[34] H. Zhao, Q. Hu, P. Zhu, Y. Wang, and P. Wang, "A recursive regularization based feature selection framework for hierarchical classification," *IEEE Trans. Knowl. Data Eng.*, vol. 33, no. 7, pp. 2833–2846, Jul. 2021.
[35] Y. Wang et al., "Coarse-to-fine: Progressive knowledge transfer-based multitask convolutional neural network for intelligent large-scale fault diagnosis," *IEEE Trans. Neural Netw. Learn. Syst.*, vol. 34, no. 2, pp. 761–774, Feb. 2023.
[36] G. A. Miller, "WordNet: A lexical database for English," *Commun. ACM*, vol. 38, no. 11, pp. 39–41, Nov. 1995.
[37] K. He, H. Fan, Y. Wu, S. Xie, and R. Girshick, "Momentum contrast for unsupervised visual representation learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 9729–9738.
[38] M. Ren et al., "Meta-learning for semi-supervised few-shot classification," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–15.
[39] L. Bertinetto, J. F. Henriques, P. Torr, and A. Vedaldi, "Meta-learning with differentiable closed-form solvers," in *Proc. Int. Conf. Learn. Represent.*, 2018, pp. 1–15.
[40] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200–2011 dataset," California Inst. Technol., Tech. Rep., 2011.
[41] H. Ye, H. Hu, D. Zhan, and F. Sha, "Few-shot learning via embedding adaptation with set-to-set functions," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 8808–8817.
[42] P. Rodríguez, I. Laradji, A. Drouin, and A. Lacoste, "Embedding propagation: Smoother manifold for few-shot classification," in *Proc. Eur. Conf. Comput. Vis.*, Aug. 2020, pp. 121–138.
[43] Z. Wang, Z. Miao, X. Zhen, and Q. Qiu, "Learning to learn dense Gaussian processes for few-shot learning," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 34, 2021, pp. 13230–13241.
[44] M. Sendera, M. Przewięźlikowski, K. Karanowski, M. Zieba, J. Tabor, and P. Spurek, "HyperShot: Few-shot learning by kernel hypernetworks," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis.*, Jan. 2023, pp. 2469–2478.
[45] P. Koniusz and H. Zhang, "Power normalizations in fine-grained image, few-shot image and graph classification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 2, pp. 591–609, Feb. 2022.
[46] X. Zhen, Y. Du, H. Xiong, Q. Qiu, C. Snoek, and L. Shao, "Learning to learn variational semantic memory," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 33, 2020, pp. 9122–9134.
[47] A. Cheraghian, S. Rahman, P. Fang, S. K. Roy, L. Petersson, and M. Harandi, "Semantic-aware knowledge distillation for few-shot class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2534–2543.
[48] M. Zhang, S. Huang, W. Li, and D. Wang, "Tree structure-aware few-shot image classification via hierarchical aggregation," in *Proc. Eur. Conf. Comput. Vis.*, 2022, pp. 453–470.
[49] X. Zhang, D. Meng, H. Gouk, and T. Hospedales, "Shallow Bayesian meta learning for real-world few-shot recognition," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 631–640.
[50] F. Zhuang et al., "A comprehensive survey on transfer learning," *Proc. IEEE*, vol. 109, no. 1, pp. 43–76, Jan. 2021.
[51] Y. Chen, Z. Liu, H. Xu, T. Darrell, and X. Wang, "Meta-baseline: Exploring simple meta-learning for few-shot learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. (ICCV)*, Oct. 2021, pp. 9062–9071.

Hong Zhao (Member, IEEE) received the Ph.D. degree from Tianjin University, Tianjin, China, in 2019, and the M.S. degree from Liaoning Normal University, Dalian, China, in 2006.

She is a Professor with the School of Computer Science and the Key Laboratory of Data Science and Intelligence Application, Minnan Normal University, Zhangzhou, China. She has authored or coauthored over 60 articles in international journals such as IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE TRANSACTIONS ON KNOWLEDGE AND DATA ENGINEERING, and IEEE TRANSACTIONS ON SIGNAL PROCESSING. Her research interests include rough sets, granular computing, and data mining for hierarchical classification.

Yuling Su is currently pursuing the M.E. degree with the School of Computer Science, Minnan Normal University, Zhangzhou, Fujian, China.

Her research interests include few-shot learning and hierarchical classification application in machine learning and granular computing.

Zhiping Wu is currently pursuing the M.E. degree from the School of Computer Science, Minnan Normal University, Zhangzhou, China.

He is an Intern with the Department of Computer Science and Technology, Nanjing University, Nanjing, China. His research interests include granular computing, machine learning, and few-shot learning for hierarchical classification.

Weiping Ding (Senior Member, IEEE) received the Ph.D. degree in computer science, Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2013.

From 2014 to 2015, he was a Post-Doctoral Researcher with the Brain Research Center, National Chiao Tung University, Hsinchu, Taiwan. In 2016, he was a Visiting Scholar with the National University of Singapore, Singapore. From 2017 to 2018, he was a Visiting Professor with the University of Technology Sydney, Ultimo, NSW, Australia. He is currently a Professor with the School of Information Science and Technology, Nantong University, Nantong, China, and also the Supervisor of Ph.D. Postgraduate with the Faculty of Data Science, City University of Macau, Macau, China. He has authored or coauthored over 200 articles in international journals, including over 80 IEEE journal articles. His 15 authored/coauthored papers have been selected as ESI Highly Cited Papers. He has coauthored four books. He has holds 27 approved invention patents, including one U.S. patent and one Australian patent. His research interests include deep neural networks, multimodal machine learning, granular data mining, and medical images analysis.

Dr. Ding served/serves as the Editorial Board Member of Knowledge-Based Systems, Engineering Applications of Artificial Intelligence, Applied Soft Computing, and the Area Editor of Information Fusion. He serves as the Associate Editor for IEEE TRANSACTIONS ON NEURAL NETWORKS AND LEARNING SYSTEMS, IEEE TRANSACTIONS ON FUZZY SYSTEMS, IEEE/CAA JOURNAL OF AUTOMATICA SINICA, IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS, IEEE TRANSACTIONS ON EMERGING TOPICS IN COMPUTATIONAL INTELLIGENCE, IEEE TRANSACTIONS ON INTELLIGENT VEHICLES, and IEEE TRANSACTIONS ON ARTIFICIAL INTELLIGENCE, INFORMATION SCIENCES, NEUROCOMPUTING. He is a Leading Guest Editor of Special Issues in several prestigious journals, including IEEE TRANSACTIONS ON EVOLUTIONARY COMPUTATION, IEEE TRANSACTIONS ON FUZZY SYSTEMS, and *Information Fusion*.