Learn to abstract via concept graph for weakly-supervised few-shot learning

Baoquan Zhang, Ka-Cheong Leung, Xutao Li, Yunming Ye*

School of Computer Science and Technology, Harbin Institute of Technology, Shenzhen, Shenzhen 518055, Guangdong, China

ARTICLE INFO

Article history:
Received 6 August 2020
Revised 16 December 2020
Accepted 5 March 2021
Available online 22 April 2021

Keywords:
Few-shot learning
Weakly-supervised learning
Meta-learning
Concept graph

ABSTRACT

In recent years, a large number of meta-learning methods have been proposed to address few-shot learning problems and have shown superior performance. However, the explicit prior knowledge (e.g., concept graph) and weakly-supervised information are rarely explored in existing methods, which are usually free or cheap to collect. In this paper, we introduce a concept graph for the weakly-supervised few-shot learning, and propose a novel meta-learning framework, namely, MetaConcept. Our key idea is to learn a universal meta-learner inferring any-level classifier, so as to boost the classification performance of meta-learning on the novel classes. Specifically, we firstly propose a novel regularization with multi-level conceptual abstraction to train a universal meta-learner to infer not only an entity classifier but also a concept classifier at different levels via the concept graph (i.e., learn to abstract). Then, we propose a meta concept inference network as the universal meta-learner for the base learner, aiming to quickly adapt to a novel task by the joint inference of the abstract concepts and a few annotated samples. We have conducted extensive experiments on two weakly-supervised few-shot learning benchmarks, namely, WS-ImageNet-Pure and WS-ImageNet-Mix. Our experimental results show that (1) the proposed MetaConcept outperforms state-of-the-art methods with an improvement of 2% to 6% in classification accuracy; (2) the proposed MetaConcept is able to yield a good performance though merely training with weakly-labeled datasets.

© 2021 Elsevier Ltd. All rights reserved.

1. Introduction

Few-Shot Learning (FSL) is a machine learning approach for understanding new concepts with a few examples. It targets at acquiring good learning performance by leveraging the prior knowledge for a novel task where its class is unfamiliar and only a little supervised information is available [1]. The study of FSL has received much attention recently because of the following features: (1) FSL is a cheap learning paradigm, which can reduce the costs of data annotations for many data-dependent applications, such as image classification [2], object detection [3], and neural architecture search [4]. (2) FSL can be directly applied to rare case learning applications, where the acquisition of annotated samples is hard or impossible due to scarcity or safety concerns, such as cold-start item recommendation [5] and drug discovery [6].

At present, most methods primarily focus on meta-learning frameworks to solve the FSL problems [2,7]. They aim to learn a base learner based on meta-knowledge from past experience so as to quickly adapt to novel tasks by just a few annotated samples. Specifically, the framework consists of two major phases: (1) learning meta-knowledge from base tasks sampled from a base class set (called meta-training phase); and (2) quickly constructing or fine-tuning a base learner by employing the learned meta-knowledge and a few annotated samples so as to obtain a task-specific model for novel tasks sampled from a novel class set (called meta-test phase). Here, the base class set and novel class set are disjoint. Although these methods have shown the superior performance on solving the FSL problem, they just focus on learning implicit meta-knowledge on finely-labeled data, and ignore explicit prior knowledge (e.g., concept graph) and weakly-supervised information. Here, the weakly-supervised information is defined as inexact supervision information, i.e. a fine-grained image with a coarse-grained label, by the definition of weak supervision in H [8]. We note that such kinds of information are very cheap to collect, e.g., from WordNet.

The two types of information have been proved to be useful in zero-shot learning [9], object localization [10] and traditional image recognition [11], respectively. However, they are rarely explored in FSL. Recently, Liu et al. [12] explored the weakly-supervised information for FSL and defined a new FSL problem, called Weakly-Supervised FSL (WSFSL). Specifically, they introduce
additional weakly-labeled data (e.g., an image of a guitar with a coarse label of the instrument) and construct a category graph for FSL. Then, a Prototype Propagation Networks (PPN) is proposed to obtain a robust prototype for few-shot classification tasks by leveraging the concept relationship, as shown in Fig. 1(a). However, the prior knowledge is not fully leveraged, e.g., they ignore the semantic information of each category and some intermediate or irrelevant abstract concepts (marked in the blue dotted line in Fig. 1(b)) in the constructed category graph. Moreover, the method does not exploit high-order message passing (shown in the red line in Fig. 1(b)). Thus, the performance improvement is limited.

In this paper, we also focus on the WSFSL problem [12] and propose a novel concept graph-based meta-learning framework (known as MetaConcept) towards the limitations mentioned above. The key idea is to learn a universal meta-learner inferring any-level classifiers (i.e., not only an entity classifier but also a concept classifier at any abstract levels) via a concept graph, so as to boost the classification performance of meta-learning on novel classes, i.e., learn to abstract. Different from the category graph used in PPN, the concept graph consists of abstract concept levels and concrete entity levels, which contains more abundant abstract concepts, relations, and semantics information, as shown in Fig. 1(b). Specifically, in the concept graph, (1) the disjoint classes share common abstract concepts; (2) the entities/concepts can be understood and represented in a common semantic space by using a word embedding model, such as Word2vec [13], fastText [14], Glove [15], and BERT [16]. Thus, the concept graph offers an explicit transfer manner for the meta-learner from the base tasks to novel tasks.

Based on this idea, in the MetaConcept, we propose two key techniques to train and model the meta-learner: (1) multi-level conceptual abstraction-based regularization. To boost the performance of meta-learning on novel classes, we construct multi-level auxiliary tasks (few-shot coarse/concept classification tasks) according to the coarse classes. Then, the valid loss of the auxiliary tasks is regarded as a regularization to train a universal meta-learner, so that the meta-learner is able to infer any-level classifiers (i.e., learn to abstract); (2) meta concept inference network. To extract the task-specific features at different levels, we divide a feature extractor into two submodules (low-level and high-level embedding module). Here, the former is shared by all tasks and the latter is task-specific. As for the classifier, we further propose a graph convolution inference module with relation learning, aiming to infer task-specific initial classifiers by the concept graph. Finally, the high-level embedding module and the classifier can be further finetuned by a few labeled samples to quickly adapt to the any-levels few-shot tasks. During meta-test, the MetaConcept framework is able to directly perform few-shot classification tasks without any weakly-supervise information. Our main contributions of this paper can be summarized as follows:

- To the best of our knowledge, this is the first work to explore the concept hierarchy via concept graph to improve the performance of the meta-learning methods for WSFSL.
- We propose a concept graph-based meta-learning framework. In this framework, a multi-level conceptual abstraction-based regularization is designed to train a universal meta-learner to learn to abstract, and a meta concept inference network is developed as the universal meta-learner to infer any-level initial classifiers.
- We have conducted extensive experiments on two datasets. The results show that the proposed MetaConcept (1) can improve the classification performance by 2–6% in terms of accuracy, and (2) still achieves good performance though it is only trained on weakly-labeled data sets.

The rest of this paper is organized as follows. In Section 2, we have a brief survey on the related works of the WSFSL. In Section 3, we define the WSFSL problem and related notations. In Section 4, we introduce our concept graph-based meta-learning framework (termed as MetaConcept). In Section 5, we evaluate our methods and make comparisons with state-of-the-art methods on two realistic datasets. In Section 6, we come to a conclusion and discuss the future work.

2. Related work

In this section, we briefly summarize related works into three categories: (1) Few-Shot Learning (FSL), (2) Zero-Shot Learning (ZSL), and (3) Graph Neural Networks (GNN).

2.1. Few-shot learning

Few-shot learning aims at learning novel concepts with a limited number of examples by making use of the prior knowledge obtained from past experiences. Currently, most methods primarily focus on meta-learning approaches to solve the FSL problems.
These methods can be grouped into three categories: metric-based approaches [17–20], optimization-based approaches [7], and graph-based approaches [12,21–23].

Metric-based approaches follow a simple nearest neighbour framework and aim at learning a common metric space shared with different tasks by minimizing the intra-class similarity while maximizing the similarity between different classes. ProtoNet [2] make use of the euclidean-based distance as a similarity measure among samples, where they make use of the similarity of query samples and the mean of support samples belonging to the same class to predict the probability of each class. Based on the simplicity of ProtoNet, the AM3 network [24] introduces novel semantic information to boost the robustness of the prototype for each class. Recently, in Li et al. [25], a novel large-scale FSL method is proposed, which constructs a semantics-based class hierarchy as a regularization of training classifier, aiming to obtain transferable feature representations for novel classes from a large number of base classes. Optimization-based approaches target at learning an effective initialization and an optimization method across different tasks. MAML [7] is a typical work in this family, which aims at learning an effective initial parameter for a base learner, so that the base learner can generalize well to novel tasks by a few fine-tuning steps. Based on this idea, many methods extend this work such as LEO [26] and MetaOptNet [27]. Graph-based approaches follow the GNN frameworks, aiming to solve the FSL problems by the supervised message passing networks. For example, a GNN which is trained end-to-end has been proposed in Satorras and Estrach [23], where the nodes are associated with images, and edges are given by a trainable similarity kernel for few-shot classification tasks. Then, EGNN [28] explores edge-labeling graph neural network for FSL, aiming to predict edge-labels, instead of node-labels, by updating node and edge features dynamically. DPCN [29] explores the distribution propagation graph network for FSL, aiming to predict labels by exploring distribution-level and instance-level relations. In [22], a novel transductive propagation network is devised for FSL, targeting at learning to propagate labels from support samples to query samples. Then, EPNet [30] extends [22] by further exploring embedding propagation to obtain a smoother manifold for labels propagation.

Different from these methods, in the paper, we focus on the WSFSL problem, instead of FSL problem. To the best of our knowledge, [12] is the first and only work defining and solving this problem. They introduce a category graph and proposes a prototype propagation network for WSFSL, aiming to propagate the prototypes on a subgraph sampled from the category graph. Different from the PPN method [12], our proposed technique can be considered as a combination of the graph-based and optimization-based approaches. Yet, it differs from existing methods in three ways. First, we introduce a concept graph as the explicit meta-knowledge of the base learner, instead of learning implicit meta-knowledge, so as to boost the classification performance of meta-learning. Second, we model a cross-modal and universal meta-learner via the concept graph, aiming to infer FSL classifiers at any level, instead of learning feature representation. Finally, our method focuses on exploring the global concept graph, not on a subgraph. This enhances the performance of the base learner on novel tasks by fully exploiting concept hierarchy on the global concept graph.

2.2. Zero-shot learning

Zero-shot learning (ZSL) is closely related to FSL, whose objective is to recognize an unseen category when no supervision information is available [31,32]. The key idea is to build semantic connections between the seen classes and unseen classes by exploiting and exploring the prior knowledge. Previous studies mainly focus on semantic embedding-based approaches to address the ZSL problem, which learn a transferable projection function between visual features and semantic representations from the auxiliary data [33]. Recently, a graph based approaches are developed for ZSL. In [9] and [34], a knowledge graph is introduced to build classifier predictor. In [35], the idea is further extended to FSL and a two-stage training framework is built. Though our method is also knowledge graph-based, there are three key differences from the previous studies: (1) we focus on addressing the WSFSL problem by introducing concept graph prior knowledge; (2) we fully explore the concept hierarchy by the joint of weakly-supervised information and the concept graph; (3) we propose a novel meta-learning framework with concept graph, which works in an end-to-end manner.

2.3. Graph neural network

Graph neural network is a type of the deep neural network, which offers a connectionist model for learning from graph-structured data end-to-end [36]. Recently, the GNN has drawn a vast interest in various domains, including knowledge graph [37] and computer vision [38,39]. Graph convolution network (GCN) is one of the classical methods in this family. In [40], the GCN has been employed for solving semi-supervised graph learning problems. It adopts a local graph convolution to represent the current node by aggregating its neighboring nodes, aiming to acquire more robust graph representation. Here, GCN has two key advantages. First, it can learn a good low-dimension embedding for node and graph from the network structure and node information. Second, it can explicitly extract multi-hop representation through node message aggregation layer-by-layer. Hence, we adopt the GCN framework to model a meta-learner aiming to learn a robust abstract and inference strategy on the concept graph for solving the WSFSL problem.

3. Preliminaries and definitions

For the FSL problem, it is difficult to learn a robust deep model by exploring only a little of supervision information. Fortunately, the weakly-labeled data and explicit concept graph are usually free or cheap to collect. In this paper, we focus on the WSFSL problem and propose a novel meta-learning framework to explore the two types of information. In this section, we firstly define the WSFSL problem, and then summarize main notations used in the paper. Here, the notations are defined by following the commonly used notations, i.e., use calligraphic fonts, bold lowercase letters, and bold uppercase letters to denote sets (e.g., \( D \)), vectors (e.g., \( z \)), and matrices (e.g., \( \mathbf{Z} \)), respectively.

Formally, we are given two types of training data set with abundant labeled data: (1) a finely-labeled data set, where the set of fine labels is denoted as a base class set \( \mathcal{C}^{\text{base}} \), which consists of \( n^{\text{base}} \) fine classes; (2) and a weakly-labeled data set (i.e., a finely-grained image with a coarse-grained label), where the set of weak labels is denoted as a coarse class set \( \mathcal{C}^{\text{weak}} \) containing \( n^{\text{level}} \) levels, i.e., \( \mathcal{C}^{\text{weak}} = \{ \mathcal{C}^{\text{weak}}_{l} \}^{l} \) \( l \) denotes the weak label level). We denote the two training data set as \( D^{\text{base}} \) and \( D^{\text{weak}} \), respectively, where the latter consists of \( n^{\text{level}} \) levels, i.e., \( \mathcal{C}^{\text{weak}} = \{ \mathcal{C}^{\text{weak}}_{l} \}^{l} \) \( l \) denotes the weak label level). Meanwhile, we are also given another finely-labeled data set \( D^{\text{met}} \) with few labeled data, where the label set is denoted as novel class set \( \mathcal{C}^{\text{novel}} \). In particular, the base class set \( \mathcal{C}^{\text{base}} \) and novel class set \( \mathcal{C}^{\text{novel}} \) are disjoint. Our goal is to learn a good classifier for novel classes with few labeled data on the two types of training data set, i.e., \( \mathcal{C}^{\text{base}} \) and \( \mathcal{C}^{\text{weak}} \). The problem is called N-way K-shot problem when the novel class set \( \mathcal{C}^{\text{novel}} \) includes \( N \) classes and each class in \( \mathcal{C}^{\text{novel}} \) contains \( K \) labeled samples. Following the definition of weak-supervised learning in H
Table 1

<table>
<thead>
<tr>
<th>Notation</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>(c^\text{base} )</td>
<td>base class set</td>
</tr>
<tr>
<td>(c^\text{weak} )</td>
<td>coarse class set</td>
</tr>
<tr>
<td>(c^\text{novel} )</td>
<td>novel class set</td>
</tr>
<tr>
<td>(n^\text{class} )</td>
<td>number of coarse class levels</td>
</tr>
<tr>
<td>(J^\text{base} )</td>
<td>sample set of base classes</td>
</tr>
<tr>
<td>(J^\text{weak} )</td>
<td>sample set of coarse classes</td>
</tr>
<tr>
<td>(J^\text{novel} )</td>
<td>sample set of novel classes</td>
</tr>
<tr>
<td>((x, y) )</td>
<td>samples</td>
</tr>
<tr>
<td>(N )</td>
<td>number of classes of novel class set</td>
</tr>
<tr>
<td>(K )</td>
<td>number of labeled samples of each novel class</td>
</tr>
</tbody>
</table>

\[ \mathcal{U} \] concept graph
\[ \mathcal{V} \] node set
\[ \mathcal{E} \] edge set
\[ \mathbf{X} \] concept semantic embedding
\[ m \] number of node
\[ \mathbf{A} \] binary adjacency matrix
\[ \mathbf{D} \] degree matrix

[8], comparing with the finely-labeled base/novel class data, the weakly-labeled data is annotated by coarse labels which can be regarded as inexact supervision [8] (i.e., only coarse-grained labels are given). Thus, we term the above FSL setting as weakly-supervised few-shot learning (WSFSL).

Different from the existing FSL problem, the key challenge of addressing the WSFSL problem is how to learn meta-knowledge from the fine-labeled data set \( J^\text{base} \) and the weakly-labeled data set \( J^\text{weak} \). Specifically, in the paper, we address the key challenge as follows:

- For the fine-labeled data set \( J^\text{base} \), we still adopt the episodic training paradigm to learn meta-knowledge. Specifically, in meta-training phase, we sample a large number of \( N \)-way \( K \)-shot tasks from the base class set \( c^\text{base} \) called “few-shot fine classification tasks”. Here, each task \( \tau \) consists of \( N \) classes sampled from \( c^\text{base} \), denoted by \( \{ c_i \}_{i=0}^{N-1} \). For each class, we sample \( K \) labeled samples from \( J^\text{base} \) as a train set (called support set \( S = \{(x_i, y_i)\}_{i=0}^{N\times K-1} \)) and \( m^\text{fine} \) labeled samples as a test set (called query set \( \mathcal{Q} = \{(x_i, y_i)\}_{i=0}^{N\times m^\text{fine}-1} \)). We then perform meta-learning on the few-shot fine classification tasks and make use of the valid loss of the query set to optimize a meta-learner, so as to learn meta-knowledge on the base class set.

- For the weakly-labeled data set \( J^\text{weak} \), we construct multi-level auxiliary tasks from the coarse class set \( c^\text{weak} \) as a regularization, so as to learn a more universal meta-learner for inferring any-level classifiers. In particular, the regularization is able to directly apply to the existing FSL methods to address the WSFSL problem, which is discussed in Section 4.1. To model the meta-learner, we further construct a concept graph from WordNet according to all classes, and propose a concept graph-based meta-learner, which is discussed in Section 4.2. Formally, the concept graph \( \mathcal{G} = (V, E) \) includes \( m \) nodes \( v_i \in V \), a number of edges \( \{(v_i, v_j)\}_{(i, j) \in E} \), a binary adjacency matrix \( \mathbf{A} \in \mathbb{B}^{m \times m} \), a degree matrix \( \mathbf{D} = \sum_j A_{ij} \), and a \( d \)-dimension concept semantic embedding \( \mathbf{X} \in \mathbb{R}^{m \times d} \). Fig. 1(b) shows an example of concept graph. In the concept graph, a leaf node denotes a concrete entity (a fine class). A non-leaf node corresponds to an abstract concept or a coarse class. An edge represents an abstract relationship.

To this end, the estimation of likelihood maximization of the concept graph-based meta-learning can be formally written as:

\[
\max_{\theta} \mathbb{E}_{\tau \sim T^{\tau}} \left[ \mathbb{E}_{(x, y) \sim \tau} \sum_{(x, y) \in \mathcal{Q}} \log(P(y|x, S, J^\text{weak}, \mathcal{G}, \theta)) \right] \tag{1}
\]

where \( \theta \) denotes the model parameters. \( T^{\tau} \) denotes a set of few-shot fine classification tasks. For clarity, the notations mentioned above are summarized in Table 1.

4. MetaConcept framework

In this section, we propose a concept graph-based meta-learning framework to address the WSFSL problem defined in Eq. (1). Specifically, in Section 4.1, we propose a regularization (termed as multi-level conceptual abstraction) to train an universal meta-learner on the fine and coarse class set. In Section 4.2, combining with the proposed regularization, we propose a meta concept inference network to model the universal meta-learner via the explicit concept graph. Finally, in Section 4.3, we discuss how to train and perform the proposed MetaConcept method.

4.1. Multi-level conceptual abstraction

People can understand an unfamiliar object by mapping it into an appropriate level of concepts (called basic-level categorization [41]), which inspires us to design a meta-learning method enabling a machine to learn to abstract. Therefore, the key challenge of the problem defined in Eq. (1) is how to learn abstract concepts via the weakly-labeled data set \( J^\text{weak} \) and the concept graph \( \mathcal{G} \) (i.e., exploring the concept hierarchy). The challenge can be addressed by multi-level concept classifications. That is, we take each coarse class with same levels as a class and apply the concept classification at each level \( l = 0, 1, \ldots, n^\text{level} - 1 \) of the coarse class set. We name the process as multi-level conceptual abstraction (MLCA).

Specifically, following the episodic training paradigm on the base class set, we construct multi-level auxiliary tasks from the coarse class set, called “few-shot coarse classification tasks”. The setting of the few-shot coarse classification task is similar to the few-shot fine classification tasks defined in Section 3. The differences have two-fold: (1) the class set of each few-shot coarse classification task is sampled from the set of coarse classes, i.e., non-leaf nodes of the concept graph; and (2) the samples \((x, y)\) of each task at level \( l \) are taken from the sample set \( J^\text{weak} \). During meta-training, for each episode, we sample batches of few-shot classification tasks from base class set and all level of coarse class set, respectively. Different from the PPN method, these tasks sampled from different levels are independent of each other, so that the concept hierarchy can be explored on global graph, instead of sub-graph. Then, all few-shot coarse classification tasks are performed in forms of regularization in the meta-training phase, i.e., training a meta-learner to infer not only an entity classifier but also a concept classifier at any level. In addition, the regularization is universal, which also can be applied to most FSL methods to directly address the WSFSL problem. To this end, when we adopt the existing FSL methods to address the WSFSL problem, the estimation of likelihood maximization of the methods can be further expressed as:

\[
\max_{\theta} \lambda_{c} \mathbb{E}_{\tau \sim T^{\tau}} \left[ \mathbb{E}_{(x, y) \sim \tau} \sum_{(x, y) \in \mathcal{Q}} \log(P(y|x, S, \theta)) \right] + \\
\lambda_{c} \sum_{l=0}^{n^\text{level}-1} \mathbb{E}_{\tau \sim T^{\tau}} \left[ \mathbb{E}_{(x, y) \sim \tau} \sum_{(x, y) \in \mathcal{Q}} \log(P(y|x, S, \theta)) \right] \tag{2}
\]
where $\mathcal{T}_f^\tau$ denotes a set of few-shot fine classification tasks, $\mathcal{T}_c^l$ denotes the set of few-shot coarse classification tasks at the abstract level $l$, and $\lambda_\tau$ and $\lambda_c$ are hyperparameters adjusting the weight of regularization, the conditional probability $P(y|x, S, \theta)$ can be implemented by most existing meta-learning methods such as MAML [7], ProtoNet [2], MetaOptNet-SVM [27], and TPN [22]. In the same manner, the estimation of likelihood maximization of the concept graph-based meta-learning method can be further expressed as:

$$
\max_{\theta} \lambda_\tau \mathbb{E}_{X \sim \mathcal{T}_c^l}[\mathbb{E}_{y \sim G} \log(P(y|x, S, G, \theta))] + \lambda_c \mathbb{E}_{X \sim \mathcal{T}_c^l}[\mathbb{E}_{y \sim G} \log(P(y|x, S, G, \theta))]
$$

Following the setting of WSFL in Liu et al. [22], $\lambda_c$ is set to one by default. In particular, the learning problem becomes more economical when $\lambda_c$ is set to zero, because the meta-learner is trained merely on the coarse class set $\mathcal{C}_\text{weak}$, which is much cheaper to obtain than base class set (fine class set). In the paper, different from [12], we discuss the two setting of WSFL to evaluate the classification performance of the meta-learning methods.

4.2. Meta concept inference network

4.2.1. Network structure and inference strategy

As defined in Eq. (3), our goal is to design a universal meta-learner for inferring any-level few-shot coarse or fine classifiers. However, learning a generic embedding network or classifier is difficult since the few-shot coarse and fine classification tasks are sampled from different levels (termed as a multi-level problem). Therefore, the key challenge is how to enable a base learner to quickly adapt to the multi-level tasks. Our intuition is that the image features should be extracted at different abstract levels for adapting to the tasks. To this end, we propose a meta concept inference network (MCIN) to address the multi-level problem, which contains two key techniques:

- **Task-specific high-level feature embedding.** Different from the existing meta-learning methods, we divide the feature embedding module of MCIN into two submodules: low-level feature embedding module $f_{\theta_l}$ and high-level feature embedding module $f_{\theta_h}$. Here, the latter is task-specific, aiming to quickly adapt to few-shot classification tasks with different levels. By doing so, the task-specific features at different abstract levels can be extracted.

- **Task-specific classifier.** Different from the existing meta-learning methods, we take the concept graph as inputs (i.e., as explicit meta-knowledge) and propose a graph convolutional inference module (the module is discussed in Section 4.2.2 in details) to directly infer initial classifier for any-level few-shot tasks. Then, the initial classifier can be further finetuned on support set $S$, so as to adapt to few-shot task with different levels.

**Network structure** Specifically, the MCIN framework is illustrated in Fig. 2, which includes a low-level feature embedding module $f_{\theta_l}$, a task-specific module consisting of a high-level feature embedding module $f_{\theta_h}$, and a softmax-based classifier $f_{\theta_c}$. Here, $f_{\theta_c}$ is a graph convolutional inference module (GCCM) $f_{\theta_c}$, with $\delta_{T}(\theta, \theta') \triangleq (\theta, \theta') - (\theta, \theta')$. Specifically, the module $f_{\theta_c}$ is shared by all few-shot tasks, which accounts for extracting transferable features such as corners, color, and textures. The module $f_{\theta_c}$ is a meta-learning module, which can quickly adapt to a task-specific embedding module $f_{\theta_c}$ for a novel task and extract task-specific object features at different abstract levels. In addition, the softmax-based classifier $f_{\theta_c}$ is also a task-specific module. However, the parameter $\theta_c$ cannot be meta-learned but inferred by the module $f_{\theta_c}$.

Inference strategy Based on the above network structure, we propose a three-steps inference strategy to achieve few-shot classification task. Formally, we take the parameter $\theta_c$, $\theta'_c$, and $\theta_{c\theta'}$ as a hidden variable, instead of an optimizable parameter. For the task $r$, the probability $P(y|x, S, G, \theta_c, \theta_{c\theta'}, \theta_{c\theta'})$ of the output $y$ can be calculated in three steps, given the query sample $x$, parameters $\theta_c$, $\theta_{c\theta'}$, and $\theta_{c\theta'}$ support set $S$, and concept graph $G$. First, the probability $P(\theta_c|g, \theta_{c\theta'})$ of the hidden variable $\theta_c$ can be calculated, given of the parameters $\theta_c$ and concept graph $G$. Second, the probability $P(\theta_{c\theta'}|\theta_c, S, \theta_{c\theta'})$ of the hidden variable set $\theta_{c\theta'}$ can be calculated, given of the parameters $\theta_c$ and $\theta_{c\theta'}$ support set $S$, and hidden variable $\theta_c$. Finally, the probability $P(y|x, \theta_c, \theta_{c\theta'}, \theta_{c\theta'})$ of the output $y$ can be calculated, given the query sample $x$, parameters $\theta_c$, and hidden variable $\theta_{c\theta'}$ and $\theta_{c\theta'}$. Hence, these procedure can be expressed as:

$$
P(y|x, S, G, \theta_c, \theta_{c\theta'}, \theta_{c\theta'}) = \int_{[\theta_{c\theta'}]} \int_{[\theta_c]} \int_{[\theta_{c\theta'}]} P(y|x, \theta_c, \theta_{c\theta'}, \theta_{c\theta'}) P(\theta_{c\theta'}) P(\theta_c|g, \theta_{c\theta'}) d\theta_c d\theta_{c\theta'} d\theta_{c\theta'}
$$

(4)

Here, the probability $P(\theta_c|g, \theta_{c\theta'})$ and $P(\theta_{c\theta'})|\theta_c, S, \theta_{c\theta'}, \theta_{c\theta'})$ are expressed in terms of delta function (i.e., point estimation) in the MCIN framework. Specifically, the three-step inference can be performed as follows:

**Concept graph inference** The goal of this step is to model the conditional probability $P(\theta_c|g, \theta_{c\theta'})$ of the hidden variable $\theta_c$. Here, we take the concept graph and the class set of few-shot classification tasks (the class set is used to select class node on concept graph) as inputs of the module $f_{\theta_c}$, to predict the initial parameter $\theta_c$ for the task-specific classifier $f_{\theta_c}$ by aggregating the abstract concepts and extracting discriminated relationship among classes on the concept graph $G$. The key of this step is to transfer the abstract concepts from semantic space to vision classifier $f_{\theta_c}$, as shown in the black lines of Fig. 2. That is,

$$
\theta_c = f_{\theta_c}(G)
$$

(5)

**Inner-loop optimization** The goal of this step is to model the conditional probability $P(\theta_{c\theta'}|\theta_c, S, \theta_{c\theta'}, \theta_{c\theta'})$ of the hidden variable $\theta_{c\theta'}$.
set $\{\theta'_e, \theta'_i\}$. After initial parameters $\theta_e$ are obtained, the hidden variable set $\{\theta'_e, \theta'_i\}$ can be determined by applying $k$-step gradient descent on the support set $S$ of the task $\tau$, which aims to adapt to few-shot classification tasks at different levels, as shown in the blue lines of Fig. 2. For example, when we apply one step of gradient descent, the parameter set $\{\theta'_e, \theta'_i\}$ can be expressed as:

$$\{\theta'_e, \theta'_i\} = \{\theta_e, \theta_i\} - \alpha_{inner} \frac{\partial L_{\text{inner}}(\{\theta_e, \theta_i\})}{\partial \{\theta_e, \theta_i\}} \quad (6)$$

where $L()$ denotes a cross-entropy loss function and $\alpha_{inner}$ is the learning rate of inner-loop optimization.

**Category prediction** The goal of this step is to predict the probability $P(y|x, \theta_e, \theta'_e, \theta'_i)$ of the output $y$. Finally, the probability estimation $\hat{y}$ of each class can be estimated by applying the task-specific module $f_{\theta'_i}(\cdot)$ and $f_{\theta'_e}(\cdot)$ obtained in previous step on the feature embedding $f_{\theta_e}(x)$ of queries samples $x$, as shown in the red lines of Fig. 2. That is,

$$\hat{y} = \text{softmax}(W_{\text{c}}^T f_{\theta'_i}(f_{\theta_e}(x)) + b_{\text{c}}) \quad (7)$$

where $W_{\text{c}}$ and $b_{\text{c}}$ are the parameters of softmax-based classifier, which are obtained from the hidden variable $\theta'_c$.

**Graph convolutional inference module**

In this section, we depict the details of the graph convolutional inference module (GCIM) $f_{\theta_0}(\cdot)$ introduced in Section 4.2. Specifically, we propose a graph convolution network with relation learning to implement the graph convolutional inference module, which includes concept-specific representation learning phase and task-specific representation/parameters learning phase, aiming to infer task-specific initial classifier by exploring the multi-hop abstract relations and semantics of concepts. As illustrated in Fig. 3, the GCIM consists of a graph embedding layer $f_{\theta_0}(\cdot)$, a relation layer $f_{\theta_1}(\cdot)$, and an output layer $f_{\theta_2}(\cdot)$, where $\theta_0$, $\theta_1$, and $\theta_2$ denote the optimizable parameters such that $\theta = (\theta_0, \theta_1, \theta_2)$.

The inference are carried out in two phases:

**Concept-specific representation learning** In the phase, the goal is to learn a concept-specific representation for each node on the concept graph, which is irrelevant to few-shot classification tasks. As shown in Eq. (8), the concept graph $G$ is fed through the graph embedding layer $f_{\theta_0}(\cdot)$, to produce the concept-specific representation $z_h$ for each node $h$:

$$z_h = f_{\theta_0}(G), \quad i = 0, 1, \ldots, m - 1 \quad (8)$$

where $m$ denotes the number of nodes of concept graph. In the paper, the graph embedding layer $f_{\theta_0}(\cdot)$ is implemented by a graph convolution with a simple layer-wide propagation rule. That is,

$$z_{h+1} = \begin{cases} \sigma(D^{-1}AXW_{ge}^h + b_{ge}^h) & \text{if } h = 0 \\ \sigma(D^{-1}AZW_{ge}^h + b_{ge}^h) & \text{otherwise} \end{cases} \quad (9)$$

where $W_{ge}^h$ and $b_{ge}^h$ are the optimizable parameters of the $h$ layer on the graph embedding layer, such that $\theta_{ge} = \begin{bmatrix} W_{ge}^1 & b_{ge}^1 \\ \vdots & \vdots \\ W_{ge}^{N_g} & b_{ge}^{N_g} \end{bmatrix}_{N_g \times N_g}$. $N_g$ is the number of layer, $Z^0$ denotes the node representation of the concept graph at $h$ layer, $X$ denotes the concept semantic embedding matrix of the concept graph, $A$ denotes the adjacency matrix of the concept graph, $D$ denotes the degree matrix of the concept graph, and $\sigma(\cdot)$ is the activation function.

**Task-specific representation/parameters learning** In the phase, our goal is to further learn a task-specific representation and parameters for each class of few-shot classification task, so as to enable the classifier to quickly adapt to a novel task at any level. Specifically, in the phase, the inference is performed on two steps: relation learning and initial classifier prediction. 1) **Relation learning.** We employ a multi-layer perceptron (MLP) to implement the relation layer $f_{\theta_1}(\cdot)$, aiming to further learn a task-specific class representation with discriminated relationship among classes for task $\tau$. Specifically, according the class set $C_{\tau}^{N_{-1}}$ of few-shot classification task, we firstly extract the concept representations $Z_{C_{\tau}}^{N_{-1}}$ from $Z_{ge}^{N_{-1}}$. Second, the representation of each class pair $C_{\ell}$ and $C_j$ of task $\tau$ are combined by the feature concatenation $\text{cat}(Z_{C_{\tau}}^{N_{-1}}, Z_{C_{\tau}}^{N_{-1}})$. Finally, these combined features are fed through the relation layer to produce a representation $r_{C_{\ell}, C_j}$ for the relationship between $C_{\ell}$ and $C_j$, known as concept relationship. That is,

$$r_{C_{\ell}, C_j} = f_{\theta_1}(Z_{C_{\tau}}^{N_{-1}}, Z_{C_{\tau}}^{N_{-1}}) = \text{MLP}(\text{cat}(Z_{C_{\tau}}^{N_{-1}}, Z_{C_{\tau}}^{N_{-1}})) \quad (10)$$

where $i, j = 0, 1, \ldots, N - 1$. Finally, the task-specific class representation $Z_{C_{\tau}}^{N_{-1}}$ with the concept relationship is produced by element-wise mean over the embedding of all classes and adding the residual connection to oneself for each class, as shown in Eq. (11).

$$Z_{C_{\tau}}^{N_{-1}} = Z_{C_{\tau}}^{N_{-1}} + \frac{1}{N} \sum_{j=0}^{N-1} r_{C_{\ell}, C_j} \quad (11)$$

2) **Initial classifier prediction.** According to the learned task-specific class representation, we then employ a single-layer graph convolution with normalization to model the output layer $f_{\theta_2}(\cdot)$ so as to predict the initial parameter $\theta_i$ for the task-specific classifier. The initial parameter $\theta_i$ is constructed by stacking the produced
task-specific class parameter of the selected classes \( \{c_i\}_{i=0}^{N-1} \) for the task \( \tau \). Specifically, \( \theta_c \) can be expressed as:

\[
\theta_c = [W_c, b_c] = \text{Stack}(\text{Norm}(D^{-1}A\Phi^{\tau_0}\Phi^{\tau_0}W_c + b_c)\beta, \{c_i\}_{i=0}^{N-1})
\]

(12)

where \( W_c \) and \( b_c \) are the optimizable parameters of the output layer, \( \theta_c = [W_c, b_c] \). \( \text{Norm}() \) is a normalization function, \( \beta \) is a hyperparameter adjusting the normalization scale, and \( \text{Stack}() \) is a matrix-formed stack operation for the weight \( W_c \) and bias \( b_c \) of the initial classifier \( f_0() \).

4.3. Overall implementation

The aim of meta-training is to learn to abstract for various fine classes (entities) via the concept graph defined by Eq. (3). Therefore, the meta-objective of our proposed MetaConcept can be expressed as:

\[
\min_{\theta_{el}, \theta_{eh}, \theta_{g}} \lambda_c \mathbb{E}_{\mathcal{T} \sim \tau'} \left[ 
\mathbb{E}_{S \sim \tau} \left[ L_{\tau}(\theta_{el}^s, \theta_{eh}^s, \theta_{g}^s) \right] + \right]
\]

\[
\mathbb{E}_{L \sim \tau} \left[ \mathbb{E}_{S \sim \tau'} \left[ L_{\tau}(\theta_{el}^s, \theta_{eh}^s, \theta_{g}^s) \right] \right]
\]

(13)

where the hidden variable set \( \{\theta'_{el}, \theta'_{eh}\} \) is performed in Eqs. (5) and (6). We update all parameters by stochastic gradient descent optimizer (SGD) under the learning rate \( \alpha_{\text{inner}}, \alpha_{\text{outer}} \) to minimize the loss as defined in Eq. (13) by applying the episode-based training strategy [19], known as outer-loop optimization.

Finally, we sum up the algorithm in Algorithm 1. For each episode, we firstly sample batches of few-shot fine classification tasks from the base class set. Then, the parameters of the task-specific module are determined by inferring initial classifier in Line 3 and fine-tuning the initial task-specific module in Line 4 for each task (Lines 2–5). Then, we sample batches of few-shot coarse classification tasks from coarse class set and perform the inference of task-specific module at each level (Lines 6–11). Finally, we calculate the loss with the MLCA regularization, and optimize the parameters \( \theta_{el}, \theta_{eh} \) and \( \theta_{g} \) by using the SGD optimizer as shown in Lines 12 and 13.

5. Performance evaluation

In order to verify the effectiveness of our proposed MetaConcept framework, in this section, we first introduce the experimental settings and then discuss the experiment results on two setting of WSFSF defined in Section 4.1, followed by our ablation study.

5.1. Datasets and settings

WS-ImageNet-Pure. The dataset [12] is a subset of 188 classes selected from the ILSVRC-12 dataset at five different levels (level-7, level-6, level-5, level-4, level-3) of the ImageNet WordNet hierarchy, where the classes from level-7 are the concrete entity classes (i.e., fine classes) and the classes from other four levels are the abstract concept classes (i.e., coarse classes). The dataset is split into two disjoint subsets following [12], i.e., a meta-training set and a meta-test set. Note that the data samples of all classes are sampled in a bottom-up manner, where the samples of any classes on level I are sampled from all the images belonging to the class in ImageNet. Further details can be found in Liu et al. [12].

WS-ImageNet-Mix. The dataset [12] is another subset of 188 classes selected from the ILSVRC-12 dataset. We still adopt the same split method in Liu et al. [12]. The dataset is similar to WS-ImageNet-Pure. The key difference is that the data samples of the abstract concept class can belong to the remaining 20% level-7 classes outside of the 80% level-7 classes used for generating few-shot fine classification tasks. The goal is to further analyze the effect of the abstract concept classes when its data samples sampled from other concrete entities not involved in the few-shot entity classification tasks. Please refer to [12] for details.

Concept graph. The concept graph can be constructed from cheap knowledge graph such as WordNet [42] for each given dataset. Specifically, we regard all categories of the dataset as the leaf nodes, and then extract their abstract concepts from knowledge graph at multiple levels as the nonleaf nodes to build the concept graph. In addition, we also extract two-hops abstract concepts which are connected with the coarse classes by using a intermediate abstract concept. For each concrete entity and the abstract concept, we use the GloVe model [15] to extract a 300-dimension concept semantic embedding as the node features of the concept graph, that the mean value of word embeddings of entity and concept names.

5.2. Implementation details

Architecture. For a fair comparison, we use a 4-layer convnet [2,12,22] with 64 channels per layer as feature extractor, which is partitioned into a low-level module with two convolutional layers and a high-level module with two convolutional layers. In the meta concept inference network, we use two-layer graph convolution to model the graph embedding layer whose dimensions are 4096 and 2048, respectively, where we add dropout layers with keep probability of 0.9. Moreover, we use a two-layer MLP to model the relation layer whose dimensions are 4096 and 2048, respectively, where we take Leaky ReLU with the negative slope of 0.1 as the activation function and add dropout layers with keep probability of 0.9. Furthermore, we use a single-layer graph convolution as the output layer with 1601 dimensions.

Training details. We adopt the SGD optimizer with an initial learning rate of \( 10^{-1} \), a momentum of 0.9, and weight decay of 0.0005 to train the proposed model with 20000 iterations, where the learning rate is reduced by 0.1 for every 500 iterations. Hyperparameters \( \beta \) and \( \lambda_c \), and inner-learning rate \( \alpha_{\text{inner}} \) are set to be 0.2, 0.75, and 0.01, respectively. For the inner update step \( k \), we set to be 5 in the meta-training phase and 10 in the meta-test phase.
In this section, we conduct two experimental setting of WSFSFL defined in Section 4.1 on the public two WSFSFL datasets (i.e., WS-ImageNet-Pure and WS-ImageNet-Mix) to show the effectiveness of the proposed MetaConcept method.

5.3. Discussion of results

We compare the proposed MetaConcept method with the recent state-of-the-art methods on the above datasets, and show the results of few-shot fine classification tasks on novel class set with four different aspects.

- We implement a number of classical or state-of-the-art methods used to addressing the FSL problem as the basis of comparison, and report the mean accuracies as the first set of results in Table 2. Note that these methods are only trained on finely-labeled base class data set. The goal is to show the effectiveness of exploring concept hierarchy by using the weakly-labeled data and concept graph.

- We present results in Liu et al. [12] for comparison as the second set of results in Table 2. These methods are related to us and are used to solve the WSFSFL problem. Moreover, we also implement the KTCH method [25] to address the WSFSFL problem. Here, we remove the semantics-based construction process of class hierarchy and replace it with the concept graph because the concept graph provides a more accuracy class hierarchy. Different from these methods, the proposed MetaConcept method focus on how to infer any-level classifier via the global concept graph. Our goal is to show the effectiveness of our method for addressing the WSFSFL problem.

- According to Eq. (2), we apply the proposed MLCA-based regularization to our implemented eight baseline methods (i.e., MAML, ProtoNet, MetaOptNet-SVM, LEO, WS-LKT, AM3-ProtoNet, TPN, and EPNet) to address the WSFSFL problem. Here, we train the models on finely-labeled base class data set and weakly-labeled data set. Finally, we report the results as the third set of results in Table 2. Our aims are that (1) taking the results as baselines of comparison, to show the effectiveness of the proposed MetaConcept method on the default setting of WSFSFL; (2) verifying the effectiveness of MLCA-based regularization on the default setting of WSFSFL for the existing FSL methods.

- We report the results of MetaConcept as shown in the last row in Table 2.

In Table 2, we first present the total number of parameters of each model and averaged running time of each episode (at meta-test phase) as an illustration of the computational cost. The reported time is evaluated on a machine with an NVIDIA 2080Ti GPU. Then, the experimental results are presented in 4-tupled values \( (A_1, A_2, B_1, B_2) \). Here, \( A_1 \) and \( A_2 \) denotes the mean classification accuracy averaged over 600 test episodes of the 5-way 1-shot and 5-way 5-shot tasks on WS-ImageNet-Pure, respectively. Similarly, \( B_1 \) and \( B_2 \) corresponds to the mean classification accuracy of the 5-way 1-shot and 5-way 5-shot tasks on WS-ImageNet-Mix, respectively. The classification accuracy is defined as the number of correct predictions divided by the total number of query samples for each episode. The 95% confidence intervals of the estimates are also shown in Table 2. From Table 2, it can be observed that the proposed MetaConcept achieves the best few-shot classification performance of \((51.06\%, 57.36\%, 47.51\%, 59.90\%)\) and achieves a significant improvement ranging from 2% to 6% on the above four tasks. This verifies the effectiveness of our proposed MetaConcept method. Meanwhile, a potential problem can be observed that MetaConcept also increases the computational cost (especially, the number of parameters of model) due to considering the additional concept graph in the GCIM module. We would like to emphasize that in this paper, we mainly focus on improving the classification performance of meta-learning on WSFSFL problem by exploiting the concept graph. Thus, this high computational cost problem will be left over for future work.
Performance analysis of exploring concept hierarchy. The comparison results of first set and last row of Table 2 exhibit the impact of exploring concept hierarchy by leveraging the concept graph and the weakly-labeled data on classification performance. We can see that the proposed MetaConcept method which explores concept hierarchy outperforms the baseline methods without this, around 3% to 12% on all tasks. Moreover, it is obvious that the improvement on the 5-way 1-shot tasks is larger than that on the 5-way 5-shot tasks, i.e., around 6% and 1% on WS-ImageNet-Pure and WS-ImageNet-Mix, respectively. This verifies the effectiveness of exploring concept hierarchy, and implies that it can significantly boost the classification performance of novel classes especially when the annotated samples are insufficient.

Performance analysis of MLCA regularization on existing FSL methods. The comparison results of first set and third set of Table 2 exhibit the impact of the MLCA regularization on classification performance for existing FSL methods. We can see that the performance improvement is remarkable for most baseline methods, for example, the FSLTK method achieves a classification accuracy improvement of 4% to 8% on all tasks. Moreover, the baseline methods achieve almost consistent performance with PPNN+ when it is combined with the MLCA regularization. This verifies the effectiveness of the MLCA regularization, which can be directly applied to the existing FSL methods and boost its classification performance.

Performance analysis of MetaConcept. From the comparison results of the second set and last row of Table 2, we can also observe that the proposed MetaConcept consistently outperforms the baseline methods with weakly-supervised strategy. Especially, compared with PPNN+ method, our proposed MetaConcept method is consistently yielding a higher classification accuracy, around 3% to 6% on all tasks. This implies that our proposed MetaConcept method is more effective than the PPNN+ method. There are three reasons for such performance gain. First, in the proposed MetaConcept method, we introduce a concept graph, which includes more abundant abstract concepts, relations, and semantic information than category graph. Second, the proposed MetaConcept method builds a MLCA regularization on a global concept graph, instead of a subgraph, which can guide the MCIN-based meta-learner to learn to abstract concepts via the global concept graph, i.e., fully exploiting the concept hierarchy. Third, the joint inference of the abstract concepts and few labeled samples is more effective for inferring task-specific classifiers than prototype propagation, due to exploring high-order relations on concept graph. The KTCH method also explore concept hierarchy, but they focus on leveraging it to learn transferable feature representation for novel classes. However, the proposed method focuses on modeling a universal meta-learner with the concept graph to infer any-level classifier. The results demonstrate the effectiveness of the proposed method, which outperforms the KTCH method, around 8% to 12% in terms of the classification accuracy. Furthermore, it is worth noting that the improved performances of the MetaConcept on WS-ImageNet-Mix is remarkable, around 4% to 6% in the classification accuracy. The observation indicates that the data samples of abstract concept classes from other concrete entities are particularly helpful for MetaConcept. It can provide more abundant abstract information for training the meta-learner to learn to abstract.

5.3.2. Results on the more economical setting of WSFSL ($\lambda_c = 0.0$)

In the section, to further verify the effectiveness of our proposed MetaConcept method, we conduct some experiments on the more economical WSFSL setting defined in Section 4.1 (i.e., $\lambda_c$ is set to zero). The experiments are similar to Section 5.3.1. The difference is that $\lambda_c$ is set to one. We report the results of MetaConcept and implemented seven baseline methods in Table 3.

Performance analysis of MetaConcept. According to the results of Tables 2 and 3, we find that (1) MetaConcept outperforms eight baseline methods, around 3% to 8% in classification accuracy on all tasks; (2) MetaConcept also outperforms the PPNN method, around 5% to 11% on all tasks, although it is trained only on the weakly-labeled data set; and (3) for the WS-ImageNet-Mix, MetaConcept trained only on the weakly-labeled data set also achieves almost consistent performance with that on the two types of data set. This further implies that the proposed MetaConcept method, exploring the concept hierarchy by leveraging concept graph and weakly-labeled data, is effective for WSFSL.

Performance analysis of MLCA on existing FSL methods. As shown in Tables 2 and 3, we find that the eight baseline methods trained on weakly-labeled data set achieves almost consistent or even better performance with that on finely-labeled base class data set, for example, the MLCA-ProtoNet method achieves a classification accuracy improvement of 1% to 4% on all tasks. This further implies that the MLCA regularization is effectiveness which is able to explore and exploit weakly-supervised information for the existing FSL methods.

5.4. Ablation study

In the section, we carry out an ablation study on the default setting of WSFSL ($\lambda_c = 1.0$) to answer the following four questions.

How does MLCA affect the performance of few-shot fine classification? We show the results of eight baseline methods and MetaConcept with MLCA and without MLCA in Tables 2 and 4, respectively. In the Tables 2 and 4, we find that (1) the performance of the eight baseline methods becomes better by applying MLCA, where, for example, the MetaOptNet-SVM achieves a classification accuracy improvement of 0.5% to 3% on all tasks; (2) the performance of ProtoNet with MLCA outperforms ProtoNet with the weakly-supervised strategy on subgraph, around 2% to 7% in classification accuracy; (3) the performance of MetaConcept becomes poor when removing the MLCA, around 4% to 9% reduction in classification accuracy. The observations indicate that MLCA is essential for MetaConcept and can improve the classification performance on few-shot classification tasks significantly.

How does MCIN affect the performance of few-shot fine classification? In Tables 2, we can also observe that the MetaConcept outperforms the seven baseline methods combined with MLCA (e.g., around 2% to 12% for AM3-ProtoNet), which testifies that the MCIN is more effective when combined with MLCA.

How does concept semantics affect the performance of few-shot fine classifications? To analyze the effects of concept semantics, we compare the performance of MetaConcept with different concept embedding, including one-hot code, Word2Vec [13], fastText [14], GloVe [15], and BERT [16]. Here, (1) the one-hot code-based MetaConcept method ignores concept semantics, aiming to infer the initial classifier by only making use of graph structure; (2) the Word2Vector, fastText, GloVe, and BERT based MetaConcept methods fully exploit concept graph including concept semantics and graph structure. The results are shown in Table 5. It can be observed that: (1) MetaConcept employing the concept semantics (except for BERT) achieves better performance than the one with one-hot code. This means that the concept semantics is useful for MetaConcept; (2) the GloVe-based MetaConcept achieves best performance. The reason may be that the GloVe focuses on global text corpus, instead of local text corpus; (3) the BERT-based MetaConcept obtains poor performance, which may be reasonable because the BERT focuses on context-based word embedding. This results in that a good representation is difficult to obtain for the class names without context.

Can the proposed MetaConcept infer any-level classifiers? To verify that MetaConcept has the ability to infer any-level classifiers, we show the classification results of MetaConcept and four baseline methods [MAML, ProtoNet, MetaOptNet-SVM, and AM3-
Table 3
Experimental results on WS-ImageNet-Pure and WS-ImageNet-Mix when $\lambda_2$ is set to zero. The best results of each set are highlighted in bold and the best results are italicized. Note that “MLCA-” denotes combining with MLCA technique.

<table>
<thead>
<tr>
<th>Method</th>
<th>WS-ImageNet-Pure</th>
<th>WS-ImageNet-Mix</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5-way 1-shot</td>
<td>5-way 5-shot</td>
</tr>
<tr>
<td>MLCA-MAML</td>
<td>33.25 ± 0.91%</td>
<td>45.49 ± 1.02%</td>
</tr>
<tr>
<td>MLCA-ProtoNet</td>
<td>34.60 ± 0.92%</td>
<td>47.89 ± 1.02%</td>
</tr>
<tr>
<td>MLCA-MetaOptNet-SVM</td>
<td>35.76 ± 0.88%</td>
<td>48.07 ± 0.95%</td>
</tr>
<tr>
<td>MLCA-LED</td>
<td>33.74 ± 0.71%</td>
<td>47.18 ± 0.72%</td>
</tr>
<tr>
<td>MLCA-FSAX</td>
<td>45.22 ± 0.90%</td>
<td>50.03 ± 0.93%</td>
</tr>
<tr>
<td>MLCA-AM3-ProtoNet</td>
<td>40.07 ± 0.89%</td>
<td>50.55 ± 0.95%</td>
</tr>
<tr>
<td>MLCA-TPN</td>
<td>39.20 ± 0.79%</td>
<td>49.10 ± 0.76%</td>
</tr>
<tr>
<td>MLCA-EPNet</td>
<td>37.65 ± 0.95%</td>
<td>47.22 ± 0.92%</td>
</tr>
<tr>
<td>MetaConcept</td>
<td>48.56 ± 0.93%</td>
<td>56.17 ± 0.93%</td>
</tr>
</tbody>
</table>

Table 4
Experiment results of MetaConcept with adding or removing MLCA.

<table>
<thead>
<tr>
<th>Setting</th>
<th>WS-ImageNet-Pure</th>
<th>WS-ImageNet-Mix</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5-way 1-shot</td>
<td>5-way 5-shot</td>
</tr>
<tr>
<td>w/ MLCA</td>
<td>51.06 ± 0.96%</td>
<td>57.36 ± 0.91%</td>
</tr>
<tr>
<td>w/o MLCA</td>
<td>42.99 ± 0.83%</td>
<td>51.59 ± 0.91%</td>
</tr>
</tbody>
</table>

Table 5
Experiment results of MetaConcept with different concept embedding.

<table>
<thead>
<tr>
<th>Method</th>
<th>WS-ImageNet-Pure</th>
<th>WS-ImageNet-Mix</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>5-way 1-shot</td>
<td>5-way 5-shot</td>
</tr>
<tr>
<td>+one-hot</td>
<td>46.59 ± 0.88%</td>
<td>55.15 ± 0.90%</td>
</tr>
<tr>
<td>+Word2Vec</td>
<td>50.35 ± 0.92%</td>
<td>56.42 ± 0.90%</td>
</tr>
<tr>
<td>+fastText</td>
<td>49.44 ± 0.88%</td>
<td>57.00 ± 0.93%</td>
</tr>
<tr>
<td>+GloVe</td>
<td>51.06 ± 0.96%</td>
<td>57.36 ± 0.91%</td>
</tr>
<tr>
<td>+BERT</td>
<td>45.11 ± 0.91%</td>
<td>54.47 ± 1.00%</td>
</tr>
</tbody>
</table>

Fig. 4. Test accuracy of MetaConcept and Baseline methods at abstract levels $l = 3, 4, 5, 6$ of two datasets, with MLCA (marked with *) and without MLCA (marked with ). Note that the meta-test set of WS-ImageNet-Pure has only one class at level-3. The results of MAML are invalid at level $l = 3$ on WS-ImageNet-Mix since the number of categories at level $l = 3$ is less than five.

ProtoNet methods) on few-shot tasks with different abstract levels, as shown in Fig. 4. It can be observed that the MetaConcept method achieves the superior performance on all few-shot tasks with different abstract levels, which indicates that MetaConcept is able to infer any-level classifiers. Besides, we also find that (1) the performance of the proposed MetaConcept method exceeds that of the baseline methods even without MLCA. This means that the meta-learner (i.e., MCIN) is effective, which can be generalized to the abstract concepts from the concrete entities; (2) the performance of the proposed MetaConcept method can be further boosted by applying the MLCA technique. This demonstrates that the effectiveness of MLCA, which can explore the concept hierarchy of the concept graph and further guide the meta-learner to learn to abstract.
How does the proposed MetaConcept work? To enable base learner to quickly adapt to the few-shot classification tasks with different levels (i.e., learn to abstract), in this paper, our proposed MetaConcept method focuses on inferring task-specific initial classifiers by leveraging the concept graph, rather than learning a universal initial classifier like MAML. To illustrate it, we visualize the inner-loop optimization process of MetaConcept and MAML. Specifically, we sample 600 episodes (5-way 1-shot test tasks) from WS-Imagenet-Pure at abstract levels $l = 7, 6, 5, 4$, respectively, and report the averaged test accuracy of the task-specific classifier learned by MetaConcept and MAML at each inner-loop update step. The experimental results are shown in Fig. 5. We observe that the MetaConcept method outperforms MAML by a large margin for all few-shot classification tasks. This indicates that MetaConcept have the ability of inferring any-level classifiers. Moreover, it can be also observed that (1) MetaConcept already achieves the superior performance before performing inner-loop optimization, i.e., the significant performance improvement is mainly introduced by the initial classifier; (2) MetaConcept can adapt to any-level few-shot classification tasks more quickly than MAML. This means that the proposed MetaConcept indeed obtains better initial classifiers than MAML, which can quickly adapt to few-shot classification tasks at any levels.

5.5. Hyperparameters analysis

In the section, we carry out detailed experiments on the default setting of WSFSI $(\lambda_c = 1.0)$ to further answer the following two research questions.

How does the partitioning strategy of feature embedding module affect the performance of few-shot fine classification? We conduct a number of experiments on WS-imagenet-Pure and WS-imagenet-Mix by applying five partitioning strategies for feature embedding module, aiming to analyze the impact of the partitioning strategy on few-shot fine classification. We show the results of our proposed MetaConcept method in Fig. 6. As shown in the Fig. 6, we can find that (1) the performance of our proposed MetaConcept method with partitioning strategy (i.e., L-H = 1-3, 2-2, and 3-1) outperforms without partitioning strategy (i.e., L-H = 0-4 and 4-0); (2) our proposed MetaConcept method achieve the best performance when partitioning the feature embedding module into low-level embedding module with two convolutional layers and high-level embedding module with two convolutional layers, i.e., L=2 and H=2. This shows that the partitioning strategy of embedding module is helpful for learning a universal meta-learner. The reason may be that (1) the low-level embedding module is a shared module, which can learn the transferable low-level feature such as corners, edge, color, and textures; (2) the high-level embedding module is a task-specific module, which can quickly adapt to a new task and extract task-specific object feature at different abstract levels.

How do the hyperparameters affect the performance of few-shot fine classification? To analyze the impacts of hyperparameters, we show the results of the MetaConcept method with different normalization scale $\beta$ and different weight of MLCA regularization $\lambda_c$ on 5-way 1/5-shot tasks of WS-Imagenet-Pure/Mix data sets. The results are shown in Fig. 7a–d. Here, the normalization scale $\beta$ is varied from 0.1 to 1.0 and the weight of MLCA regularization $\lambda_c$ is varied from 0.25 to 2. From Fig. 7, we observe that (1) the hyperparameter $\beta$ has an important impact on few-shot fine classification tasks, especially the 5-way 5-shot classification task; (2) the MetaConcept method can achieve a better performance when we set a smaller scale for the normalization of classifier weights, around $\beta = 0.1 \sim 0.3$; 3) as for the weight $\lambda_c$ of MLCA regular-
ization, MetaConcept can obtain consistent performance improvements on the four tasks when we set $\lambda_c$ to 0.5 – 1.0.

6. Conclusions

In this paper, we introduce a concept graph and propose a novel concept graph-based meta-learning framework, named as MetaConcept, for tackling weakly-supervised few-shot learning (WSFSL) problem. The MetaConcept framework consists of two key components, i.e., a regularization based on multi-level conceptual abstraction and a meta concept inference networks. Specifically, the regularization with multi-level conceptual abstraction is designed to explore the weakly-supervision information of the coarse class set, which can be directly applied to most existing meta-learning methods to addressing the WSFSL problem. The meta concept inference networks is designed to infer the task-specific classifier for any-level few-shot classification tasks by making use of the concept graph. Extensive experiments demonstrate that the MetaConcept method outperforms the state-of-the-art methods and the two components are effective for boosting the classification performance of meta-learning on novel classes. In the future work, we can consider: (1) a novel Bayesian inference method to make use of the concept hierarchy prior for further investigation of the potential on using the concept graph; (2) a lightweight graph convolutional inference module to decrease the computational cost; and (3) an extended MetaConcept framework to few-shot object detection and few-shot semantic segmentation tasks.

Declaration of Competing Interest

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

Acknowledgments

This work was supported by the Shenzhen Science and Technology Program under Grants nos. JCYJ2018050718323045 and JCYJ2020010913014456.

References
