Improved Continually Evolved Classifiers for Few-Shot Class-Incremental Learning

Ye Wang, Guoshuai Zhao, *Member, IEEE*, and Xueming Qian, *Member, IEEE*

***Abstract*— *Few-shot class-incremental learning* (FSCIL) aims to continually learn new classes using a few samples while not forgetting the old classes. The scarcity of new training data will seriously destroy the model's stability and plasticity. Continually Evolved Classifiers (CEC) (Zhang et al., 2021), a kind of framework, maintains the stability by freezing the encoder and achieves the plasticity by evolving the classifier along with a pseudo incremental learning scheme. However, the performance of CEC is limited due to 1) inequitable information gains between classifier weights and test features, and 2) inefficient learning task construction strategy. To address the first issue, we propose a *Knowledge-guided Relation Refinement Module* (KRRM) to update both the classifier weights and test features. The main function of KRRM is achieved through cross-attention to propagate the knowledge represented by old encoded data. To address the second issue, we design a *Pseudo Incremental relation Refinement Learning* (PIRL) that utilizes a novel hard concepts mining strategy to mine hard concept tasks globally and locally. By successfully addressing the two issues, our proposed method, named Improved Continually Evolved Classifiers (CEC+), extends the potential of CEC without introducing any additional parameters. More precisely, extensive experiments on CIFAR100, *mini*ImageNet, and Caltech-UCSD Birds-200-2011, demonstrate that our proposed method surpasses prior state-of-the-art methods.**

***Index Terms*— Lifelong learning, few-shot class-incremental learning, image recognition, cross-attention mechanism.**

I. Introduction

DEEP Neural Networks (DNNs) have achieved remarkable success on many vision tasks [1], [2], [3], [4]. However, such a model is designed to process only predefined classes, which limits it's application on many practical image recognition scenarios where the number of recognition targets keeps growing. Especially, when classes of new coming targets are different from the previous ones, this learning paradigm is called *Class-Incremental Learning* (CIL) [5]. Despite the advance of current CIL methods [6], [7], the success of these methods lies in the availability of sufficient annotated training samples for new classes, which requires a significant investment of time and effort. Additionally, in some practical scenarios like identifying rare bird species, training samples for new classes are scarce. Regarding such a challenge scenario, *few-shot class-incremental learning*(FSCIL) is proposed to explore efficient solutions to help the model learn new classes with few annotated training samples incrementally.

FSCIL simulates real-world scenarios and sets up a series of sequentially incoming learning sessions. The first session, dubbed the base session, consists of lots of training samples. In contrast, the following sessions, called incremental sessions, only have a few training data for each class. In each session, only the data of the current session is available while the model needs to classify all encountered classes. The challenge lies in that is the scarcity of new training data will seriously destroy the model's stability and plasticity.

To address these issues, most existing methods [1], [8], [9] freeze the encoder in incremental sessions to maintain the stability and design various modules or strategies to improve the plasticity. For example, Continually Evolved Classifiers (CEC) [1], a superior method, proposes an Adaption Module (AM) that evolves the classifier by propagating the context information between classifier weights, and equips a pseudo incremental learning strategy (PIL) to learn the AM's parameters. However, the performance of CEC is still limited due to *1) inequitable information gains between classifier weights and test features, 2) inefficient learning task construction strategy.*

To address the first issue, we propose a Knowledge-guided Relation Refinement Module (KRRM) to update both the classifier weights and test features by knowledge propagation. In incremental sessions, the frozen encoder inevitably represents new classes weak. Despite updating the classifier weights can help the classifier select representative features for each test sample, poorly represented test samples still make the relation between the classifier weights and the test features not discriminative. To enable the classifier classify new classes well, it's optimal to update both the classifier weights and test features. Considering that some general features of new classes also exist in old classes. For example, the zebra has a similar stripe type to the tiger and a similar body shape

Manuscript received 9 March 2023; revised 20 May 2023 and 7 June 2023; accepted 21 June 2023. Date of publication 30 June 2023; date of current version 6 February 2024. This work was supported in part by the NSFC under Grant 62272380 and Grant 62103317; and in part by the Science and Technology Program of Xi'an, China, under Grant 21RGZN0017. This article was recommended by Associate Editor S. Wang. *(Corresponding author: Xueming Qian.)*

Ye Wang is with the SMILES Laboratory, School of Information and Communications Engineering, Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: xjtu2wangye@stu.xjtu.edu.cn).

Guoshuai Zhao is with the SMILES Laboratory, School of Information and Communications Engineering, Faculty of Electronic and Information Engineering, Xi'an Jiaotong University, Xi'an 710049, China, and also with Shaanxi Yulan Jiuzhou Intelligent Optoelectronic Technology Company Ltd., Xi'an 710000, China (e-mail: guoshuai.zhao@xjtu.edu.cn).

Xueming Qian is with the Ministry of Education Key Laboratory for Intelligent Networks and Network Security, SMILES Laboratory, School of Information and Communication Engineering, Xi'an Jiaotong University, Xi'an 710049, China, and also with Shaanxi Yulan Jiuzhou Intelligent Optoelectronic Technology Company Ltd., Xi'an 710000, China (e-mail: qianxm@mail.xjtu.edu.cn).

This article has supplementary material provided by the authors and color versions of one or more figures available at https://doi.org/10.1109/TCSVT.2023.3291054.

Digital Object Identifier 10.1109/TCSVT.2023.3291054

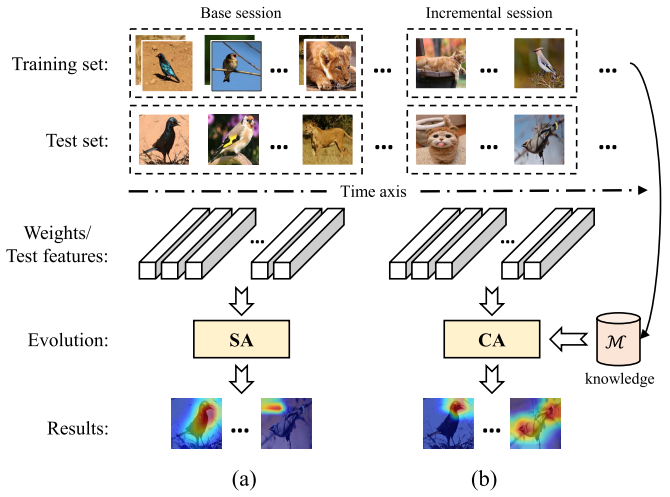


Fig. 1. Compared to the (a) baseline, (b) our proposed method makes more discriminative decisions by using old knowledge to augment both classifier weights and test features. **SA:**self-attention, **CA:**cross-attention, $\mathcal{M}$ refers to memory used to store knowledge.

to the horse. We represent the old knowledge in the form of prototypes for memory efficient and propose a *Knowledge-guided Relation Refinement Module* (KRRM) to augment both the classifier weights and test features by propagating the knowledge, where the knowledge propagation is achieved by cross-attention mechanism. By such a way, we can not only help the classifier weights to select representative features for each test sample, but also can utilize the general feature existed in the old knowledge to augment the test feature, thus making the relations between the classifier weights and test features more discriminative. As shown in Figure 1, compared to CEC, more discriminative decisions can be made with this module conducted. Additionally, apart from the prototypes, we also explore other old knowledge forms, such as trainable prototypes or features. Interestingly, we find that though there exists a trade-off between stability and plasticity under different forms of old knowledge, both can achieve excellent performance.

To address the second issue, we design a Pseudo Incremental relation Refinement learning (PIRL) using a hard concept mining strategy to construct more effective learning tasks. To enable the ability of KRRM, it's optimal to train it under the incremental setting. However, the incremental classes only include scarce training samples and the old data is not available making such training impossible. When facing this difficulty, the pseudo incremental learning (PIL) proposed by CEC mimics the real incremental setting and constructs a series of pseudo incremental tasks to learn their adaption module, where each task consists of several pseudo base classes sampled from the base session and pseudo new classes obtained by rotating the pseudo base classes. However, this strategy is contradictory and sub-optimal. The intrinsic reason is that the sampled pseudo base classes only consists of few training samples similar to real new classes, the constructed pseudo incremental tasks actually only consist of several pseudo new classes which do not match the real incremental setting, thus compromising the model's performance. To construct more effective pseudo learning tasks, we propose the *Pseudo Incremental relation Refinement learning* (PIRL). In contrast to CEC, PIRL first samples several classes from the base session as pseudo new classes and the remaining classes as pseudo base classes. Each pseudo new class is composed of a support set and a query set, which can be treated as the training and test sets of real new class, respectively. The pseudo new classes are then rotated and combined with rotated classes to construct local learning tasks. By using the local learning tasks to optimize the KRRM, the model's plasticity is improved. However, optimizing only the model's plasticity can affect its stability, so we next construct global learning tasks by combining the pseudo old and new classes. By introducing the information of base classes, impact of local learning tasks on the model's stability is mitigated. Furthermore, considering that a delicately pre-trained model may still classify these tasks well, we combine the query features and corresponding top-K classifier weights to construct hard concept tasks from local and global learning tasks.[1] By using these hard concept tasks to optimize the KRRM, the ability of the KRRM is enhanced.

By solving the two issues, our proposed method, dubbed Improved Continually Evolved Classifiers (CEC+), successfully extends the potential of CEC without introducing any additional parameters. We conduct extensive experiments on three popular FSCIL benchmark datasets, CIFAR100, *mini*ImageNet, and Caltech-UCSD Birds-200-2011. The qualitative and quantitative results demonstrate CEC+ surpasses prior state-of-the-art methods.

In summary, our main contributions are as follows:

- We propose Improved Continually Evolved Classifiers (CEC+) that refines the relations between classifier weights and test features by knowledge propagation.
- We propose a Knowledge-guided Relation Refinement Module (KRRM) that utilizes cross-attention to adaptively mines valuable information from the stored knowledge to augment classifier weights and test features.
- We design a Pseudo Incremental relation Refinement Learning (PIRL) using hard concepts mining strategy to mine hard concept tasks to augment the ability of KRRM.

II. Related Work

In this section, we first review several studies about few-shot learning (FSL) and class-incremental learning (CIL) as they present preliminary knowledge of our work. Then, we briefly introduce the recent research focused on few-shot class-incremental learning (FSCIL).

A. Few-Shot Learning

Few-shot learning (FSL) aims to learn a classifier to classify new classes using a few samples. Current FSL can be categorized into three parts: metric-based, optimization-based, and hallucination-based. The metric-based methods measure the relation between the support and query sets by leveraging fixed metrics, learning transferable deep metrics, or Graph Neural

[1] In FSCIL, we often use the prototype given by the mean feature of the training data to represent each class. As a result, the classifier weights initialized by prototypes capture the concepts of their corresponding classes. Furthermore, given that the top-K classifier weights of each query feature tend to be highly similar, we regard them as hard concepts, and the tasks constructed using these weights and query features as hard concept tasks.

Networks. (1) Leveraging fixed metrics [10], [11], [12], [13]. For example, Vinyals et al. [10] propose a matching network that leverages cosine to measure the relation. Snell et al. [11] propose a prototypical network that calculates the prototype feature by averaging features of the support set and demonstrates that the Euclidean distance is better than the cosine used in [10]. Zhu et al. [13] propose a method that projects features into subspace and utilizes the Wasserstein distance to perform relation measuring. (2) Learning transferable deep metric [14], [15], [16]. Selecting an optimal metric is often trivial and relies on the expert-knowledge, Sung et al. [14] propose a classical method named relation network that utilizes the MLP to learn the relation between prototypes and query features automatically. Considering that the comparison ability of relation network [14] is limited to the inherent local connectivity of CNN, Wu et al. [15] propose a method that utilizes a deformable feature extractor, self-correlation, and cross-correlation to solve this problem. Compared with previous relation networks, Cao et al. [16] propose to introduce more information, such as appearance and mutual information, to model the relation between the query and support samples. (3) Graph Neural Network [17], [18], [19], [20]. Unlike previous metric-based methods, Graph Neural Network introduces the relation information to guide the label propagation. For example, Yang et al. [17] propose a bi-directional graph neural network that makes the relation and features guide each other to make discriminative relation measuring. Chen et al. [19] propose a method that introduces class-level knowledge to calibrate the relation measured by the instance-level graph. The optimization-based methods [21], [22], [23] focus on learning a good initial model that can quickly adapt to new classes using a few samples. For example, Finn et al. [21] propose a classical and famous method named MAML, consisting of a meta-learner using the support set and a fixed learning rate to conduct model fast adaption. After the emergence of MAML, many MAML-based works [24], [25], [26] are proposed. For example, Rusu et al. [24] decouple the gradient-based adaptation procedure from the underlying high-dimensional space of model parameters to a low-dimensional space to make the model generalize to new tasks easier. Baik et al. [26] propose task-and-layer-wise attenuation on the compromised initialization to reduce the adverse effects of forcibly sharing the initialization in MAML. The hallucination or generation-based methods focus on learning a generation model or module to generate classification weights [27] or fake samples [28], [29]. For example, Dong et al. [27] propose to utilize the attention mechanism and fuse the information provided by both support and query set to generate the classification weights to classify query samples. Xu et al. [29] propose to use the conditional variational autoencoder to generate more representative features, while Dong et al. [28] propose to generate the adversarial images to improve the representation ability of the model.

In contrast to FSL methods that mainly focus on adapting to new classes, we also prioritize the stability of the model.

B. Class-Incremental Learning

Class-incremental learning (CIL) aims to learn new classes without forgetting previously learned classes. However, due to the limitation in using old data, the model's parameters are overwritten by the data of new classes in incremental sessions, which leads to the notorious catastrophic forgetting problem. To address this issue, current CIL methods can be roughly divided into three groups: regularization-based, rehearsal-based, and isolation-based. (1) The regularization-based methods [5], [30], [31], [32] constrain dramatic changes in the model's parameters to resist the catastrophic forgetting problem. For example, Li et al. [5] propose LwF, a representative work that applies knowledge distillation on the output, while Douillard et al. [30] distill the features of each layer of the model to indirectly constrain the parameters' change. (2) The rehearsal-based methods [33], [34], [35], [36] replay old data to recall the memory of the model when learning new classes. To select more representative old data, various data sampling strategies are introduced or designed. focus on designing various data sampling strategies to select representative old data. For example, Rebuffi et al. [33] propose a representative work named iCaRL that leverages the herding strategy to select representative samples of each old classes. Hu et al. [35] propose a curiosity-driven strategy that selects representative samples by the uncertainty and novelty. (3) The isolation-based methods [37], [38] focus on introducing additional parameters for the model to learn new knowledge. For example, Yan et al. [37] propose a method that extends the parameters of each layer for the model to new classes.

Current CIL methods are train-based, and sufficient training samples are available for these works to learn new classes. However, these methods often fail when there are only limited training samples available for new classes. Unlike CIL methods, our proposed method is train-free, which means it does not require additional training during incremental sessions, and can achieve excellent incremental performance with only a few samples.

C. Few-Shot Class-Incremental Learning

Few-shot class-incremental learning (FSCIL) is first proposed by Tao et al. [39] and aims to continually learn new classes using a few samples while not forgetting the old classes. Compared to CIL, FSCIL is more challenging due to only limited number of training samples available in incremental sessions. As a result, this research topic has attracted the attention of many researchers in recent years. The challenges in FSCIL are that scarce training samples make the model suffer from the notorious catastrophic forgetting problem and make the relation measuring challenging in incremental sessions. To mitigate the catastrophic forgetting problem, some works propose to perform knowledge distillation on the relation [40], or the semantic information [41]. Compared with such a strategy, many works [1], [8], [9], [42] have validated that freezing the encoder in incremental sessions is an effective solution. To construct discriminative relations between prototypes and test features, Zhu et al. [42] propose

to utilize the old prototypes to update global prototypes and use an MLP to perform relation measuring. Zhang et al. [1] propose to propagate context information between old and new prototypes to achieve the evolution of the global classifier. Hersche et al. [9] propose to solve this problem by storing old features and replaying them with new features to fine-tune a fully-connected layer to make the model output discriminative representation. Considering that the learning objective of previous methods is inconsistent with the objective in real incremental sessions may compromise the performance, Chi et al. [43] propose a meta-learning-based method that mimics the multi-step incremental setting and constructs pseudo incremental tasks to make the model learn to optimize itself using a few training samples. Zhou et al. [44] argue that previous methods make the updated model similar to the old one unnecessary. To make the model compatible with new classes, they propose to utilize MixUp [45] to squeeze the embedding space of old classes and reserve the squeezed space for new class adaption.

In this paper, we propose a method that improves CEC to the new state-of-the-art performance by utilizing old knowledge and cross-attention mechanism to achieve equitable information gain for classifier weights and test features.

III. Problem Description

Few-shot class-incremental learning aims to learn a classifier in phases to classify all encountered classes, where the classes contained in different phases are disjoint and the training samples for new classes are scarce. Let $\{\mathcal{D}^0, \ldots, \mathcal{D}^i (i > 0)\}$ denote the data streams, where $\mathcal{D}^0$ is termed the base session, $\mathcal{D}^i (i > 0)$ is termed the incremental session. The label spaces of different sessions satisfy $\mathcal{C}^i \cap \mathcal{C}^j = \emptyset (i \neq j)$. Each session $\mathcal{D}^i$ consists of a training set $\mathcal{D}^i_{train}$ and a test set $\mathcal{D}^i_{test}$. Specifically, sufficient annotated samples available in $\mathcal{D}^0_{train}$ while few training samples available in $\mathcal{D}^i_{train} (i > 0)$. For example, in the popular benchmark dataset *mini*ImageNet takes the incremental setting named 5-way-5-shot, which means $\mathcal{D}^i_{train} (i > 0)$ consists of 5 new classes, and each class only consists of 5 training samples. For each session, only the training set of the current session is available for model learning. In contrast, test sets of encountered classes are used to evaluate the model's performance. For example, in session i, only $\mathcal{D}^i_{train}$ is available, while the model is evaluated on $\{\mathcal{D}^0_{test}, \ldots, \mathcal{D}^i_{test}\}$. In this paper, we consider this problem as an incremental relation measuring as CEC [1]. In FSCIL, scarce training samples lead to weak representation for new classes, making this incremental relation measuring problem challenging.

IV. Method

Our proposed method shares the similar learning framework with CEC [1], but deviates CEC from two aspects. The first aspect is that we replace the Adaption Module (AM) of CEC with our proposed Knowledge-guided Relation Refinement Module (KRRM). The second is that we replace the Pseudo Incremental Learning (PIL) of CEC with our proposed Pseudo Incremental relation Refinement Learning (PIRL). To explain our proposed method, namely "Improved Continually Evolved Classifiers (CEC+)", we first provide a brief overview of the CEC [1] in part IV-A, before presenting our proposed method in part IV-B.

A. Continually Evolved Classifiers

The main idea of CEC [1] is to update the old and new classifiers by propagating the context information between them, the main function is achieved by AM which is a Transformer block. The learning framework of CEC consists of three stages, the feature pre-training, the pseudo incremental learning, and the classifier learning stages.

1) Feature Pre-Training: This stage is mainly used to learn the parameters θ_e of the encoder. Let the θ_c denotes the parameters of the fully connected layer, $x \in \mathcal{D}^0_{train}$ denotes the training data, $Y \in \mathcal{C}^0$ denotes ground truth. The θ_e and θ_c is optimized by

$$\theta_e^*, \theta_c^* = \arg\min_{\theta_e, \theta_c} \mathcal{L}_{CE}(P, Y), \tag{1}$$

where P denotes the prediction results and is computed by

$$P = \text{softmax}(s\Phi(\theta_c, f(x))). \tag{2}$$

Here, s is the scale factor used to control the peakiness of softmax distribution [46], Φ denotes the cosine classifier and $\Phi(a, b) = \frac{a \cdot b}{||a||_2 ||b||_2}$, $f(x)$ refers to the embedding feature of x.

2) Pseudo Incremental Learning: The PIL is used to learn the parameters of the AM. PIL mimics the incremental setting and constructs a series of pseudo-incremental tasks using data sampled from the base session. Concretely, several classes from the base session are first sampled as the pseudo base classes. Then, the data of these sampled classes is rotated to form the pseudo new classes. Notably, both the pseudo base classes and pseudo new classes have a support set and a query set denoted as $\{S_o, Q_o\}$ and $\{S_n, Q_n\}$, respectively. Next, the prototypes μ_o and μ_n are computed based on the class-wise mean features of S_o and S_n, and used to initialize the old and new classifiers. After that, the AM is applied to update μ_o and μ_n. Finally, PIL uses the updated classifier to make predictions for Q_o and Q_n, and computes the loss to optimize the AM.

3) Classifier Learning: After finishing the pseudo incremental learning, CEC freezes the model to mitigate the catastrophic forgetting problem. In each new incremental session, CEC first uses the prototypes given by the class-wise mean features of training data of new coming classes to initialize a new classifier. Then, CEC applies the trained AM and takes the parameters of old and new classifiers as input to update all classifiers. In the inference stage, the updated classifiers is deployed to make predictions for each test sample.

B. Improved Continually Evolved Classifiers

In CEC+, we also adopt the three-stage learning framework to perform few-shot class-incremental learning. Unlike CEC, our proposed method update both the classifier initialized by prototypes and test features to construct more discriminative relations. The main function is achieved by the Knowledge-guided Relation Refinement Module (KRRM). Furthermore,

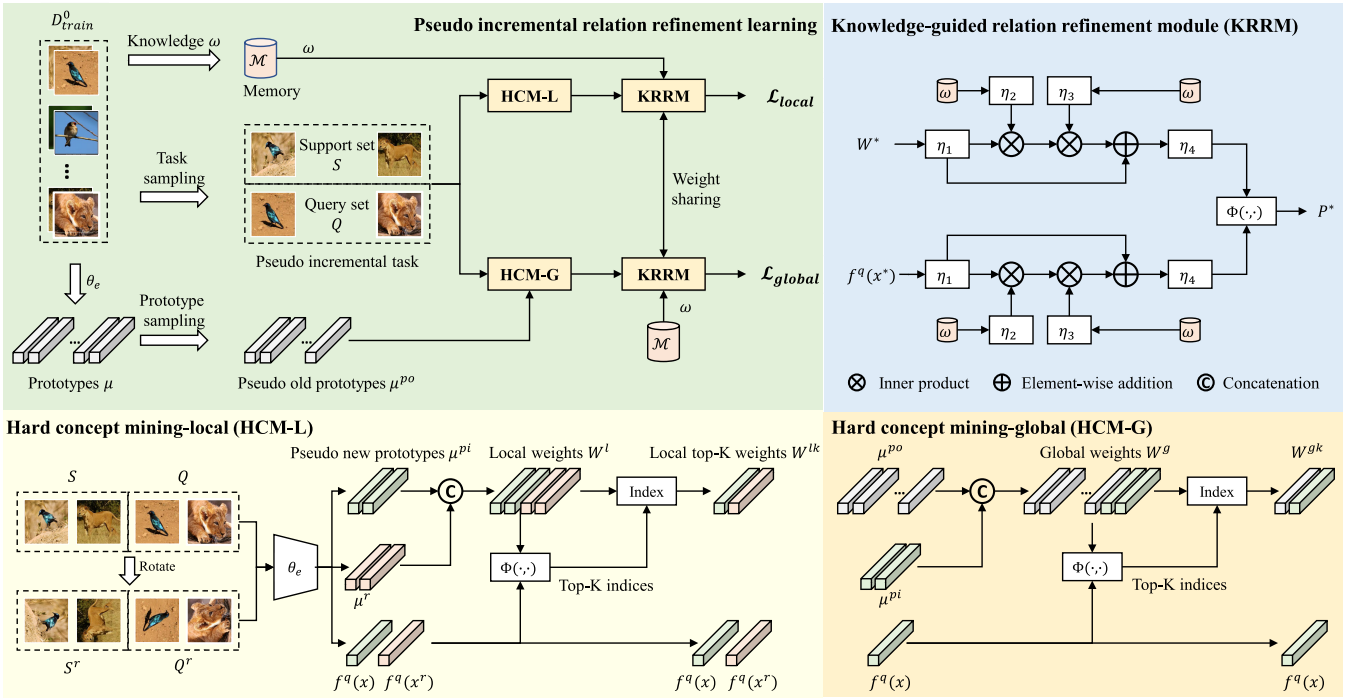


Fig. 2. The learning scheme of our proposed method, where η_* refers to linear layer, the knowledge ω is given by the prototypes of old data. Our proposed training scheme mimics the real inference process using hard concept mining strategy to make the KRRM learn to refine the relation between prototypes and query features by knowledge propagation.

we extend the PIL and propose the Pseudo Incremental relation Refinement Learning (PIRL). We leave the detailed descriptions about the KRRM and PIRL in part IV-C and IV-D, respectively.

C. Knowledge-Guided Relation Refinement Module

The knowledge-guided relation refinement module (KRRM) is used to refine the relations between classifier weights and test features, which is achieved by knowledge propagation. Concretely, let ζ represent the classifier weights or test features for the sake of the following descriptions. In the knowledge propagation process, to mine valuable information from ω, the relation e between ζ and the knowledge ω represented by the prototypes of old classes is first computed by

$$e = \text{softmax}(\frac{\eta_1(\zeta) \cdot T(\eta_2(\omega))}{\sqrt{d}}), \tag{3}$$

where η_1 and η_2 represent the linear transformation functions, $\cdot$ represents the inner product, and T represents the transpose operation. Then, the relation e is used to weight the ω to suppress the unimportant and strengthen the important information of ω for ζ, the formula is given by

$$\omega' = e \cdot \eta_3(\omega), \tag{4}$$

where ω' refers to the weighted knowledge, η_3 represents the linear transformation function. In the end, based on the computed ω', ζ is augmented by

$$\hat{\zeta} = \eta_4(\zeta + \omega'), \tag{5}$$

where η_4 represents the linear transformation function. After finishing the knowledge propagation process, the relation between the updated classifier weights and test features is measured by using Eq. (2).

Why the knowledge-guided relation refinement module works. KRRM adaptively enhances the target by leveraging the relation between the target and old knowledge. Specifically, for new classes, this mechanism effectively enhances the similarity information in same-class classifier weights and test features, which is necessary for the model to make more discriminative predictions and thus improves the model's plasticity to new classes. Although introducing a significant amount of old knowledge into new classes may affect the model's classification ability on old classes to some extent, subsequent experiments demonstrate that the benefits of KRRM on model's plasticity outweigh its impact on model's stability.

D. Pseudo Incremental Relation Refinement Learning

To learn the KRRM's parameters, it's optimal to train it under the incremental setting. However, the scarcity of training samples and the limitation of using old data make it difficult to perform this learning process in incremental sessions. To solve this problem, as shown in Figure 2, we mimic the incremental setting and propose the pseudo incremental relation measuring learning (PIRL) to borrow the treasure from the base session. Concretely, PIRL consists of four steps, *knowledge construction*, *initial learning task construction*, *hard concept mining*, and *module optimization*.

1) Knowledge Construction: To prepare for future knowledge propagation, the knowledge ω is first extracted from D^0_{train} by using the trained encoder $f(;\theta_e)$. To reduce the memory consumption, we use the combination of the mean

Algorithm 1 Pseudo Incremental Relation Refinement Learning

Require: Training data of base session D^0_{train}, encoder $f(;\theta_e)$, a randomly initialized KRRM. μ refers to the prototypes of all classes contained in D^0_{train}.
Ensure: A trained KRRM.

while not done **do**
  $\omega \leftarrow$ Extract knowledge from D^0_{train} using $f(;\theta_e)$
  $S, Q \leftarrow$ Sample the support set and the query set to construct the pseudo incremental task
  $\mu^{po} \leftarrow$ Sample pseudo base prototypes from μ
  $S^r, Q^r \leftarrow$ Rotate S and Q
  $f^s(x), f^q(x), f^s(x^r), f^q(x^r) \leftarrow$ Encode S, Q, S^r, Q^r using $f(;\theta_e)$
  $\mu^{pi}, \mu^r \leftarrow$ Compute the mean features of $f^s(x)$ and $f^s(x^r)$
  $W^l \leftarrow$ Get the local classifier weights by concatenating μ^{pi}, μ^r
  $W^{lk} \leftarrow$ Use W^l and Eq. (2) to classify Q and Q^r, then get the classifier weights of top-K predictions
  $\{\omega, f^q(x), f^q(x^r), W^{lk}\} \leftarrow$ Construct the local hard concept task
  $W^g \leftarrow$ Get the global classifier weights by concatenating μ^{po}, μ^{pi}
  $W^{gk} \leftarrow$ Use W^g and Eq. (2) to classify Q, then get the classifier weights of top-K predictions
  $\{\omega, f^q(x), W^{gk}\} \leftarrow$ Construct the global hard concept task
  $\mathcal{L}_{local} \leftarrow$ Compute the loss relevant to $\{\omega, f^q(x), f^q(x^r), W^{lk}\}$
  $\mathcal{L}_{global} \leftarrow$ Compute the loss relevant to $\{\omega, f^q(x), W^{gk}\}$
  $\mathcal{L} \leftarrow$ Get the total objective by weighting $\mathcal{L}_{local}$ and $\mathcal{L}_{global}$
  optimize KRRM with $\mathcal{L}$ and SGD
end while

feature of each class contained in D^0_{train} to represent the ω, though other forms, such as some representative features or trainable prototypes can represent ω.

2) Initial Learning Task Construction: In this step, we first sample several classes from $\mathcal{C}^0$ as the pseudo new classes. Then, for each sampled pseudo new class, we sample a few data from $\mathcal{D}^0_{train}$ to construct the support set S and the query set Q, respectively. Because the form of $\{S, Q\}$ is similar to the incremental session, where the S can be treated as the training set and Q can be test set in real incremental session, we call $\{S, Q\}$ the pseudo incremental task. Next, for each pseudo incremental task, the prototypes of other classes are sampled and treated as the pseudo base prototypes μ^{po}. In the end, PIRL uses the combination of $\{\omega, S, Q, \mu^{po}\}$ to construct a series of initial learning tasks.

3) Hard Concept Mining: Because the data contained in each initial learning task basically belongs to the base session, the pre-trained model can classify these tasks well. Therefore, directly using these initial learning tasks to optimize the KRRM is helpless. To solve this problem, we design the hard concept mining strategy to mine hard concept tasks from local and global perspectives.

Hard Concept Mining-Local (HCM-L): Following CEC [1], we first rotate the data of each pseudo incremental task. Let S^r and Q^r denote the rotated S and Q, respectively. Then, we obtain the embedding features $f^s(x)$, $f^q(x)$, $f^s(x^r)$, $f^q(x^r)$ by employing the encoder $f(;\theta_e)$ to encode the data of S, Q, S^r, and Q^r, respectively. Next, we compute the mean feature of each class of $f^s(x)$ and treat them as the pseudo new prototypes μ^{pi}. In the meantime, we compute the mean features μ^r of $f^s(x^r)$. After that, the local classifier weights W^l is obtained by concatenating μ^{pi} and μ^r and used to make prediction for $f^q(x)$ and $f^q(x^r)$ with Eq. (2). In the end, the top-K classifier weights W^{lk} which correspond to the top-K predictions are indexed from μ^l and used to construct the local hard concept task $\{\omega, f^q(x), f^q(x^r), W^{lk}\}$.

Hard Concept Mining-Global (HCM-G): We first get the global classifier weights W^g by concatenating μ^{pi} and the pseudo base prototypes μ^{po}. Then, W^g is used to make prediction for $f^q(x)$. Next, the top-K classifier weights W^{gk} which correspond to the top-K predictions are indexed from W^g. In the end, the combination of $\{\omega, f^q(x), W^{gk}\}$ is used to construct the global hard concept task.

4) Module Optimization: To improve the model's plasticity, we utilize the constructed local hard concept tasks to optimize the KRRM. Concretely, based on $\{\omega, f^q(x), f^q(x^r), W^{lk}\}$, the KRRM first uses ω and Eq. (3), (4), (5) to augment $f^q(x)$, $f^q(x^r)$, and W^{lk}, respectively. The corresponding augmented results we denote as $\hat{f}^q(x)$, $\hat{f}^q(x^r)$, and $\hat{W}^{lk}$, respectively. Then, the relation P^q between $\hat{f}^q(x)$ and $\hat{W}^{lk}$ is measured by using Eq. 2. In the meanwhile, the relation P^r between $\hat{f}^q(x^r)$ and $\hat{W}^{lk}$ is measured by using Eq. 2. With computed P^q and P^r, the loss $\mathcal{L}_{local}$ based on local hard concept tasks is given by

$$\mathcal{L}_{local} = \mathcal{L}_{CE}(P^q, Y^q) + \mathcal{L}_{CE}(P^r, Y^r), \tag{6}$$

where $\mathcal{L}_{CE}$ refers to the cross-entropy loss function, Y^q and Y^r refer to the relative label of the data of the query set and the rotated query set, respectively.

Optimizing the model's plasticity alone may affect its stability. To address this issue, we utilize the constructed global hard concept tasks to further optimize the KRRM to mitigate the destabilizing effects. Concretely, based on $\{\omega, f^q(x), W^{gk}\}$, the KRRM first uses ω and Eq. (3), (4), (5) to augment $f^q(x)$, and W^{gk}, respectively. The corresponding augmented results we denote as $\hat{f}^q(x)$ and $\hat{W}^{gk}$, respectively. Then, the relation P^q between $\hat{f}^q(x)$ and $\hat{W}^{gk}$ is measured by using Eq. 2. With computed P^q and Eq. 6, the loss $\mathcal{L}_{global}$ based on global hard concept tasks is computed.

Overall, the KRRM is optimized by

$$\mathcal{L} = \sum \lambda_1 \mathcal{L}_{local} + \lambda_2 \mathcal{L}_{global}, \tag{7}$$

where λ_1 and λ_2 are used to control the influences of $\mathcal{L}_{local}$ and $\mathcal{L}_{global}$.

V. Experiments

A. Datasets

CIFAR100. CIFAR100 [48] consists of 60,000 images from 100 classes, where each class consists of 500 training images and 100 test images. Following [39], we split this dataset into 60 base classes and 40 incremental classes, where the 40 incremental classes are equally divided into eight incremental sessions. Each incremental session takes the setting 5-way-5-shot, which means each incremental session consists of 5 classes, and each class consists of 5 support images.

***mini*ImageNet.** *mini*ImageNet is the subset of ImageNet [49]. This dataset consists of 100 classes, where each class consists of 500 training images and 100 test images. Following [39], we split this dataset into 60 base classes and 40 incremental classes, where the 40 incremental classes are equally divided into eight incremental sessions. Each incremental session takes the setting 5-way-5-shot.

Caltech-UCSD Birds-200-2011. CUB200 [50] is a fine-grained dataset containing 11,788 images from 200 classes, where each class consists of approximately 30 training images and 30 test images. Following [39], we split this dataset into 100 base classes and 100 incremental classes, where the 100 incremental classes are equally divided into ten incremental sessions. Each incremental session takes the setting 10-way-5-shot.

B. Implementation Details

Our work is implemented using PyTorch [51] library, we employ ResNet18 as the backbone for all benchmark datasets. Following CEC [1], we first pretrain the model with the standard training paradigm, then perform pseudo incremental relation learning to learn the parameter of the knowledge-guided relation refinement module.

1) Pretraining: On CUB200, we pre-train the model for 50 epoch with a batch size of 128. We choose SGD as the optimizer with a learning rate of 0.03, a weight decay of 0.0001, and a momentum of 0.9. We decay the learning rate with a factor of 0.1 per 10 epochs. On *mini*ImageNet and CIFAR100, we pre-train the model 100 epoch with a batch size of 64. We choose SGD as the optimizer with a learning rate of 0.1, a weight decay of 0.0005, and a momentum of 0.9. We decay the learning rate with a factor of 0.1 per 40 epochs. Following Zhu et al. [42], random resized crop, random horizontal flip, and color jitter are used to augment the training data.

2) Pseudo Incremental Relation Refinement Learning: We set the max training epoch to 50 and randomly sample 200 pseudo incremental tasks from $\mathcal{D}^0_{train}$ using setting 25-way-1-shot in each epoch, *i.e.* , we randomly sample 25 classes, and sample 1 sample for each sampled class to construct the support set and 15 samples for each sampled class to construct the query set. We adopt the SGD as the optimizer with an initial learning rate of 0.0002 and a weight decay of 0.0001. We set both λ_1 and λ_2 to 1.

C. Baselines

To validate the effectiveness of our proposed method, we compare our proposed method with some classical CIL methods (iCaRL* [33], EEIL* [47], and NCM* [46]) and previous FSCIL methods (TOPIC [39], SPPR [42], CEC [1], F2M [8], C-FSCIL [9], MetaFSCIL [43] and FACT [44]).

The descriptions of these methods are presented as follows:

- **iCaRL** selects parts of representative old data by the herding strategy [52] and replays the selected old data in incremental sessions to mitigate the catastrophic forgetting problems.
- **EEIL** extends iCaRL and proposes a balanced fine-tuning strategy the samples equal number of training samples of old and new classes to further finetune the model.
- **NCM** learns a unified classifier to balance the bias between old and new data by incorporating cosine normalization, less-forget constraint, and inter-class separation.
- **TOPIC** constrains the topology of feature space to mitigate the catastrophic forgetting problem.
- **SPPR** measures the relations between global prototypes and test features by a MLP, where the global prototypes is given by concatenating old and new prototypes and updated by the relations with old prototypes.
- **CEC** propagates context information between old and new classifiers initialized by prototypes to achieve the evolution of classifiers.
- **F2M** fine-tunes the model in incremental sessions within the base training objective's flat local minima found by adding random noise to the encoder's parameters to achieve the balance between stability and plasticity.
- **C-FSCIL** selects parts of old features and replay them with new data to finetune the fully-connected layer to update the outputs of the frozen encoder in the incremental sessions.
- **MetaFSCIL** samples the pseudo incremental sequence instead of the single pseudo incremental task to optimize the model.
- **FACT** utilizes the MixUp [45] to squeeze the embedding space of old classes and reserve for new classes.

D. Comparison Results

The results of the baseline methods and our proposed method on three benchmark datasets are shown in Table I, we can see that

- On three benchmark datasets for FSCIL, the classical class-incremental learning methods iCaRL, EEIL, and NCM show more significant performance degradation than FSCIL methods, such as TOPIC, as the incremental process proceeds.
- On three benchmark datasets, our proposed method achieves the highest accuracy on each session compared to other FSCIL methods.
- On CIFAR100 and *mini*ImageNet, it can be seen from the results of SPPR and SPPR† that method with high performance on the first session seems to have a relatively larger performance degradation on the following sessions. Surprisingly, our proposed method achieves the highest performance in the first session and relatively smaller performance degradation in the following sessions. Partic-

TABLE I

COMPARISON WITH STATE-OF-THE-ART METHODS ON CIFAR100, *mini*IMAGENET AND CUB200, † DENOTES OUR REPRODUCED RESULT, ∗ INDICATES RESULTS COPIED FROM TOPIC. SINCE THE BEST PERFORMANCE OF SPPR ON CUB200 IS THE RESULT SHOWN IN THEIR PAPER, WE USE THEIR PUBLICLY AVAILABLE RESULT DIRECTLY. OUR PROPOSED METHOD ACHIEVES THE BEST PERFORMANCE ON ALMOST ALL SESSIONS OF EACH BENCHMARK DATASET

Method	Backbone	sessions(CIFAR100)								
		0	1	2	3	4	5	6	7	8
C-FSCIL[9]	ResNet-12	77.47	72.40	67.47	63.25	59.84	56.95	54.42	52.47	50.47
NCM* [46]	ResNet-20	64.10	53.05	43.96	36.97	31.61	26.73	21.23	16.78	13.54
iCaRL* [33]		64.10	53.28	41.69	34.13	27.93	25.06	20.41	15.48	13.73
EEIL* [47]		64.10	53.11	43.71	35.15	28.96	24.98	21.01	17.26	15.85
TOPIC[39]		64.10	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37
MetaFSCIL[43]		74.50	70.10	66.84	62.77	59.48	56.52	54.36	52.56	49.97
FACT[44]		74.60	72.09	67.56	63.52	61.38	58.36	56.28	54.24	52.10
SPPR[42]	ResNet-18	63.97	65.86	61.31	57.60	53.39	50.93	48.27	45.36	43.32
SPPR†[42]		77.64	72.80	67.36	63.20	59.10	55.78	52.56	50.01	47.52
F2M[8]		71.45	68.10	64.43	60.80	57.76	55.26	53.53	51.57	49.35
CEC†[1]		**81.30**	76.37	72.17	67.84	64.78	61.78	59.54	57.37	55.02
Ours		81.25	**77.23**	**73.30**	**69.41**	**66.69**	**63.93**	**62.16**	**59.62**	**57.41**

Method	Backbone	sessions(*mini*ImageNet)								
		0	1	2	3	4	5	6	7	8
C-FSCIL[9]	ResNet-12	76.40	71.14	66.46	63.29	60.42	57.46	54.78	53.11	51.41
NCM* [46]	ResNet-18	61.31	47.80	39.31	31.91	25.68	21.35	18.67	17.24	14.17
iCaRL* [33]		61.31	46.32	42.94	37.63	30.49	24.00	20.89	18.80	17.21
EEIL* [47]		61.31	46.58	44.00	37.29	33.14	27.12	24.10	21.57	19.58
TOPIC[39]		61.31	50.09	45.17	41.16	37.48	35.52	32.19	29.46	24.42
SPPR[42]		61.45	63.80	59.53	55.53	52.50	49.60	46.69	43.79	41.92
SPPR†[42]		79.28	74.10	68.69	64.27	60.28	56.79	53.68	50.83	48.39
F2M[8]		72.05	67.47	63.16	59.70	56.71	53.77	51.11	49.21	47.84
MetaFSCIL[43]		72.04	67.94	63.77	60.29	57.58	55.16	52.90	50.79	49.19
FACT[44]		72.56	69.63	66.38	62.77	60.60	57.33	54.34	52.16	50.49
CEC†[1]		82.25	76.91	72.57	69.17	66.23	63.32	60.37	58.28	56.90
Ours		**82.65**	**77.82**	**73.59**	**70.24**	**67.74**	**64.82**	**61.91**	**59.96**	**58.35**

Method	sessions(CUB200, w ResNet-18)										
	0	1	2	3	4	5	6	7	8	9	10
NCM* [46]	68.68	57.12	44.21	28.78	26.71	25.66	24.62	21.52	20.12	20.06	19.87
iCaRL* [33]	68.68	52.65	48.61	44.16	36.62	29.52	27.83	26.26	24.01	23.89	21.16
EEIL* [47]	68.68	53.63	47.91	44.20	36.30	27.46	25.93	24.70	23.95	24.13	22.11
TOPIC[39]	68.68	62.49	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.28
SPPR[42]	68.68	61.85	57.43	52.68	50.19	46.88	44.65	43.07	40.17	39.63	37.33
MetaFSCIL[43]	75.90	72.41	68.78	64.78	62.96	59.99	58.30	56.85	54.78	53.82	52.64
F2M[8]	77.13	73.92	70.27	66.37	64.34	61.69	60.52	59.38	57.15	56.94	55.89
FACT[44]	75.90	73.23	70.84	66.13	65.56	62.15	61.74	59.83	58.41	57.89	56.94
CEC†[1]	78.12	74.52	71.71	68.00	66.44	63.82	62.67	61.52	59.71	59.41	58.41
Ours	**79.46**	**76.11**	**73.12**	**69.31**	**67.97**	**65.86**	**64.50**	**63.83**	**62.20**	**62.00**	**60.97**

ularly, on CIFAR100, the accuracy of last session of our proposed method surpasses SPPR† and CEC† by **9.89**% and **2.39**% respectively. On *mini*ImageNet, the accuracy of the last session of our proposed method surpasses SPPR† and CEC† by **9.96**% and **1.45**% respectively.

- On CUB200, our proposed method outperforms the second best method CEC† on each session, where the accuracy of the last session of our proposed method surpasses CEC† by **2.56**%.

The results demonstrate that our proposed method sets new state-of-the-art performance.

E. Ablation Study

Our proposed method relies on the knowledge-guided relation refinement module (KRRM) to refine the relation between the classifier weights and test features and pseudo incremental relation refinement learning (PIRL) to mine hard concept tasks from local and global perspectives to learn the parameters of KRRM. To validate the effectiveness of KRRM and PIRL, we use the performance given by the pre-trained model as the baseline and conduct several ablation studies on CUB200. From Table II, we can see that

- Compared to the baseline, though using using only hard concept mining-global (HCM-G) slightly drops the performance in the first five sessions, it improves the

TABLE II

ABLATION STUDIES ON CUB-200-2011 USING 10-WAY-5-SHOT WITH RESNET18, WHERE PIRL IS THE PSEUDO INCREMENTAL RELATION REFINEMENT LEARNING, KRRM IS THE KNOWLEDGE-GUIDED RELATION REFINEMENT MODULE, HCM-G/HCM-L REFERS TO THE GLOBAL/LOCAL HARD CONCEPT MINING

PIRL		KRRM	sessions										
HCM-G	HCM-L		0	1	2	3	4	5	6	7	8	9	10
			79.05	75.51	72.80	68.81	66.91	64.51	62.85	61.63	59.78	59.15	58.18
√			78.88	75.38	72.32	68.49	66.73	64.52	63.21	62.01	60.50	60.12	58.99
	√		79.23	75.68	72.97	68.82	67.04	64.65	63.21	62.27	60.69	60.53	59.41
√	√		79.32	76.09	73.08	69.20	67.75	65.45	63.89	63.06	61.21	61.09	60.08
		√	**79.48**	76.03	72.94	69.17	67.40	64.95	63.46	62.32	60.64	60.07	58.99
√		√	79.12	75.44	72.32	68.59	67.30	65.45	64.00	62.97	61.41	60.99	59.82
	√	√	79.45	76.01	**73.16**	68.91	67.24	64.75	63.42	62.36	60.95	60.68	59.60
√	√	√	79.46	**76.11**	73.12	**69.31**	**67.97**	**65.86**	**64.50**	**63.83**	**62.20**	**62.00**	**60.97**

performance in the last six sessions by a relatively larger margin. The primary reason is that applying HCM-G enhances the model's plasticity to new classes at the expense of stability on old classes. In the initial few sessions, the benefit brought by HCM-G is suppressed due to the dominance of old class test data. However, as the learning process progressed and more new class test data is introduced, the advantage of HCM-G gradually became apparent, leading to its superior performance over the baseline in the last six sessions.

- Compared to the baseline, using using only hard concept mining-local (HCM-L) improves the performance on each session. Particularly, the accuracy of the last session given by HCM-L outperforms the baseline by **1.23**%.
- Compared to using only HCM-G or HCM-L, the combination of HCM-G and HCM-L achieves better performance across all sessions. Particularly, the accuracy of the last session given by the combination of HCM-G and HCM-L outperforms that given by HCM-L by **0.67**%.
- Compared to the baseline, using only KRRM improves the performance on the first session by **0.43**% and the performance on the last session by **0.81**%.
- Compared to using only the KRRM, though using HCM-G to train the KRRM slightly drops the performance on the first five sessions, it improves the performance on the last six sessions by a relatively larger margin. Particularly, the accuracy of the last session given by the combination of KRRM and HCM-G outperforms that given by KRRM by **0.83**%.
- Compared to using only the KRRM, though using HCM-L to train the KRRM slightly drops the performance on the first seven sessions, it improves the performance on the last four sessions by a relatively larger margin. Particularly, the accuracy of the last session given by the combination of KRRM and HCM-L outperforms that given by KRRM by **0.61**%.
- Compared to previous configurations, using PIRL, consisting of HCM-L and HCM-G, to learn the parameters of the KRRM almost achieves the highest performance on each incremental session.

In summary, the results demonstrate the effectiveness of PIRL and KRRM.

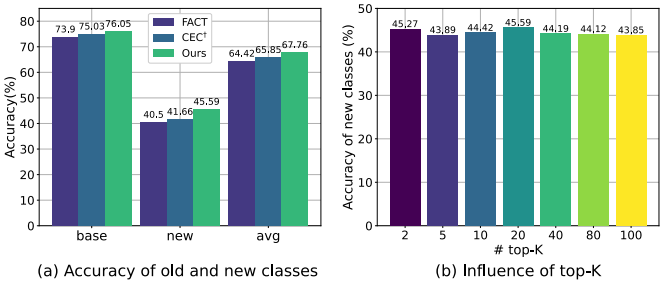


Fig. 3. Analysis on (a) different performance measures, and (b) the influence of top-K.

F. Analysis

1) Performance Measure: To explore the ability of new class adaption and forgetting resistance of our proposed method, we report the accuracy of base and new classes as well as the average accuracy on CUB200 and compare with two previous state-of-the-art methods, CEC† [1] and FACT [44]. As shown in Figure 3(a), our proposed method outperforms FACT by a margin of **2.15**% and CEC by a margin of **1.02**% on base classes. On new classes, our proposed method outperforms FACT by a margin of **5.09**% and CEC by a margin of **3.93**%. As for the average accuracy, our proposed method outperforms FACT by a margin of **3.34**% and CEC by a margin of **1.91**%. The results demonstrate that our proposed method has better new class adaption and forgetting resistance abilities than CEC and FACT.

2) Top-K: In this paper, we use the top-K setting to mine hard concept tasks. To explore the influence of top-K, we report the accuracy of new classes on CUB200 given by different top-K settings. As we can see from Figure 3(b), when top-K is set to 2, the accuracy is **45.27**%. However, the accuracy drops significantly by **1.38**% when we set top-K to 5. Interestingly, the model's performance keeps improving when we increase the value of top-K from 5 to 20 and achieves the highest accuracy **45.59**% when the value of top-K is set to 20. In contrast, the model's performance keeps decreasing when we increase the value of top-K from 20 to 100 (refers to not using the top-K setting) and gets the lowest accuracy **43.85**% when the value of top-K is set to 100. In summary, using top-K can improve the model's performance in new classes.

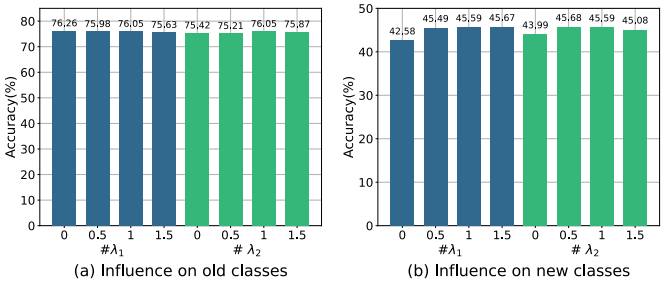


Fig. 4. The influence of $\mathcal{L}_{local}$ and $\mathcal{L}_{global}$ on old and new classes.

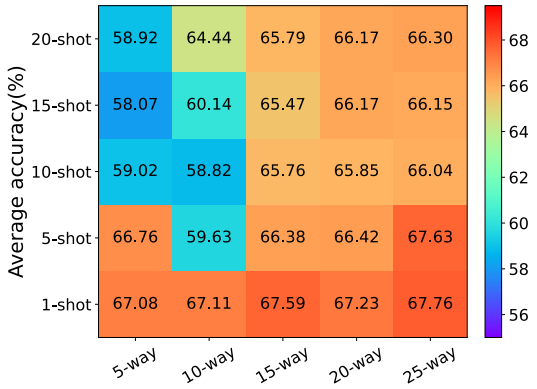


Fig. 5. The influence of task sampling setting on CUB200. Our proposed method prefers to a large way and a small shot.

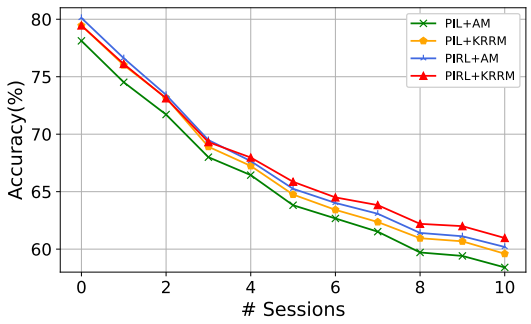


Fig. 6. Module comparison with CEC. PIL (Pseudo Incremental Learning) and AM (Adaption Module) are proposed by CEC. PIRL (Pseudo Incremental relation Refinement Learning) and KRRM (Knowledge-guided Relation Refinement Module) are proposed by us. Each module proposed by us is more effective than CEC.

TABLE III

THE INFLUENCE OF DIFFERENT FORMS OF OLD KNOWLEDGE, WHERE F-PROTOTYPE AND T-PROTOTYPE REPRESENT FIXED AND TRAINABLE PROTOTYPES, RESPECTIVELY

Knowledge form	base	new	avg	budget(KB)
f-prototype	76.05	**45.59**	67.76	≈200
t-prototype	**77.13**	43.59	67.72	≈200
feature	76.40	44.30	**67.84**	≈1000

Particularly, setting the value of top-K to 20 is an optimal choice.

3) The Influences of $\mathcal{L}_{local}$ and $\mathcal{L}_{global}$: To further explore the influence of $\mathcal{L}_{local}$ and $\mathcal{L}_{global}$ on old and new classes on CUB200, we first fix λ_2 to 1 and change λ_1 among $\{0, 0.5, 1, 1.5\}$, then we fix the optimal λ_1 and change λ_2 among $\{0, 0.5, 1, 1.5\}$. As we can see from Figure 4, increasing λ_1 to a relatively larger value significantly improves the performance on new classes but slightly decreases the performance on old classes. When λ_2 is increased to a relatively larger value, we observe an improvement in performance for both old and new classes. The results validate that $\mathcal{L}_{local}$ improves the model's stability at the expense of the model's stability while $\mathcal{L}_{global}$ can mitigate the influence of the $\mathcal{L}_{local}$ on the model's stability.

4) Sampling Setting: To explore the influence of pseudo incremental task sampling setting on average accuracy, we experiment with different combinations of the number of ways and shots. As we can see from Figure 5, when we fix the number of ways, we find that setting the number of shots to a small value achieves better performance than setting the number of shots to a large value. In contrast, when fixing the number of shots, we find that using a large number of ways results in better performance than using a small number of ways. Particularly, using the setting of 5-way-20-shot to sample pseudo incremental tasks yields the worst performance, while using the setting of 25-way-1-shot to sample pseudo incremental tasks leads to the best performance. The main reason may step from that larger way makes the task harder compared to smaller way, and a small shot provides limited prior information, which forces the KRRM to learn stronger relation refinement ability compared to larger shot.

5) Module Comparison With CEC: In this paper, we boost CEC from two aspects. Firstly, we improve the AM and propose the Knowledge-guided Relation Refinement Module (KRRM) to update the classifier weights and test features by cross-attention mechanism. Secondly, we extend the PIL and propose the Pseudo Incremental relation Refinement Learning (PIRL) using the hard concept mining strategy to mine hard concept task. To validate whether both of them are more effective than CEC, we use the combination of PIL and AM as the baseline. As shown in Figure 6, compared with to the baseline, the combination of PIL and our proposed KRRM achieves better performance in each session. Moreover, using our proposed PIRL to learn the parameters of AM also achieves better performance in each session compared to the baseline. Although the performance of the combination of PIRL and KRRM drops slightly in the first four sessions compared to the combination of PIRL and AM, it exhibits a relatively larger improvement in the following sessions. Overall, these results demonstrate that our proposed KRRM and PIRL are more effective than AM and PIL proposed by CEC.

6) Knowledge Form: To investigate the influence of different knowledge forms, we change the knowledge form among fixed prototypes, trainable prototypes, and features, where the feature selection strategy is referenced to [33], and the number of selected features is is set to 5 as in [40]. As we can see from Table III, using the form of fixed prototypes achieves the highest accuracy of **45.59**% on new classes. Compared with the form of fixed prototypes, using trainable prototypes achieves better performance on the base classes but drops

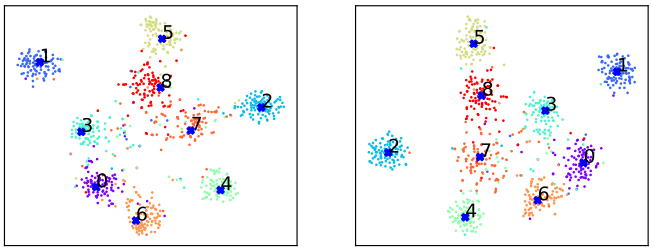


Fig. 7. t-SNE [53] visualization of data embeddings and classifier weights updated by CEC and our proposed method on CIFAR100, where five old classes (0-4) and four new classes (5-8) are randomly selected. Compared to CEC, our proposed method constructs more discriminative relations between the classifier weights and test features.

the performance on new classes. Compared with the form of fixed prototypes or trainable prototypes, representing the old knowledge in the form of features consumes more memory but can achieve a balance between the performance given by the fixed prototypes and trainable prototypes.

7) Visualization of Adaption: To further observe our proposed method's new class adaption ability, we use t-SNE [53] to embed test features and classifier weights updated by our proposed method and CEC to low dimensions and plot. The results in Figure 7. From Figure 7(a), we can observe that the classifier weights updated by CEC for some new classes, such as class 6 and class 8, are located far away from the corresponding clusters composed of the test features. This infers to that the relations established by CEC may only offer discriminative information for some of the new classes. In contrast, the relative positions of classifier weights and test features updated by our proposed method are more harmonious, as shown in Figure 7(b). The results demonstrate that our proposed method achieves better plasticity than CEC.

VI. Conclusion

In this paper, we propose the Improved Continually Evolved Classifiers (CEC+) to boost CEC by addressing its two shortcomings: 1) inequitable information gains between classifier weights and test features, and 2) inefficient learning task construction strategy. To address the first issue, we propose a Knowledge-guided Relation Refinement Module (KRRM), which updates both the classifier weights and test features through knowledge propagation. To tackle the second issue, we propose a Pseudo Incremental relation Refinement Learning (PIRL) scheme, which constructs hard concept tasks by combining query features and corresponding Top-K classifier weights. Comprehensive experiments on three widely used few-shot class-incremental learning benchmark datasets demonstrate that our proposed approach achieves state-of-the-art performance, and the proposed modules are more effective than those of CEC.

Practical incremental learning scenarios are often complex and diverse. Alongside the challenge of insufficient samples for new classes, dirty data may also exist in the incoming data, such as distorted data, which have a severe impact on the model's plasticity and stability. Relying on humans to remove such dirty data will consumes substantial time and effort. Therefore, designing an incremental learning method capable of achieving continual learning with few samples and dirty data is crucial for efficient model evolution. In recent years, some researchers have proposed effective methods [54], [55] for incremental image quality assessment, producing quality assessment results close to human-level, which provides a good foundation for cleaning dirty data. Hence, in future work, we will explore how to integrate existing incremental image quality assessment methods with the approach proposed in this paper to achieve efficient model evolution.

References

[1] C. Zhang, N. Song, G. Lin, Y. Zheng, P. Pan, and Y. Xu, "Few-shot incremental learning with continually evolved classifiers," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 12450–12459.

[2] X. Li, S. Lai, and X. Qian, "DBCFace: Towards pure convolutional neural network face detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 1792–1804, Apr. 2022.

[3] H. Jin, S. Lai, and X. Qian, "Occlusion-sensitive person re-identification via attribute-based shift attention," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 4, pp. 2170–2185, Apr. 2022.

[4] C. Liu, Y. Liang, Y. Xue, X. Qian, and J. Fu, "Food and ingredient joint learning for fine-grained recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 6, pp. 2480–2493, Jun. 2021.

[5] Z. Li and D. Hoiem, "Learning without forgetting," in *Proc. ECCV*, 2016, pp. 614–629.

[6] Z. Wang et al., "Learning to prompt for continual learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 139–149.

[7] Z. Wang et al., "DualPrompt: Complementary prompting for rehearsal-free continual learning," in *Proc. ECCV*, 2022, pp. 631–648.

[8] G. Shi, J. Chen, W. Zhang, L. M. Zhan, and X. M. Wu, "Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima," in *Proc. NeurIPS*, vol. 3, 2021, pp. 6747–6761.

[9] M. Hersche, G. Karunaratne, G. Cherubini, L. Benini, A. Sebastian, and A. Rahimi, "Constrained few-shot class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9047–9057.

[10] O. Vinyals, C. Blundell, T. Lillicrap, and D. Wierstra, "Matching networks for one shot learning," in *Proc. NIPS*, vol. 29, 2016, pp. 1–9.

[11] J. Wang and Y. Zhai, "Prototypical Siamese networks for few-shot learning," in *Proc. IEEE 10th Int. Conf. Electron. Inf. Emergency Commun. (ICEIEC)*, Jul. 2020, pp. 178–181.

[12] W. Jiang, K. Huang, J. Geng, and X. Deng, "Multi-scale metric learning for few-shot learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 3, pp. 1091–1102, Mar. 2021.

[13] H. Zhu and P. Koniusz, "EASE: Unsupervised discriminant subspace learning for transductive few-shot learning," in *Proc. CVPR*, 2022, pp. 9078–9088.

[14] F. Sung, Y. Yang, L. Zhang, T. Xiang, P. H. Torr, and T. M. Hospedales, "Learning to compare: Relation network for few-shot learning," in *Proc. CVPR*, 2018, pp. 1199–1208.

[15] Z. Wu, Y. Li, L. Guo, and K. Jia, "Parn: Position-aware relation networks for few-shot learning," in *Proc. ICCV*, 2019, pp. 6659–6667.

[16] C. Cao and Y. Zhang, "Learning to compare relation: Semantic alignment for few-shot learning," *IEEE Trans. Image Process.*, vol. 31, pp. 1462–1474, 2022.

[17] L. Yang, L. Li, Z. Zhang, X. Zhou, E. Zhou, and Y. Liu, "DPGN: Distribution propagation graph network for few-shot learning," in *Proc. CVPR*, 2020, pp. 13390–13399.

[18] 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. 2022.

[19] C. Chen, X. Yang, C. Xu, X. Huang, and Z. Ma, "ECKPN: Explicit class knowledge propagation network for transductive few-shot learning," in *Proc. CVPR*, 2021, pp. 6596–6605.

[20] T. Yu, S. He, Y.-Z. Song, and T. Xiang, "Hybrid graph neural networks for few-shot learning," in *Proc. AAAI Conf. Artif. Intell.*, 2022, vol. 36, no. 3, pp. 3179–3187.

[21] C. Finn, P. Abbeel, and S. Levine, "Model-agnostic meta-learning for fast adaptation of deep networks," in *Proc. 34th Int. Conf. Mach. Learn.*, vol. 70, Aug. 2017, pp. 1126–1135.

[22] J. Oh, H. Yoo, C. Kim, and S. Y. Yun, "BOIL: Towards representation change for few-shot learning," in *Proc. ICLR*, 2021, pp. 1–24.

[23] N. Fei, Z. Lu, T. Xiang, and S. Huang, "MELR: Meta-learning via modeling episode-level relationships for few-shot learning," in *Proc. ICLR*, 2021, pp. 1–20.

[24] A. A. Rusu et al., "Meta-learning with latent embedding optimization," in *Proc. ICLR*, 2019, pp. 1–17.

[25] M. A. Jamal and G. J. Qi, "Task agnostic metalearning for few-shot learning," in *Proc. CVPR*, 2019, pp. 11719–11727.

[26] S. Baik, S. Hong, and K. M. Lee, "Learning to forget for meta-learning," in *Proc. CVPR*, 2020, pp. 2379–2387.

[27] Y. Guo and N. M. Cheung, "Attentive weights generation for few shot learning via information maximization," in *Proc. CVPR*, 2020, pp. 13499–13508.

[28] J. Dong, Y. Wang, J. H. Lai, and X. Xie, "Improving adversarially robust few-shot image classification with generalizable representations," in *Proc. CVPR*, 2022, pp. 9025–9034.

[29] J. Xu and H. Le, "Generating representative samples for few-shot classification," in *Proc. CVPR*, 2022, pp. 9003–9013.

[30] A. Douillard, M. Cord, C. Ollion, T. Robert, and E. Valle, "PodNet: Pooled outputs distillation for small-tasks incremental learning," in *Proc. ECCV*, 2020, pp. 86–102.

[31] X. Hu, K. Tang, C. Miao, X. S. Hua, and H. Zhang, "Distilling causal effect of data in class-incremental learning," in *Proc. CVPR*, 2021, pp. 3957–3966.

[32] S. Wang, W. Shi, S. Dong, X. Gao, X. Song, and Y. Gong, "Semantic knowledge guided class-incremental learning," *IEEE Trans. Circuits Syst. Video Technol.*, early access, Mar. 28, 2023, doi: 10.1109/TCSVT.2023.3262739.

[33] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "iCaRL: Incremental classifier and representation learning," in *Proc. CVPR*, 2017, pp. 2001–2010.

[34] H. Liu, X. Zhu, Z. Lei, D. Cao, and S. Z. Li, "Fast adapting without forgetting for face recognition," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 8, pp. 3093–3104, Aug. 2021.

[35] Q. Hu, Y. Gao, and B. Cao, "Curiosity-driven class-incremental learning via adaptive sample selection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 32, no. 12, pp. 8660–8673, Dec. 2022.

[36] H. Lin, S. Feng, X. Li, W. Li, and Y. Ye, "Anchor assisted experience replay for online class-incremental learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 33, no. 5, pp. 2217–2232, May 2023.

[37] S. Yan, J. Xie, and X. He, "DER: Dynamically expandable representation for class incremental learning," in *Proc. CVPR*, 2021, pp. 3014–3023.

[38] Y. Liu, B. Schiele, and Q. Sun, "Adaptive aggregation networks for class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 2544–2553.

[39] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, and Y. Gong, "Few-shot class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 12180–12189.

[40] S. Dong, X. Hong, X. Tao, X. Chang, X. Wei, and Y. Gong, "Few-shot class-incremental learning via relation knowledge distillation," in *Proc. AAAI Conf. Artif. Intell.*, 2021, vol. 35, no. 2, pp. 1255–1263.

[41] 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.

[42] K. Zhu, Y. Cao, W. Zhai, J. Cheng, and Z.-J. Zha, "Self-promoted prototype refinement for few-shot class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2021, pp. 6797–6806.

[43] Z. Chi, L. Gu, H. Liu, Y. Wang, Y. Yu, and J. Tang, "MetaFSCIL: A meta-learning approach for few-shot class incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 14146–14155.

[44] D.-W. Zhou, F.-Y. Wang, H.-J. Ye, L. Ma, S. Pu, and D.-C. Zhan, "Forward compatible few-shot class-incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2022, pp. 9036–9046.

[45] H. Zhang, M. Cisse, Y. N. Dauphin, and D. Lopez-Paz, "Mixup: Beyond empirical risk minimization," 2017, *arXiv:1710.09412*.

[46] S. Hou, X. Pan, C. C. Loy, Z. Wang, and D. Lin, "Learning a unified classifier incrementally via rebalancing," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2019, pp. 831–839.

[47] F. M. Castro, M. J. Marín-Jiménez, N. Guil, C. Schmid, and K. Alahari, "End-to-end incremental learning," in *Proc. ECCV*, 2018, pp. 233–248.

[48] A. Krizhevsky, "Learning multiple layers of features from tiny images," Univ. Toronto, Toronto, ON, Canada, Tech. Rep. TR-2009, 2009.

[49] O. Russakovsky et al., "ImageNet large scale visual recognition challenge," *Int. J. Comput. Vis.*, vol. 115, no. 3, pp. 211–252, Dec. 2015.

[50] C. Wah, S. Branson, P. Welinder, P. Perona, and S. Belongie, "The Caltech-UCSD birds-200–2011 dataset," California Inst. Technol., Pasadena, CA, USA, Tech. Rep. CNS-TR-2011-001, 2011.

[51] A. Paszke et al., "PyTorch: An imperative style, high-performance deep learning library," in *Proc. NIPS*, vol. 32, 2019, pp. 1–12.

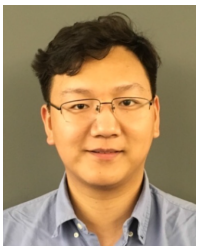
[52] M. Welling, "Herding dynamical weights to learn," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, Jun. 2009, pp. 1121–1128.

[53] L. Van der Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, no. 86, pp. 2579–2605, 2008.

[54] W. Zhang, D. Li, C. Ma, G. Zhai, X. Yang, and K. Ma, "Continual learning for blind image quality assessment," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 2864–2878, Mar. 2023.

[55] W. Zhang, K. Ma, G. Zhai, and X. Yang, "Task-specific normalization for continual learning of blind image quality models," 2021, *arXiv:2107.13429*.

Ye Wang received the B.S. degree from Huazhong Agricultural University, Wuhan, China, in 2017, and the M.S. degree from the Chinese Academy of Agricultural Mechanization Sciences, Beijing, China, in 2020. He is currently pursing the Ph.D. degree with Xi'an Jiaotong University, Xi'an, China. His current research interests include few-shot learning and few-shot class-incremental learning.



Guoshuai Zhao (Member, IEEE) received the B.S. degree from Heilongjiang University, Harbin, China, in 2012, and the M.S. and Ph.D. degrees from Xi'an Jiaotong University, Xi'an, China, in 2015 and 2019, respectively. He is currently an Assistant Professor with the School of Software Engineering, Xi'an Jiaotong University, where he is also with the Key Laboratory for Intelligent Networks and Network Security, Ministry of Education, and the SMILES Laboratory. He is mainly engaged in the research of social media big data analysis and recommendation systems.

Xueming Qian (Member, IEEE) received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree from the School of Electronics and Information Engineering, Xi'an Jiaotong University, in 2008. He was a Visiting Scholar with Microsoft Research Asia from 2010 to 2011. He was an Assistant Professor with Xi'an Jiaotong University, where he was an Associate Professor from 2011 to 2014, and is currently a Full Professor. He is also the Director of the Smiles Laboratory, Xi'an Jiaotong University. He received the Microsoft Fellowship in 2006. He received Outstanding Doctoral Dissertations of Xi'an Jiaotong University and Shaanxi Province in 2010 and 2011, respectively. His research interests include social media big data mining and search. His research is supported by the National Natural Science Foundation of China, Microsoft Research, and Ministry of Science and Technology.