Contents lists available at ScienceDirect

Alexandria Engineering Journal

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

Original article

Prompt-based learning for few-shot class-incremental learning☆

Jicheng Yuan [a,b], Hang Chen [c], Songsong Tian [b], Wenfa Li [a,*], Lusi Li [d], Enhao Ning [a], Yugui Zhang [b]

[a] School of Intelligent Science and Technology, University of Science and Technology Beijing, Beijing, 100083, China
[b] Institute of Semiconductors, Chinese Academy of Sciences, Beijing, 100083, China
[c] Zhejiang Chuchi Technology Co., Ltd, Hangzhou, 310000, China
[d] Department of Computer Science, Old Dominion University, Norfolk, VA 23529, USA

ARTICLE INFO

Keywords:
Few-shot class-incremental learning
Prompt learning
Prototype classifier
Catastrophic forgetting

ABSTRACT

Few-Shot Class-Incremental Learning (FSCIL) aims to enable deep neural networks to incrementally learn new tasks from a limited number of labeled samples, while retaining knowledge of previously learned tasks, mimicking the way humans learn. In this paper, we introduce a novel approach called Prompt Learning for FSCIL (PL-FSCIL), which leverages the power of prompts alongside a pre-trained Vision Transformer (ViT) model to effectively tackle the challenges of FSCIL. Our approach explores the feasibility of directly applying visual prompts in FSCIL, using a simplified model architecture. PL-FSCIL integrates two key prompts: the Domain Prompt and the FSCIL Prompt. Both are tensors incorporated into the attention layer of the ViT network to enhance its capabilities. The Domain Prompt helps the model adapt to new data domains, while the FSCIL Prompt, in combination with a prototype classifier, boosts the model's ability to handle incremental tasks. We evaluate the performance of PL-FSCIL on well-established benchmark datasets, including CIFAR-100 and CUB-200. The results demonstrate competitive performance, highlighting the method's promising potential for real-world applications, particularly in scenarios where high-quality labeled data is scarce. The source code is at: https://github.com/JichengYuan81/PL-FSCIL.

1. Introduction

Recent advancements in computing technology and data have significantly improved deep neural networks (DNNs) in computer vision [1,2]. However, DNNs still struggle when new categories emerge with limited high-quality data, primarily due to catastrophic forgetting [3] and overfitting [4]. Addressing these challenges is central to incremental learning (IL) and few-shot learning (FSL). IL focuses on mitigating catastrophic forgetting, often branching into Task-IL, Domain-IL, and Class-IL [5]. FSL aims to enable models to learn new classes with minimal samples, balancing rich representation from extensive training and adaptation to new, scarce data. In this context, [6] introduced Few-Shot Class-Incremental Learning (FSCIL) to merge IL and FSL challenges, leveraging pre-existing knowledge to enhance AI's adaptive learning for real-world applications.

Recently, several FSCIL methods have used pre-trained ResNet [2] as the backbone, fine-tuning the network on base classes to adapt to new data distributions [7–11]. However, these methods are inefficient, limited by the pretrained encoders' feature extraction capabilities and the network's capacity, leading to performance plateaus. To overcome these limitations, highly generalizable pre-trained models like Vision Transformer (ViT) [12] have emerged, surpassing ResNet in various tasks. While fine-tuning ViT directly is computationally intensive, prompt learning offers an efficient alternative, eliminating additional training and saving resources [13]. Prompt learning has gained attention for its effectiveness in guiding pre-trained models to generate desired outputs.

In this paper, inspired by the versatility of pre-trained ViT models, we propose leveraging prompts for Few-Shot Class-Incremental Learning tasks to efficiently integrate new knowledge into existing models without the need for extensive retraining. We introduce PL-FSCIL, an innovative strategy that employs prompts to enhance model performance in FSCIL scenarios. PL-FSCIL consists of three key components: a Domain Prompt, an FSCIL Prompt, and a Prototype Classifier. The Domain Prompt is designed to be domain-specific, enabling the pre-trained model to adapt its feature representation capabilities to the current dataset's domain by being seamlessly incorporated into

☆ This study was supported by the Natural Science Foundation of china [Grant number 61972040].

* Corresponding author.

E-mail addresses: yuanjicheng@xs.ustb.edu.cn (J. Yuan), 13306815831@189.cn (H. Chen), tiansongsong@semi.ac.cn (S. Tian), liwenfa@ustb.edu.cn (W. Li), lusili@cs.odu.edu (L. Li), ningenhao@xs.ustb.edu.cn (E. Ning), zhangyugui@semi.ac.cn (Y. Zhang).

https://doi.org/10.1016/j.aej.2025.02.008
Received 1 January 2025; Received in revised form 27 January 2025; Accepted 2 February 2025
Available online 18 February 2025

the model. In contrast, the FSCIL Prompt is task-specific, appending task-related information beyond mere domain details by introducing prompts tailored to few-shot tasks. This dual-prompt approach allows the model to dynamically adjust to new classes while maintaining its ability to recognize previously learned classes. Both types of prompts are integrated into the appropriate self-attention layers of the Transformer through a technique known as prefix-tuning. Additionally, we replace the conventional Softmax classifier with a Prototype Classifier, which is derived from the feature outputs of each class within the training set and eliminates the need for gradient-based optimization. This combination of domain and task-specific prompts, along with the prototype-based classification, ensures that PL-FSCIL effectively manages incremental learning tasks with minimal computational overhead.

In summary, our main contributions can be outlined as follows:

- We propose leveraging prompt learning in FSCIL by innovatively utilizing ViT models. Our Domain Prompt and FSCIL Prompt enhance these models, balancing simplicity and effectiveness in tackling FSCIL challenges.
- We introduce an innovative prompt regularization mechanism that enforces orthogonality between the Domain Prompt and FSCIL Prompt. By leveraging the Frobenius norm, this mechanism enables the FSCIL Prompt to assimilate diverse, task-specific knowledge.
- Our work introduces a streamlined yet powerful baseline for FSCIL tasks. Incorporating a prototype classifier and undergoing comprehensive evaluation on multiple benchmark datasets, PL-FSCIL achieves notable improvements over existing FSCIL methodologies.

2. Related work

2.1. Few-shot class-incremental learning

In the domain of Few-Shot Class-Incremental Learning, researchers have focused on gradually introducing new categories with limited data while preserving the ability to recognize previously learned categories. TOPIC [6] was the first to propose the FSCIL task and effectively addressed the issue of catastrophic forgetting by maintaining the topological relationships between features of both old and new categories using a neural gas structure. CEC [7] employed independent classifiers to differentiate the categories and leveraged a graph model to propagate contextual information across these classifiers. CABD [14] introduced a class-aware bilateral distillation structure, which simultaneously addresses both catastrophic forgetting and overfitting. However, such replay-based methods are challenging to apply in scenarios with strict data privacy constraints. For non-replay methods, many techniques leverage data augmentation [15–17] and class enhancement, such as FACT [18], SAVC [19], and M2SD [20]. These approaches improve overall classification performance by generating virtual classes that help compact the embedding space of known categories. Additionally, some studies [9,21–23] have incorporated meta-learning into FSCIL to enhance the model's adaptability to real incremental scenarios, by simulating pseudo-incremental scenarios during the base class phase. Other methods utilize parameter regularization mechanisms, such as F2M [24], which mitigates catastrophic forgetting by targeting flat minima, WaRP [25], which compresses most old knowledge into key parameters to accommodate new category learning, and LDC [26], which offers a unified framework that simultaneously preserves old class distributions and estimates new class distributions with limited samples. For a more comprehensive review, refer to the related surveys [27].

2.2. Prompt learning

Prompt learning was initially developed in the field of Natural Language Processing (NLP) [28,29] to enhance the performance of various downstream tasks by leveraging pre-trained language models, typically through fine-tuning for specific tasks [30,31]. In recent years, this approach has been extended to the visual domain, leading to the creation of vision-language models such as CLIP [32] and CoOp [33]. These models utilize textual prompts like "a photo of [cls]" to improve task performance. In computer vision, Visual Prompt Tuning (VPT) [34] for ViT involves injecting noise directly into input images as prompts, demonstrating strong performance on certain datasets while requiring fewer trainable parameters compared to full fine-tuning. Within the realm of incremental learning, prompt-based methods such as DualPrompt [35] and L2P [36] capture both task-invariant and task-specific knowledge but still face challenges in few-shot class-incremental learning. For Few-Shot Class-Incremental Learning (FSCIL), recent multimodal prompt studies like M-FSCIL [37], UACL [38], FSPT-FSCIL [39] and IOSPL [40] have emerged. However, these approaches involve complex architectures that require further optimization.

3. Prerequisites

3.1. Problem formalization

FSCIL is usually consists of a sequence sessions $\mathcal{D} = \{\mathcal{D}^0, \mathcal{D}^1, \ldots, \mathcal{D}^m\}$, where $\mathcal{D}^i = \{\mathcal{D}^i_{train}, \mathcal{D}^i_{test}\}$ denotes the training and testing dataset for sessions $\{0, 1, \ldots, m\}$. For session i, we have its training set $\mathcal{D}^i_{train}$ with the corresponding label space of $\mathcal{Y}^i$. The training data across different sessions are disjoint, i.e. $\forall i, j$ and $i \neq j, \mathcal{Y}^i \cap \mathcal{Y}^j = \emptyset$. During testing, the testing set $\mathcal{D}^i_{test}$ at session i includes test data from all previous and current classes, i.e., the label space of $\mathcal{Y}^0 \cup \mathcal{Y}^1 \ldots \cup \mathcal{Y}^i$. In addition, for the base session $(i = 0)$, a sufficient amount of training data is provided and for the following incremental sessions $(i > 0)$, only a limited amount of data is provided. We organize the limited instances in $\mathcal{D}^i_{train}$ as N-way K-shot training set, where there are N classes in the dataset, and each class has K training images.

3.2. Prompt-based learning

Prompt-based learning, or prompting, was first introduced in NLP for transfer learning by adding extra instructions to pre-trained models for conditional downstream tasks [28]. Prompt Tuning [41], a recent technique, attaches prompt parameters to frozen transformer-based models [42] to perform downstream tasks. These prompts are typically prepended to the input sequence to guide model predictions. A brief illustration of Prompt Tuning is provided below.

In a pre-trained ViT, the input embedding layer converts an image into a sequence-like output $\mathbf{X} \in \mathbb{R}^{N \times C}$, where N includes patches plus a CLS token, and C is the embedding dimension. For downstream tasks, the frozen backbone acts as a feature extractor. Prompt parameters $\mathbf{P} \in \mathbb{R}^{L_p \times C}$, with sequence length L_p, are prepended to the sequence, and the extended embedding is passed through the model for classification.

4. Method

4.1. Prompt pool design

In this section, we primarily discuss the design of two types of prompts: Domain Prompt $(\mathbf{P}_D)$ and FSCIL Prompt $(\mathbf{P}_F)$. The overview of our proposed PL-FSCIL is shown in Fig. 1.

Domain Prompt. A shared parameter for all tasks, the domain knowledge, represents the overall understanding of the dataset. It is structured as a tensor $\mathbf{P}_D \in R^{L_D \times C}$, where L_D denotes the length of the domain prompt and C is the embedding dimension. This tensor is randomly initialized at the start of training with the goal of providing a

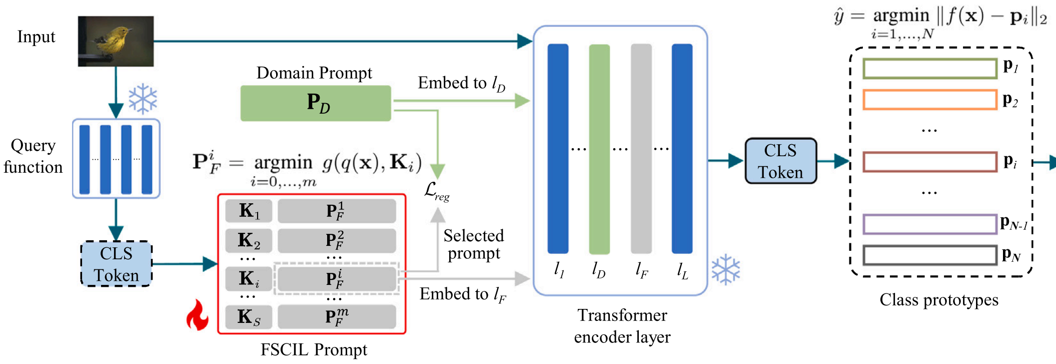
Input
Query function
CLS Token
Domain Prompt
$\mathbf{P}_D$
Embed to l_D
$\mathbf{P}_F^i = \underset{i=0,\ldots,m}{\text{argmin}}\ g(q(\mathbf{x}), \mathbf{K}_i)$
$\mathcal{L}_{reg}$
Selected prompt
Embed to l_F
$\mathbf{K}_1$ $\mathbf{P}_F^1$
$\mathbf{K}_2$ $\mathbf{P}_F^2$
$\mathbf{K}_i$ $\mathbf{P}_F^i$
$\mathbf{K}_S$ $\mathbf{P}_F^m$
FSCIL Prompt
l_I l_D l_F l_L
Transformer encoder layer
CLS Token
$\hat{y} = \underset{i=1,\ldots,N}{\text{argmin}}\ \|f(\mathbf{x}) - \mathbf{p}_i\|_2$
$\mathbf{p}_I$
$\mathbf{p}_2$
$\mathbf{p}_i$
$\mathbf{p}_{N-1}$
$\mathbf{p}_N$
Class prototypes
'Bird'
Output

Fig. 1. Overview of PL-FSCIL. In this architecture, an input image is first processed via a query function, aligning it with the relevant FSCIL Prompt to leverage task-specific incremental knowledge. Simultaneously, the Domain Prompt imbues the input with dataset-specific knowledge, acting as a reservoir of domain acumen. To diversify the knowledge assimilated by the FSCIL Prompt and maintain orthogonality with the Domain Prompt content, a Prompt Regularization Loss $\mathcal{L}_{reg}$ is imposed. Then the input image, coupled with the dual prompts, is integrated into a pre-defined layer of the network, culminating in the emission of a CLS token, which is used for subsequent classification by the prototype classifier.

unified knowledge representation that can be leveraged across all tasks.

FSCIL Prompt. The FSCIL Prompt is task-specific, necessitating the integration of session knowledge. Thus, we define $\mathbf{P}_F \in R^{m \times L_F \times C}$, where m denotes the number sessions in FSCIL task, and L_F represents the length of the FSCIL Prompt and C is the embedding dimension as Domain Prompt. The prompt corresponding to the ith session is denoted as $\mathbf{P}_F^i \in R^{L_F \times C}$. Different from $\mathbf{P}_D$, each $\mathbf{P}_F^i$ is associated with a unique Prompt Key $\mathbf{K}_i \in R^C$, which is a parameter that can be learned.

To ensure task-specific, we uniformly initialize $\mathbf{K}_i$ and establish a query function $q(\cdot)$ to search for the best matching key, then selecting the appropriate FSCIL Prompt to use. Furthermore, we update the corresponding $\mathbf{K}_i$ to match instances of input features by the matching loss $\mathcal{L}_{dis}$ during training. Inspired by [36], we can directly use the whole pre-trained model as a frozen feature extractor to get the query features: $q(\mathbf{x}) = f(\mathbf{x})[0]$ (the feature vector corresponding to [**class**] token). The form of $\mathcal{L}_{dis}$ is as follows:

$$\mathcal{L}_{dis}\left(\mathbf{x}, \mathbf{K}_i\right) = g(q(\mathbf{x}), \mathbf{K}_i), \tag{1}$$

where $\mathbf{x}$ is the input image, and $g(\cdot)$ can be either Euclidean distance or cosine distance.

During testing, the appropriate $\mathbf{P}_F^i$ is selected by calculating the distance between the input sample and various task keys $\mathbf{K}_i$, adding task-specific knowledge to the model.

$$\mathbf{P}_F^i = \underset{i=0,\ldots,m}{\text{argmin}}\ g(q(\mathbf{x}), \mathbf{K}_i), \tag{2}$$

where $\mathbf{P}_F^i$ represents the prompt selected specifically for $\mathbf{x}$ from $\mathbf{P}_F$.

Prompt regularization mechanism. To ensure the Domain Prompt captures dataset-wide knowledge and the FSCIL Prompt focuses on task-specific knowledge, we introduce a prompt regularization mechanism to enforce orthogonality between the two prompts. This orthogonality prevents updates to the FSCIL Prompt from interfering with the Domain Prompt's comprehensive knowledge, aiding in the separation of generalized and task-specific knowledge. This approach helps avoid catastrophic forgetting and enhances positive forward knowledge transfer. Considering that the Frobenius norm can measure orthogonality by assessing the degree of difference in element values, for the $\mathbf{P}_D$ and any session's P_F^i, the regularization loss function can be defined as follows:

$$\mathcal{L}_{reg}(i) = \|\mathbf{P}_D \cdot \mathbf{P}_F^{i\ T}\|, \tag{3}$$

Herein, $\|\cdot\|$ represents the Frobenius norm. We can use $\mathcal{L}_{reg}$ to measure the degree of difference between $\mathbf{P}_D$ and P_F^i. If they are orthogonal, the value of $\mathcal{L}_{reg}$ is zero. As $\mathcal{L}_{reg}$ increases, more similar prompt knowledge is considered to be contained in $\mathbf{P}_D$ and P_F^i.

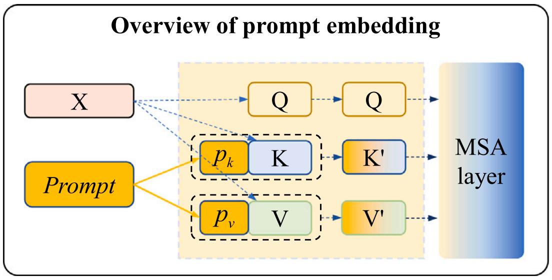

Fig. 2. Before being passed to the MSA layer, the prompt is split evenly and appended as a prefix to the keys and values of the hidden features.

4.2. Embedding prompts in vision transformer

In the domain of ViTs, a prompt represents an auxiliary input that can direct the model's attention mechanism toward particular elements of the input data. This section explores the integration of prompts into ViT architectures.

In the Transformer layer l within the ViT, the input $\mathbf{X}^{(l)}$ is first linearly transformed into three components: query $\mathbf{Q}^{(l)}$, key $\mathbf{K}^{(l)}$, and value $\mathbf{V}^{(l)}$, which are then used to compute the attention weights and the output feature. In the original ViT model, the self-attention operation can be expressed as:

$$\begin{aligned}\mathbf{X}^{(l)} &= \text{Attention}(\mathbf{Q}^{(l)}, \mathbf{K}^{(l)}, \mathbf{V}^{(l)}) \\ &= \text{Softmax}\left(\frac{\mathbf{Q}^{(l)}(\mathbf{K}^{(l)})^T}{\sqrt{d_k}}\right)\mathbf{V}^{(l)},\end{aligned} \tag{4}$$

where d_k is the dimension of the key vectors.

The overall flow of prompt embedding into the ViT is illustrated in Fig. 2, we employ prefix-tuning [43], a method that constructs a set of task-related virtual tokens as Prefix before the input tokens. During training, only the parameters of the prefix are updated, while other parameters in the model remain fixed. Specifically, for Domain Prompt or FSCIL Prompt, we inject the prompt into the key and value components, anticipating the prompt to match the shape of the key and value tensors post-permutation. That is $\mathbf{P} = \left[\mathbf{P}_k; \mathbf{P}_v\right]$, and the modified self-attention operation becomes:

$$\mathbf{K}' = [\mathbf{P}_k; \mathbf{K}^{(l)}], \quad \mathbf{V}' = [\mathbf{P}_v; \mathbf{V}^{(l)}], \tag{5}$$

$$\begin{aligned}\mathbf{X}^{(l)} &= \text{Attention}(\mathbf{Q}^{(l)}, \mathbf{K}', \mathbf{V}') \\ &= \text{Softmax}\left(\frac{\mathbf{Q}^{(l)}\mathbf{K}'^T}{\sqrt{d_k}}\right)\mathbf{V}',\end{aligned} \tag{6}$$

where $\mathbf{P}_k$ and $\mathbf{P}_v$ represent the key and value components of the prompt, and $[;]$ denotes concatenation operation. By concatenating the prompt with the key and value tensors, the prompt can affect the calculation of attention weights and the output feature.

Unlike the original self-attention operation in transformer models that only utilize input data, the introduction of prefix-tuning allows self-attention operations to incorporate prompt information. This enhancement enables the model to target particular aspects of input data as directed by the prompt, thus rendering the model more controllable and adaptable.

For the embedded positions of two prompts, VPT [34] offers both deep and shallow strategies, namely embedding prompts in all layers or only in the first layer. In this work, we adopt an intermediate approach, empirically embedding $\mathbf{P}_D$ and $\mathbf{P}_F$ at specific selected layers $\mathbf{X}^{(l_D)}$ and $\mathbf{X}^{(l_F)}$, respectively.

4.3. Prototype classifier

In the testing phase, our approach uses a Prototype Classifier, where each class's prototype is defined as the mean feature vector of its samples. New samples are compared to these prototypes and classified based on proximity. While the prototype classifier isn't novel, it achieves strong results when paired with our proposed prompts.

Prototype Construction. For each category, we compute the mean of the CLS tokens of all samples in the output of the pre-trained ViT model as the category prototype, as represented by the following process:

$$\mathbf{p}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} f(\mathbf{x}_{ij}), \tag{7}$$

where $f(\mathbf{x}_{ij})$ denotes the output of the pre-trained ViT model for the jth sample of the ith category, i.e., the feature representation.

Classification via Distances. The classification decision for a given test instance is based on its distance to category prototypes. Employing Euclidean distance, we assign the sample to its nearest prototype's category:

$$\hat{y} = \underset{i=1,\dots,N}{\operatorname{argmin}} \|f(\mathbf{x}) - \mathbf{p}_i\|_2, \tag{8}$$

where $\mathbf{x}$ is the input sample, and N is the total number of categories.

4.4. Overall loss function

For all sessions, the overall loss function is represented as follows:

$$\min_{\mathbf{P}_D,\mathbf{P}_F} \sum_{i=0}^{m} (\mathcal{L}(\Gamma(\mathbf{x}), y) + \lambda \mathcal{L}_{dis}(\mathbf{x}, \mathbf{K}_i) + \alpha \mathcal{L}_{reg}(i)), \tag{9}$$

where $\mathbf{x} \in D^i_{train}$, $\Gamma(\mathbf{x})$ denotes the output of the model, with the overall loss function comprising the cross-entropy loss $\mathcal{L}$, the matching loss $\mathcal{L}_{dis}$ as defined in Eq. (1), and the prompt regularization loss $\mathcal{L}_{reg}$. The terms λ and α serve as scalar factors to balance the contribution of the matching and regularization losses, respectively.

4.5. Algorithms for PL-FSCIL

The design and embedding of two prompts can be found in Algorithm 1.

5. Experiments

In this section, we present the experimental setup and evaluation of our proposed method for FSCIL. We first describe the datasets and implementation details, followed by a comparison with state-of-the-art approaches. Finally, we conduct an ablation study and analyze the contributions of different components of the PL-FSCIL model.

Algorithm 1 Training and model optimization with Domain Prompt and FSCIL Prompt

Input: Training Inputs: Dataset D_{train}, pretrained ViT (L layers), Domain Prompt ($\mathbf{P}_D$, embedded at l_D), FSCIL Prompt ($\mathbf{P}_F$, embedded at l_F), Incremental Learning tasks (S), query function $q(\cdot)$

Output: Optimized $\mathbf{P}_D$ and $\mathbf{P}_F$

```
Initialize $\mathbf{P}_D$, $\mathbf{P}_F$ and the corresponding Prompt Keys $\mathbf{K}$ using PyTorch's default methods.
for $i \Leftarrow 1$ to $|S|$ do
    for all $(\mathbf{x}, y)$ in $D^i_{train}$ do
        $\mathbf{Z}^{(0)} \Leftarrow \mathbf{x}$
        for $l \Leftarrow 1$ to $L$ do
            if $l$ in $l_D$ then
                $\mathbf{Z}^{(l)} \Leftarrow \Gamma^{(l)}(\mathbf{Z}^{(l-1)}; \mathbf{P}_D)$
            else if $l$ in $l_F$ then
                $\mathbf{Z}^{(l)} \Leftarrow \Gamma^{(l)}(\mathbf{Z}^{(l-1)}; \mathbf{P}^{(i)}_F)$
            else
                $\mathbf{Z}^{(l)} \Leftarrow \Gamma^{(l)}(\mathbf{Z}^{(l-1)})$
            end if
        end for
        Calculate $\mathcal{L}_{dist}$, given by $\mathcal{L}_{dis}(\mathbf{x}, \mathbf{K}_i) = g(q(\mathbf{x}), \mathbf{K}_i)$
        Update $\mathbf{P}_D$ and $\mathbf{P}_F$ using $\mathcal{L}_{reg}$, according to $\min_{\mathbf{P}_D,\mathbf{P}_F} \mathcal{L}(\Gamma(\mathbf{x}), y) + \lambda \mathcal{L}_{dis}(\mathbf{x}, \mathbf{K}_i) + \alpha \mathcal{L}_{reg}(i)$
    end for
end for
```

Table 1
Experimental setup for the three datasets (in %).

Dataset	Base classes	Incremental sessions setup	Sessions
CIFAR-100	60	5-way, 5-shot	8
MiniImageNet	60	5-way, 5-shot	8
CUB-200	100	10-way, 5-shot	10

5.1. Dataset

We utilize three benchmark datasets for our experiments: CUB-200 [44], CIFAR-100 [45] and MiniImageNet [46]. CUB-200 is a popular benchmark image dataset for fine-grained classification and recognition research. It has a total of 11,788 bird images from 200 classes, with 5994 images for training and 5794 images for testing. CIFAR-100 comprises 100 classes, each with 600 RGB images of size 32 × 32 pixels. Each class has 500 training images and 100 testing images. MiniImageNet contains 60,000 RGB images of size 84 × 84 pixels, which are derived from the ImageNet-1k dataset. It has the same number of classes and samples as CIFAR-100, but with more complex content for FSCIL research. During training sessions, we follow the setting in [7] to train our models. For detailed dataset settings refer to Table 1.

5.2. Implementation details

Our experiments were conducted on a platform with PyTorch 1.12.1, using four NVIDIA GeForce RTX 3090 GPUs. We used a pre-trained ViT Base/16 model with an input size of 224 × 224 pixels. Both the Domain Prompt and FSCIL Prompt lengths were set to 10, and the class prototype dimensionality matched the number of classes in each dataset. During training, we used actual sample labels and cross-entropy loss for error calculation, with no loss computed during the prototype construction phase.

Evaluation protocol. After each session, we evaluate the Top 1 accuracy and the average accuracy (AA) across all sessions. We also use the performance dropping rate (PD) [7] to quantify the absolute reduction in accuracy during the final session compared to the base

Table 2
Evaluation on CUB-200. Presented in this table are the accuracy of each session, overall average accuracy across all sessions, and performance drop rate. Methods denoted with an asterisk (*) are based on the CLIP model (in %).

Methods	Accuracy in each session ↑											AA ↑	PD ↓
	0	1	2	3	4	5	6	7	8	9	10		
TOPIC [6]	68.68	62.49	54.81	49.99	45.25	41.40	38.35	35.36	32.22	28.31	26.28	43.92	42.40
SPPR [47]	68.68	61.85	57.43	52.68	50.19	46.88	44.65	43.07	40.17	39.63	37.33	49.32	31.35
CEC [7]	75.85	71.94	68.50	63.50	62.43	58.27	57.73	55.81	54.83	53.52	52.28	61.33	23.57
F2M [48]	81.07	78.16	75.57	72.89	70.86	68.17	67.01	65.26	63.36	61.76	60.26	69.49	20.81
MetaFSCIL [9]	75.90	72.41	68.78	64.78	62.96	59.99	58.30	56.85	54.78	53.82	52.64	61.93	23.26
ERDR [8]	75.90	72.14	68.64	63.76	62.58	59.11	57.82	55.89	54.92	53.58	52.39	61.52	23.51
FACT [49]	75.90	73.23	70.84	66.13	65.56	62.15	61.74	59.83	58.41	57.89	56.94	64.42	18.96
ALICE [11]	77.40	72.70	70.60	67.20	65.90	63.40	62.90	61.90	60.50	60.60	60.10	65.75	17.30
LIMIT [50]	75.89	73.55	71.99	68.14	67.42	63.61	62.40	61.35	59.91	58.66	57.41	65.48	18.48
CLOM [51]	79.57	76.07	72.94	69.82	67.80	65.56	63.94	62.59	60.62	60.34	59.58	67.17	19.99
DSN [52]	80.86	78.18	75.57	72.68	71.42	70.12	69.16	67.94	66.99	65.10	63.21	71.02	17.65
NC-FSCIL [53]	80.45	75.98	72.30	70.28	68.17	65.16	64.43	63.25	60.66	60.01	59.44	67.28	21.01
M-FSCIL* [37]	81.04	79.73	76.62	73.30	71.22	68.90	66.87	65.02	63.90	62.49	60.40	69.95	20.64
IOSPL* [40]	84.30	83.24	80.86	**79.25**	**77.74**	72.42	72.15	69.88	68.85	67.42	66.79	74.81	17.51
CA-CLIP* [54]	85.16	79.20	77.49	70.97	70.82	69.12	66.33	62.54	60.55	61.05	60.33	69.41	24.83
LIMIT+V-Swin-T* [55]	82.59	81.09	79.46	76.68	76.94	**75.12**	**74.59**	**73.14**	**73.40**	**73.17**	**73.34**	**76.32**	**9.25**
PL-FSCIL	**85.16**	**85.40**	**82.75**	75.22	77.22	73.25	72.39	70.24	67.97	68.33	69.86	75.25	15.30

Table 3
Evaluation on CIFAR-100. Presented in this table are the accuracy of each session, overall average accuracy across all sessions, and performance drop rate. Methods denoted with an asterisk (*) are based on the CLIP model (in %).

Methods	Accuracy in each session ↑									AA ↑	PD ↓
	0	1	2	3	4	5	6	7	8		
TOPIC [6]	64.10	55.88	47.07	45.16	40.11	36.38	33.96	31.55	29.37	42.62	34.73
SPPR [47]	64.10	65.86	61.36	57.34	53.69	50.75	48.58	45.66	43.25	54.51	20.85
CEC [7]	73.07	68.88	65.26	61.19	58.09	55.57	53.22	51.34	49.14	59.53	23.93
F2M [48]	64.71	62.05	59.01	55.58	52.55	49.96	48.08	46.28	44.67	53.65	**20.04**
MetaFSCIL [9]	74.50	70.10	66.84	62.77	59.48	56.52	54.36	52.56	49.97	60.79	24.53
ERDR [8]	74.40	70.20	66.54	62.51	59.71	56.58	54.52	52.39	50.14	60.77	24.26
C-FSCIL [56]	77.50	72.45	67.94	63.80	60.24	57.34	54.61	52.41	50.23	61.84	27.27
ALICE [11]	79.00	70.50	67.10	63.40	61.20	59.20	58.10	56.30	54.10	63.21	24.90
LIMIT [50]	73.81	72.09	67.87	63.89	60.70	57.77	55.67	53.52	51.23	61.84	22.58
CLOM [51]	74.20	69.83	66.17	62.39	59.26	56.48	54.36	52.16	50.25	60.57	23.95
DSN [52]	73.00	68.83	64.82	62.64	59.36	56.96	54.04	51.57	50.00	60.14	23.00
NC-FSCIL [53]	82.52	76.82	73.34	69.68	66.19	62.85	60.96	59.02	56.11	67.50	26.41
M-FSCIL* [37]	85.55	**80.94**	**77.27**	**73.51**	69.16	66.44	62.01	59.04	55.06	69.89	30.49
UACL* [38]	76.13	72.80	68.67	65.17	62.65	60.41	58.72	57.00	54.82	64.04	21.31
LIMIT+V-Swin-T* [55]	82.07	78.49	75.90	73.27	72.36	71.20	70.60	69.39	67.69	73.44	14.38
PL-FSCIL	**85.73**	74.54	74.77	72.42	**72.98**	**72.87**	**72.49**	**71.62**	**75.13**	**74.73**	**10.60**

session, which is formulated as:

$$\mathrm{PD} = \mathcal{A}_0 - \mathcal{A}_M, \tag{10}$$

where $\mathcal{A}_0$ represents the classification precision in the base session, and $\mathcal{A}_M$ denotes the precision in the last session.

5.3. Comparison with the state-of-the-art

In this section, we conduct comprehensive experiments to compare the performance of our proposed PL-FSCIL approach against state-of-the-art FSCIL approaches. Considering that the utilization of the CLIP model inherently results in improved performance, to make the comparison fair and precise, we first compare a range of ResNet-based models, namely TOPIC [6], SPPR [47], CEC [7], F2M [48], MetaFSCIL [9], ERDR [8], C-FSCIL [56], FACT [49], ALICE [11], LIMIT [50], CLOM [51], DSN [52], and NC-FSCIL [53]. We then assess models based on the CLIP architecture, specifically M-FSCIL [37], IOSPL [40], UACL [38], and CA-CLIP [54].

There may be a data leakage in FSCIL due to ViT's pretraining on ImageNet [12] potentially including new categories. Therefore, we focus on analyzing its performance on the cub200 and cifar100 datasets.

Results on CUB-200. Table 2 shows the FSCIL experiment results on the CUB-200 dataset, where our PL-FSCIL model achieves an average accuracy of 75.25% and a low performance drop rate of 15.30%. This outstrips the second-ranked ResNet-based model, DSN [52], which achieves a mean accuracy of 71.02%, showcasing the superiority of our method. To ensure fairness in our comparative analysis, we also compared PL-FSCIL with the latest CLIP-based models using transformer architecture, the results show that our method maintains strong performance in the first three sessions. Although its accuracy slightly lags behind Swin-T's method in the later sessions, the performance remains highly competitive.

Results on CIFAR-100. Table 3 presents the FSCIL experiment results on the CIFAR-100 dataset, where the PL-FSCIL model achieves the highest initial accuracy of 85.73% in session 0 and maintains the highest accuracy of 75.13% in session 8. It also secures the highest overall average accuracy (AA) of 74.73% across all sessions, demonstrating its robust performance. The PL-FSCIL approach prioritizes a balanced and synergistic method between initial learning and memory retention, crucial for long-term learning scenarios. Although its early-stage performance lags behind M-FSCIL [37], PL-FSCIL eventually surpasses it by over 5 percentage points in the final session. And its performance dropping rate is the lowest, indicating the model's ability to maintain high performance despite the incremental increase in classes.

Results on MiniImageNet. As shown in Fig. 3, we can observe that our method outperforms all other methods at each encountered learning session on MiniImageNet dataset.

Compared with DualPrompt Our approach introduces the prototype classifier along with an FSCIL prompt design, enabling more

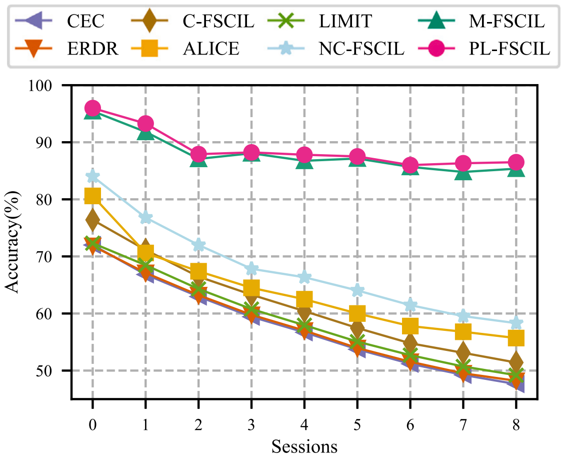

Fig. 3. Comparison results on MiniImageNet.

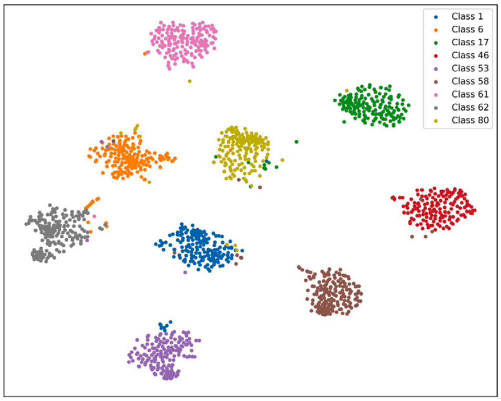

Fig. 4. The t-SNE visualization on the CIFAR100 dataset of the embeddings learned by our methods. Dots with different colors represent data points from different classes.

Table 4
Comparative performance analysis of PL-FSCIL and DualPrompt.

Methods	CUB-200		CIFAR-100		MiniImageNet	
	AA ↑	PD ↓	AA ↑	PD ↓	AA ↑	PD ↓
DualPrompt	64.56	30.59	46.44	60.36	72.14	25.24
PL-FSCIL	**75.25**	**15.30**	**74.73**	**10.60**	**88.84**	**9.48**

effective handling of few sample data scenarios. In comparison, the DualPrompt method requires a considerable amount of data for new categories to operate effectively. Table 4 presents a comparative analysis of the performance metrics for both methods across three standard datasets: CUB-200, CIFAR-100, and MiniImageNet. It is clear that the PL-FSCIL method surpasses DualPrompt in every evaluated aspect on these datasets. Specifically, PL-FSCIL achieves consistently higher Average Accuracy and PD, with performance improvements exceeding 10% on each dataset.

To further demonstrate that our prompt learning strategy can better overcome catastrophic forgetting and learn incremental classes, we visualize the embedding space on the CIFAR100 dataset using the widely used t-SNE [57] tool in Fig. 4. We randomly selected 6 classes from the base class and 3 classes from the new class. As can be seen from the figure, our method can precisely separate different classes and tightly cluster the same class (see Fig. 4).

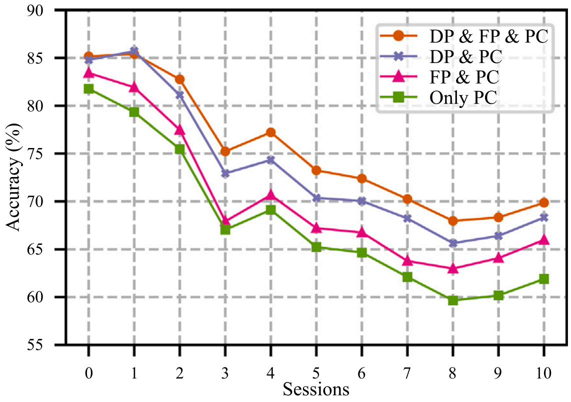

Fig. 5. Visualization of ablation experiments impact on accuracy in CUB200 dataset.

5.4. Ablation study and analysis

Is the Domain Prompt truly efficacious? In Tables 2 and 3, PL-FSCIL demonstrates impressive results on the base classes (session 0) compared to other methods. This can be attributed to PL-FSCIL's use of a pre-trained ViT model and the Domain Prompt, which together enhance domain adaptation for new datasets. To validate the Domain Prompt's efficacy, we select four classic classification datasets: CIFAR-10 [45], STL-10 [58], Flowers-102 [59], Caltech-256 [60]. We benchmark against the pre-trained ResNet18 [2] and the Visual Prompt Tuning (VPT) model [34]. In these experiments, we remove the FSCIL Prompt module and replace the prototype classifier with an MLP classifier.

Table 5 shows the experimental results, including the Top 1 accuracy on each dataset, the number of trainable parameters, and the computational complexity of the models. Compared to ResNet18, which is commonly used in other FSCIL methods, Domain Prompt and VPT model not only achieve superior accuracy but also require fewer parameters, despite they elevate computational complexity to some degree. Additionally, compared to VPT model, Domain Prompt not only yields competitive results in terms of accuracy with fewer parameters but also demonstrates lower computational complexity (16.86 G Macs). This demonstrates the efficacy of incorporating Domain Prompt into the PL-FSCIL framework for adapting to new datasets, while simultaneously achieving equilibrium among high classification accuracy, model compactness, and computational efficiency.

The impact of different components. We have investigated the impact of Domain Prompt, FSCIL Prompt, and Prototype Classifier on the performance of PL-FSCIL. Table 6 provides an overview of the contributions of the different components in terms of average accuracy and performance dropping rate. A checkmark in the table represents the inclusion of a particular component in the experimental setup. Employing a prototype classifier is fundamental for PL-FSCIL's classification capabilities. Integrating a Domain Prompt individually surpasses the performance gain of using a FSCIL Prompt alone. Employing both prompts enables the model to achieve peak performance. Fig. 5 provides a more detailed view of the ablation experiments, breaking down the accuracy per session of the model. It is evident that using solely a prototype classifier is insufficient for the model to learn accurate class representations, as this imprecision is already apparent in the base classes.

Effectiveness of Prompt Regularization. As shown in Eq. (11), we define the orthogonality $\Omega_\perp$ between the domain prompt and the FSCIL prompt as the mean of the Frobenius norms of each component in $\mathbf{P}_F$ with respect to $\mathbf{P}_D$.

$$\Omega_\perp\left(\mathbf{P}_D,\mathbf{P}_F\right) = \frac{1}{S}\sum_{j=1}^{S} \|\mathbf{P}_D \cdot \mathbf{P}_F^{j\ T}\| \tag{11}$$

Table 5
Comparison of top-1 classification accuracy (Acc), number of trainable parameters (Params), and computational complexity (GFLOPs) for different backbones across various datasets (in %).

Models	CIFAR-10		STL-10		Flowers-102		Caltech-256		GFLOPs (Mac) ↓
	Acc ↑	Params ↓	Acc ↑	Params ↓	Acc ↑	Params ↓	Acc ↑	Params ↓	
ResNet18 [2]	91.68	11.18 M	87.83	11.18 M	84.65	11.23 M	72.05	11.31 M	**1.82 G**
VPT [34]	96.89	99.85 K	98.98	99.85 K	**97.38**	308.84 K	91.69	428.03 K	17.71 G
Domain prompt	**97.82**	**23.05 K**	**99.19**	**23.05 K**	97.35	**93.80 K**	**91.85**	**213.00 K**	16.86 G

Table 6
Ablation study on CUB200 dataset: impact of Domain Prompt (DP), FSCIL Prompt (FP), and prototype classifier (PC) on average accuracy and performance dropping rate (in %).

DP	FP	PC	AA ↑	PD ↓
✓	✓	✓	**75.25**	**15.30**
✓		✓	73.43	16.46
	✓	✓	70.19	17.44
		✓	67.86	19.88

Table 7
Impact of prompt regularization on model performance across CUB-200 and CIFAR-100 datasets (in %).

α	CUB-200		CIFAR-100	
	AA ↑	PD ↓	AA ↑	PD ↓
0.000	74.29	15.51	72.03	24.39
0.001	**75.25**	15.30	72.19	24.55
0.005	75.03	15.04	72.39	24.69
0.010	74.80	15.07	**72.66**	**24.21**
0.050	74.01	**14.81**	72.20	24.23

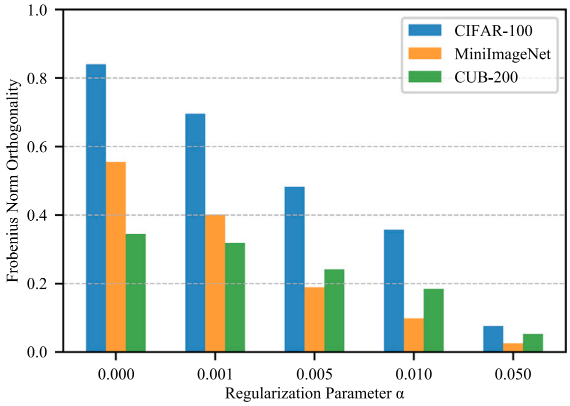

Fig. 6. Frobenius norm orthogonality between two Prompts at varying regularization parameters.

Following the previously stated definition, when $\Omega_{\perp}$ approaches 1, it indicates lower orthogonality, while a value nearing 0 signifies greater orthogonality. Fig. 6 displays the Frobenius norm orthogonality between two prompts across three datasets at varying regularization parameters. Consistently, they all exhibit the same trend whereby the orthogonality $\Omega_{\perp}$ progressively increases with the rise of α. This corroborates the efficacy of the proposed Prompt regularization mechanism.

In addition, in order to reveal the influence of the prompt regularization coefficient α on the model performance, we conduct further ablation experiments. As Table 7 illustrates, a modest α enhances the AA, with the highest recorded for CUB-200 at $\alpha = 0.001$ and for CIFAR-100 at $\alpha = 0.010$. This indicates that a slight prompt regularization contributes to learning precision. Conversely, excessive regularization ($\alpha = 0.050$) slightly diminishes AA, hinting at a threshold beyond which knowledge distinction becomes counterproductive. These results collectively affirm that the prompt regularization mechanism effectively induces orthogonality, enhancing the model's ability to discriminate between general and task-specific knowledge, showcasing its significance for balancing domain and task-specific knowledge.

Sensitive Study of Prompt Length. We conduct a meticulous study of our method on the CUB-200 dataset to investigate the impact of prompt length on performance. We set both L_D and L_F to the same length. Fig. 7 illustrates the trend of overall average accuracy and performance dropping rate with increasing prompt length. It can be observed that as the prompt length increases, the initial increase in model performance is the most notable. Subsequently, the performance fluctuates and peaks when the prompt length reaches 10.

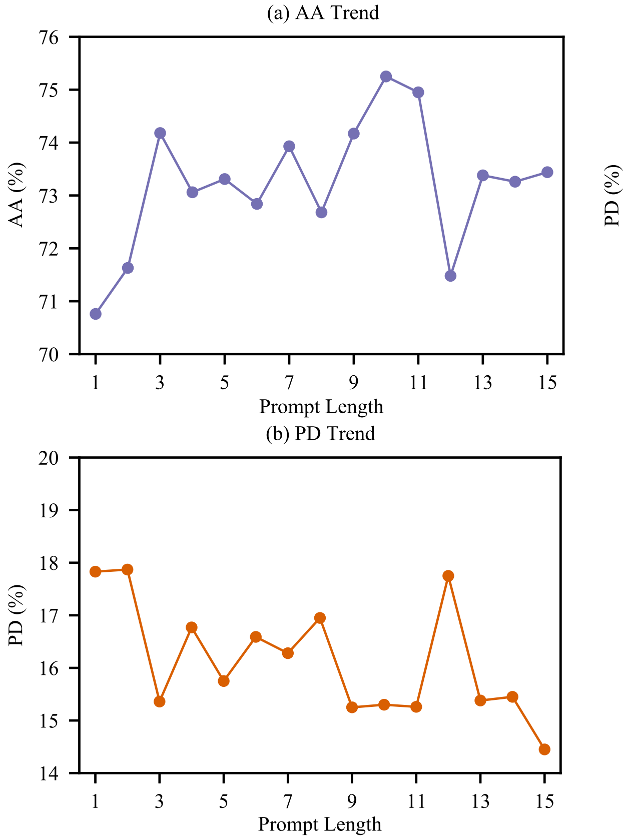

Fig. 7. Trend plots of AA and PD for the model on the CUB-200 dataset as the prompt length increases.

6. Conclusion

In this paper, we introduce PL-FSCIL, a novel method designed to address the challenges of Few-Shot Class-Incremental Learning (FSCIL) by effectively integrating prompts within a pre-trained ViT. The proposed approach enables the model to efficiently learn and adapt to new tasks and domains with limited data. Extensive evaluation on benchmark datasets demonstrates the better precision and objectivity of PL-FSCIL, with the ablation study confirming the critical role of each component in the model's success. Notably, the Domain Prompt and

FSCIL Prompt enhance the model's ability to extract meaningful features from new data and tasks, significantly improving its performance in few-shot learning scenarios.

The success of PL-FSCIL sets a new benchmark for FSCIL tasks and paves the way for future research in prompt-based learning within computer vision. However, there are certain limitations to the approach. One notable challenge is the high computational cost of the ViT model, which may hinder its applicability in resource-constrained environments. Additionally, the simplicity of the prototype classifier may not be sufficient in scenarios with complex data distributions, potentially affecting the model's overall performance. In future work, we aim to address these limitations by refining the prototype classifier and exploring more efficient ways to incorporate prompts. We also plan to extend PL-FSCIL to real-time object recognition tasks and conduct research on medical (e.g., chest X-rays, retinal images) and industrial datasets. These efforts will help improve the model's scalability and robustness in diverse real-world applications.

CRediT authorship contribution statement

Jicheng Yuan: Writing – review & editing, Software, Data curation, Conceptualization. **Hang Chen:** Software, Resources, Formal analysis. **Songsong Tian:** Writing – original draft, Methodology, Data curation. **Wenfa Li:** Writing – review & editing, Resources. **Lusi Li:** Writing – review & editing, Supervision, Methodology. **Enhao Ning:** Software, Project administration, Formal analysis. **Yugui Zhang:** Software, Formal analysis.

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.

Acknowledgment

This study was supported by the Natural Science Foundation of China [Grant number 61972040].

References

[1] Y. LeCun, Y. Bengio, G. Hinton, Deep learning, Nature 521 (7553) (2015) 436–444.

[2] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2016, pp. 770–778.

[3] M. McCloskey, N.J. Cohen, Catastrophic interference in connectionist networks: The sequential learning problem, in: Psychology of Learning and Motivation, vol. 24, Elsevier, 1989, pp. 109–165.

[4] H. Zou, T. Hastie, Regularization and variable selection via the elastic net, J. R. Stat. Soc. Ser. B Stat. Methodol. 67 (2) (2005) 301–320.

[5] G.M. Van de Ven, A.S. Tolias, Three scenarios for continual learning, 2019, arXiv preprint arXiv:1904.07734.

[6] X. Tao, X. Hong, X. Chang, S. Dong, X. Wei, Y. Gong, Few-Shot Class-Incremental Learning, in: 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR, IEEE, Seattle, WA, USA, 2020, pp. 12180–12189, http://dx.doi.org/10.1109/CVPR42600.2020.01220.

[7] C. Zhang, N. Song, G. Lin, Y. Zheng, P. Pan, Y. Xu, Few-shot incremental learning with continually evolved classifiers, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 12455–12464.

[8] H. Liu, L. Gu, Z. Chi, Y. Wang, Y. Yu, J. Chen, J. Tang, Few-shot class-incremental learning via entropy-regularized data-free replay, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXIV, Springer, 2022, pp. 146–162.

[9] Z. Chi, L. Gu, H. Liu, Y. Wang, Y. Yu, J. Tang, Metafscil: A meta-learning approach for few-shot class incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 14166–14175.

[10] G. Zheng, A. Zhang, Few-Shot Class-Incremental Learning with Meta-Learned Class Structures, in: 2021 International Conference on Data Mining Workshops, ICDMW, 2021, pp. 421–430, http://dx.doi.org/10.1109/ICDMW53433.2021.00058.

[11] C. Peng, K. Zhao, T. Wang, M. Li, B.C. Lovell, Few-shot class-incremental learning from an open-set perspective, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXV, Springer, 2022, pp. 382–397.

[12] A. Dosovitskiy, L. Beyer, A. Kolesnikov, D. Weissenborn, X. Zhai, T. Unterthiner, M. Dehghani, M. Minderer, G. Heigold, S. Gelly, J. Uszkoreit, N. Houlsby, An image is worth 16x16 words: Transformers for image recognition at scale, in: International Conference on Learning Representations, 2021, URL https://openreview.net/forum?id=YicbFdNTTy.

[13] T. Brown, B. Mann, N. Ryder, M. Subbiah, J.D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, et al., Language models are few-shot learners, Adv. Neural Inf. Process. Syst. 33 (2020) 1877–1901.

[14] L. Zhao, J. Lu, Y. Xu, Z. Cheng, D. Guo, Y. Niu, X. Fang, Few-shot class-incremental learning via class-aware bilateral distillation, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 11838–11847.

[15] J. Kalla, S. Biswas, S3c: Self-supervised stochastic classifiers for few-shot class-incremental learning, in: European Conference on Computer Vision, Springer, 2022, pp. 432–448.

[16] E. Ning, C. Wang, H. Zhang, X. Ning, P. Tiwari, Occluded person re-identification with deep learning: a survey and perspectives, Expert Syst. Appl. 239 (2024) 122419.

[17] E. Ning, Y. Wang, C. Wang, H. Zhang, X. Ning, Enhancement, integration, expansion: Activating representation of detailed features for occluded person re-identification, Neural Netw. 169 (2024) 532–541.

[18] D.W. Zhou, F.Y. Wang, H.J. Ye, L. Ma, S. Pu, D.C. Zhan, Forward compatible few-shot class-incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9046–9056.

[19] Z. Song, Y. Zhao, Y. Shi, P. Peng, L. Yuan, Y. Tian, Learning with fantasy: Semantic-aware virtual contrastive constraint for few-shot class-incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2023, pp. 24183–24192.

[20] J. Lin, Z. Wu, W. Lin, J. Huang, R. Luo, M2SD: Multiple mixing self-distillation for few-shot class-incremental learning, in: Proceedings of the AAAI Conference on Artificial Intelligence, vol. 38, (4) 2024, pp. 3422–3431.

[21] H. Ran, W. Li, L. Li, S. Tian, X. Ning, P. Tiwari, Learning optimal inter-class margin adaptively for few-shot class-incremental learning via neural collapse-based meta-learning, Inf. Process. Manage. 61 (3) (2024) 103664.

[22] D.W. Zhou, H.J. Ye, L. Ma, D. Xie, S. Pu, D.C. Zhan, Few-shot class-incremental learning by sampling multi-phase tasks, IEEE Trans. Pattern Anal. Mach. Intell. 45 (11) (2022) 12816–12831.

[23] K. Zhu, Y. Cao, W. Zhai, J. Cheng, Z.J. Zha, Self-promoted prototype refinement for few-shot class-incremental learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 6801–6810.

[24] G. Shi, J. Chen, W. Zhang, L.M. Zhan, X.M. Wu, Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima, Adv. Neural Inf. Process. Syst. 34 (2021) 6747–6761.

[25] D.Y. Kim, D.J. Han, J. Seo, J. Moon, Warping the space: Weight space rotation for class-incremental few-shot learning, in: The Eleventh International Conference on Learning Representations, 2023.

[26] B. Liu, B. Yang, L. Xie, R. Wang, Q. Tian, Q. Ye, Learnable distribution calibration for few-shot class-incremental learning, IEEE Trans. Pattern Anal. Mach. Intell. 45 (10) (2023) 12699–12706.

[27] S. Tian, L. Li, W. Li, H. Ran, X. Ning, P. Tiwari, A survey on few-shot class-incremental learning, Neural Netw. 169 (2024) 307–324.

[28] P. Liu, W. Yuan, J. Fu, Z. Jiang, H. Hayashi, G. Neubig, Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing, ACM Comput. Surv. 55 (9) (2023) 1–35.

[29] T. Gao, A. Fisch, D. Chen, Making pre-trained language models better few-shot learners, 2020, arXiv preprint arXiv:2012.15723.

[30] C. Buck, J. Bulian, M. Ciaramita, W. Gajewski, A. Gesmundo, N. Houlsby, W. Wang., Ask the right questions: Active question reformulation with reinforcement learning, in: International Conference on Learning Representations, 2018, URL https://openreview.net/forum?id=S1CChZ-CZ.

[31] J. Dodge, G. Ilharco, R. Schwartz, A. Farhadi, H. Hajishirzi, N. Smith, Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping, 2020, arXiv preprint arXiv:2002.06305.

[32] A. Radford, J.W. Kim, C. Hallacy, A. Ramesh, G. Goh, S. Agarwal, G. Sastry, A. Askell, P. Mishkin, J. Clark, et al., Learning transferable visual models from natural language supervision, in: International Conference on Machine Learning, PMLR, 2021, pp. 8748–8763.

[33] K. Zhou, J. Yang, C.C. Loy, Z. Liu, Learning to prompt for vision-language models, Int. J. Comput. Vis. 130 (9) (2022) 2337–2348.

[34] M. Jia, L. Tang, B.C. Chen, C. Cardie, S. Belongie, B. Hariharan, S.N. Lim, Visual prompt tuning, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXXIII, Springer, 2022, pp. 709–727.

[35] Z. Wang, Z. Zhang, S. Ebrahimi, R. Sun, H. Zhang, C.Y. Lee, X. Ren, G. Su, V. Perot, J. Dy, et al., Dualprompt: Complementary prompting for rehearsal-free continual learning, in: Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI, Springer, 2022, pp. 631–648.

[36] Z. Wang, Z. Zhang, C.Y. Lee, H. Zhang, R. Sun, X. Ren, G. Su, V. Perot, J. Dy, T. Pfister, Learning to prompt for continual learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 139–149.

[37] J. Li, Y. Bai, Y. Lou, X. Linghu, J. He, S. Xu, T. Bai, Memory-based label-text tuning for few-shot class-incremental learning, 2022, arXiv preprint arXiv:2207.01036.

[38] J. Zhu, J. Zhao, J. Zhou, L. He, J. Yang, Z. Zhang, Uncertainty-aware few-shot class-incremental learning, in: ICASSP 2023-2023 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP, IEEE, 2023, pp. 1–5.

[39] H. Ran, X. Gao, L. Li, W. Li, S. Tian, G. Wang, H. Shi, X. Ning, Brain-inspired fast-and slow-update prompt tuning for few-shot class-incremental learning, IEEE Trans. Neural Netw. Learn. Syst. (2024).

[40] I.U. Yoon, T.M. Choi, S.K. Lee, Y.M. Kim, J.H. Kim, Image-object-specific prompt learning for few-shot class-incremental learning, 2023, arXiv preprint arXiv:2309.02833.

[41] B. Lester, R. Al-Rfou, N. Constant, The power of scale for parameter-efficient prompt tuning, 2021, arXiv preprint arXiv:2104.08691.

[42] C. Raffel, N. Shazeer, A. Roberts, K. Lee, S. Narang, M. Matena, Y. Zhou, W. Li, P.J. Liu, Exploring the limits of transfer learning with a unified text-to-text transformer, J. Mach. Learn. Res. 21 (140) (2020) 1–67.

[43] X.L. Li, P. Liang, Prefix-tuning: Optimizing continuous prompts for generation, in: Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), 2021, pp. 4582–4597.

[44] C. Wah, S. Branson, P. Welinder, P. Perona, S. Belongie, The caltech-ucsd birds-200–2011 dataset, Calif. Inst. Technol. (2011).

[45] A. Krizhevsky, G. Hinton, et al., Learning multiple layers of features from tiny images, Handb. Syst. Autoimmune Dis. 1 (4) (2009).

[46] O. Vinyals, C. Blundell, T. Lillicrap, D. Wierstra, et al., Matching networks for one shot learning, Adv. Neural Inf. Process. Syst. 29 (2016).

[47] K. Zhu, Y. Cao, W. Zhai, J. Cheng, Z.J. Zha, Self-Promoted Prototype Refinement for Few-Shot Class-Incremental Learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2021, pp. 6801–6810.

[48] G. shi, J. Chen, W. Zhang, L.M. Zhan, X.M. Wu, Overcoming Catastrophic Forgetting in Incremental Few-Shot Learning by Finding Flat Minima, in: Advances in Neural Information Processing Systems, vol. 34, Curran Associates, Inc., 2021, pp. 6747–6761.

[49] D.W. Zhou, F.Y. Wang, H.J. Ye, L. Ma, S. Pu, D.C. Zhan, Forward Compatible Few-Shot Class-Incremental Learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9046–9056.

[50] D.W. Zhou, H.J. Ye, L. Ma, D. Xie, S. Pu, D.C. Zhan, Few-Shot Class-Incremental Learning by Sampling Multi-Phase Tasks, IEEE Trans. Pattern Anal. Mach. Intell. (2022) 1–16, http://dx.doi.org/10.1109/TPAMI.2022.3200865.

[51] Y. Zou, S. Zhang, Y. Li, R. Li, Margin-based few-shot class-incremental learning with class-level overfitting mitigation, in: A.H. Oh, A. Agarwal, D. Belgrave, K. Cho (Eds.), Advances in Neural Information Processing Systems, 2022, URL https://openreview.net/forum?id=hyc27bDixNR.

[52] B. Yang, M. Lin, Y. Zhang, B. Liu, X. Liang, R. Ji, Q. Ye, Dynamic support network for few-shot class incremental learning, IEEE Trans. Pattern Anal. Mach. Intell. (2022).

[53] Y. Yang, H. Yuan, X. Li, Z. Lin, P. Torr, D. Tao, Neural collapse inspired feature-classifier alignment for few-shot class-incremental learning, in: International Conference on Learning Representations, 2023, URL https://openreview.net/forum?id=y5W8tpojhtJ.

[54] Y. Xu, S. Huang, H. Zhou, CA-CLIP: category-aware adaptation of CLIP model for few-shot class-incremental learning, Multimedia Syst. 30 (3) (2024) 1–14.

[55] W. Qiu, S. Fu, J. Zhang, C. Lei, Q. Peng, Semantic-visual guided transformer for few-shot class-incremental learning, in: 2023 IEEE International Conference on Multimedia and Expo, ICME, IEEE, 2023, pp. 2885–2890.

[56] M. Hersche, G. Karunaratne, G. Cherubini, L. Benini, A. Sebastian, A. Rahimi, Constrained Few-Shot Class-Incremental Learning, in: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2022, pp. 9057–9067.

[57] L. Van der Maaten, G. Hinton, Visualizing data using t-SNE, J. Mach. Learn. Res. 9 (11) (2008).

[58] A. Coates, A. Ng, H. Lee, An analysis of single-layer networks in unsupervised feature learning, in: Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, JMLR Workshop and Conference Proceedings, 2011, pp. 215–223.

[59] M.E. Nilsback, A. Zisserman, Automated flower classification over a large number of classes, in: 2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing, IEEE, 2008, pp. 722–729.

[60] G. Griffin, A. Holub, P. Perona, Caltech-256 Object Category Dataset, California Institute of Technology, 2007.