Tripartite Weight-Space Ensemble for Few-Shot Class-Incremental Learning

Juntae Lee Munawar Hayat Sungrack Yun
Qualcomm AI Research†
{juntlee,hayat,sungrack}@qti.qualcomm.com

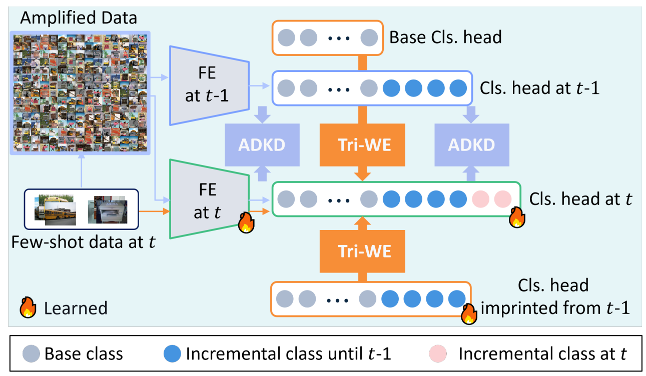


Figure 1. **Overview of the proposed method** (FE: feature extractor, Cls. head: classification head). Through Tri-WE, at tth incremental session, Cls. head is composed by interpolating the weights of the Cls. heads covering the classes until base, $(t-1)$th, tth sessions, respectively. For ADKD regularization loss, the input few-shot data are amplified to reliably mine generalized knowledge of the model of the previous $(t-1)$th session. After incremental learning, deployed components are outlined in green.

Abstract

Few-shot class incremental learning (FSCIL) enables the continual learning of new concepts with only a few training examples. In FSCIL, the model undergoes substantial updates, making it prone to forgetting previous concepts and overfitting to the limited new examples. Most recent trend is typically to disentangle the learning of the representation from the classification head of the model. A well-generalized feature extractor on the base classes (many examples and many classes) is learned, and then fixed during incremental learning. Arguing that the fixed feature extractor restricts the model's adaptability to new classes, we introduce a novel FSCIL method to effectively address catastrophic forgetting and overfitting issues. Our method enables to seamlessly update the entire model with a few examples. We mainly propose a tripartite weight-space ensemble (Tri-WE). Tri-WE interpolates the base, immediately previous, and current models in weight-space, especially for the classification heads of the models. Then, it collaboratively maintains knowledge from the base and previous models. In addition, we recognize the challenges of distilling generalized representations from the previous model from scarce data. Hence, we suggest a regularization loss term using amplified data knowledge distillation. Simply intermixing the few-shot data, we can produce richer data enabling the distillation of critical knowledge from the previous model. Consequently, we attain state-of-the-art results on the miniImageNet, CUB200, and CIFAR100 datasets.

1. Introduction

The ability of a model to continually adapt with the availability of new data is highly desirable for many real-world applications. Few-shot class-incremental learning (FSCIL) has recently gained significant research attention as evidenced by several studies [17, 25, 29, 42, 45], because it offers a promising learning paradigm. It enables the model to flexibly expand its classification capabilities to new concepts using only a few training examples. FSCIL starts by training a model on numerous base classes with abundant labeled data. In each subsequent incremental session, the model encounters only a limited number of labeled data for new classes, along with a single prototype for each of the previously learned classes. Hence, FSCIL faces multiple challenges, including the retention of previously learned knowledge and the prevention of overfitting to the limited examples of new data.

To address these challenges, the most recent works are based on separating learning the representation from the classification. They rely on constructing a robust feature embedding space using plentiful base class data. In subsequent incremental learning where only a few labeled data for new classes are available, the embedding space remains fixed, and the focus is shifted to learning the classifier for the new classes. This strategy is intended to avoid accumulated model collapse or overfitting due to the limited data. For example, [25, 42] refrain from altering any base

†Qualcomm AI Research is an initiative of Qualcomm Technologies, Inc. and/or its subsidiaries.

model parameters and use mean features for each class to define their classifiers. In contrast, [17, 29] employ ensembling multiple classifiers to improve generalization. [45] introduces an extra module to assess the reliability of outputs from the base and incrementally learned models for a weighted sum of the two outputs. Moreover, [1, 40] fixes the feature embedding space and also pre-defines classifier templates, requiring prior knowledge of the total number of incremental classes to be encountered at the outset of the base class learning. However, as indicated in Fig. 2, updating the entire model can lead to a drop in base class performance but may enhance the model's adaptability to new classes. Hence, if we address the catastrophic forgetting and overfitting more effectively, we can harness the benefits of updating the entire model.

To this end, we propose a novel framework for FSCIL as depicted in Fig. 1. Mainly, inspired by Wise-FT [37], we develop *tripartite weight-space ensemble (Tri-WE)* to continually update the weights of the model. We interpolate the classification heads of models from the base, intermediately previous, and current sessions within the weight space, thereby achieving a balance between adaptability to new classes and retention of knowledge of previous ones. This approach allows for a continual update of knowledge without an abrupt shift in the decision boundary. Moreover, to address catastrophic forgetting, knowledge distillation (KD) has been a common regularization technique in class incremental learning. However, its application in a few-shot scenario is prone to overfitting to the few available examples, and then recent cutting-edge methods have less used KD for FSCIL. To counter this, we also introduce *amplified data KD (ADKD) regularization loss* where we intermix few-shot training datasets, enriching the data source for distilling the knowledge from the previous model. It simply ensures that the model prevents catastrophic forgetting by transferring the generalized knowledge for previous classes. We achieve state-of-the-art (SOTA) results on three benchmarks: miniImageNet, CUB200, and CIFAR100.

Our contributions are summarized as three-fold:

- We introduce the Tri-WE, wherein the classification heads from base, preceding, and current models are synergistically interpolated within the weight space, promoting a knowledge continuum.
- Tackling distillation of biased knowledge resulting from a scarcity of data in KD, we develop ADKD sub-loss term that enhances the generality of the distilled knowledge.
- Leveraging these proposed components, we attain SOTA results across on miniImageNet, CUB200, and CIFAR100. Comprehensive experiments demonstrate the significance and efficacy of our contributions.

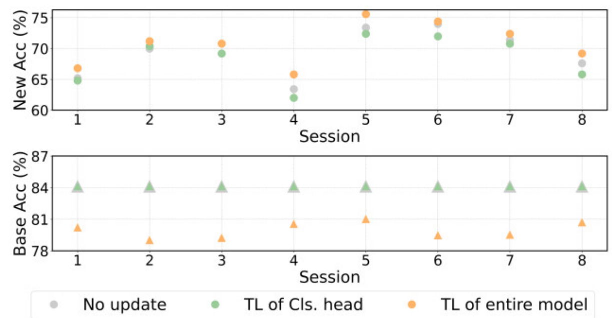


Figure 2. **Few-shot (10-way 5-shot miniImageNet) transfer learning (TL) starting from the base session model on each session.** Classification accuracy scores on incremented 10 classes (top) & 40 base classes (bottom) are shown for each session.

2. Related Works

Class-incremental learning (CIL) aims to seamlessly integrate new classes without access to previously seen data, preventing the catastrophic forgetting problem. To ensure knowledge retention, rehearsal of selective previous samples [3, 28] has been developed. Also, in [35, 39], the network is dynamically expanded to accommodate new learnings. LwF [22] and iCaRL [27] used KD and nearest class mean (NCM) classifiers, respectively, to mitigate the catastrophic forgetting. UCIR [15] introduced a novel feature distillation loss, and PODNet [12] extended distillation beyond the final embedding output, incorporating the pooled outputs from the model's intermediate layers. In [2, 38, 44], the class imbalance problem is tackled. However, these methods depend on ample training data, an aspect not feasible in FSCIL, necessitating new strategies that adapt swiftly with minimal data while preserving existing knowledge.

FSCIL fuses CIL and few-shot learning. It gained attention after TOPIC [32] initiated the concept, leading to numerous subsequent methods. Recent works align with two strategies. One approach completely fixes the learnable weights of the model after the base session's many-shot many-class training, and then incrementally extends the classifier by adding mean features of new classes. CEC [42] enhances the base model by simulating the FSCIL scenarios sampled from the base data again, while ALICE [25] employs diverse techniques to boost the generalization capability of the base model. Alternatively, some methods [17, 29, 40, 45] update a part of the feature extractor or the added class-mean-initialized classifiers in each session. SoftNet [18] finds the weights less impactful for the catastrophic forgetting at the base session, and focuses on updating them in the incremental learning. In WaRP [19], the important weights are decided in every session, and they keep frozen in the following sessions. GKEAL [47] updates the weight of the classification head using least square solver in the view of analytic learning. S3C [17] employed the stochastic classifier as the extreme ensemble of classifiers. NS-FSCIL [40] and OrCo [1] assume the base session knows the amount of total incremental classes and incremental session starts with mapping each class to a classifier predefined in the base session. NS-FSCIL optimizes the last module of the feature encoder and OrCo does the last module and the selected pre-defined classifier. In SAVC [29], a class is recognized by a set of classifiers, each tailored to a specific input augmentation, resulting in a swift expansion of deployed model size during incremental learning. In BiDistill [45], every prediction is made by the ensemble of the base and last models.

Network ensemble has long been an important technique in deep learning in order to reduce overfitting and improve generalization ability [10, 13]. It involves the training of multiple neural networks and combining them for the final prediction. Various ensemble methods have been developed to get different but collaborative models such as random-weight initialization [21], different training data [30], and capturing multiple snapshots in training schedules [16]. While these methods have consistently demonstrated improved performance over single-model approaches, they often come at the cost of increased computational overhead, both in terms of training and inference. Recently, Model Soup [36] suggested interpolating the weights of a bunch of networks finetuned with different hyperparameters configurations. Also, to mitigate the overfitting of the finetuned CLIP [26], WiseFT [37] combines the weights of fine-tuned and initial CLIP models. These approaches can attain better generalization capability without increasing the computational cost of deployed models.

3. Proposed Method

In this section, we describe the proposed Tri-WE and ADKD for FSCIL. The pipeline is illustrated in Fig. 3. We also provide the process of the base session training.

Problem setting. FSCIL aims to incrementally scale the model's class recognition ability by learning new classes with limited training data while preserving its knowledge of previously learned, old classes. In specific, we suppose that a model $f_{\theta,\phi}$ consists of the feature extractor g_θ and the classification head (cosine classifier) h_ϕ where θ and ϕ denote learnable weights of them, respectively. Initially, the model is pre-trained on the base training set in the base session $\mathcal{S}^{(0)}$. Then, the model progresses a series of T incremental sessions, $\{\mathcal{S}^{(1)},\dots,\mathcal{S}^{(T)}\}$. Current session $\mathcal{S}^{(t)}$ includes a class set $\mathcal{C}^{(t)}$ and the corresponding training dataset $\mathcal{D}^{(t)}=\{(x,y)\}$ where x is a sample and y is its label.

In the base session $\mathcal{S}^{(0)}$, the class set $\mathcal{C}^{(0)}$ includes a substantial number of N_0 base classes with ample training data for each class. In contrast, in the incremental session $\mathcal{S}^{(t)}$ ($t \geq 1$), the class set $\mathcal{C}^{(t)}$ contains N classes with only a small K number of training data available for every class (N-way K-shot setting). Notice that $\mathcal{C}^{(i)}$ and $\mathcal{C}^{(j)}$ do not overlap when $i \neq j$, $\forall i,j$. Following the prior FSCIL works [32, 42], the model at each session $\mathcal{S}^{(t)}$ only has access to $\mathcal{D}^{(t)}$ and an additional prototype (the averaged feature of the K examples per class) buffer $\mathcal{M}$ memorizing the classes of the base and previous $(t-1)$ sessions. A single prototype is maintained for every class in $\mathcal{M}$. Once the incremental learning at the session $\mathcal{S}^{(t)}$ concludes, the model is evaluated on query data from all the seen N_t classes thus far, that is $\mathcal{C}^{(0:t)}=\mathcal{C}^{(0)}\cup\mathcal{C}^{(1)}\cdots\cup\mathcal{C}^{(t)}$.

3.1. Tripartite weight-space ensemble (Tri-WE)

The base session model is generalized by numerous classes with sufficient data, and hence its weight space is conducive to adapting to new classes. Although the weight space is somewhat compromised, the intermediate previous session's model has knowledge for all the classes encountered thus far. To capitalize on the strengths of both models, we suggest the weight-space ensemble of three models from the sessions $\mathcal{S}^{(0)}$, $\mathcal{S}^{(t-1)}$, and $\mathcal{S}^{(t)}$.

However, the feature extractor, which contains many layers, is prone to overfitting or collapsing when only a few examples are available. Hence, while reducing the learning rate for the feature extractor $g_\theta^{(t)}$, we only ensemble the classification heads h_{ϕ_0}, $h_{\phi_\text{old}}^{(t)}$, $h_{\phi_\text{all}}^{(t)}$ in the weight space. Here, h_{ϕ_0} is the classification head from the base session model. $h_{\phi_\text{old}}^{(t)}$ is initialized by the classification head $h_\phi^{(t-1)}$ from the session $\mathcal{S}^{(t-1)}$, and then specialized to the old classes $\mathcal{C}^{(0:t-1)}$. $h_{\phi_\text{all}}^{(t)}$ covers all the target classes $\mathcal{C}^{(0:t)}$, where the old class part is initialized similarly to $h_{\phi_\text{old}}^{(t)}$. Then, as depicted in Fig. 3, Tri-WE interpolates the classification head weight ϕ^n for nth class by

$$\phi^n = \begin{cases} \bar{\alpha}_1\phi_0^n + \bar{\alpha}_2\phi_\text{old}^n + \bar{\alpha}_3\phi_\text{all}^n & \text{if } n \leq N_0, \\ \bar{\alpha}_4\phi_\text{old}^n + \bar{\alpha}_5\phi_\text{all}^n & \text{if } N_0 < n \leq N_{t-1}, \\ \phi_\text{all}^n & \text{otherwise,} \end{cases} \tag{1}$$

where $1 \leq n \leq N_t$, $\phi^n \in \mathbb{R}^d$, and d is the dimensionality of ϕ^n. Then, the weight of the classification head $h_\phi^{(t)}$ to classify N_t classes consists of $\{\phi^n\}_{n=1}^{N_t}$. Notice that, we also add two learnable scalar weights α_1 and α_2 to automatically adjust the trade-offs between the three classification

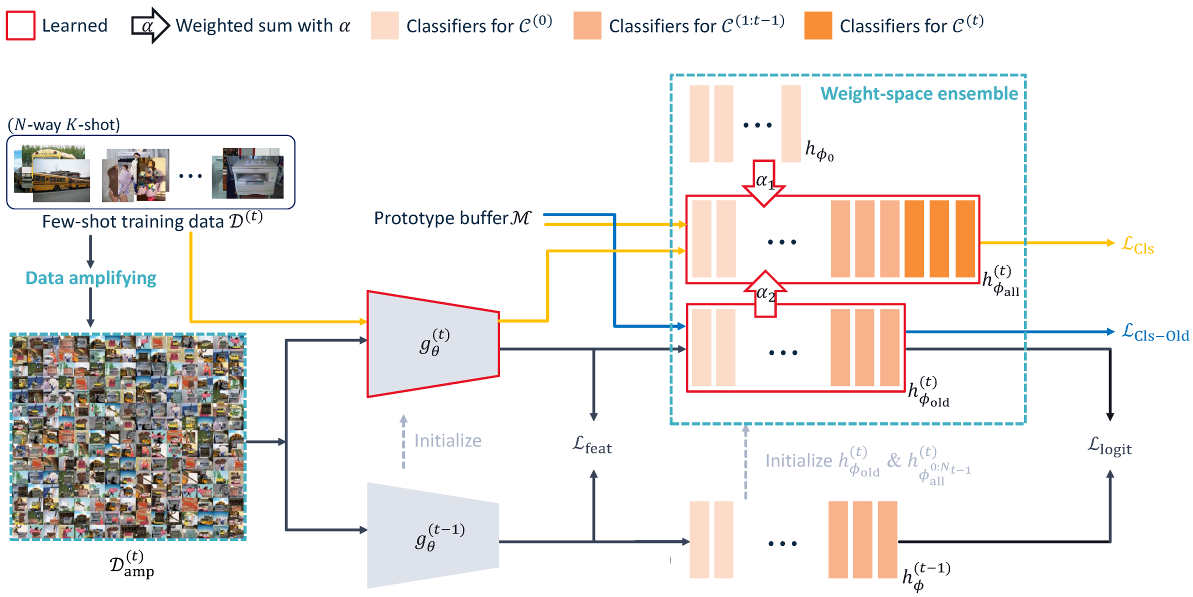


Figure 3. **The pipeline of the proposed method.** At the tth incremental session $S^{(t)}$, the model is initialized using the resulting model of session $S^{(t-1)}$. Then, a set of N-way K-shot few-shot examples $D^{(t)}$ and the prototype buffer $\mathcal{M}$ are given. The weights ϕ_0, ϕ_{old}, ϕ_{all} of classification heads h_{ϕ_0}, $h^{(t)}_{\phi_{\text{old}}}$, $h^{(t)}_{\phi_{\text{all}}}$ are interpolated with learnable scalars α_1 and α_2. The main loss $\mathcal{L}_{\text{Cls}}$ is computed on the resulting classification head $h^{(t)}_\phi$'s output. Also, classification loss over the old classes, appeared until the session $S^{(t-1)}$, is computed on the output of $h^{(t)}_{\phi_{\text{old}}}$. Additionally, $D^{(t)}$ is amplified to $D^{(t)}_{\text{amp}}$, and then KD losses $\mathcal{L}_{\text{feat}}$ and $\mathcal{L}_{\text{logit}}$ are computed on the amplified dataset $D^{(t)}_{\text{amp}}$. The learned components are outlined in red.

heads. Then, $\{\bar\alpha_1, \bar\alpha_2, \bar\alpha_3, \bar\alpha_4, \bar\alpha_5\}$ are obtained by

$$\begin{aligned}
\bar\alpha_1 &= \alpha_1/(\alpha_1+\alpha_2+1),\\
\bar\alpha_2 &= \alpha_2/(\alpha_1+\alpha_2+1),\\
\bar\alpha_3 &= 1/(\alpha_1+\alpha_2+1),\\
\bar\alpha_4 &= \alpha_2/(\alpha_2+1),\\
\bar\alpha_5 &= 1/(\alpha_2+1).
\end{aligned} \tag{2}$$

When $t=1$ (i.e. $\phi_{\text{old}} = \phi_0$), we set $\alpha_2 = 0$.

Then, the main classification loss $\mathcal{L}_{\text{Cls}}$ is computed on the output from the weight-space ensembled classification head $h^{(t)}_\phi$ by

$$\begin{aligned}
\mathcal{L}_{\text{Cls}} = \mathbb{E}_{(x,y)\sim\mathcal{D}^{(t)}}&[\text{CE}(h^{(t)}_\phi(g^{(t)}_\theta(x)), y]\\
&+ \mathbb{E}_{(\mathbf{p},y)\sim\mathcal{M}}[\text{CE}(h^{(t)}_\phi(\mathbf{p}), y],
\end{aligned} \tag{3}$$

where CE is the cross-entropy loss function, and $\mathbf{p}$ denotes the old class prototype in $\mathcal{M}$.

Using the buffer $\mathcal{M}$, we also calculate the old class-specific loss $\mathcal{L}_{\text{Cls-Old}}$ for the classification head $h^{(t)}_{\phi_{\text{old}}}$ by

$$\mathcal{L}_{\text{Cls-Old}} = \mathbb{E}_{(\mathbf{p},y)\sim\mathcal{M}}[\text{CE}(h^{(t)}_{\phi_{\text{old}}}(\mathbf{p}), y)]. \tag{4}$$

In this loss, $h^{(t)}_{\phi_{\text{old}}}$ focuses on preserving the discriminative ability regarding all the old classes on top of the current feature extractor $g^{(t)}_\theta$. Hence, we do not update h_{ϕ_0} and instead use it to retain the knowledge for the original, well-designed base class distribution from the base session model.

Unlike the naive ensemble approaches where multiple classification heads independently make predictions and combine them in the testing phase, our weight-space ensemble deploys only $h^{(t)}_\phi$ collaboratively utilizing the knowledge from the base and previous session models.

3.2. Total training loss

KD has been a widely used regularization technique in CIL to prevent the catastrophic forgetting problem [27]. Specifically, when new classes are introduced, an additional distillation loss is combined with the main classification loss to ensure that the current model replicates the old class behavior of the previous model. Nevertheless, in FSCIL, the limited data source can lead to the extraction of knowledge biased toward those data, rather than generalized knowledge of the old classes. To mitigate this issue, we simply generate an amplified data source $\mathcal{D}^{(t)}_{\text{amp}}$ from the few-shot training data by randomly mixing pairs of data from $\mathcal{D}^{(t)}$ employing simple data augmentation techniques such as [41, 43]. The ADKD loss is then computed as

$$\mathcal{L}_{\text{ADKD}} = \mathcal{L}_{\text{feat}} + \mathcal{L}_{\text{logit}}, \tag{5}$$

where $\mathcal{L}_{\text{feat}}$ and $\mathcal{L}_{\text{logit}}$ are feature-level and logit-level KD losses which are defined by

$$\mathcal{L}_{\text{feat}} = \mathbb{E}_{x\sim\mathcal{D}_{\text{amp}}^{(t)}} \, ||g_{\theta}^{(t-1)}(x) - g_{\theta}^{(t)}(x)||_2 \tag{6}$$

$$\mathcal{L}_{\text{logit}} = \mathbb{E}_{x\sim\mathcal{D}_{\text{amp}}^{(t)}} \, - f_{\theta,\phi}^{(t-1)}(x) \log f_{\theta,\phi^{0:N_{t-1}}}^{(t)}(x). \tag{7}$$

The previous session model $f_{\theta,\phi}^{(t-1)}$ including $h_{\phi}^{(t-1)}$ and $g_{\theta}^{(t-1)}$ is kept frozen during this process, unlike $h_{\phi_{\text{old}}}^{(t)}$.

Finally, the total loss function used to optimize the current model is

$$\mathcal{L} = \mathcal{L}_{\text{Cls}} + \gamma_1 \mathcal{L}_{\text{Cls-Old}} + \gamma_2 \mathcal{L}_{\text{ADKD}}, \tag{8}$$

where γ_1 and γ_2 are hyper-parameters that balance the respective losses, $\mathcal{L}_{\text{Cls-Old}}$ and $\mathcal{L}_{\text{ADKD}}$. We provide the pseudo code of the overall process during an incremental session in the supplementary materials.

3.3. Base session training

In FSCIL, it is broadly acknowledged that a model with strong generalization to base classes is beneficial for learning new classes [25, 29, 40, 42]. To ensure fairness when assessing the proposed components for incremental learning, we strive for base session performance on par with the prior SOTA methods [29, 40, 45]. For this purpose, we employ the base session training techniques of ALICE [25] (see the literature for details). On top of it, we also added a geometric classification head in parallel with the main classification head. In specific, an input is geometrically transformed into B types, and we use them as the proxy labels of transformed inputs. Then, the geometric classification head is learned to classify each into one of the B proxy labels. This auxiliary geometric classification helps the model encode the inherent structure of data so that the model's generalization capability on new tasks is improved. This auxiliary geometric classification strategy is not used in the incremental sessions. Through this, as aftermentioned in Sec. 4, the base session model achieves comparable or superior performance to previous SOTA methods on three benchmark datasets. A more detailed description is in the supplementary materials.

4. Experimental Results

Datasets. Consistent with the setting of [32], we conduct our experiments on three benchmark datasets: miniImageNet [33], CIFAR100 [20] and CUB200 [34]. miniImageNet is a subset of ImageNet [8] that includes 60,000 images from 100 classes, each with resolution 84×84. CIFAR100 comprises 60,000 tiny images of size 32×32 across 100 categories. CUB200, which is focused on a fine-grained classification, contains 200 bird species with similar appearance where the image size is 224×224. We follow the splits from [32]. For miniImageNet and CIFAR100, 60 categories are selected as base classes while the remaining are split into 8 incremental sessions with only 5 training examples per novel class (i.e., 5-way 5-shot). As for the CUB200 dataset, 100 categories are designated as the base training set, while the rest are divided into 10-way 5-shot tasks across 10 sessions.

Evaluation protocol. In line with the FSCIL literature, we employ the top-1 accuracy (Acc) for all the observed classes at the end of each incremental learning session. We also provide the average of Acc over all the sessions.

Implementation details. We implement our method using the PyTorch library. In keeping with the previous works [18, 25, 29, 45], ResNet18 [14] serves as our feature extractor. The entire model is trained via the SGD optimizer. For the base session, the learning rates are initialized by 0.01, 0.001, and 0.01 for miniImageNet, CUB200, and CIFAR100 respectively, and decayed by 0.1 at 60 and 70 epochs. In each incremental session, the learning rate for the weight-space ensembled ϕ_{all} is 0.1, while it is 0.001 for the rest. Hence, the feature extractor is minimally updated in incremental learning. For the ADKD loss, the given NK training examples are amplified to $16NK$ for each class. Also, the learnable scalars α_1 and α_2 are initially set to 1.0. Loss weights λ_1 and λ_2 are empirically determined to be 1.2 and 10.0, respectively. Further details are in Sec. S-5 of the supplementary materials.

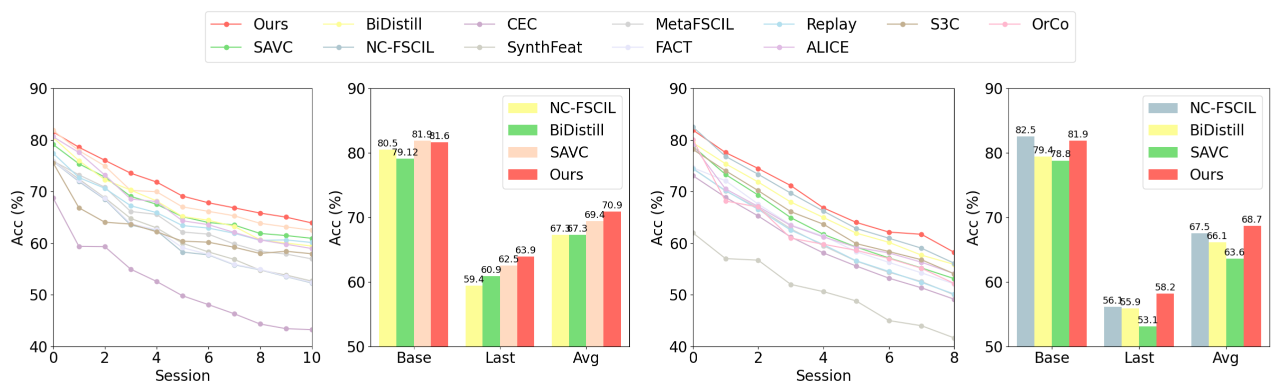
4.1. Comparative assessment

On the three benchmark datasets, we first compare the proposed method with the existing approaches. As indicated in Table 1 on the miniImageNet dataset, our method outperforms the compared in all the sessions, achieving a clear SOTA in terms of the average Acc by a margin of 1.82% at least. Also, BiDistill and SAVC require the base session model and a considerable number of classifiers per class, respectively, even in the testing phase. NC-FSCIL and OrCo benefit from knowing the total number of incremental classes at the outset of the base session. Whereas, we attain clear SOTA with a smaller-scale deployed model compared to BiDistill and SAVC, and without the information advantage of NC-FSCIL and OrCo.

Moreover, Fig. 4 shows the Acc curve according to sessions on the CUB200 and CIFAR100 datasets. As in the miniImageNet dataset, our method consistently surpasses the compared methods by large gaps in the incremental sessions on the CUB 200 dataset. In specific, despite lower base session Acc than SAVC, our method outperforms it in the last session. Also, while NC-FSCIL merely finetuned the last module of the feature encoder in incremental sessions, OrCo attempted to also optimize the pre-set classifiers. Nevertheless, the additional update of only the classifiers is less effective. On the CIFAR100 dataset, we can

Table 1. **Comparative results on miniImageNet dataset.** Session-wise Acc (%) and the average of them (Avg) are reported. In each case, the best and second results are bold-faced and underlined, respectively. Last sess. impro. denotes the performance improvement of our method at the last session.

Method	Venue	Acc in each session (%) 0	1	2	3	4	5	6	7	8	Avg	Last sess. impro.
iCaRL [27]	CVPR 2017	61.31	46.32	42.94	37.63	30.49	24.00	20.89	18.80	17.20	33.29	+43.13
TOPIC [32]	CVPR 2020	61.31	50.09	45.17	41.16	37.48	35.52	32.19	29.46	24.42	39.64	+35.71
ERL++ [11]	AAAI 2021	61.70	57.58	54.66	51.72	48.66	46.27	44.67	42.81	40.79	49.87	+19.34
IDLVQ [4]	ICLR 2020	64.77	59.87	55.93	52.62	49.88	47.55	44.83	43.14	41.84	51.16	+18.29
CEC [42]	CVPR 2021	72.00	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	57.74	+12.50
SynthFeat [5]	ICCV 2021	61.40	59.80	54.20	51.69	49.45	48.00	45.20	43.80	42.10	50.63	+18.03
MetaFSCIL [6]	CVPR 2022	72.04	67.94	63.77	60.29	57.58	55.16	52.9	50.79	49.19	58.85	+10.94
FACT [46]	CVPR 2022	72.56	69.63	66.38	62.77	60.60	57.33	54.34	52.16	50.49	60.70	+9.64
Replay [23]	ECCV 2022	71.84	67.12	63.21	59.77	57.01	53.95	51.55	49.52	48.21	58.02	+11.92
ALICE [25]	ECCV 2022	80.60	70.60	67.40	64.50	62.50	60.00	57.80	56.80	55.70	63.99	+4.43
S3C [17]	ECCV 2022	76.55	71.74	67.66	64.52	61.51	58.09	55.36	53.04	51.34	62.20	+8.79
WaRP [19]	ICLR 2023	72.99	68.10	64.31	61.30	58.64	56.08	53.40	51.72	50.65	59.69	+9.48
SoftNet [18]	ICLR 2023	76.63	70.13	65.92	62.52	59.49	56.56	53.71	51.72	50.48	60.79	+9.65
NC-FSCIL [40]	ICLR 2023	<u>84.02</u>	<u>76.80</u>	72.00	67.83	66.35	<u>64.04</u>	<u>61.46</u>	<u>59.54</u>	<u>58.31</u>	<u>67.82</u>	+1.82
GKEAL [47]	CVPR 2023	73.59	68.90	65.33	62.29	59.39	56.70	54.20	52.59	51.31	60.48	+10.40
BiDistill [45]	CVPR 2023	74.65	70.43	66.29	62.77	60.75	57.24	54.79	53.65	52.22	61.42	+7.91
SAVC [29]	CVPR 2023	81.12	76.14	<u>72.43</u>	<u>68.92</u>	<u>66.48</u>	62.95	59.92	58.39	57.11	67.05	+3.02
OrCo [1]	CVPR 2024	83.30	75.32	71.52	68.16	65.62	63.11	60.20	58.82	58.08	67.12	+2.05
CLOSER [24]	ECCV 2024	76.02	71.61	67.99	64.69	61.70	58.94	56.23	54.52	53.33	62.78	+6.80
Ours		**84.13**	**81.41**	**76.65**	**73.59**	**70.10**	**65.13**	**63.42**	**61.02**	**60.13**	**70.62**	



(a) CUB200

(b) CIFAR100

Figure 4. **Comparative results on CUB200 and CIFAR100 datasets.** For each dataset, the curve of the session-wise Acc (%) is first provided. And also, the base, last, average Acc are reported on our method comparing with three recent SOTA methods, NC-FSCIL [40], SAVC [29], YourSelf [31] (Best viewed in color).

see the similar tendency. NC-FSCIL's setting is somewhat favorable, as it allows the pre-definition of classifiers for the incremental classes with knowledge of all class numbers in advance at the base session. This advantage is particularly beneficial on the CIFAR100 dataset, where some classes are semantically similar (e.g., possum and mouse), and the low resolution (32×32) of the images could otherwise complicate classification. The detailed results are available in Sec. S-4 of the supplementary materials. These findings demonstrate our method's superior ability to adapt to new classes with limited training data while maintaining seamless integration with the old classes.

4.2. Ablation analysis

We extensively analyze the components of the proposed method on the miniImageNet or CUB200 datasets.

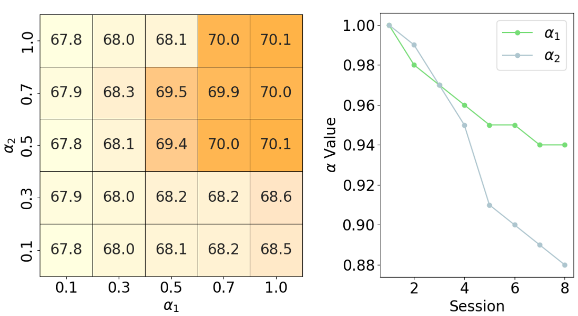
Tripartite strategy. We conduct the ablation experiments on the Tri-WE in Table 2. Before this, initially, we contrast "Naive" with the "No WE" configuration, the latter incorporating the ADKD loss. The naive approach, lacking both WE and ADKD loss, exhibits a significant performance decline due to simultaneous updates of the feature extractor and classification head. However, No WE achieves respectable results thanks to the ADKD loss (more discussed in the following sections). Next, as demonstrated

Table 2. **Ablation analysis for Tri-WE.** Varying the combination of classification heads ensembled in weight space (No WE, Dual-WE, Tri-WE), session-wise and average Acc (%) are provided on miniImageNet. Also, “Naive” means that none of WE and ADKD is used.

	Classification head weight			Acc in each session (%)									Avg
	ϕ_0	ϕ_{old}	ϕ_{all}	0	1	2	3	4	5	6	7	8	
Naive				84.13	80.55	71.26	60.93	50.05	44.08	36.42	19.78	16.73	51.55
No WE			✓	84.13	78.95	74.09	70.06	66.56	63.07	60.32	58.08	56.01	67.92
Dual-WE		✓	✓	84.13	80.25	75.47	71.58	67.62	64.52	61.11	60.17	58.93	69.31
	✓		✓	84.13	79.54	74.99	70.13	66.60	63.43	60.85	59.89	57.31	68.54
Tri-WE	✓	✓	✓	84.13	81.41	76.65	73.59	70.1	65.13	63.42	61.02	60.13	70.62

Table 3. **Impact on increasing the layers ensembled in weight space.** Applying the proposed Tri-WE to the convolutional blocks of the feature extractor (ResNet18), session-wise and average Acc (%) are provided on miniImageNet.

Weight-space ensembled	Acc in each session (%)									Avg
	0	1	2	3	4	5	6	7	8	
Blocks 1-4	84.13	75.25	70.25	67.54	64.85	61.34	59.74	56.38	53.45	65.88
Blocks 1-4 + clf. head	84.13	76.08	72.13	68.42	65.83	62.60	60.19	57.75	55.29	66.94
Blocks 2-4 + clf. head	84.13	77.84	73.42	69.22	65.85	62.75	60.56	57.34	55.33	67.38
Blocks 3-4 + clf. head	84.13	78.59	74.43	69.80	66.25	63.13	60.46	58.04	55.65	67.83
Block 4 + clf. head	84.13	78.80	74.21	69.64	66.01	63.32	61.50	58.17	55.88	67.96
Clf. head only	84.13	81.41	76.65	73.59	70.1	65.13	63.42	61.02	60.13	70.62



(a) Fixed (α_1, α_2) (b) Learned α_1, α_2

Figure 5. **Analysis on learnable scalars α_1 and α_2 on miniImageNet.** (a) Comparison with multiple combinations of fixed (not-learnable) α_1 and α_2 in terms of averaged Acc over all the sessions, (b) session-wise learned α_1 and α_2 in ours.

in Table 2, the absence of the weight-space ensemble for the classification head leads to increased catastrophic forgetting. And, from “Dual-WE,” we can see that the model from the intermediate previous session is more important than the base session model, as the reduced learning rate for the feature extractor mitigates drastic changes, helping to preserve the integrity of the feature space. Accordingly, the knowledge for incrementally-appeared classes is more helpful. Lastly, the complete Tri-WE clearly outperforms No WE by 2.70% in average Acc, effectively retaining the essential knowledge of the base and previous session models.

WE in classification head. We also evaluate the effect of weight-space ensemble varying the number of interpolated layers in Table 3. In specific, we apply the proposed Tri-WE on the convolutional blocks of the feature extractors. Here, we use different learnable scalars depending on the convolutional blocks, rather than sharing them across the convolutional blocks and classification head. We observed a tendency for the decline in performance as the number of weight-space-ensembled layers increases. This is because that deciding how to merge a lot of layers is difficult with only a few examples. Also, only merging feature extractor (1st row of Table 3) shows the lowest performance. It emphasizes that the head's weight-space is crucial to prevent catastrophic forgetting.

Table 4. **Analysis on data amplification schemes.** The last and average Acc (%) are reported on each scheme on miniImageNet.

	w/o AD	AD scheme			
		CutOut [9]	RandAug [7]	MixUp [43]	CutMix [41]
Last Acc	58.08	58.12	58.26	59.14	60.13
Avg Acc	69.05	69.13	69.09	70.03	70.62

Learnable scalars α_1&α_2. Fig. 5(a) shows the averaged Acc on diverse combinations of (α_1, α_2) where both are fixed after initialization in all the sessions. Compared to all the fixed cases, the learnable scalars show better result (70.62% in the averaged Acc). Also, as Fig. 5(b), we show the learned α_1 and α_2 in each session in our method where we initialize both as 1.0 in every session. We can see that the learned α_1 and α_2 are different depending on sessions. Accordingly, our learnable scalar approach is useful for bal-

Table 5. **Analysis on loss terms of $\mathcal{L}_{\text{ADKD}}$.** The last and average Acc (%) are reported ablating the loss terms on miniImageNet.

$\mathcal{L}_{\text{feat}}$	$\mathcal{L}_{\text{logit}}$	Last Acc	Avg Acc
✓		59.42	70.01
	✓	58.77	68.41
✓	✓	60.13	70.62

ancing the trade-off among the three classification heads. We also provide session-wise results on the fixed cases in the supplementary materials.

Effect of data amplification. While Tri-WE already attains SOTA (i.e. w/o AD in Table 4), incorporating our ADKD loss provides further increase (69.05% *vs* 70.62% in average Acc). Also, to see the influence of the choice of data amplification scheme in the ADKD loss, we compare multiple data augmentation methods for our data amplification. To facilitate the use of ADKD in follow-up works, we consider four widely-used data augmentation methods, CutOut [9], RandAug [7], CutMix [41], MixUp [43]. In the first two, the images are not inter-mixed. Whereas, two images are mixed for data augmentation. In Table 4, we can see that CutOut and RandAug are not effective compared to "w/o AD" where no data amplification is considered. With only a few examples, those methods cannot yield a diversity of data. Next, in comparison with MixUp, CutMix shows a better result. MixUp mixes two images smoothly, and then the original semantics are somewhat preserved. This is less useful in KD. Namely, CutMix encourages the model to focus more on general image recognition ability rather than high-level semantics, which is more helpful in preventing the catastrophic forgetting problem in FSCIL. Hence, we employ CutMix as the default option. In addition, the impact of the degree of data amplification is addressed in Sec. S-1 of the supplementary materials.

Two loss terms of $\mathcal{L}_{\text{ADKD}}$. As shown in the first two rows of Table 5, we ablate $\mathcal{L}_{\text{feat}}$ and $\mathcal{L}_{\text{logit}}$, respectively. From this study, we can see that $\mathcal{L}_{\text{feat}}$ is slightly more important than $\mathcal{L}_{\text{logit}}$ since Tri-WE already helps to prevent catastrophic forgetting in the side of the classification head. Then, using both is most effective.

$\mathcal{L}_{\text{Cls-Old}}$. As we update the feature extractor as well as the classification head in the incremental learning, we also adapt ϕ_{old} to the updated feature extractor using the loss $\mathcal{L}_{\text{Cls-Old}}$. To verify this, we conduct ablating the feature extractor update and $\mathcal{L}_{\text{Cls-Old}}$. As shown in Table 6, the performance is degraded without this loss term when the feature extractor is updated. Also, when the feature extractor is frozen, updating ϕ_{old} is less meaningful. Finally, updating the feature extractor with ϕ_{old} is most beneficial.

New class adaptability. To see the efficacy of our method in terms of the trade-off between base and newly introduced classes, we provide the individual accuracy of the base and novel classes, and the harmonic mean (HM). Since most previous works have addressed the joint accuracy of the base and novel classes, we can compare our method with only a few recent methods [17, 33, 42, 45, 46] on the CUB200 dataset, following [17, 46]. Fig. 6 shows that our approach is more adept at integrating new classes without compromising the performance on the base classes.

Table 6. **Analysis on $\mathcal{L}_{\text{Cls-Old}}$ and updating the feature extractor g_θ.** The last and average Acc (%) are reported, on miniImageNet.

$\mathcal{L}_{\text{Cls-Old}}$	Update g_θ	Last Acc	Avg Acc
	✓	59.27	70.15
✓	✓	60.13	70.62
		58.86	69.41
✓		58.88	69.40

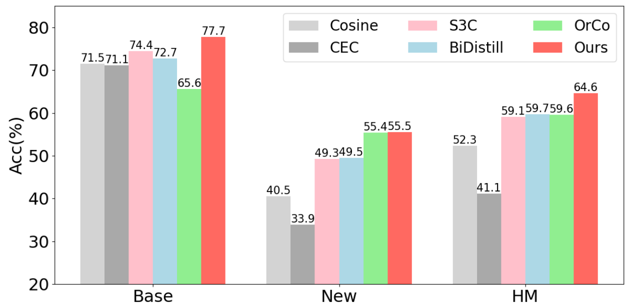


Figure 6. **Analysis on the new class adaptability on CUB200.** The average of accuracy scores are provided for base and novel classes, denoted by Base and New respectively. And, harmonic means are also compared.

5. Conclusions

We explored FSCIL which facilitates the continuous acquisition of new concepts from a few examples. To this end, we suggested a novel FSCIL method including Tri-WE and ADKD regularization loss. The proposed method can effectively prevent overfitting to the few data and catastrophic forgetting of old concepts in the model. Tri-WE strategically interpolates the weights of the classification heads from the base, immediate previous, and current models, which collectively preserves the knowledge from the earlier models without overwhelming updates. Enriching the few examples for KD, the ADKD loss facilitates the transfer of essential knowledge from the previous model, rather than the knowledge overfitted to the few examples. We conducted extensive analyses on the proposed components and achieved state-of-the-art performance on the miniImageNet, CUB200, and CIFAR100 datasets.

References

[1] Noor Ahmed, Anna Kukleva, and Bernt Schiele. Orco: Towards better generalization via orthogonality and contrast for

few-shot class-incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2024.

[2] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[3] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. *arXiv preprint arXiv:1812.00420*, 2018.

[4] Kuilin Chen and Chi-Guhn Lee. Incremental few-shot learning via vector quantization in deep embedded space. In *Int. Conf. Learn. Represent.*, 2020.

[5] Ali Cheraghian, Shafin Rahman, Sameera Ramasinghe, Pengfei Fang, Christian Simon, Lars Petersson, and Mehrtash Harandi. Synthesized feature based few-shot class-incremental learning on a mixture of subspaces. In *Int. Conf. Comput. Vis.*, 2021.

[6] Zhixiang Chi, Li Gu, Huan Liu, Yang Wang, Yuanhao Yu, and Jin Tang. Metafscil: A meta-learning approach for few-shot class incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[7] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *IEEE Conf. Comput. Vis. Pattern Recog. Worksh.*, 2020.

[8] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2009.

[9] Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

[10] Thomas G Dietterich. Ensemble methods in machine learning. In *International Workshop on Multiple Classifier Systems*, 2000.

[11] Songlin Dong, Xiaopeng Hong, Xiaoyu Tao, Xinyuan Chang, Xing Wei, and Yihong Gong. Few-shot class-incremental learning via relation knowledge distillation. In *AAAI*, 2021.

[12] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Eur. Conf. Comput. Vis.*, 2020.

[13] Lars Kai Hansen and Peter Salamon. Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12(10): 993–1001, 1990.

[14] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2016.

[15] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[16] Gao Huang, Yixuan Li, Geoff Pleiss, Zhuang Liu, John E Hopcroft, and Kilian Q Weinberger. Snapshot ensembles: Train 1, get m for free. In *Int. Conf. Learn. Represent.* 2017.

[17] Jayateja Kalla and Soma Biswas. S3c: Self-supervised stochastic classifiers for few-shot class-incremental learning. In *Eur. Conf. Comput. Vis.*, 2022.

[18] Haeyong Kang, Jaehong Yoon, Sultan Rizky Hikmawan Madjid, Sung Ju Hwang, and Chang D Yoo. On the soft-subnetwork for few-shot class incremental learning. In *Int. Conf. Learn. Represent.*, 2023.

[19] Do-Yeon Kim, Dong-Jun Han, Jun Seo, and Jaekyun Moon. Warping the space: Weight space rotation for class-incremental few-shot learning. In *Int. Conf. Learn. Represent.*, 2023.

[20] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009.

[21] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Adv. Neural Inform. Process. Syst.* 2012.

[22] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Trans. Pattern Anal. Mach. Intell.*, 40(12):2935–2947, 2017.

[23] Huan Liu, Li Gu, Zhixiang Chi, Yang Wang, Yuanhao Yu, Jun Chen, and Jin Tang. Few-shot class-incremental learning via entropy-regularized data-free replay. In *Eur. Conf. Comput. Vis.*, 2022.

[24] Junghun Oh, Sungyong Baik, and Kyoung Mu Lee. Closer: Towards better representation learning for few-shot class-incremental learning. In *Eur. Conf. Comput. Vis.*, 2024.

[25] Can Peng, Kun Zhao, Tianren Wang, Meng Li, and Brian C Lovell. Few-shot class-incremental learning from an open-set perspective. In *Eur. Conf. Comput. Vis.*, 2022.

[26] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *Int. Conf. Machine Learning*, 2021.

[27] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2017.

[28] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. 2019.

[29] Zeyin Song, Yifan Zhao, Yujun Shi, Peixi Peng, Li Yuan, and Yonghong Tian. Learning with fantasy: Semantic-aware virtual contrastive constraint for few-shot class-incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

[30] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *IEEE Conf. Comput. Vis. Pattern Recog.* 2015.

[31] Yu-Ming Tang, Yi-Xing Peng, Jingke Meng, and Wei-Shi Zheng. Rethinking few-shot class-incremental learning: Learning from yourself. In *Eur. Conf. Comput. Vis.*, 2024.

[32] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[33] Oriol Vinyals, Charles Blundell, Tim Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *NeurIPS*, 2016.

[34] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011.

[35] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[36] Mitchell Wortsman, Gabriel Ilharco, Samir Ya Gadre, Rebecca Roelofs, Raphael Gontijo-Lopes, Ari S Morcos, Hongseok Namkoong, Ali Farhadi, Yair Carmon, Simon Kornblith, et al. Model soups: averaging weights of multiple fine-tuned models improves accuracy without increasing inference time. In *Int. Conf. Machine Learning*, 2022.

[37] Mitchell Wortsman, Gabriel Ilharco, Jong Wook Kim, Mike Li, Simon Kornblith, Rebecca Roelofs, Raphael Gontijo Lopes, Hannaneh Hajishirzi, Ali Farhadi, Hongseok Namkoong, and Ludwig Schmidt. Robust fine-tuning of zero-shot models. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[38] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2019.

[39] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[40] Yibo Yang, Haobo Yuan, Xiangtai Li, Zhouchen Lin, Philip Torr, and Dacheng Tao. Neural collapse inspired feature-classifier alignment for few-shot class incremental learning. In *Int. Conf. Learn. Represent.*, 2023.

[41] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Int. Conf. Comput. Vis.*, 2019.

[42] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2021.

[43] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. *arXiv preprint arXiv:1710.09412*, 2017.

[44] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2020.

[45] Linglan Zhao, Jing Lu, Yunlu Xu, Zhanzhan Cheng, Dashan Guo, Yi Niu, and Xiangzhong Fang. Few-shot class-incremental learning via class-aware bilateral distillation. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.

[46] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, Liang Ma, Shiliang Pu, and De-Chuan Zhan. Forward compatible few-shot class-incremental learning. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2022.

[47] Huiping Zhuang, Zhenyu Weng, Run He, Zhiping Lin, and Ziqian Zeng. Gkeal: Gaussian kernel embedded analytic learning for few-shot class incremental task. In *IEEE Conf. Comput. Vis. Pattern Recog.*, 2023.