MICS: Midpoint Interpolation to Learn Compact and Separated Representations for Few-Shot Class-Incremental Learning

Solang Kim[1], Yuho Jeong[2]*, Joon Sung Park[1], Sung Whan Yoon[1]†
[1]Graduate School of Artificial Intelligence, Ulsan National Institute of Science and Technology (UNIST), Republic of Korea
[2]Nuvilab Inc., Republic of Korea
solangii@unist.ac.kr yuho.jeong@nuvilab.com furuju@unist.ac.kr shyoon8@unist.ac.kr

Abstract

Few-shot class-incremental learning (FSCIL) aims to learn a classification model for continually accepting novel classes with a few samples. The key of FSCIL is the joint success of the following two training stages: Base training stage to classify base classes and Incremental training stage with sequential learning of novel classes. However, recent efforts show a tendency to focus on one of the stages, or separately design strategies for each stage, so that less effort has been paid to devise a consistent strategy across the consecutive stages. In this paper, we first emphasize the particular aspects of the successful FSCIL algorithm that are worthwhile to consistently pursue during both stages, i.e., intra-class compactness and inter-class separability of the representation, which allows a model to reserve feature space in between current classes for preparing the acceptance of novel classes in the future. To achieve these aspects, we propose a mixup-based FSCIL method called MICS, which theoretically guarantees to enlarge the thickness of the margin space between different classes, leading to outstanding performance on the existing benchmarks. Code is available at https://github.com/solangii/MICS.

1. Introduction

Deep visual models offer exceptional performance under the assumption of stationary training data with even largely distributed image categories. However, this condition rarely holds in the real-world setting, where the class distribution deviates over time. A naive approach that stores past data samples and retrains the models on demand faces significant memory burden and computational overhead. Moreover, fitting the model to the current data without memorizing past data often results in catastrophic forgetting of prior knowledge and overfitting to the current samples.

In the computer vision context, class-incremental learning (CIL) addresses the challenge of learning novel image categories without relying on past samples [3,10,13,20,31]. However, CIL methods often struggle to acquire the acceptable classification ability for novel classes while preserving the knowledge of past classes. These challenges are exacerbated in the few-shot class-incremental learning (FSCIL) setting, where training data for novel classes is severely limited, intensifying the problems of forgetting and overfitting.

Many of the prior FSCIL methods ranging from the early work including iCaRL [20] and TOPIC [22] to relatively recent works such as FSLL [18], WaRP [12] and Soft-SubNet [11] primarily attempt to develop the strategies working in the *incremental stage* which focus on accepting few-shot novel categories without forgetting base classes. Their philosophy is to adopt particular strategies for preventing forgetting via distillation [20], graph-based memorization [22], theoretical intuitions on parameter space [11, 12, 18].

Nevertheless, a group of recent studies [19, 21, 35] has observed the effort done in the base stage is probably more impactful than the effort in the incremental stage. Surprisingly, a straightforward 'Baseline' method described in [21] which simply freezes the feature extractor after being trained in the base stage, sometimes outperforms most of the aforementioned prior works where their emphasis is on the incremental stage. The baseline employs a simple method called Nearest Class Mean (NCM) in the incremental stage where the classifiers are computed as the per-class mean feature vectors. It implies that a well-trained representation during the base stage might be sufficient to handle the incremental learning stages.

After the finding, the recent research emphasis has shifted towards effective *base training*, where its goal is to train a well-prepared base model that can successfully accept future novel classes via a simple NCM method. To this end, algorithms called FACT [35] and ALICE [19], utilize class mixup augmentation in the base training where the constructed mixups are treated as fake, or auxiliary novel classes. When the class mixups are considered as auxiliary

*He contributed to this work when he was a graduate student at UNIST.
†Sung Whan Yoon is the corresponding author.

classes, the model then reserves the marginal feature space between the base classes, which will be served for the actual novel classes. Along the same lines, a recent work named CLOM [11] employs the margin-based loss term during the base training. Although the branch of works pursuing the margin space between base classes currently shows state-of-the-art FSCIL performance, they are still restricted to relying on the frozen representation, which cannot learn further features from novel classes, and the expansion of their concepts to the incremental stages is not trivially anticipated. Moreover, none of these works provide theoretical evidence demonstrating how their method thickens the margin space.

Our interest is in the branch of FSCIL methods reserving margin between classes in the representation space to provide theory-and-practice fourfold claims. Specifically, we start with two intriguing empirical examinations: **i)** A particular aspect of FSCIL methods, i.e., *intra-class compactness* and *inter-class separability* in the representation space, is strongly related to FSCIL performance. **ii)** Prior state-of-the-art methods to reserve the margin space suffer from catastrophic performance degradation when their methodologies are extended to the incremental stages. Along with the findings, a theoretical link between mixup and FSCIL is provided: **iii)** We first formulate the mathematical condition of the mixup-based FSCIL method to enlarge the thickness of margin space between classes with the lens of *boundary thickness* [28]. **iv)** Based on the findings, we propose a mixup-based FSCIL method called Midpoint Interpolation for Compact and Separated Representation (MICS) with the following strengths: expandability to the incremental stage, the theoretical rationale for the enlarged margin space, and the remarkable FSCIL performance.

2. Related Work

2.1. Mixup methods

A baseline method called Mixup [33] selects a pair of image samples and linearly interpolates both images at the input space. On the other hand, Manifold mixup [23] reveals that the interpolation is much more effective when it takes place at a deeper layer, so it performs interpolation at the hidden layers of deep models. CutMix [30] involves cutting a portion of an image sample and pasting it to substitute a portion of a different image. Other methods exemplified by AdaMixUp [8] and ACAI [1] determine the mixing ratio by learning adaptive mixing policies. When imposing the label on the generated mixup sample, the prior methods commonly interpolate two original class labels to form the label of mixup sample. In other words, they do not treat the mixup as a virtual or auxiliary class without imposing a virtual class label on it. However, mixup-based FSCIL methods including FACT [35], ALICE [19] and our MICS explicitly impose a virtual class label for mixup sample.

2.2. Class-Incremental Learning (CIL)

CIL involves training a classification model through a series of sessions, each containing a distinct set of image classes. An early group of CIL algorithms, including [2, 3, 17, 20], aims to retain knowledge from past sessions via distillation loss that is computed by a portion of memorized previous samples, i.e., the exemplar-based method. Another strategy, such as [10, 16, 26], focuses on reducing the task imbalance between past and current tasks to mitigate strong bias towards recent sessions. Another branch of methods, including [4, 13, 31], freezes the crucial model parameters during incremental stages to preserve past knowledge.

2.3. Few-Shot Class-Incremental Learning (FSCIL)

FSCIL handles CIL when novel classes are given with a few samples [5, 15, 34]. We here categorize prior works into three parts: **i)** Methods in incremental stage, **ii)** Methods in base stage, **iii)** Methods for large-margin representation.

Methods in incremental stage: This group of methods mainly focuses on relieving the forgetting issue during the incremental stages via distillation by iCaRL [20], topology-graph-based memorization of feature distributions by TOPIC [22], attentive updates and expansions of model parameters by FSLL [18], Soft-SubNet [11], WaRP [12] and DSN [27], and efforts on classifiers rather than the representation by CEC [32], SPPR [36], and NC-FSCIL [29]. For the methods with in-depth intuitions on model parameter spaces, their key essence lies in identifying the important parameters and partially freezing them during incremental stages. FSLL [18] selects the large magnitude parameters. Soft-SubNet [11] learns adaptive soft masks to selectively update the subnetwork. WaRP [12] transforms model parameters into a smaller sub-space to keep them unchanged. DSN [27] which is slightly apart from the aforementioned methods, dynamically expands model nodes to accept novel classes. For the approaches focusing on classifier updates, CEC [32] and SPPR [36] concentrate on adjusting both base and novel classifiers during incremental stages. NC-FSCIL [29] assigns predefined classifiers and learns an extractor to align features with these classifiers. In the surge of efforts in the incremental stage, researchers generally agree that the active update-based approaches are required to tackle the realistic lifelong learning scenario with a large number of incremental stages, but they are sometimes inferior to the recent efforts in the base stage that makes the research emphasis partially shifted to the better base stage learning.

Methods in base stage: A straightforward 'Baseline' method [21] with a frozen feature extractor, after being trained in the base stage, sometimes outperforms most of the efforts where their emphasis is on the incremental stage. Stepping on the observation, some recent works primarily focus on preparing the base stage training to facilitate the better acceptance of novel classes in the future via flat-

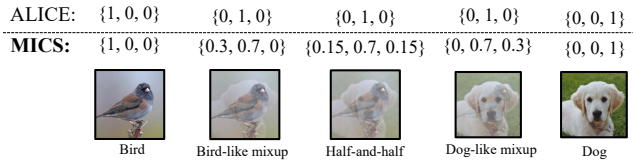


Figure 1. Soft labels of mixup samples for MICS and ALICE.

minima searching F2M [21], margin-based loss by CLOM [37], mixup augmentations by FACT [35] and ALICE [19]. When focusing on the details, F2M [21] finds a flatter minimum during the base stage which lets the model robust to distribution shifts and allows fine-tuning inside the flatter region during incremental sessions. CLOM [37] adopts margin loss that adaptively tunes the margin value according to the paired classes. FACT [35] and ALICE [19] adopt mixup methods and treat class mixups as virtual or auxiliary classes so that the margin representation space between base classes is largely reserved for future novel classes. In the incremental stages, CLOM, FACT and ALICE freeze the representation and rely on the Nearest Mean Classifier (NCM) method where the novel classifiers are naively computed with per-class averaged features, i.e., prototypes.

2.4. Novelty of MICS over Relevant Works

Comparison to margin-based method: DSN [27] primarily focuses on model expansion, but it points out the importance of margin space of representation. In incremental stages, DSN recalls the past class distributions to acquire sufficient margin space, which is backward-compatible. In contrast, MICS does not adjust the model and runs both in the base and incremental stages. Also, MICS does not recall the past distribution but utilizes mixup methods to enlarge the margin between current classes for future classes, which can be viewed as forward-compatible. When emphasizing the novelty beyond CLOM, MICS utilizes Mixup to enlarge margin spaces rather than using margin-based loss.

Comparison to mixup-based FSCIL: MICS is significantly different from FACT [35], and ALICE [19] in that MICS updates models via mixup samples both in base training and incremental learning. When focusing on a technical viewpoint, the ways of labeling mixups and employing virtual classifiers greatly differ from the prior works. MICS imposes a soft label by considering how much the mixup images look novel or similar to the paired base classes. As illustrated in Fig. 1, when mixing a pair of 'bird' and 'dog' images, MICS can control the mixup images probabilities of {bird, virtual, dog} by considering how much the images look like 'bird', novel, or 'dog'. However, ALICE assigns a hard label to the mixup with 100% confidence. Also, as shown in Fig. 2, MICS introduces virtual classifiers at the midpoint of base classifiers. However, FACT and ALICE introduce an extra learnable classifier for the virtual class, which is not in the middle ground of the original classes. That enables MICS to directly embed mixup samples at the midpoint classifier to enlarge the in-between space.

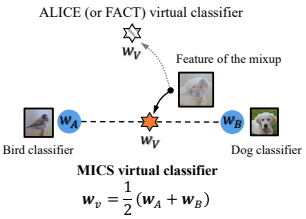


Figure 2. Midpoint classifier of MICS for virtual class

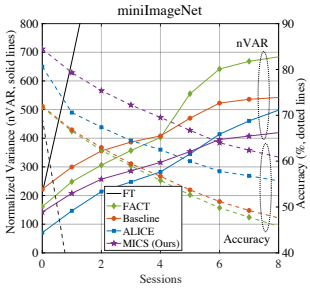


Figure 3. Compactness and separability vs. Accuracy

3. Preliminaries

Here, we present intriguing observations regarding the importance of *intra-class compactness* and *inter-class separability* in representation and the catastrophic performance degradation when the most relevant methods, i.e., CLOM and ACLIE, are extended to incremental stages.

3.1. Compact and Separable Representations

To support the importance of the intra-class compactness and inter-class separability of representation, we quantify the aspects with the normalized Variance (nVAR), which is the mean of per-class variance of features divided by the squared distance to the nearest interfering class centroid:

$$\text{nVAR}(t) = \mathop{\mathbb{E}}_{k \in \mathcal{C}_t} \left[\mathop{\mathbb{E}}_{(\mathbf{x},y) \in \mathrm{D}_{t,k}} \left[\frac{||\mathbf{c}_k - f_\theta(\mathbf{x})||^2}{||\mathbf{c}_k - \mathbf{n}_k||^2} \right] \right], \quad (1)$$

where t is the session number, $\mathcal{C}_t$ is the set of class indices up to the session t, $\mathcal{D}_{t,k}$ is the union set of samples from class k up to session t, k is the class index, $\mathbf{c}_k$ is the class centroid, i.e., prototype of class k, $f_\theta(\mathbf{x})$ is the feature vector of input $\mathbf{x}$, and $\mathbf{n}_k$ is the nearest interfering prototype of class k. When nVAR is small, the representation is intra-class compact and inter-class separated. As shown in Fig. 3, we computed the nVAR values (denoted as solid lines) of the prior methods, including Fine-tuning (FT), FACT, Baseline, and ALICE for the miniImageNet FSCIL benchmark. When we briefly show the resulting performance of MICS, it shows the outstanding FSCIL performance and the nVAR value at the final session. More details are in the Experiments section. **Our first observation is** that FSCIL methods with better FSCIL accuracies show smaller nVAR values, indicating larger margins in representation space.

3.2. Extension to Incremental Stages

We examine what happens when ALICE and CLOM are extended to the incremental session, where they are de-

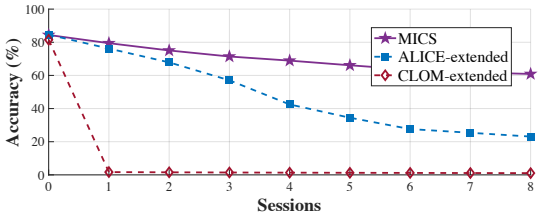


Figure 4. MICS vs. ALICE/CLOM-extended for miniImageNet

signed to pursue large margin representation during the base stage, i.e., their methods in the base stage are simply applied to incremental stages. To alleviate dramatic model changes, we only change 30% of model parameters with larger magnitude by borrowing the technique in FSLL [18]. As shown in Fig. 4, we observe that the performance catastrophically collapses (referring 'ALICE/CLOM-extended'). **Our second observation** is that the extension to incremental sessions is not trivial for the cutting-edge margin-based methods, but MICS successfully applies to incremental stages.

3.3. Notations and Problem Setting for FSCIL

In the FSCIL setting, a model is trained by handling a sequence of *training sessions* $\{\mathcal{D}^{(0)}, \mathcal{D}^{(1)}, \ldots, \mathcal{D}^{(t)}\}$, where $\mathcal{D}^{(t)} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{|\mathcal{D}^{(t)}|}$ is the dataset of session t with input image $\mathbf{x}_i$ and its class label y_i. Label set $\mathcal{C}^{(t)}$ contains class labels of session t. Also, different sessions contain disjoint label sets without overlapping classes. The initial session, when $t = 0$, contains a relatively large amount of training samples of $\mathcal{C}^{(0)} = N_b$ base classes, and it is called the *base training stage*. The following sessions, when $t > 0$, are called *incremental sessions* and contain $\mathcal{C}^{(t)} = N$ classes for each. In each incremental session, K training samples for each class are given, so-called N-way, K-shot setting for each session. In each session t, we can only use the data samples in $\mathcal{D}^{(t)}$ of the current session to train the FSCIL model. It is not allowed to access the past data samples of the past sessions. For the model, $f_\theta(\cdot)$ is the feature extractor parameterized by θ. Also, $\mathbf{w}_k$ is the classifier for class k. The model should prepare additional $|\mathcal{C}^{(t)}|$ classifiers for novel classes, i.e., $\mathbf{W}^{(t)} = \{\mathbf{w}_k\}_{k=|\mathcal{C}_{t-1}|+1}^{|\mathcal{C}_{t-1}|+|\mathcal{C}^{(t)}|}$. After session t, the FSCIL model with updated feature extractor $f_\theta(\cdot)$ and classifiers $\mathbf{W}_t = \bigcup_{\tau=1}^{t} \mathbf{W}^{(\tau)}$, is evaluated on the test samples from all encountered classes in $\mathcal{C}_t = \bigcup_{\tau=0}^{t} \mathcal{C}^{(\tau)}$, which is the union set of the class indices.

4. Proposed Method: MICS

We present our method, called Midpoint Interpolation for Compact and Separated Representation (MICS), in detail. The notations in the Preliminaries section are used. We generally describe the training procedures for each session. For session t, feature extractor $f_\theta(\cdot)$ and classifier $\mathbf{W}_{t-1}$ are given from the past session. When $t = 0$, the feature extractor is randomly initialized, and the classifier set is an empty set, i.e., $\mathbf{W}_{-1} = \varnothing$. Dataset $\mathcal{D}^{(t)}$ is then given for training $|\mathcal{C}^{(t)}|$ novel classes. Before starting the session, we should initialize the classifiers $\mathbf{W}^{(t)}$ for $|\mathcal{C}^{(t)}|$ novel classes. When $t = 0$, which is base training, the classifiers are initialized with additional learnable weights. When $t > 0$, which is an incremental session, the classifiers are set to be the per-class averaged features of the given few-shot samples. We are then ready to describe the mixup process and the labeling process of the mixup sample. The training procedures of MICS for each session consist of the following three steps: **i)** Construction of mixup samples, **ii)** Establishment of midpoint classifiers, and **iii)** Training of the model.

4.1. Construction of Mixup Samples

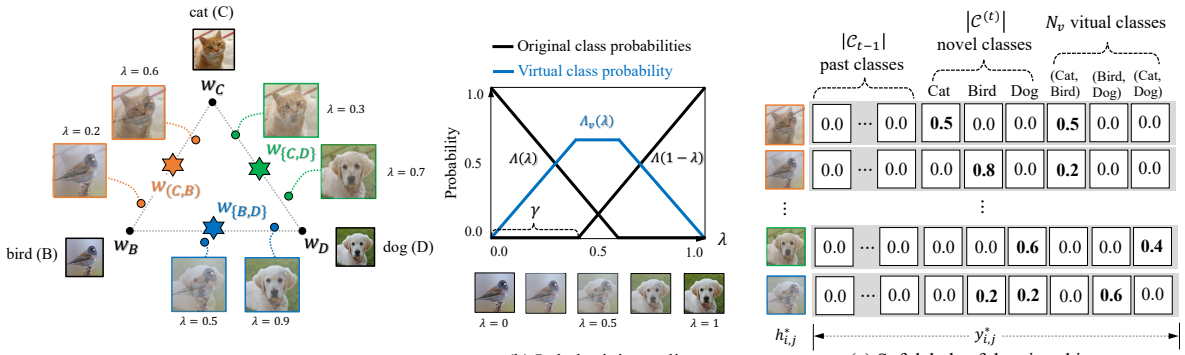
To construct the mixups, we randomly pick $|\mathcal{B}|$ different pairs of two different samples from $\mathcal{D}^{(t)}$, where $\mathcal{B}$ is the set of the picked pairs. Let us denote the number of different combinations of the paired classes in $\mathcal{B}$ as N_v. MICS then treats the N_v combinations as the N_v virtual classes.

Mixup process: Without losing the generality, let us focus on a pair of different samples, i.e., $\{(\mathbf{x}_i, y_i), (\mathbf{x}_j, y_j)\} \in \mathcal{B}$. MICS utilizes the Manifold mixup method, where the feature combination is done at the hidden layer of the model. For a simple formulation, let us denote the feature extractor with a two-step process: $f_\theta(\mathbf{x}) = g(h(\mathbf{x}))$, where $h(\cdot)$ is the hidden-layer feature and $g(\cdot)$ is the forward path through the remaining layers. When we combine the features at the hidden layer, the signal of the mixup sample is:

$$h^*_{i,j} = \lambda h(\mathbf{x}_i) + (1-\lambda) h(\mathbf{x}_j), \tag{2}$$

where $\lambda \in (0, 1)$ is drawn from Beta(α, α) distribution, where $\alpha > 0$ Here, we describe the process based on Manifold mixup [23], but MICS is not limited to a particular method. By processing the mixup signal through the remaining layer, the final feature $g(h^*_{i,j})$ of the mixup sample is obtained. In the Experiments section, we provide the ablation to utilize other mixup methods. Fig. 5(a) illustrates the mixup process when there are three novel classes, i.e., 'cat', 'bird', and 'dog' generating $|\mathcal{B}| = 6$ mixup samples.

Soft-labeling process: MICS then computes the soft label $y^*_{i,j}$ of the mixup sample by considering how much the mixup is virtual or similar to the original classes y_i and y_j. When the mixup is around half-and-half of the given images, i.e., $\lambda \simeq 0.5$, the virtual class probability is set to be high, and the original class probabilities become low. Otherwise, when λ is near 0 or 1, the virtual probability is low, and the original class probabilities are set to be high. To follow the behavior, Fig. 5(b) shows how MICS determines the virtual class probability and original class probabilities for varying λ. Here, we adopt additional parameter γ that determines the steepness of decreasing behavior of original class probabilities as λ goes to 0.5. MICS assigns $\Lambda(\lambda)$ and



(a) Construction of mixup samples (b) Label mixing policy (c) Soft labels of the virtual images

Figure 5. The mixup process of MICS. (a) We have $|C^{(t)}| = 3$ novel classes, i.e., 'cat', 'bird', and 'dog'. $|\mathcal{B}| = 6$ mixup samples are constructed using the novel classes. (b) When the mixup is bird-like or dog-like mixup, then the probabilities for the 'bird' and 'dog' are high (black-colored functions, i.e., $\Lambda(\lambda)$ and $\Lambda(1-\lambda)$). For a half-and-half mixup, the probability for the virtual class is high (a blue-colored function, i.e., $\lambda_v = 1 - \Lambda(\lambda) - \Lambda(1-\lambda)$). (c) Constructed mixup samples and their soft labels.

$\Lambda(1-\lambda)$ probabilities to the paired original classes, and $\Lambda_v(\lambda) = 1 - \Lambda(\lambda) - \Lambda(1-\lambda)$ to the virtual class by following the equations:

$$\begin{aligned}\Lambda(\lambda) &= \max\big((1-\lambda-\gamma)/(1-\gamma), 0\big)\\ \Lambda(1-\lambda) &= \max\big((\lambda-\gamma)/(1-\gamma), 0\big)\\ \Lambda_v(\lambda) &= 1 - \Lambda(\lambda) - \Lambda(1-\lambda)\end{aligned} \tag{3}$$

In Fig. 5(c), the resulting soft labels of the example mixup samples are shown. For each mixup sample, zero values are assigned to the past class indices. For the pair of classes that participated in the mixup process, we assign $\Lambda(\lambda)$ and $\Lambda(1-\lambda)$ probabilities. Finally, MICS assigns $\Lambda_v(\lambda)$ probability to the corresponding virtual class index. Let us denote the computed soft label for the mixup sample as $y^*_{i,j}$.

4.2. Establishment of Midpoint Classifiers

MICS sets the virtual classifiers, i.e., N_v, as the midpoint between the paired classes. For a mixup that combines a pair of samples $\{(\mathbf{x}_i, y_i), (\mathbf{x}_j, y_j)\}$, the virtual classifier $\mathbf{w}_{(y_i,y_j)}$ is set to be the middle point of $\mathbf{w}_{y_i}$ and $\mathbf{w}_{y_j}$:

$$\mathbf{w}_{(y_i,y_j)} = (\mathbf{w}_{y_i} + \mathbf{w}_{y_j})/2. \tag{4}$$

As shown in Fig. 5(a), three virtual classifiers for the mixups of 'cat', 'bird', and 'dog' classes are denoted as $\mathbf{w}_{(C,B)}$, $\mathbf{w}_{(C,D)}$, and $\mathbf{w}_{(B,D)}$, which are at the midpoints.

4.3. Training of the Model

We prepare $|\mathcal{B}|$ mixup samples and the corresponding soft labels. The parameter θ of the feature extractor $f_\theta(\cdot)$ and the classifiers $\mathbf{W}^{(t)}$ are updated to minimize the cross-entropy loss $\mathcal{L}_{\text{CE}}$ for the mini-batch of mixup samples:

$$\mathcal{L} = \frac{1}{|\mathcal{B}|} \sum_{\{(\mathbf{x}_i,y_i),(\mathbf{x}_j,y_j)\}\in\mathcal{B}} \mathcal{L}_{\text{CE}}\big(\hat{y}^*_{i,j}, y^*_{i,j}\big), \tag{5}$$

$$\text{where } \hat{y}^*_{i,j} = \frac{\exp\big(\delta\big(g(h^*_{i,j}), \mathbf{w}_{(y_i,y_j)}\big)/\tau\big)}{\sum_l \exp\big(\delta\big(g(h^*_{i,j}), \mathbf{w}_l\big)/\tau\big)}.$$

Also, $\delta(\cdot,\cdot)$ is a similarity metric and τ is the temperature. When $t > 0$, i.e., an incremental session, MICS updates a subset of the feature extractor's parameters with a lower absolute value to relieve forgetting. Let us denote θ_ϵ as ϵ-subset of the parameters with lower absolute value, where $0 \le \epsilon \le 1$. When $\epsilon = 0$, MICS freezes the feature extractor, i.e., the mixup process is only applied in base training. We found the best value of $\epsilon = 0.01, 0.3, 0.3$ for CIFAR-100,

Algorithm 1 Training process of MICS for each session

Input: Session t; Dataset $\mathcal{D}^{(t)}$; Feature extractor $f_\theta(\cdot)$; Past classifiers $\mathbf{W}_{t-1}$, Mixup hyperparameters α and γ; Ratio ϵ.

Initialize $|\mathcal{C}^{(t)}|$ novel classifiers $\mathbf{W}^{(t)}$ with:
  the additional learnable weights ($t = 0$) and
  the per-class prototypes ($t > 0$).
for $\tau = 1, \cdots, I$ **do**
  Randomly pick a set of paired samples, i.e., $\mathcal{B}$
  Compute Manifold mixup samples (Eq. 2)
  Compute labels $\Lambda(\lambda)$, $\Lambda(1-\lambda)$, $\Lambda_v(\lambda)$ (Eq. 3)
  Compute midpoint classifiers (Eq. 4)
  Calculate the loss (Eq. 5)
  Update $\mathbf{W}^{(t)}$ and the θ_ϵ with small absolute values:
    $\epsilon = 1$ ($t = 0$) and
    $0 \le \epsilon < 1$ ($t > 0$).
end for
Refine $\mathbf{W}^{(t)}$ to the per-class prototypes

Method	Accuracy in each session (%)									PD ↓
	0	1	2	3	4	5	6	7	8	
Finetune	69.37	47.41	11.00	8.48	4.79	6.00	4.59	5.41	6.72	62.65
Baseline	69.37	64.34	60.33	57.23	54.18	51.35	48.87	47.08	45.56	23.81
Rebalance [10]	61.31	47.80	39.31	31.91	25.68	21.35	18.67	17.24	14.17	47.14
iCaRL [20]	61.31	46.32	42.94	37.63	30.49	24.00	20.89	18.80	17.21	44.10
FSLL [18]	66.48	61.75	58.16	54.16	51.10	48.53	46.54	44.20	42.28	24.20
CEC [32]	72.00	66.83	62.97	59.43	56.70	53.73	51.19	49.24	47.63	24.37
FACT* [35]	71.78	66.54	62.39	58.96	55.80	52.65	49.82	47.78	45.80	25.98
CLOM [37]	73.08	68.09	64.16	60.41	57.41	54.29	51.54	49.37	48.00	25.08
NC-FSCIL [29]	84.02	76.80	72.00	67.83	66.35	64.04	61.46	59.54	58.31	25.71
WaRP [12]	72.99	68.10	64.31	61.30	58.64	56.08	53.40	51.72	50.65	**22.34**
ALICE [19]	80.6	70.6	67.4	64.5	62.5	60.0	57.8	56.8	55.7	24.9
MICS (Ours)	**84.40**	**79.48**	**75.09**	**71.40**	**68.89**	**66.16**	**63.57**	**61.79**	**60.74**	23.66

Table 1. The evaluation for the FSCIL benchmark with the miniImageNet dataset. MICS uses $\epsilon = 0.3$. * indicates FACT of [35] without AutoAugment of [6] for a fair comparison. The results of MICS with AutoAugment are given in the Supplementary material.

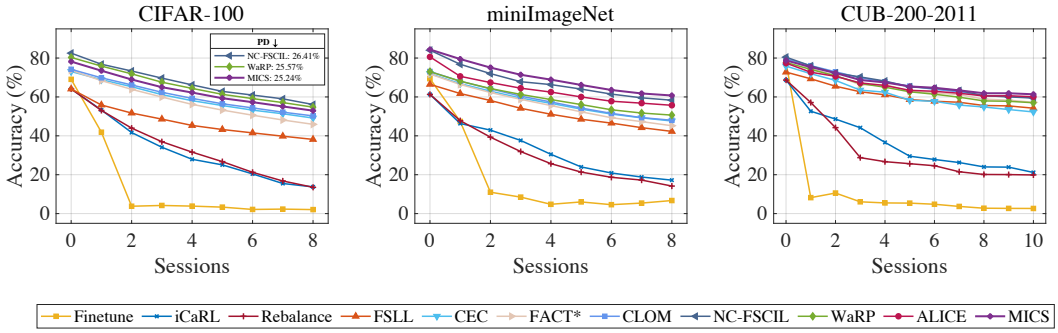


Figure 6. Results on three benchmark datasets: CIFAR-100, miniImageNet, and CUB-200-2011. In the figure of CIFAR-100, we exclude ALICE of [19] because they use a large backbone, ResNet-18 (with 11M parameters), which has 40 times more parameters than ResNet-20 (with 0.27M parameters) of other methods. MICS uses $\epsilon = 0.01, 0.3, 0.3$ for CIFAR-100, miniImageNet, and CUB-200-2011 respectively.

miniImageNet, and CUB-200-2011 respectively.

After the training, MICS re-computes the novel classifiers to be the per-class averaged feature vectors and drops the virtual classifiers. Consequently, MICS obtains the updated feature extractor $f_\theta(\cdot)$ and the classifiers $\mathbf{W}_t = \{\mathbf{w}_k\}_{k=1}^{|\mathcal{C}_t|}$. Alg. 1 presents the pseudocode of the details learning process of MICS for a learning session.

4.4. Boundary Thickness

We borrow the concept of Boundary Thickness [28] to explain how MICS enlarges the margin space. In brief, boundary thickness is the thickness of the uncertain margin space between two differently-labeled samples. When the thickness of the uncertain region is large, we can say that the model shows a large margin space between classes. To focus on the representation space, we have slightly changed and simplified the original concept of the boundary thickness to measure the thickness in the representation space. In addition, we normalize the thickness with the distance between paired samples to remove the effect of the power.

For a C-way classification task with input, output and representation space, i.e., $\mathcal{X}$, $\mathcal{Y}$ and $\mathcal{H}$, respectively, let us denote the embedding function as $g(x) : \mathcal{X} \to \mathcal{H}$ and the prediction function as $f(h) : \mathcal{H} \to [0,1]^C$.

Definition 1. *(Normalized Boundary Thickness in Representation Space) For $\alpha \in (0,1)$, the boundary thickness $\Theta(f,\alpha)$ in representation space $\mathcal{H}$ is defined as follows:*

$$\Theta(f,\alpha) = \mathbb{E}_{(x_i,x_j)}\left[\int_0^1 \mathbf{I}\{|\Delta_{ij} f(h^*_{ij})| < \alpha\} d\lambda\right], \quad (6)$$

where $\mathbf{I}\{\cdot\}$ is an indicator function and $\Delta_{ij} f(h) = f(h)_i - f(h)_j$ is the gap between the probabilities for classifying embedded feature h to class i and j. Also, $h^*_{ij} = \lambda h_i + (1-\lambda) h_j$, where $h_i = g(x_i)$ and $h_j = g(x_j)$.

Let us then denote the normalized boundary thickness of the model learned by Manifold mixup as $\Theta(f_{\text{Mixup}}, \alpha)$.

Also, we denote the normalized boundary thickness of MICS as $\Theta(f_{\text{MICS}}, \alpha, \Lambda)$. We then provide the condition for satisfying $\Theta(f_{\text{Mixup}}, \alpha) \leq \Theta(f_{\text{MICS}}, \alpha, \Lambda)$.

Theorem 1. *For all $\alpha \in (0, 1)$, MICS achieves larger normalized boundary thickness than Manifold mixup, i.e., $\Theta(f_{Mixup}, \alpha) \leq \Theta(f_{MICS}, \alpha, \Lambda)$, when the following holds:*

$$\lambda - \Lambda(1 - \lambda) \geq 1 - \lambda - \Lambda(\lambda). \tag{7}$$

When designing $\Lambda(\cdot)$ as a linear function by introducing hyperparameter γ as described in 'Proposed Method' section, we can obtain the following condition for larger thickness:

Corollary 1. *For all $\alpha \in (0, 1)$, MICS with linear function $\Lambda(\cdot)$ shows larger normalized boundary thickness than Manifold mixup when the following holds: $\gamma \geq 0.25$.*

Proofs are in Supplementary.

5. Experiments

5.1. Datasets and Benchmarks

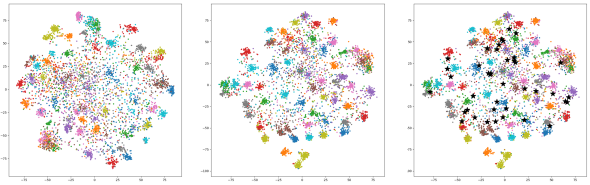
Our evaluation is done on three benchmark datasets, including CIFAR-100, miniImageNet, and CUB-200-2011. CIFAR-100 [14] consists of 60,000 RGB images of size 32×32 from 100 classes. miniImageNet [24] consists of 60,000 RGB images of size 84×84 from 100 classes. The CIFAR-100 and miniImageNet benchmarks show the same configurations: 500 training and 100 testing images are given for each class. There exist 60 base classes and 40 novel classes. The base class samples are used only in base training, and the novel class samples are used in session. There are eight incremental sessions, and a 5-way 5-shot setting is applied for each incremental session. CUB-200-2011 [25] consists of 11,788 RGB images of size 224×224 from 200 classes. It contains 5,994 samples for training and 5,794 samples for testing. There are 100 base classes and the remaining 100 novel classes. Ten incremental stages are given with a 10-way 5-shot setting for each.

5.2. Implementation Details

Following the standard setting in [22], we use ResNet-20 [9] for CIFAR-100, ResNet-18 for miniImageNet, and ImageNet [7] pre-trained ResNet-18 for CUB-200-2011. We use the cosine similarity for $\delta(\cdot, \cdot)$, and the temperature τ is optimized for each benchmark. We optimize the model using momentum Stochastic Gradient Descent (SGD) for every session. Random crop, random horizontal flip, and random scaling are applied for data augmentation during training. Further details are in Supplementary.

5.3. Performance Comparisons

In Table 1 and Fig. 6, we report the performance for the FSCIL benchmarks. Also, we also evaluate the performance dropping rate (PD), which measures the gap between the accuracies of base training and the last session.



(a) **Baseline**, 60 base classes (b) **MICS**, 60 base classes (c) **MICS**, after the final sessions

Figure 7. t-SNE plots for the representations of (a) Baseline, (b) MICS, and (c) MICS at the final incremental session.

	miniImageNet			
Baseline	FACT	ALICE	Manifold Mixup	**MICS**
Boundary Thickness	0.908	0.792	0.897	**0.934**

Table 2. Normalized Boundary Thickness

The exact numbers for the CIFAR-100 and CUB-200-2011 cases are in the Supplementary material. MICS records the notable performance in all sessions and the PD values for all benchmarks. Specifically, MICS outperforms the previous state-of-the-art method by +2.43% on miniImageNet, and +1.27% on CUB-200-2011 for the last session accuracies. For the miniImagNet case, when compared with the existing mixup-based methods, MICS shows significant gains, i.e., +5.04% beyond ALICE [19] and +14.94% beyond FACT [35].

5.4. Analysis for Compactness and Separability

We emphasize that MICS pursues the intra-class compact and inter-class separated representation. In Fig. 7, we visualize the representation of MICS by evaluating the t-distributed Stochastic Neighbor Embedding (t-SNE) plot of the feature vectors. The analysis is done for the CIFAR-100 benchmark. When comparing the representation of Baseline and MICS in Fig. 7(a) and 7(b), respectively, MICS clearly shows more intra-class compact and inter-class separated representation at base training. It confirms that MICS indeed reserves in-between spaces of the past classes. After the final sessions, novel classifiers depicted with black asterisks are located in the reserved spaces.

From another point of view, we evaluate the normalized variance (nVAR) value of MICS. A more compact and separable representation shows a smaller nVAR value. The exact formulation for nVAR is in Eq. 1. In Fig. 3, we plot the nVAR values with dotted lines and the FSCIL accuracies with solid lines for the miniImageNet benchmarks. MICS, with purple lines, achieves the highest FSCIL accuracy and the smallest nVAR value at the final session. ALICE [19] shows rapidly increasing nVAR as session goes on. However, MICS shows moderately increasing nVAR values, which implies that our method is better for keeping the representation compact and separated. We have also attached the results for other benchmarks to Supplementary.

			miniImageNet	
Mixup	**Soft Labeling**	**Midpoint**	Final accuracy	nVAR
✓	✗	✗	2.07%	1.691×10^4
✓	✓	✗	1.34%	1.746×10^5
✓	✓	✓	**60.74%**	**421.4**

Table 3. Ablations for the components of MICS ($\epsilon = 0.3$)

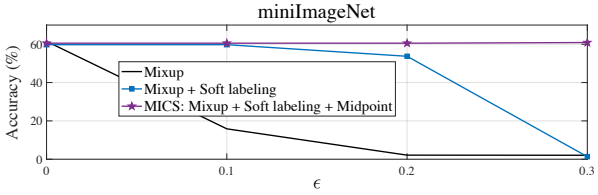


Figure 8. The last session accuracies with ϵ-ratio

	Base Session		**Incremental Session**	
Benchmark	α	γ	α	γ
miniImageNet	1.0	0.4	0.7	0.3
CIFAR-100	0.5	0.5	0.5	0.5
CUB-200-2011	0.2	0.3	0.2	0.3

Table 4. Best mixup hyperparameters for different benchmarks

Also, we empirically confirm that MICS improves boundary thickness over Manifold mixup and even other FSCIL methods such as ALICE and FACT in Table 2.

5.5. Ablation Study

Effectiveness of Components of MICS: Herein, we verify how much the components of MICS, i.e., Manifold mixup, Soft labeling policy, Midpoint classifiers, are effective. Table 3 shows the performance of models when we add the components one by one. 'Mixup' is a model with the Manifold mixup method in base training and incremental sessions. It does not employ virtual classifiers. 'Mixup + Soft Labeling' indicates a model that adopts virtual classifiers that are not midpoint classifiers. The final version, 'Mixup + Soft Labeling + Midpoint' is MICS. When we see the accuracies of the last session, 'Mixup' and 'Mixup + Soft Labling' show severe catastrophic forgetting due to fine-tuning. However, MICS shows the highest accuracy and the smallest nVAR in the last session. Fig. 8 shows the forgetting behavior by changing the update ratio ϵ. When the feature extractor is frozen, i.e., $\epsilon = 0$, our three models show similar accuracies, but MICS only shows robust performance when adopting fine-tuning, i.e., $\epsilon > 0$. We conjecture that MICS shows the most compact and separated representation so that it can allow the training of the representation, which slightly changes the feature distribution.

Mixup Hyperparameters: α controls the sampling of λ from the Beta distribution, and γ determines the steepness of labeling functions. In Table 4, we show the best choices of the hyperparameters for the three FSCIL benchmarks.

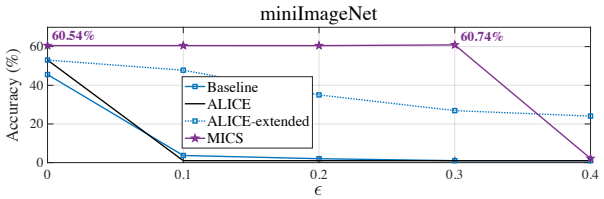


Figure 9. The last session accuracies with ϵ-ratio

When the dataset contains more similar classes, such as CUB-200-2011, then a small α is preferred. When α is small, a mixup is close to the original paired classes, which focuses on the training of original images. Also, γ is larger than 0.25, which coincides with our mathematical claims.

ϵ-Ratio for Model Updates: In each incremental stage, ϵ-ratio parameters with smaller absolute values are selected to be fine-tuned. It is well-known that the existing FSCIL methods that freeze the feature extractor suffer from severe forgetting when fine-tuning parameters during incremental sessions. We evaluate the final accuracies on the miniImageNet benchmark. As shown in Fig. 9, Baseline and ALICE of [19] show the drastic performance degradation when the fine-tuning is adopted. When we extend the mixup policy of ALICE to the incremental sessions (denoted as 'ALICE-extended'), it shows moderate degradation but still suffers from performance degradation as ϵ increases. However, MICS shows consistent and acceptable performance up to $\epsilon = 0.3$. When picking the best case, MICS shows the best final accuracy at $\epsilon = 0.3$ with +0.20% gain compared with $\epsilon = 0$ case that freezes the feature extractor.

6. Conclusion

We propose a FSCIL learning strategy called MICS, which operates not only in the pre-training stage but also incremental stage. Using feature mixup and a novel label mixing policy with a virtual class, MICS enables the model to learn intra-class compact and inter-class separated representations, which leads to improved performance and robustness to network updates.

Acknowledgements

This work was supported by the Institute of Information & communications Technology Planning & Evaluation (IITP) grant funded by the Korea government (MSIT) (No. 2020-0-01336, Artificial Intelligence Graduate School Program (UNIST)), (No. 2021-0-02201, Federated Learning for Privacy Preserving Video Caching Networks) (No. 2023-RS-2022-00156361, Innovative Human Resource Development for Local Intellectualization support program), and the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2021R1C1C1012797).

References

[1] David Berthelot, Colin Raffel, Aurko Roy, and Ian Goodfellow. Understanding and improving interpolation in autoencoders via an adversarial regularizer. In *International Conference on Learning Representations (ICLR)*, 2019. 2

[2] Pietro Buzzega, Matteo Boschini, Angelo Porrello, Davide Abati, and Simone Calderara. Dark experience for general continual learning: a strong, simple baseline. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:15920–15930, 2020. 2

[3] Francisco M Castro, Manuel J Marín-Jiménez, Nicolás Guil, Cordelia Schmid, and Karteek Alahari. End-to-end incremental learning. In *European Conference on Computer Vision (ECCV)*, pages 233–248, 2018. 1, 2

[4] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *European Conference on Computer Vision (ECCV)*, pages 532–547, 2018. 2

[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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 8661–8670, 2021. 2

[6] Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation strategies from data. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 113–123, 2019. 6

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 248–255. Ieee, 2009. 7

[8] Hongyu Guo, Yongyi Mao, and Richong Zhang. Mixup as locally linear out-of-manifold regularization. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 33, pages 3714–3722, 2019. 2

[9] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016. 7

[10] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 831–839, 2019. 1, 2, 6

[11] 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. *ArXiv*, abs/2209.07529, 2022. 1, 2

[12] Do-Yeon Kim, Dong-Jun Han, Jun Seo, and Jaekyun Moon. Warping the space: Weight space rotation for class-incremental few-shot learning. In *International Conference on Learning Representations (ICLR)*, 2023. 1, 2, 6

[13] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526, 2017. 1, 2

[14] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 7

[15] Anna Kukleva, Hilde Kuehne, and Bernt Schiele. Generalized and incremental few-shot learning by explicit learning and calibration without forgetting. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9020–9029, 2021. 2

[16] Sang-Woo Lee, Jin-Hwa Kim, Jaehyun Jun, Jung-Woo Ha, and Byoung-Tak Zhang. Overcoming catastrophic forgetting by incremental moment matching. *Advances in Neural Information Processing Systems (NeurIPS)*, 30, 2017. 2

[17] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 40(12):2935–2947, 2017. 2

[18] Pratik Mazumder, Pravendra Singh, and Piyush Rai. Few-shot lifelong learning. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 35, pages 2337–2345, 2021. 1, 2, 4, 6

[19] Can Peng, Kun Zhao, Tianren Wang, Meng Li, and Brian C Lovell. Few-shot class-incremental learning from an open-set perspective. In *European Conference on Computer Vision (ECCV)*, pages 382–397, 2022. 1, 2, 3, 6, 7, 8

[20] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2001–2010, 2017. 1, 2, 6

[21] Guangyuan Shi, Jiaxin Chen, Wenlong Zhang, Li-Ming Zhan, and Xiao-Ming Wu. Overcoming catastrophic forgetting in incremental few-shot learning by finding flat minima. *Advances in Neural Information Processing Systems (NeurIPS)*, 34:6747–6761, 2021. 1, 2, 3

[22] Xiaoyu Tao, Xiaopeng Hong, Xinyuan Chang, Songlin Dong, Xing Wei, and Yihong Gong. Few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12183–12192, 2020. 1, 2, 7

[23] Vikas Verma, Alex Lamb, Christopher Beckham, Amir Najafi, Ioannis Mitliagkas, David Lopez-Paz, and Yoshua Bengio. Manifold mixup: Better representations by interpolating hidden states. In *International Conference on Machine Learning (ICML)*, pages 6438–6447. PMLR, 2019. 2, 4

[24] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Daan Wierstra, et al. Matching networks for one shot learning. *Advances in Neural Information Processing Systems (NeurIPS)*, 29, 2016. 7

[25] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 7

[26] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 374–382, 2019. 2

[27] Boyu Yang, Mingbao Lin, Yunxiao Zhang, Binghao Liu, Xiaodan Liang, Rongrong Ji, and Qixiang Ye. Dynamic support network for few-shot class incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2022. 2, 3

[28] Yaoqing Yang, Rajiv Khanna, Yaodong Yu, Amir Gholami, Kurt Keutzer, Joseph E Gonzalez, Kannan Ramchandran, and Michael W Mahoney. Boundary thickness and robustness in learning models. *Advances in Neural Information Processing Systems (NeurIPS)*, 33:6223–6234, 2020. 2, 6

[29] 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. *arXiv preprint arXiv:2302.03004*, 2023. 2, 6

[30] 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 *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 6023–6032, 2019. 2

[31] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning (ICML)*, pages 3987–3995. PMLR, 2017. 1, 2

[32] Chi Zhang, Nan Song, Guosheng Lin, Yun Zheng, Pan Pan, and Yinghui Xu. Few-shot incremental learning with continually evolved classifiers. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 12455–12464, 2021. 2, 6

[33] Hongyi Zhang, Moustapha Cisse, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (ICLR)*, 2018. 2

[34] Hanbin Zhao, Yongjian Fu, Mintong Kang, Qi Tian, Fei Wu, and Xi Li. Mgsvf: Multi-grained slow vs. fast framework for few-shot class-incremental learning. *IEEE Transactions on Pattern Analysis and Machine Intelligence (T-PAMI)*, 2021. 2

[35] 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 *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9046–9056, 2022. 1, 2, 3, 6, 7

[36] Kai Zhu, Yang Cao, Wei Zhai, Jie Cheng, and Zheng-Jun Zha. Self-promoted prototype refinement for few-shot class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 6801–6810, 2021. 2

[37] Yixiong Zou, Shanghang Zhang, Yuhua Li, and Ruixuan Li. Margin-based few-shot class-incremental learning with class-level overfitting mitigation. *arXiv preprint arXiv:2210.04524*, 2022. 3, 6