Mixture Uniform Distribution Modeling and Asymmetric Mix Distillation for Class Incremental Learning

Sunyuan Qiang[1], **Jiayi Hou**[2], **Jun Wan**[1,3,4*], **Yanyan Liang**[1*], **Zhen Lei**[3,4], **Du Zhang**[1]

[1] Macau University of Science and Technology
[2] Lafayette College
[3] National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences
[4] School of Artificial Intelligence, University of Chinese Academy of Sciences
jun.wan@ia.ac.cn, yyliang@must.edu.mo

Abstract

Exemplar rehearsal-based methods with knowledge distillation (KD) have been widely used in class incremental learning (CIL) scenarios. However, they still suffer from performance degradation because of severely distribution discrepancy between training and test set caused by the limited storage memory on previous classes. In this paper, we mathematically model the data distribution and the discrepancy at the incremental stages with *mixture uniform distribution* (MUD). Then, we propose the *asymmetric mix distillation* method to uniformly minimize the error of each class from distribution discrepancy perspective. Specifically, we firstly promote mixup in CIL scenarios with the incremental mix samplers and incremental mix factor to calibrate the raw training data distribution. Next, mix distillation label augmentation is incorporated into the data distribution to inherit the knowledge information from the previous models. Based on the above augmented data distribution, our trained model effectively alleviates the performance degradation and extensive experimental results validate that our method exhibits superior performance on CIL benchmarks.

Introduction

Processing real-world data streams is fundamental for humans to perceive the world and acquire experience. Although deep neural networks (DNNs) have exhibited significant improvements in computer vision tasks (Krizhevsky, Sutskever, and Hinton 2012; He et al. 2016), they suffer from severe performance degradation in processing streaming data. Such a nuisance phenomenon is known as catastrophic forgetting (McCloskey and Cohen 1989; Goodfellow et al. 2014), where the knowledge learned at previous stages is severely lost when learning future tasks. Therefore, class incremental learning (CIL) (Rebuffi et al. 2017; Belouadah, Popescu, and Kanellos 2021) is widely researched to relieve this problem, where the model learns new classes incrementally.

Owing to promising performances, rehearsal-based approaches (Rebuffi et al. 2017; Chaudhry et al. 2019) with knowledge distillation (KD) (Hinton, Vinyals, and Dean 2015; Li and Hoiem 2016) are widely utilized in the community. Such methods typically store limited exemplars of previous stages and train jointly with the samples of the current stage. Then, the KD is used to transfer the knowledge from the previous model to the current one. However, despite incorporating the benefits of KD methods, the insufficient memory for old classes still leads to the distortions of the training data distribution, which limits the model performance and can be considered as a major cause of catastrophic forgetting. Some recent works (Wu et al. 2019; Zhao et al. 2020) have also noted this problem, but in-depth analysis and modeling of distribution discrepancy is lacking.

In this work, we first analyze the above phenomenon by modeling the mixture uniform marginal data distribution and KL divergence is employed to measure the mismatch. The training data distribution deviates significantly from the target test distribution with growing incremental tasks, which motivates us to explore a novel data distribution model in CIL from the discrepancy perspective. Moreover, inspired by the mixup (Zhang et al. 2018, 2021), recent research communities also applied this training strategy in CIL (Mi et al. 2020; Bang et al. 2021; Zhu et al. 2021; Zhou et al. 2022). We extend our modeling framework to mixup and find that discrepancy remains as in Corollary 1 and Fig. 2(b). To this end, we propose the asymmetric mix distillation method, which minimizes the objective error on our well-designed asymmetric *Mix-Distill-Aug* data distribution model. Such a data distribution enables us to alleviate the mismatch under our modeling framework while preventing past knowledge from being forgotten, as shown in Fig. 2.

The asymmetric *Mix-Distill-Aug* data model is composed of three components: incremental mix samplers, incremental mix distillation, and incremental mix factor. Specifically, two pairs of data are firstly sampled from incremental mix samplers respectively to ensure the observation of each data samples and increase the focus on the rehearsal memory data samples. Then, the learnt knowledge of the old model is transferred to the current training stage via incremental mix distillation. The linear interpolation factors are obtained by the incremental mix factor procedure, which further slightly calibrate for mismatches in the distributions. Finally, our asymmetric *Mix-Distill-Aug* data model is obtained by linearly interpolating two pairs of distillation-based samples

[*] Corresponding author

with the above mix factors as shown in Eq. 6 and Alg. 1. Extensive experiments on three benchmark datasets, CIFAR100, ImageNet100, and CUB200, validate the effectiveness of our proposed method. In particular, we achieve about 70% and 74% average accuracy on CIFAR100 dataset and ImageNet100 over 10 stages, respectively. The main contributions of our work are as follows:

- We firstly propose to model the CIL data distribution with the mixture uniform distribution, then derive a measure of distribution discrepancy between training and test data distributions with KL divergence.
- The asymmetric mix distillation is proposed to minimize the objective error on our well-designed *Mix-Distill-Aug* data model, which consists of three components: incremental mix samplers, incremental mix distillation, and incremental mix factor.
- Extensive experiments on benchmarks showed that our method outperforms existing methods. We also conduct ablation experiments with different distribution models to validate the effectiveness of the modeling framework.

Related Work

Class incremental learning (CIL) (Belouadah, Popescu, and Kanellos 2021) can be viewed as a branch of continual learning (CL) (van de Ven and Tolias 2019). Generally, the CL methods (Lange et al. 2022) can be divided into three themes: regularization-based methods, parameter isolation methods, and rehearsal methods. (1). *Regularization* based methods (Kirkpatrick et al. 2016; Li and Hoiem 2016; Zenke, Poole, and Ganguli 2017; Aljundi et al. 2018) constrained the model by adding an extra regularization term to prevent forgetting previous knowledge. (2). *Parameter isolation* methods (Mallya, Davis, and Lazebnik 2018; Hung et al. 2019) fixed previous model parameters and incrementally allocated additional model parameters to alleviate the forgetting, which are not suitable for CIL scenario due to unavailable task IDs. (3). *Rehearsal* based methods (Rebuffi et al. 2017; Chaudhry et al. 2019; Shin et al. 2017) retained a small subset of previous data or synthesize the pseudo samples with generative models for jointly training.

Among them, the rehearsal based methods with knowledge distillation (KD) (Rebuffi et al. 2017; Castro et al. 2018; Hou et al. 2019; Douillard et al. 2020; Yan, Xie, and He 2021; Li, Wan, and Yu 2022; Kang, Park, and Han 2022; Liu, Schiele, and Sun 2021) received great attention with superior performance in CIL. LwF (Li and Hoiem 2016) firstly introduced the knowledge distillation into continual learning while (Rebuffi et al. 2017) extended it to rehearsal based CIL with promising performance. Later, different variants of distillation loss to keep knowledge from the old model to the current model were widely used in CIL, such as less-forget constraint (Hou et al. 2019), pooled outputs distillation (Douillard et al. 2020), and importance weighted feature maps distillation (Kang, Park, and Han 2022). Some recent works (Yan, Xie, and He 2021; Douillard et al. 2022; Wang et al. 2022) also proposed dynamic architecture based methods in CIL, which add additional learnable parameters for learning new classes. However, growing classes can lead to severe model overhead. Our proposed method is also based on rehearsal based methods with KD, but we further explore the discrepancy between training and test data distribution in CIL and extend the mixup in CIL to propose a novel method from distribution discrepancy perspective. In the following, we discuss the related work of imbalance problems and mixup strategy in CIL.

Imbalance in CIL. The performance of DNNs on class imbalanced datasets has received extensive attention (Dong, Gong, and Zhu 2019). Recently, many works (Castro et al. 2018; Hou et al. 2019; Wu et al. 2019; Zhao et al. 2020; He, Wang, and Chen 2021) have found the imbalanced problem of rehearsal strategy in CIL. An additional fine-tuning stage with balanced subdataset is added in training process (Castro et al. 2018). (Hou et al. 2019) introduced the margin ranking loss to tackle the imbalance. BiC method (Wu et al. 2019) and weight aligning (WA) (Zhao et al. 2020) are proposed to correct the final biased classification layer. However, all of the above works lack the analysis of imbalance mismatches between training and test data distribution in CIL scenario.

Mixup in CIL. Mixup (Zhang et al. 2018) is a widely used training strategy in the recent research community to improve model generalization and robustness (Zhang et al. 2021). In CIL, (Mi et al. 2020) simply applied mixup in both samples in the current stages and replay exemplars. (Zhu et al. 2021) proposed class augmentation, which utilizes the mixup to augment the original classes by synthesizing auxiliary virtual classes. (Bang et al. 2021) proposed to use cutmix (Yun et al. 2019) to alleviate the side effects caused by class distribution, while in (Zhou et al. 2022), manifold mixup (Verma et al. 2019) is applied to fuse instance as a virtual new class for few-shot CIL. However, we find that directly using mixup strategy cannot effectively alleviate the problem of data distribution mismatch under our proposed framework and the performance improvement of this method is still limited.

Preliminaries

Problem Formulation. In class incremental learning scenarios, the training streaming dataset $\{\mathcal{D}_t\}_{t=1}^{T}$ consists of N_t sample pairs at each stage t, $\{(x_{t,i}, y_{t,i}) \,|\, x_{t,i} \in \mathcal{X}_t, y_{t,i} \in \mathcal{Y}_t\}_{i=1}^{N_t}$, where $x_{t,i}$ and $y_{t,i}$ is the i^{th} sample at the t^{th} stage sampled from data and label space $\mathcal{X}_t$ and $\mathcal{Y}_t$, respectively. The label spaces at different stages are disjoint $\mathcal{Y}_i \cap \mathcal{Y}_j = \emptyset, i \neq j$. At stage t, a deep neural network model with parameter θ is trained using the available N_t samples from the dataset $\mathcal{D}_t$ while the trained model is tested on all seen classes $\mathcal{Y}_{1:t}$. Due to the absence of datasets $\mathcal{D}_{1:t-1}$ of previous stages, the model tends to minimize the objective error of current classes $\mathcal{Y}_t$ instead of all seen classes $\mathcal{Y}_{1:t}$, leading to the dilemma of catastrophic forgetting. In this paper, we adopt the rehearsal strategy in CIL setting (Rebuffi et al. 2017), where a small amount of memory buffer is available $\mathcal{M}_t^{N_{\text{mem}}} \subset \{\mathcal{D}_1, \mathcal{D}_2, ..., \mathcal{D}_{t-1}\}$ at stage t, and N_{mem} denotes the size of the memory buffer.

ERM and VRM. The supervised learning aims to find the hypothesis $f \in \mathcal{F}$ that builds the connection between inputs data x and label y. Given an objective function $\ell : \mathcal{Y} \times \mathcal{Y} \mapsto$

$\mathbb{R}$, the expected risk error $R(f) = \int \ell(f(x), y)\mathrm{d}P(x, y)$ is intractable due to the unknown data distribution $P(x, y)$. The empirical risk error (ERM) (Vapnik 1991) $R_{\text{erm}}(f) = \frac{1}{n}\sum_{i=1}^{n} \ell(f(x_i), y_i)$ is introduced to minimize the error on the empirical data distribution $(x_i, y_i) \sim P_\delta(x, y)$ to approximate the expected risk. However, the converged model obtained by ERM is only reliable under the assumption that the empirical data distribution P_δ is close to the true data distribution P. Severely biased prediction occurs when the true data distribution is poorly approximated. Later, vicinal risk minimization (VRM) (Chapelle et al. 2000) $R_{\text{vrm}}(f) = \frac{1}{m}\sum_{i=1}^{m} \ell(f(\tilde{x}_i), \tilde{y}_i)$ is introduced to fit a richer distribution, which extends the empirical delta distribution P_δ to the vicinity distribution of data points P_ν, where $(\tilde{x}_i, \tilde{y}_i) \sim P_\nu(\tilde{x}, \tilde{y})$. One of the widely used strategies is that $(\tilde{x}_i, \tilde{y}_i)$ can be computed by a linear interpolation operation in mixup (Zhang et al. 2018). Specifically, given training samples (x_i, y_i), (x_j, y_j) randomly sampled from the empirical data distribution, the mixup based vicinal point $(\tilde{x}_{i,j}, \tilde{y}_{i,j})$ is calculated as follows:

$$\begin{aligned} \tilde{x}_{i,j} &= \lambda \cdot x_i + (1-\lambda) \cdot x_j, \\ \tilde{y}_{i,j} &= \lambda \cdot y_i + (1-\lambda) \cdot y_j, \end{aligned} \tag{1}$$

where λ is usually drawn from Beta distribution $Beta(\alpha, \alpha)$.

Data Distribution Modeling

Mathematically, the empirical data distribution $P_{\delta_t}(x, y)$ is formed by training samples from training dataset $\mathcal{D}_t$ and memory buffer $\mathcal{M}_t$ at stage t in CIL. Then, the marginal distribution $P_{\delta_t}(y)$ is formalized as follows:

$$P_{\delta_t}(y) = \int P_{\delta_t}(x, y)\mathrm{d}x. \tag{2}$$

In rehearsal-based CIL (Rebuffi et al. 2017), the number of instances of each class is close in the current dataset $\mathcal{D}_t$ and in memory buffer $\mathcal{M}_t$, respectively. Therefore, one can further describe the marginal distribution by a mixture uniform distribution in Eq. 3. Here, we omit the dirac δ function flags for simplicity.

$$\begin{aligned} P_t(y) = & \frac{N_{\text{mem}}}{N_{\text{mem}} + N_t} \cdot \mathbf{U}(0, |\mathcal{Y}_{1:t-1}|] \\ & + \frac{N_t}{N_{\text{mem}} + N_t} \cdot \mathbf{U}(|\mathcal{Y}_{1:t-1}|, |\mathcal{Y}_{1:t}|], \end{aligned} \tag{3}$$

where $\mathbf{U}$ denotes the uniform distribution. $|\mathcal{Y}_{1:t-1}|$ denotes the number of classes in previous tasks, while $|\mathcal{Y}_{1:t}|$ denotes the number of all classes observed so far. N_{mem} and N_t denote the total number of samples in memory buffer and current new task, respectively. It usually minimizes the objective function ℓ to penalize the difference between predictions $f(x)$ and ground truth label y with ERM. On the contrary, the marginal class distribution of test data is usually balanced and the model is required to have the same preference for each class. Then, the test distribution can be directly considered as a uniform distribution, $P_t^{\text{test}}(y) = \mathbf{U}(0, |\mathcal{Y}_{1:t}|]$. The mismatch with Eq. 3 (P_t^{test} and P_t) often leads to severe bias in model predictions. To this end, we simply utilize KL divergence to analytically measure the discrepancy between above marginal distributions with the mixture uniform distribution.

In addition to directly perform the ERM on mixture uniform data distribution in Eq. 3, *knowledge distillation* (Hinton, Vinyals, and Dean 2015; Rebuffi et al. 2017; Douillard et al. 2020) is widely used in CIL to transfer learned knowledge from the old models to the current one. Among them, we treat the knowledge learned from logits as a label function, augmenting the original data samples into *Distill-Aug* samples.

$$\begin{aligned} \hat{x} &= x, \\ \hat{y} &= \left[\sigma(f_{\theta_{t-1}}(\hat{x}))^{\mathcal{Y}_{1:t-1}}; y^{\mathcal{Y}_t}\right], \end{aligned} \tag{4}$$

where $f_{\theta_{t-1}}$ denotes the learned model at stage $t-1$. (x, y) are sampled from empirical training distribution $P_t(x, y)$, $[\,\cdot\,;\cdot\,]$ denotes the concatenate operation, and activation function $\sigma(\cdot)$ converts the predicted logits to the probability. The new virtual label $\hat{y}$ of each data sample $\hat{x}$ is composed of the distilled label from the old model and the ground truth label in the corresponding label space. To analyze the insufficiency in marginal data distribution of knowledge distillation, we treat the distilled labels $\hat{y}$ as inherited from the original labels y and raise the following definition.

Definition 1 (*Distill-Aug*)**.** *The virtual sample pair $(\hat{x}, \hat{y})$ generated by Eq. 4 is defined as a distilled augmented (Distill-Aug) sample, which is a robust sample of original class y in model's feature learning.*

The *Distill-Aug* definition provides us with a way to analyze the data distribution independently of knowledge distillation in CIL. As the *Distill-Aug* samples belong to the original labels, we can derive that the class marginal distribution $P_{t\text{-distill}}$ follows the same distribution in Eq. 3, causing the discrepancy between the training and test distributions remains. Thus, minimizing the error on *Distill-Aug* data model $P_{t\text{-distill}}$ still focuses on the current classes $\mathcal{Y}_t$ instead of all seen classes $\mathcal{Y}_{1:t}$.

Asymmetric Mix Distillation

In this section, we introduce the proposed asymmetric mix distillation, which minimizes the risk error on the asymmetric *Mix-Distill-Aug* data distribution model. Three key components, incremental mix samplers, incremental mix distillation, and incremental mix factor are used to construct the data model. In the following, we describe each component in detail and the complete training procedure is summarized in Alg. 1.

Incremental Mix Samplers

The incremental mix samplers include two sampling strategies for later performing mixup linear interpolation operations and distillation label augmentation over two pairs of samples. We propose incremental reverse sampler and combine instance sampler to form the final incremental mix samplers, which both focus on the rehearsal memory data and ensure that each sample is observed over training. *Instance*

Sampler: As shown in Eq. 3, the coefficient of the data distribution model is determined by the memory buffer size N_{mem} and sample size of current task N_t, which can be viewed as a common sampling method called instance sampler (Kang et al. 2020). We simplify the Eq. 3 with probability coefficient $p = N_{\text{mem}}/(N_{\text{mem}} + N_t)$ to obtain our instance sampler based data model P_t. The instance sampler commonly traverses the entire dataset once to ensure that each sample is observed. *Incremental Reverse Sampler:* With the mixture uniform distribution modeling framework, we firstly formally give the data distribution model determined by q as follows.

$$\begin{aligned} Q_t(y) = q \cdot \mathbf{U}(0, |\mathcal{Y}_{1:t-1}|] \\ + (1-q) \cdot \mathbf{U}(|\mathcal{Y}_{1:t-1}|, |\mathcal{Y}_{1:t}|]. \end{aligned} \tag{5}$$

We design q to be a quadratic ratio of the number of classes $\frac{(|\mathcal{Y}_{1:t-1}|)^2}{(|\mathcal{Y}_{1:t-1}|)^2+(|\mathcal{Y}_t|)^2}$ to maintain focus on the rehearsal memory samples in the increasing incremental stages. In the next subsection, we discuss the results of our mixed data model based on the above sampling strategies, and find that such a model can greatly alleviate the discrepancy between training and test data distributions.

Incremental Mix Distillation

With Eq. 1 and Eq. 4, we formulate our incremental mix distillation as follows, where mixed samples are fed into the old model to obtain knowledge information.

$$\begin{aligned} &\tilde{x}_{i,j} = \lambda \cdot x_i + (1-\lambda) \cdot x_j, \\ &\tilde{y}_{i,j} = \left[\sigma(f_{t-1}(\tilde{x}_{i,j}))^{\mathcal{Y}_{1:t-1}}; [\lambda \cdot y_i + (1-\lambda) \cdot y_j]^{\mathcal{Y}_t}\right]. \end{aligned} \tag{6}$$

Inspired by (Xu, Chai, and Yuan 2021), we extend mixup strategy with *Distill-Aug* definition and raise the *Mix-Distill-Aug* below in order to analyze the marginal data distribution.

Definition 2 (*Mix-Distill-Aug*)**.** *The virtual sample pair* $(\tilde{x}_{i,j}, \tilde{y}_{i,j})$ *generated by Eq. 6 with mixing factor* λ *is defined as a Mix-Distill-Aug sample, which is a robust sample of class* y_i *(class* y_j*) iff* $\lambda \geq 0.5 (\lambda < 0.5)$ *in model's feature learning.*

Corollary 1. *When mixing factor* $\lambda \in [0, 1]$ *sampled from a symmetric distribution, the virtual dataset composed of Mix-Distill-Aug sample pairs* $(\tilde{x}_{i,j}, \tilde{y}_{i,j})$ *in Eq. 6 follows the same marginal distribution* P_t *in Eq. 3, where* (x_i, y_i) *and* (x_j, y_j) *are both randomly sampled from instance sampler* P_t.

As in Corollary 1, we derive that performing mixup strategy with incremental mix distillation results in the same distribution P_t. We refer such data model as base *Mix-Distill-Aug* model $(\tilde{x}_{i,j}, \tilde{y}_{i,j}) \sim \tilde{P}_{t\text{-base}}$, and experimentally find that the performance improvement of the model is limited in Fig. 2.

Corollary 2. *When mixing factor* $\lambda \in [0, 1]$ *sampled from a symmetric distribution, the virtual dataset composed of Mix-Distill-Aug sample pairs* $(\tilde{x}_{i,j}, \tilde{y}_{i,j})$ *in Eq. 6 follows a new marginal distribution* $\widetilde{P}_{t\text{-sym}}$ *in Eq. 7, where* (x_i, y_i) *and* (x_j, y_j) *are randomly sampled from instance sampler* P_t *and incremental reverse sampler* Q_t*, respectively.*

Algorithm 1: Asymmetric Mix Distillation

```
Input: Training data $\mathcal{D}_t \cup \mathcal{M}_t = \{x_i, y_i\}_{i=1}^{N_t+N_{\text{mem}}}$; Incremental mix samplers $P_t$ and $Q_t$; Previous model parameters $\theta_{t-1}$; Stage $t$, $(t > 1)$.
Output: Model parameters $\theta_t$;
for $k$ steps do
    Sample minibatch $(x_i, y_i)$ from $P_t$ in Eq. 3;
    Sample minibatch $(x_j, y_j)$ from $Q_t$ in Eq. 5;
    Sample the factors $\lambda$ from asymmetric $\widetilde{P}_\lambda$ in Eq. 8;
    Construct the asymmetric Mix-Distill-Aug data model $(\tilde{x}_{i,j}, \tilde{y}_{i,j}) \sim \tilde{P}_{t\text{-asym}}$ with $(x_i, y_i)$, $(x_j, y_j)$, $\theta_{t-1}$, and $\lambda$ in Eq. 6;
    Train $\theta_t$ by minimizing the objective error in Eq. 10;
end for
```

$$\begin{aligned} \widetilde{P}_{t\text{-sym}}(y) = \frac{p+q}{2} \cdot \mathbf{U}(0, |\mathcal{Y}_{1:t-1}|] \\ + \frac{2-p-q}{2} \cdot \mathbf{U}(|\mathcal{Y}_{1:t-1}|, |\mathcal{Y}_{1:t}|)], \end{aligned} \tag{7}$$

where p and q denote the coefficient of distribution P_t and Q_t, respectively. As in Corollary 2, with our proposed incremental mix samplers, the new distribution data model $\widetilde{P}_{t\text{-sym}}$, termed symmetric *Mix-Distill-Aug* model, increases the focus on rehearsal memory data with the probability coefficient $\frac{p+q}{2}$, alleviating the gap between the training and the test data distribution. The distance measured by KL divergence over incremental stages is shown in Fig. 2(b), we can see that the $\widetilde{P}_{t\text{-sym}}$ model greatly reduces the distribution distance compared to $\widetilde{P}_{t\text{-base}}$, but a small gap still exists in the growing incremental stages. In the next subsection, we introduce the incremental mix factor to further slightly calibrate the data distribution model.

Incremental Mix Factor

As discussed above, small gap still exists in the symmetric *Mix-Distill-Aug* model with probability coefficient $\frac{p+q}{2}$ in Eq. 7. In Corollary 2, the mixing factors are sampled from a symmetric distribution P_λ such as beta distribution with parameter α, $Beta(\alpha, \alpha)$. We propose a novel mixing factor method, termed incremental mix factor, which the linear interpolation factors λ are sampled from the asymmetric distribution $\widetilde{P}_\lambda$ to calibrate the symmetric data distribution model under our modeling framework at each stage t. And the asymmetric mix factors distribution $\widetilde{P}_\lambda$ is defined as an extension of the original factor distribution.

$$\widetilde{P}_\lambda = m \cdot P_{\lambda(0.5,1)} + (1-m) \cdot P_{\lambda(0,0.5)}, \tag{8}$$

where $P_{\lambda(a,b)}$ maps the original beta sampling results from $[0, 1]$ to $[a, b]$ with linear function $\tilde{\lambda} = \lambda \cdot (b-a) + a$. Therefore, we divide the sampling space $[0, 1]$ with $[0, 0.5]$ and $[0.5, 1]$. By designing a reasonable probability m, we can

Dataset	CIFAR100			ImageNet100		CUB200	
stages	5	10	20	5	10	10	20
iCaRL	$67.31_{\pm 1.26}$	$64.32_{\pm 1.28}$	$60.19_{\pm 1.18}$	72.23	66.78	65.91	56.90
BiC	$67.01_{\pm 1.52}$	$63.98_{\pm 1.03}$	$60.89_{\pm 1.60}$	73.21	64.98	67.12	59.78
WA	$69.28_{\pm 0.74}$	$67.45_{\pm 1.07}$	$66.31_{\pm 1.27}$	73.73	67.41	65.73	58.45
PODNet	$64.03_{\pm 0.79}$	$54.66_{\pm 0.65}$	$46.49_{\pm 1.46}$	75.18	67.75	53.19	45.29
SSIL	$63.73_{\pm 1.29}$	$57.44_{\pm 1.03}$	$52.02_{\pm 1.23}$	71.94	64.05	66.02	58.01
AFC	$65.99_{\pm 1.09}$	$60.42_{\pm 1.42}$	$54.91_{\pm 1.12}$	76.36	69.73	60.71	56.64
Ours	$72.08_{\pm 0.81}$	$70.18_{\pm 0.38}$	$68.16_{\pm 0.89}$	77.24	74.37	71.07	62.66

Table 1: Performance comparison of CIFAR100, ImageNet100, and CUB200 benchmarks. Average accuracies (%) over all stages are reported. For CIFAR100 benchmarks, we run experiments using three different class orders and report their averages and standard deviations.

control the preferences for the samplers P_t and Q_t according to *Mix-Distill-Aug* definition.

Corollary 3. *When mixing factor $\lambda \in [0,1]$ sampled from an asymmetric distribution in Eq. 8, the virtual dataset composed of Mix-Distill-Aug sample pairs $(\tilde{x}_{i,j}, \tilde{y}_{i,j})$ in Eq. 6 follows a new marginal distribution $\widetilde{P}_{t\text{-asym}}$ in Eq. 9, where (x_i, y_i) and (x_j, y_j) are randomly sampled from instance sampler P_t and incremental reverse sampler Q_t, respectively.*

$$\begin{aligned}\widetilde{P}_{t\text{-asym}}(y) = p_{\text{asym}} \cdot \mathbf{U}(0, |\mathcal{Y}_{1:t-1}|] \\ + (1-p_{\text{asym}}) \cdot \mathbf{U}(|\mathcal{Y}_{1:t-1}|, |\mathcal{Y}_{1:t}|],\end{aligned} \tag{9}$$

where $p_{\text{asym}} = p \cdot m + q \cdot (1-m)$. p and q denote the coefficient of distribution P_t and Q_t, respectively. m denotes the coefficient of asymmetric $\tilde{P}_\lambda$. In this paper, we set m to be $\left(q - \frac{|\mathcal{Y}_{1:t-1}|}{|\mathcal{Y}_{1:t}|}\right)/(q-p)$, to build asymmetric *Mix-Distill-Aug* data model with zero distribution discrepancy, as shown in Fig. 2(b).

Summary

In this subsection, we briefly give the summary of our proposed incremental mix distillation method and complete training procedure. In the first stage $t = 1$, giving training data (x_i, y_i),$(x_j, y_j) \sim \mathcal{D}_1$, we simply train the model with parameters θ_1 on mixup samples $(\tilde{x}_{i,j}, \tilde{y}_{i,j})$ in Eq. 1. $\min_{\theta_1} \ell(f_{\theta_1}(\tilde{x}_{i,j}), \tilde{y}_{i,j})$, where the objective function ℓ is set to binary cross entropy as in (Rebuffi et al. 2017). In the following incremental stages $t > 1$, giving training data $\mathcal{D}_t \cup \mathcal{M}_t$, we firstly sample data pairs (x_i, y_i) and (x_j, y_j) from the incremental mix samplers P_t and Q_t in Eq. 3 and Eq. 5, respectively. Then, the linear interpolation factor λ is sampled from asymmetric distribution $\tilde{P}_\lambda$ in Eq. 8. With the previous model $f_{\theta_{t-1}}$, the above data pairs, and the linear interpolation factor, we construct the asymmetric *Mix-Distill-Aug* data distribution model $(\tilde{x}_{i,j}, \tilde{y}_{i,j})$ with Eq. 6.

$$\min_{\theta_t} \ell(f_{\theta_t}(\tilde{x}_{i,j}), \tilde{y}_{i,j}). \tag{10}$$

Similarly, we minimize the error on asymmetric *Mix-Distill-Aug* samples with the binary cross entropy in Eq. 10 and the details are shown in Alg. 1. Our proposed method shows excellent performance without additional network parameters and training computation. In the inference, we adopt the NME strategy (Rebuffi et al. 2017) for predictions.

Experiments

In this section, we follow the previous evaluation protocols in CIL and conduct extensive experiments to validate the effectiveness of our method.

Datasets and Evaluation Protocols

Datasets We employ CIFAR100 (Krizhevsky and Hinton 2009), ImageNet100 (Deng et al. 2009), and CUB200 (Wah et al. 2011) for evaluation. CIFAR100 is 32×32 color images datasets containing 50,000 training images and 10,000 testing images with 100 classes. ImageNet100 is a subset of the ImageNet large dataset, which we selected the same 100 classes according to previous CIL evaluation protocols (Rebuffi et al. 2017; Douillard et al. 2020). CUB200 is a dataset containing 200 classes of fine-grained visual birds with 11,788 images in total.

Protocols For different comparison methods, we keep the same backbone network for fair comparison. Following the recent CIL works (Rebuffi et al. 2017), we employ the modified ResNet32 backbone for CIFAR100 datasets, and the standard ResNet18 (He et al. 2016) for ImageNet100 and CUB200 datasets. The specific evaluation settings are described below. (1) *CIFAR100:* We split the 100 classes into 5, 10, and 20 stages with a total memory size of 2,000. (2) *ImageNet100:* The 100 classes are split into 5 and 10 stages with a total memory size of 2,000. (3) *CUB200:* Following the (Yu et al. 2020; Zhou, Ye, and Zhan 2021), the backbone network pretrained on ImageNet is required for CUB200 dataset. We split the 200 classes into 10 and 20 stages and the memory buffer size is set to 3 images per class.

Comparison Results

We compare our method with various models, iCaRL (Rebuffi et al. 2017), BiC (Wu et al. 2019), WA (Zhao et al. 2020), PODNet (Douillard et al. 2020), SSIL (Ahn et al. 2021), and AFC (Kang, Park, and Han 2022). The average

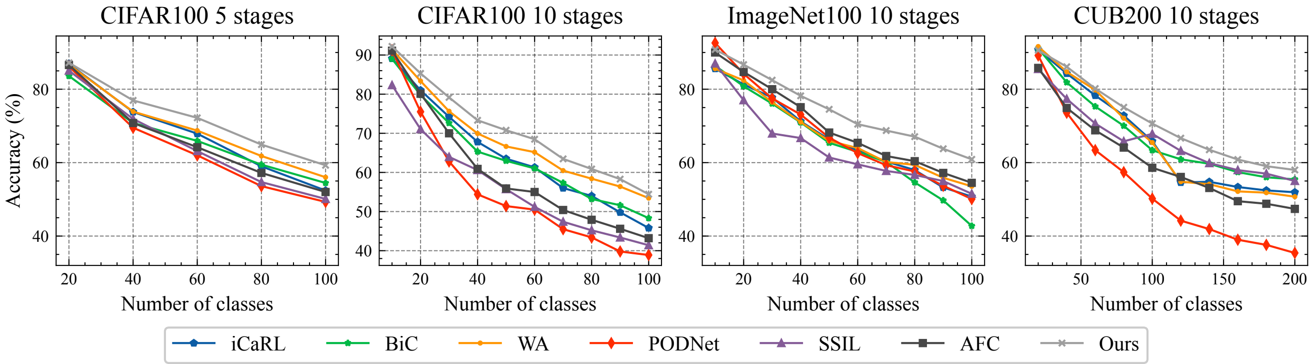
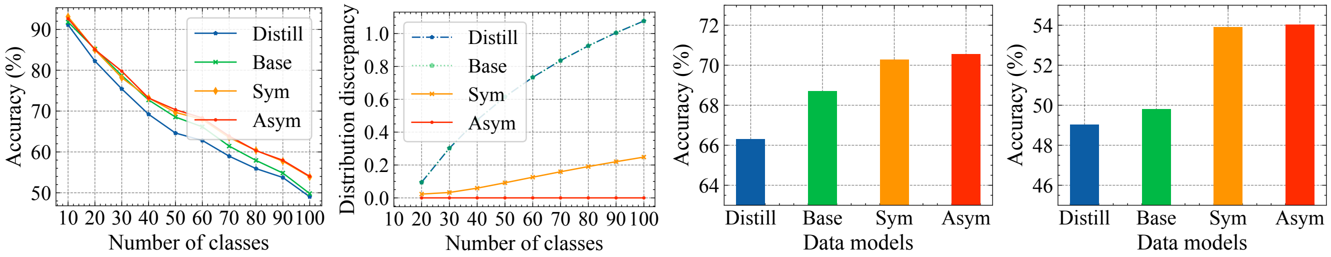


Figure 1: The visualization results of performance comparison for each stage.



(a) Accuracy over stages.
(b) Distribution discrepancy.
(c) Average accuracy.
(d) Last accuracy.

Figure 2: The visualization results of subfigures all come from 10 steps CIFAR100 dataset evaluation experiment. Distill: *Distill-Aug* data model; Base: base *Mix-Distill-Aug* data model; Sym: symmetric *Mix-Distill-Aug* data model; Asym: asymmetric *Mix-Distill-Aug* data model. (a) The accuracy over 10 stage. (b) Discrepancy between training and test distribution measured by KL divergence. (c) The average accuracy of all stages. (d) The accuracy of last stage.

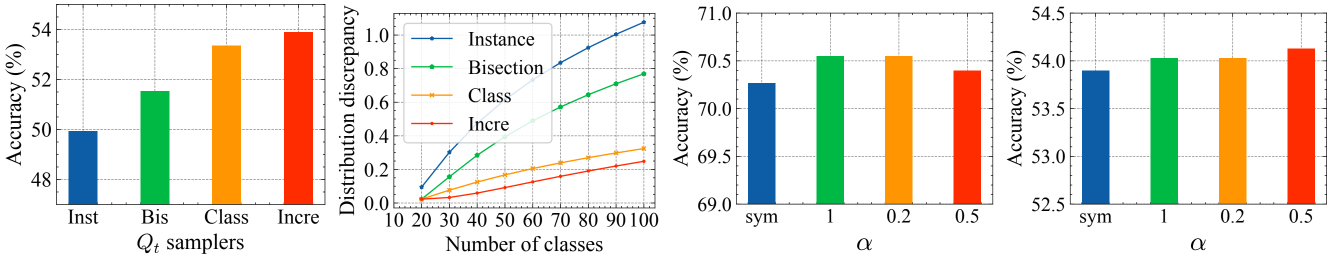
accuracies (Rebuffi et al. 2017) over all stages are reported for quantitative evaluation. The implementation details of our method are added to the supplementary material.

Table 1 and Fig 1 summarizes the comparison results of different benchmarks. For CIFAR100 benchmarks, we run experiments using three different class orders and report their averages and standard deviations. The experimental results show that our method surpasses all previous models on average accuracies and per-step accuracy. Particularly, we achieve 2.8% and 1.85 % improvement with 5 stages and 20 steps in both short and long increment stages, respectively. In the setting of CIFAR100 of 10 steps, we surpass the state-of-the-art method from 67.45% to 70.18% (+2.73%), which proves the effectiveness of our method. In the supplementary material, we also evaluate on another common protocol with base training stage (Douillard et al. 2020) and achieve competitive results. As for the ImageNet100 benchmarks, we can see that our method achieve about 77.24% and 74.37% in 5 stages and 10 stages protocols, which outperforms the other methods. We also add comparative results for top 5 average accuracy (Rebuffi et al. 2017) in the supplementary material. For CUB200 benchmarks, due to the extreme scarcity of rehearsal memory, m is directly set to 0.5 in this case to prevent repeated memory samples from dominating. We can see that our method still outperforms other methods. In the protocols of 200 classes with 10 classes per step (20 stages), we improve from 59.78% to 62.66% (+2.88%), which further validates the effectiveness of our method.

Ablation Study

In this subsection, we conduct exhaustive ablation study to validate our proposed data model. We also discuss the performance by varying samplers with different discrepancy measured by KL divergence. Moreover, we study the effect of different asymmetric mixing factor distributions.

Data Distribution Models. Table 2 and Fig. 2 summarizes the results of ablation experiments on data distribution models with CIFAR100 10 stages benchmark. We use the basic knowledge distillation label augmentation data model *Distill-Aug* in Definition 1 as the baseline model. The base *Mix-Distill-Aug* data model in Corollary 1 improves the average accuracy from 66.31% to 68.71% and last accuracy from 49.02% to 49.81%, respectively, which shows that mix distillation is an effective strategy with mixup. However, as shown in Fig. 2(b), the distribution discrepancy of such data model still increases significantly as the classes accumulate in the incremental stages, and the improvement of last accuracy is limited compared with the average accuracy. As for the symmetric *Mix-Distill-Aug* data model in Corollary 2, it significantly improves the last accuracy by 4.88%. The distribution distance curve is also greatly reduced compared with the above two models. We reduce the distribution gap



(a) Last accuracy. (b) Distribution discrepancy. (c) Average accuracy. (d) Last accuracy.

Figure 3: (a) The accuracy of last stage on CIFAR100 with different Q_t samplers. (b) Discrepancy between training and test distribution measured by KL divergence on CIFAR100 with different Q_t samplers. (c) The average accuracy of all stages on CIFAR100 with different asymmetric mixing factor distribution. (d) The accuracy of last stage on CIFAR100 with different asymmetric mixing factor distribution.

Models	Avg	Last	Δ
Distill-Aug	66.31	49.02	-
base *Mix-Distill-Aug*	68.71	49.81	+0.79
sym *Mix-Distill-Aug*	70.26	53.90	+4.88
asym *Mix-Distill-Aug*	70.55	54.03	+5.01

Table 2: Performance comparison of CIFAR100 10 stages benchmark with different data distribution models.

Q_t	q	Avg	Last
Instance	$\frac{N_{\text{mem}}}{N_{\text{mem}}+N_t}$	68.27	49.93
Bisection	0.5	69.27	51.53
Class-balance	$\frac{\lvert\mathcal{Y}_{1:t-1}\rvert}{\lvert\mathcal{Y}_{1:t}\rvert}$	69.95	53.35
Incre-reverse	$\frac{(\lvert\mathcal{Y}_{1:t-1}\rvert)^2}{(\lvert\mathcal{Y}_{1:t-1}\rvert)^2+(\lvert\mathcal{Y}_t\rvert)^2}$	70.26	53.90

Table 3: Performance comparison of CIFAR100 10 stages benchmark with different Q_t samplers.

while also increasing the accuracy, showing that a smaller distribution gap between training and test sets may lead to higher model accuracy. With the asymmetric mix factor distribution, the distribution distance is narrowed to zero at all incremental stages, thereby uniformly minimizing the error of each class under our MUD modeling framework. Our final asymmetric *Mix-Distill-Aug* data model in Corollary 3 further slightly improves the average and last accuracy to 70.55% and 54.03%, respectively, which validates the effectiveness of our modeling framework and the proposed method. From the accuracies over 10 stages in Fig. 2(a), the models we designed have obvious improvement compared with the baseline model, and the detail visualization results of average and last accuracy are shown in Fig. 2(c) and 2(d). Moreover, as shown in Fig. 2, on CIFAR100 10 stages benchmark, from base to asym data model, the discrepancy curve measured by KL divergence shifts down gradually, and both the average accuracy and the last accuracy increase. We empirically find that such discrepancy may lead to the degradation of our data model, and the results validate the effectiveness of the method and modeling framework.

Samplers and Distribution Discrepancy. To assess the effectiveness of the incremental reverse sampler and further analyze the modeling framework, we vary the coefficient q of Q_t to build different symmetric *Mix-Distill-Aug* data model. Table 3 summarizes the results with different coefficient q settings. Compared with the basic instance sampler, we significantly improve the last accuracy from 49.93% to 53.90% with incremental reverse sampler, which demonstrates the effectiveness of our proposed sampler method. Additionally, as shown in Fig. 3(a) and 3(b), the distribution distance decreases with varying q from instance sampler to incremental reverse sampler, and the accuracy also gradually increases, which again validates the effectiveness of our proposed framework. We argue that under our modeling framework and data model settings, large distribution discrepancy may lead to model degradation.

Mixing Factor Distribution. Fig. 3(c) and 3(d) visualize the performance results for different mix factor distributions $\tilde{P}_\lambda$ with varying hyper-parameter α. The sym refers to the symmetric *Mix-Distill-Aug* data model with $Beta(1, 1)$ mix factor distribution. We can see that both average and last accuracy are improved slightly compared with the sym data model, and we simply choose 0.2 as our method setting. We also evaluate some other $\tilde{P}_\lambda$ distribution settings and presented in the supplementary material.

Conclusion

In this work, we formulate the marginal class data distribution with mixture uniform distribution (MUD) and systematically analyze the distribution discrepancy between training and test set for class incremental learning scenario. We propose the asymmetric mix distillation to uniformly minimize the model error of each class by extending mixup strategy with three key components. The incremental mix samplers and the mix factor calibrate the raw data distribution from distribution discrepancy perspective, and the mix distillation transfers previous knowledge to the current stage with label augmentation. Extensive experiments show that our method obtains superior performance on CIL benchmarks.

Acknowledgments

This work was supported by the National Key Research and Development Plan under Grant 2021YFE0205700, CCF-Zhipu AI Large Model Fund (CCF-Zhipu202318), the External cooperation key project of Chinese Academy Sciences 173211KYSB20200002, the Science and Technology Development Fund of Macau (0025/2019/AKP, 0004/2020/A1, 0070/2020/AMJ), Guangdong Provincial Key R&D Programme 2019B010148001, and Open Research Projects of Zhejiang Lab No. 2021PF0AB01.

References

Ahn, H.; Kwak, J.; Lim, S.; Bang, H.; Kim, H.; and Moon, T. 2021. SS-IL: Separated Softmax for Incremental Learning. In *2021 IEEE/CVF International Conference on Computer Vision, ICCV 2021*, 824–833. IEEE.

Aljundi, R.; Babiloni, F.; Elhoseiny, M.; Rohrbach, M.; and Tuytelaars, T. 2018. Memory Aware Synapses: Learning What (not) to Forget. In *Computer Vision - ECCV 2018 - 15th European Conference*, volume 11207, 144–161. Springer.

Bang, J.; Kim, H.; Yoo, Y.; Ha, J.; and Choi, J. 2021. Rainbow Memory: Continual Learning With a Memory of Diverse Samples. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021*, 8218–8227. Computer Vision Foundation / IEEE.

Belouadah, E.; Popescu, A.; and Kanellos, I. 2021. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*, 135: 38–54.

Castro, F. M.; Marín-Jiménez, M. J.; Guil, N.; Schmid, C.; and Alahari, K. 2018. End-to-End Incremental Learning. In *Computer Vision - ECCV 2018 - 15th European Conference*, volume 11216, 241–257. Springer.

Chapelle, O.; Weston, J.; Bottou, L.; and Vapnik, V. 2000. Vicinal Risk Minimization. In *Advances in Neural Information Processing Systems 13, 2000*, 416–422. MIT Press.

Chaudhry, A.; Rohrbach, M.; Elhoseiny, M.; Ajanthan, T.; Dokania, P. K.; Torr, P. H. S.; and Ranzato, M. 2019. Continual Learning with Tiny Episodic Memories. *CoRR*, abs/1902.10486.

Deng, J.; Dong, W.; Socher, R.; Li, L.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009)*, 248–255. IEEE Computer Society.

Dong, Q.; Gong, S.; and Zhu, X. 2019. Imbalanced Deep Learning by Minority Class Incremental Rectification. *IEEE Trans. Pattern Anal. Mach. Intell.*, 41(6): 1367–1381.

Douillard, A.; Cord, M.; Ollion, C.; Robert, T.; and Valle, E. 2020. PODNet: Pooled Outputs Distillation for Small-Tasks Incremental Learning. In *Computer Vision - ECCV 2020 - 16th European Conference*, volume 12365, 86–102. Springer.

Douillard, A.; Ramé, A.; Couairon, G.; and Cord, M. 2022. DyTox: Transformers for Continual Learning with DYnamic TOken eXpansion. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, 9275–9285. IEEE.

Goodfellow, I. J.; Mirza, M.; Da, X.; Courville, A. C.; and Bengio, Y. 2014. An Empirical Investigation of Catastrophic Forgeting in Gradient-Based Neural Networks. In *2nd International Conference on Learning Representations, ICLR 2014*.

He, C.; Wang, R.; and Chen, X. 2021. A Tale of Two CILs: The Connections Between Class Incremental Learning and Class Imbalanced Learning, and Beyond. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops, CVPR Workshops 2021*, 3559–3569. Computer Vision Foundation / IEEE.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016*, 770–778. IEEE Computer Society.

Hinton, G. E.; Vinyals, O.; and Dean, J. 2015. Distilling the Knowledge in a Neural Network. *CoRR*, abs/1503.02531.

Hou, S.; Pan, X.; Loy, C. C.; Wang, Z.; and Lin, D. 2019. Learning a Unified Classifier Incrementally via Rebalancing. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, 831–839. Computer Vision Foundation / IEEE.

Hung, S. C. Y.; Tu, C.; Wu, C.; Chen, C.; Chan, Y.; and Chen, C. 2019. Compacting, Picking and Growing for Unforgetting Continual Learning. In *Advances in Neural Information Processing Systems 32, NeurIPS 2019*, 13647–13657.

Kang, B.; Xie, S.; Rohrbach, M.; Yan, Z.; Gordo, A.; Feng, J.; and Kalantidis, Y. 2020. Decoupling Representation and Classifier for Long-Tailed Recognition. In *8th International Conference on Learning Representations, ICLR 2020*. OpenReview.net.

Kang, M.; Park, J.; and Han, B. 2022. Class-Incremental Learning by Knowledge Distillation with Adaptive Feature Consolidation. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, 16071–16080. Computer Vision Foundation / IEEE.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N. C.; Veness, J.; Desjardins, G.; Rusu, A. A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; Hassabis, D.; Clopath, C.; Kumaran, D.; and Hadsell, R. 2016. Overcoming catastrophic forgetting in neural networks. *CoRR*, abs/1612.00796.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical report, University of Toronto, Toronto, Ontario.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. ImageNet Classification with Deep Convolutional Neural Networks. In *Advances in Neural Information Processing Systems 25, 2012*, 1106–1114.

Lange, M. D.; Aljundi, R.; Masana, M.; Parisot, S.; Jia, X.; Leonardis, A.; Slabaugh, G. G.; and Tuytelaars, T. 2022. A Continual Learning Survey: Defying Forgetting in Classification Tasks. *IEEE Trans. Pattern Anal. Mach. Intell.*, 44(7): 3366–3385.

Li, K.; Wan, J.; and Yu, S. 2022. CKDF: Cascaded Knowledge Distillation Framework for Robust Incremental Learning. *IEEE Trans. Image Process.*, 31: 3825–3837.

Li, Z.; and Hoiem, D. 2016. Learning Without Forgetting. In *Computer Vision - ECCV 2016 - 14th European Conference*, volume 9908, 614–629. Springer.

Liu, Y.; Schiele, B.; and Sun, Q. 2021. RMM: Reinforced Memory Management for Class-Incremental Learning. In *Advances in Neural Information Processing Systems 34: Annual Conference on Neural Information Processing Systems 2021, NeurIPS 2021*, 3478–3490.

Mallya, A.; Davis, D.; and Lazebnik, S. 2018. Piggyback: Adapting a Single Network to Multiple Tasks by Learning to Mask Weights. In *Computer Vision - ECCV 2018 - 15th European Conference*, volume 11208, 72–88. Springer.

McCloskey, M.; and Cohen, N. J. 1989. Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In *Psychology of Learning and Motivation*, volume 24, 109–165. Academic Press.

Mi, F.; Kong, L.; Lin, T.; Yu, K.; and Faltings, B. 2020. Generalized Class Incremental Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR Workshops 2020*, 970–974. Computer Vision Foundation / IEEE.

Rebuffi, S.; Kolesnikov, A.; Sperl, G.; and Lampert, C. H. 2017. iCaRL: Incremental Classifier and Representation Learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, 5533–5542. IEEE Computer Society.

Shin, H.; Lee, J. K.; Kim, J.; and Kim, J. 2017. Continual Learning with Deep Generative Replay. In *Advances in Neural Information Processing Systems 30, 2017*, 2990–2999.

van de Ven, G. M.; and Tolias, A. S. 2019. Three scenarios for continual learning. *CoRR*, abs/1904.07734.

Vapnik, V. 1991. Principles of Risk Minimization for Learning Theory. In *Advances in Neural Information Processing Systems 4, 1991*, 831–838. Morgan Kaufmann.

Verma, V.; Lamb, A.; Beckham, C.; Najafi, A.; Mitliagkas, I.; Lopez-Paz, D.; and Bengio, Y. 2019. Manifold Mixup: Better Representations by Interpolating Hidden States. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019*, volume 97, 6438–6447. PMLR.

Wah, C.; Branson, S.; Welinder, P.; Perona, P.; and Belongie, S. 2011. Caltech-UCSD Birds-200-2011. Technical report, California Institute of Technology.

Wang, F.; Zhou, D.; Ye, H.; and Zhan, D. 2022. FOSTER: Feature Boosting and Compression for Class-Incremental Learning. In *Computer Vision - ECCV 2022 - 17th European Conference*, volume 13685, 398–414. Springer.

Wu, Y.; Chen, Y.; Wang, L.; Ye, Y.; Liu, Z.; Guo, Y.; and Fu, Y. 2019. Large Scale Incremental Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019*, 374–382. Computer Vision Foundation / IEEE.

Xu, Z.; Chai, Z.; and Yuan, C. 2021. Towards Calibrated Model for Long-Tailed Visual Recognition from Prior Perspective. In *Advances in Neural Information Processing Systems 34, NeurIPS 2021*, 7139–7152.

Yan, S.; Xie, J.; and He, X. 2021. DER: Dynamically Expandable Representation for Class Incremental Learning. In *IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021*, 3014–3023. Computer Vision Foundation / IEEE.

Yu, L.; Twardowski, B.; Liu, X.; Herranz, L.; Wang, K.; Cheng, Y.; Jui, S.; and van de Weijer, J. 2020. Semantic Drift Compensation for Class-Incremental Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, 6980–6989. Computer Vision Foundation / IEEE.

Yun, S.; Han, D.; Chun, S.; Oh, S. J.; Yoo, Y.; and Choe, J. 2019. CutMix: Regularization Strategy to Train Strong Classifiers With Localizable Features. In *2019 IEEE/CVF International Conference on Computer Vision, ICCV 2019*, 6022–6031. IEEE.

Zenke, F.; Poole, B.; and Ganguli, S. 2017. Continual Learning Through Synaptic Intelligence. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017*, volume 70, 3987–3995. PMLR.

Zhang, H.; Cissé, M.; Dauphin, Y. N.; and Lopez-Paz, D. 2018. mixup: Beyond Empirical Risk Minimization. In *6th International Conference on Learning Representations, ICLR 2018*. OpenReview.net.

Zhang, L.; Deng, Z.; Kawaguchi, K.; Ghorbani, A.; and Zou, J. 2021. How Does Mixup Help With Robustness and Generalization? In *9th International Conference on Learning Representations, ICLR 2021*. OpenReview.net.

Zhao, B.; Xiao, X.; Gan, G.; Zhang, B.; and Xia, S. 2020. Maintaining Discrimination and Fairness in Class Incremental Learning. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020*, 13205–13214. Computer Vision Foundation / IEEE.

Zhou, D.; Wang, F.; Ye, H.; Ma, L.; Pu, S.; and Zhan, D. 2022. Forward compatible few-shot class-incremental learning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022*, 9046–9056. Computer Vision Foundation / IEEE.

Zhou, D.; Ye, H.; and Zhan, D. 2021. Co-Transport for Class-Incremental Learning. In *MM '21: ACM Multimedia Conference, 2021*, 1645–1654. ACM.

Zhu, F.; Cheng, Z.; Zhang, X.; and Liu, C. 2021. Class-Incremental Learning via Dual Augmentation. In *Advances in Neural Information Processing Systems 34 , NeurIPS 2021*, 14306–14318.