Balancing between Forgetting and Acquisition in Incremental Subpopulation Learning

Mingfu Liang[1], Jiahuan Zhou[2,(✉)], Wei Wei[1], and Ying Wu[1]

[1] Northwestern University, Evanston, USA
{mingfuliang2020,weiwei2022}@u.northwestern.edu, yingwu@northwestern.edu
[2] Peking University, Beijing, China
jiahuanzhou@pku.edu.cn

Abstract. The subpopulation shifting challenge, known as some subpopulations of a category that are not seen during training, severely limits the classification performance of the state-of-the-art convolutional neural networks. Thus, to mitigate this practical issue, we explore incremental subpopulation learning (ISL) to adapt the original model via incrementally learning the unseen subpopulations without retaining the seen population data. However, striking a great balance between subpopulation learning and seen population forgetting is the main challenge in ISL but is not well studied by existing approaches. These incremental learners simply use a pre-defined and fixed hyperparameter to balance the learning objective and forgetting regularization, but their learning is usually biased towards either side in the long run. In this paper, we propose a novel two-stage learning scheme to explicitly disentangle the acquisition and forgetting for achieving a better balance between subpopulation learning and seen population forgetting: in the first "gain-acquisition" stage, we progressively learn a new classifier based on the margin-enforce loss, which enforces the hard samples and population to have a larger weight for classifier updating and avoid uniformly updating all the population; in the second "counter-forgetting" stage, we search for the proper combination of the new and old classifiers by optimizing a novel objective based on proxies of forgetting and acquisition. We benchmark the representative and state-of-the-art non-exemplar-based incremental learning methods on a large-scale subpopulation shifting dataset for the first time. Under almost all the challenging ISL protocols, we significantly outperform other methods by a large margin, demonstrating our superiority to alleviate the subpopulation shifting problem.[3]

1 Introduction

For the classification task in computer vision, a category is always consisted of many fine-grained sub-classes which can be called subpopulations. For example, the category "dog" has subpopulations including "Dalmatians", "Poodles" and "Terriers", etc (as shown in Fig. 1). However, such kinds of large-scale subpopulations severely limit the discriminative ability of learned models. Recently,

[3] Code is released in https://github.com/wuyujack/ISL

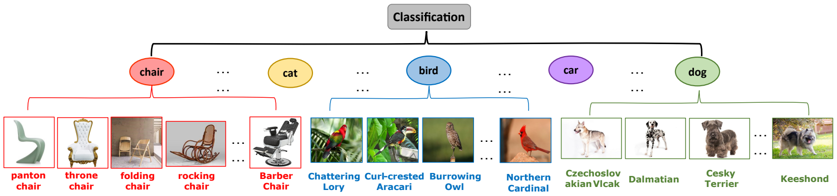

Fig. 1. Subpopulations [28] are widely existed in the real world. A visual category (colored ellipse) contains a large number of subpopulations (denoted by each image) which are semantically similar and share common visual characteristics [28] to be in the same category, while they also have large differences in appearances, shape, context, etc. Each subpopulation [28] is also a distribution with sufficient variations, e.g., cover thousands of distinct objects belonging to this subpopulation in nature.

Santurkar et al. [28] studied how well a model generalizes to subpopulations that are unseen during training, i.e., whether the model can recognize "Dalmatians" as "dogs" even their training data for "dogs" comprise only the dog's breeds like "Poodles" and "Terriers". Their observations demonstrate that the classification accuracy on those unseen subpopulations drops significantly (mostly more than 30%) compared to the seen population. Such a critical issue is defined as *subpopulation shifting*, caused by the large intra-class variations within a category, or more specifically, the large inter-subclass variations between different subpopulations of a common category in the natural world.

To tackle such a subpopulation shifting problem, a naive solution is to comprehensively collect sufficient data from all subpopulations for learning. However, due to the visual complexity of a category in nature, it is hard to completely cover all the subpopulations during data collection. Therefore, in recent years, more efforts have been paid to leverage the *incremental learning* (IL) technique to improve the generalization ability of an offline learned classification model against the online unseen data. While existing incremental learning methods mostly focus on the data from unseen categories but simply ignore the subpopulation shifting problem within a seen category from the training phase.

Recently, a few works spotlight a scenario where the distributions of seen categories are shifting while the label space is fixed, called incremental domain learning (IDL) [32,12], which mostly targets on two specific cases. Firstly, changing visual domains (e.g., from photo-style to painting-style) of the seen category, known as continual domain adaptation (CDA) [33]; Secondly, adding new poses and environment conditions (e.g., illumination, background) to the seen category, denoted as the new instance (NI) setting [18,20]. However, none of them recognize the critical subpopulation shifting problem caused by the large inter-subclass variations within every specific visual category. Therefore, it is worthwhile to provide a first and comprehensive study of tackling the subpopulation shifting problem in an incremental learning manner, e.g., incrementally learning to recognize the unseen subpopulations of "dog" as "dog", without retaining the data of seen population. We call this *incremental subpopulation learning* (ISL)

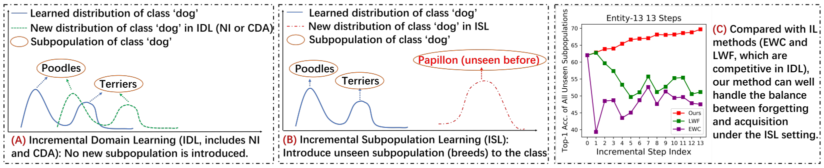

Fig. 2. (A) and (B) show the difference between the ISL and incremental domain learning (IDL): in IDL (includes NI and CDA), the new distribution is only the manipulation of the existing subpopulations' distributions (e.g., the same subpopulation in different visual domains), but no new unseen subpopulations are introduced; Instead in ISL, the new distribution is the totally new and unseen subpopulation [28] that is not existed in the distribution of a category before. Concrete examples are in our supplementary. (C) shows our method can gradually acquire the unseen subpopulations during ISL.

and more discussions about the differences with the aforementioned incremental learning settings can be found in Fig. 2 and Section 2.

However, as commonly observed in the incremental learning research, a model may be quickly adapted to acquire the unseen task while forgetting the seen tasks gradually, especially without retaining the previous learning data [6,42]. Such a phenomenon significantly limits the final discriminative ability of the model. Thus, *balancing the forgetting and acquisition appropriately in incremental learning* still remains a challenging problem and may be more critical in ISL. The reason is that the unseen subpopulations share common visual characteristics to be grouped in the same category [28]. Such a correlation makes the model being easily transferred to unseen subpopulations in finetuning [28] while forget the seen subpopulations. Currently, general IL solutions design various forgetting regularizations to jointly optimize the acquisition and forgetting [15,14,36,11,42,8,17]. However, they heavily rely on a controller hyperparameter predetermined before incremental learning starts and fixed afterward [6]. Since the relation between forgetting and acquisition is not explicitly modeled, the hyperparameter needs to be subtly tuned based on a held-out test set from each incremental learning phase. This not only introduces large amounts of manual trials and errors, but also has no guarantee to obtain a great balance in the long run, especially when we can not access previous test sets [6].

Therefore, in this paper, we propose a novel two-stage learning scheme to tackle the above forgetting issue from an adversarial perspective, also as a preliminary baseline for ISL. In the first "gain-acquisition" stage, we progressively learn a new classifier using only the learning data from unseen subpopulations without explicitly regularizing the forgetting issue. To do so, we explore the possibility of the feature extractor sharing during incremental learning and achieve a better stability-and-plasticity trade-off [6] by progressively reducing prediction error on the hard samples and classes. To explicitly defy the forgetting issue, we propose a second "counter-forgetting" stage to further achieve a better balance between forgetting and acquisition by encouraging them to compete against each other and linearly combining the old and new classifiers based on an additive parameter α. To achieve this, we leverage a novel objective function to model the confrontation by two proxy estimations of forgetting and acquisition respectively, then search for the proper α by optimizing this objective function.

Our proposed method disentangles the acquisition and forgetting in our two-stage learning scheme to explicitly guarantee the acquisition of knowledge from unseen subpopulations as well as mitigate the forgetting issue of seen subpopulations. To verify this, for the first time, we elaborately design extensive experimental protocols to investigate ISL on the large-scale datasets, i.e., BREEDS [28], which are recently proposed to precisely simulate the subpopulation shifting condition. Extensive empirical results demonstrate that our proposed method outperforms the existing incremental learning approaches by a significant margin under almost all the protocols. Moreover, our further discussions and analyses also show the effectiveness of leveraging incremental learning to alleviate the subpopulation shifting problem. To sum up, our contributions are three-fold: (1) We conduct a first extensive experimental study of representative incremental learning methods on incremental subpopulation learning (ISL) based on a recently proposed large-scale benchmark tailored to subpopulation shifting; (2) We propose a novel two-stage non-exemplar-based (NEB) ISL method to explicitly disentangle the acquisition and forgetting in ISL for achieving a better balance, which outperforms the representative NEB methods by a large margin under different and challenging ISL protocols; (3) We empirically show that incremental learning is promising for alleviating the challenging subpopulation shifting problem, which is worthwhile for future study. Our empirical analyses further enlighten the challenges and future research direction for ISL.

2 Related Works

To highlight the necessity of our proposed ISL, it is essential to compare several related learning scenarios, including our Incremental Subpopulation Learning (ISL), Incremental Domain Learning (IDL) [32,12] that includes New Instance (NI) [18,20] and Continual Domain Adaptation (CDA) [33] settings, Class-Incremental Learning (CIL) [21,6] and Incremental Implicitly-Refined Classification (IIRC) [1]. Please also refer to our supplementary for more discussions.

ISL v.s. IDL (includes NI and CDA): As mentioned in Sec. 1, the input distribution or domain in IDL [32,12] is shifting while the label space is fixed. However, IDL does not propose to introduce any new unseen subpopulations to a category (see Fig. 2). The general IDL methods [32,12] can hardly model the data variation caused by subpopulation shifting and can not balance the forgetting and acquisition without retaining old data. More specifically, NI [18] adds new patterns to the same object by changing the object's poses and image conditions (e.g., illumination). CDA [33] means continually adapting a model to new visual domains (e.g., from photo style to other styles' images). Thus, both NI and CDA can be considered as specific cases of IDL. Although our ISL also does not change the label space and can be generally viewed as a specific case of IDL, ISL has its specific research targets and challenges that are different from the above-mentioned settings. Given that NI and CDA are explicitly framed to specify their identity, it is also of great necessity to frame ISL explicitly.

From the data perspective, datasets [18,33] used in NI or CDA do not have a large scale and precise label hierarchy to define the subpopulations within a category, thus they can not precisely simulate the subpopulation shifting [28]. CORe50 [18] used in NI does have a hierarchy (10 categories, each one has 5 classes), but each class is only a distinct object. The objects are hand held in different views and the environmental conditions are changed to get their new instances. Such a dataset is insatiable to create the subpopulation given its limited diversity and scale (see Fig. 1), and can not simulate the subpopulation shift as the new instances are still belonging to existing seen objects (see Fig. 2 (A)). Recently, [28] proposed a large-scale BREEDS dataset to create the desired hierarchy with large amounts of human efforts and identify for the first time the subpopulation shifting problem to the community. Thus, existing studies in IDL (NI or CDA) are not suitable for ISL. To the best of our knowledge, ISL has rarely been mentioned or studied in the IL literature. The BREEDS benchmark paves the way for our timely study of the subpopulation shifting problem.

ISL v.s. General IL (includes CIL [6] and IIRC [1]): ISL also differs from CIL. In CIL, we continually learn new classes that are disjoint with previous ones. Thus we have a clear boundary between new and old classes and can fix the old classifiers to avoid detrimental updates [6]. In contrast, we only have a fixed size, unified classifier in ISL, and it is unavoidable to update the whole decision boundary. Recently, IIRC [1] is proposed to incrementally learn new classes and also refine the label hierarchy between the seen subclasses and their specific class: A model first learns several classes (e.g., “cat”), where the training data for each class comprises several subclasses. Then the model encounters both new class samples (e.g., “cow”) and the seen sample with its subclass label. The model needs to learn the different granularity of labels of a class and the relation between them. Differently, in ISL, we do not introduce new classes, and the new subpopulation needs to be strictly *unseen and disjoint* to the old ones.

Incremental Learning Methods. Given whether the training images can be retained, existing IL methods are divided into exemplar-based (EB) [5,24,13,11] [36,31,3,16] and non-exemplar-based (NEB) methods. However, storing old training data is not privacy-preserved in the real world [6]. NEB methods mostly aim to design better forgetting regularization constraints on parameters [14,40,19,2,39] and model outputs [8,41,35,15]. However, the former needs a well-defined metric to identify the important parameter, which is hard to design [6]; the latter's performance depends largely on the old and new task correlation [6]. Other kinds of NEB methods learn a generative model (GAN) [30,34,4,34] to generate the old images for retraining or dynamically extend the models [23,22,38,5,37]. However, the former requires the GAN to be capable of IL and generate high-quality images, which is still challenging; the latter requires growing memory and is undesirable in the real world. Moreover, all the above NEB methods mostly couple the forgetting and acquisition into a joint optimization problem, where their balance is controlled based on finely-tuned hyperparameters.

Differently, our proposed method disentangles the acquisition and forgetting to explicitly and adaptively control them in a data-driven manner in ISL, which significantly outperforms the representative NEB methods mentioned above.

3 Method

3.1 Terminology and Problem Formulation.

Following the terminology in [28], the term "population" is concretely defined as class or super-class, e.g., "cat" and "dog", and the "subpopulation" is defined as the subclass of a specific class, e.g., different "dog" breeds. All subclasses are under the same visual domain (i.e., natural image). Before incremental learning, we have a *base step* to train a model to learn many diverse classes with sufficient data, where the model is called the *base step's model*. Each class is learned by a dataset comprised of different subclasses, e.g., subclasses of "dog" like "Poodles" and "Terriers". These subclasses are labeled as class "dog". Then in each *incremental step*, the model encounters unseen subclasses of existing classes, and it incrementally learns to predict the class label of these unseen subclasses.

Formally, let $t = 0$ denote the *base step*, and let $t = 1, 2, .., T$ denote the *incremental steps*. The training dataset of the t-th incremental step is $D_t^{train} = \left\{X_t^{train}, S_t^{train}, Y\right\} = \left\{x_{t,j}^{train}, s_{t,j}^{train}, y_j\right\}_{j=1}^{N_t}$, where x, s, y denote the inputs, subclass labels and class labels, respectively. Note that the only supervision is the class label, while the subclass labels will not be used during training but to ensure unseen subclasses differ from all seen subclasses, i.e., $S_t^{train} \cap (\cup_{i=0}^{t-1} S_i^{train}) = \varnothing$. The set of class labels is the same over all steps. At each step we have a corresponding held-out test set $D_t^{test} = \{X_t^{test}, S_t^{test}, Y\}$ to evaluate the performance on the current step, and we also only use the class label Y for evaluation. The model, e.g., CNN, comprises the feature extractor f_θ and classifier G_ϕ, parameterized by θ and ϕ respectively, where G_ϕ refers to the last linear layer of the CNN. After T steps, the model is tested on all the previous steps' held out test sets $D_t^{test}, t = 0, ..., T$ to evaluate the performance over all the learned subclasses.

3.2 A Novel Two-stage Learning Scheme

Here we introduce the proposed two-stage learning method. We argue that the learned CNN feature extractor is capable of extracting discriminative features for each class. Then the potential reason of misclassifying the unseen subclasses is that the final classifier emphasizes the feature that is less discriminative for the unseen subclasses because the classifier may have already biased to the seen subclasses. Therefore, we conjecture that the subpopulation shifting may be alleviated by appropriately updating the classifier to emphasize the proper feature for the unseen subpopulation. To explore this idea, we consider to share a fixed feature extractor after the *base step* and only learn the new classifier as a novel baseline tailored to ISL. Since feature extractor sharing may lead to concerns of stability and plasticity trade-off, thus in our Stage-1 we introduce

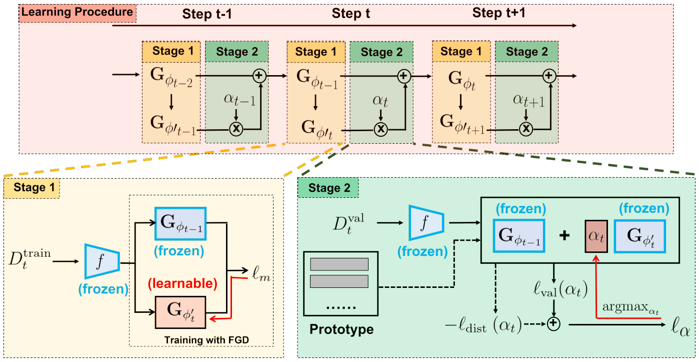

Fig. 3. During the incremental subpopulation learning procedure, in each incremental step t we obtain the classifier G_{ϕ_t} for the model F_t by two stages. In Stage-1, we learn a new classifier $\mathrm{G}_{\phi'_t}$ via functional gradient descent (FGD) of Eqn. 3. In Stage-2, we obtain G_{ϕ_t} by combining the new and old classifiers linearly via a proper α_t solved by Eqn. 10 to balance the acquisition and forgetting approximately.

the gradient-based boosting [26] idea to alleviate the issue by progressively reducing the prediction error. In Stage-2 we design a specific objective function to approximately model the balance of acquisition and forgetting and leverage it to achieve our target. Fig. 3 provides a systematic view of our design.

Stage-1: Gain-Acquisition. Suppose $Y = \{1, \ldots, C\}$ is the class label for each incremental step in ISL, and $y^k \in \mathbb{R}^C$ is the one-hot vector to represent the class k. We define the margin [27,26] of a sample x to an arbitrary class k as:

$$\mathcal{M}\left(y^k, F(x)\right) = \min_{l \neq k} \frac{1}{2} < y^k - y^l, F(x) > \tag{1}$$

where $F(x) = G_\phi\left(f_\theta(x)\right)$ denotes the model prediction given a sample x and $F(x) \in \mathbb{R}^C$; $\langle ., . \rangle$ denotes the dot-product and $\frac{1}{2}\left\langle y^k - y^l, F(x) \right\rangle$ is the l-th margin component of class k, where $y^k F(x)$ is the k-th element of the model's prediction vector. Now we define the margin of the model given the training data $D = \{X, Y\}$ (we omit t and the subclass label S_t as it is not used as supervision):

$$M(D, F) = \min_{(x_i, y^{c_i}) \in D} \mathcal{M}\left(y^{c_i}, F(x_i)\right), \tag{2}$$

where y^{c_i} is the one-hot vector of the ground-truth class label c_i given a sample x_i. $M(D, F)$ measures the distance between the closest sample to each ground-truth class's decision boundary given the model and training data, and we want to encourage the model to have a large $M(D, F)$ such that given a sample, the model prediction of its ground-truth class can be far away from other classes [27,26]. Hence we define our margin-enforce objective function [27] as:

$$\ell_m(F) = \frac{1}{|D|} \cdot \sum_{(x_i, y^{c_i}) \in D} L_M\left[y^{c_i}, F(x_i)\right], \tag{3}$$

where $|D|$ denotes the size of dataset and the $L_M[.,.]$ should be a differentiable and monotonically decreasing function such that by minimizing Eqn. 3, it is equivalent to maximize the margin defined by Eqn. 2. This margin-enforce property has a guarantee that the optimal model obtained by minimizing Eqn. 3 may have good generalization, as demonstrated in the boosting theory [26]. We consider the default choice in the multi-class boosting theory [27,26] as:

$$L_M[y^c, F(x)] = \sum_{k=1,k\neq c}^{C} e^{-\frac{1}{2}\left[<y^c,F(x)>-<y^k,F(x)>\right]}. \tag{4}$$

Then a new model is obtained by the largest decrease of Eqn. 3. Such a decrease can be determined by the directional derivative [9] of Eqn. 3 along the functional $g : \mathcal{X} \to \mathbb{R}^C$, where $\mathcal{X}$ denotes the input space of the model $F(x)$:

$$\delta\ell_m[F; g] = \left.\frac{\partial\ell_m[F(x) + \epsilon g]}{\partial\varepsilon}\right|_{\varepsilon=0}, \tag{5}$$

called the functional gradient (FG) [9]. By first order Taylor expansion we have:

$$-\delta\ell_m[F; g] = \frac{1}{2|D|}\sum_{i=1}^{|D|} w_i < g\left(x_i\right), y^{c_i} - \sum_{k\neq c_i} y^k \tau_k\left(x_i, c_i\right) >, \tag{6}$$

where $w_i = \sum_{k\neq c_i} e^{-\frac{1}{2}<y^{c_i}-y^k,F(x_i)>}$ and $\tau_k\left(x_i, c_i\right) = \frac{e^{-\frac{1}{2}<y^{c_i}-y^k,F(x_i)}}{\sum_{k\neq c_i} e^{-\frac{1}{2}<y^{c_i}-y^k,F(x_i)>}}$. Detailed derivation of Eqn. 6 can be found in [27,26]. Finally, we achieve a new model by maximizing the negative functional gradient, i.e., Eqn. 6:

$$g(x) = \arg\max_g \sum_{i=1}^{|D|} w_i < g\left(x_i\right), y^{c_i} - \sum_{k\neq c_i} y^k \tau_k\left(x_i, c_i\right) > . \tag{7}$$

Such an update is known as functional gradient descent (FGD) [26] on Eqn. 3. In our work, we integrate the above learning mechanism into incremental subpopulation learning (ISL): Assume we have a model F_{t-1} after $t-1$ incremental steps. For the t-th step, we only use the current step's training data to learn a new model F_t by optimizing Eqn. 3 via FGD, which is exactly to solve the Eqn. 7 and then $F_t = g$. Since we want to explore the possibility of only updating the classifier of CNN, thus for the model F_t, only the classifier is learnable and the feature extractor f_θ is frozen after the *base step* and shared over each *incremental step*, illustrated in Fig. 3. Thus by FGD, we actually obtain a new classifier $\mathrm{G}_{\phi'_t}$ for F_t. To solve the Eqn. 7, we initialize g as F_{t-1} and minimize the negative of Eqn. 7 by stochastic gradient descent (SGD). We empirically observe that this learning mechanism works smoothly and converges to high accuracy (>90%) on the unseen subpopulation training data after several epochs, which essentially relieves the stability-plasticity concern. This is due to the merit of margin-enforce loss [29,26] that can progressively reduce the training error.

Formally, it relates to the reweighting mechanism [26] presented in Eqn. 7 and we show that it is also beneficial to ISL: (1) The w_i is a decreasing function of the margin components of the ground-truth class c_i given the sample x_i. Then the w_i is close to 0 when the smallest margin component of class c_i is large and positive, where from Eqn. 1 this means the sample x_i is with a large margin and is easy to be predicted since the model prediction of the class c_i, i.e., $y^{c_i}F(x_i)$, is much larger than other classes. Thus, the classifier may receive small updates from easy samples as w_i are small, and large updates from hard samples (i.e., with small margins). *This reweighting mechanism may avoid uniformly updating the decision boundary given every training sample. Meanwhile, it lets the model to focus on hard samples to reduce error progressively. This is crucial for ISL since it avoids unnecessary updates and lead to less forgetting.* (2) The other reweighting can be observed from $< g\left(x_i\right), y^{c_i} - \sum_{k \neq c_i} y^k \tau_k\left(x_i, c_i\right) >$. We change it to $\sum_{k \neq c} \tau_k(x_i, c_i) \left[< g(x_i), y^{c_i} - y^k >\right]$ by taking the $\tau_k\left(x_i, c_i\right)$ outside. $\tau_k\left(x_i, c_i\right)$ is a weighted average over the margin components of the class c_i, and it will weigh a class k (hard class) more for updating the classifier if the model prediction of it, i.e., $y^k F(x_i)$, is close to the prediction of the class c_i. Hence this reweighting may also avoid uniformly updating the class decision boundaries.

Stage-2: Counter-Forgetting. Although we connect general boosting with incremental learning and show that it is favorable for ISL due to its reweighting mechanisms, the above learning mechanism can not entirely defy the forgetting. The new classifier $G_{\phi'_t}$ is only trained with the current step's unseen subpopulation data while we do not explicitly impose any forgetting control. Therefore, we propose to obtain the final classifier G_{ϕ_t} for model F_t by linear addition:

$$G_{\phi_t} = G_{\phi_{t-1}} + \alpha_t \cdot G_{\phi'_t}. \tag{8}$$

This is also inspired from boosting mechanism, but it is totally different from boosting since the α_t here is for controlling the learning and forgetting, while in boosting we use it to progressively reduce the training error [29]. As the classifier of CNN is a linear layer and in ISL, we do not introduce new classes and the size of this layer is fixed. Thus the linear combination of two linear classifiers is equal to linear combine the weight of them, i.e., $\phi_t = \phi_{t-1} + \alpha_t \cdot \phi'_t$. The key challenge is to determine the proper α_t without storing previous training images since it is infeasible to measure the forgetting only by the current step's training data. To tackle this challenge, we consider obtaining the proxy of forgetting by measuring the relative distance distortion of the class representative prototype between the last step's classifier $G_{\phi_{t-1}}$ and new classifier $G_{\phi'_t}$ under different α_t. The insight is: Since the feature extractor is shared, it can provide consistent transformation for each class's data in the feature space. Thus the class prototype is fixed and consistent for each class during incremental learning. The forgetting can now be disentangled and measured by the distance distortion between the prototype and the changed decision boundary, since the class prediction error is directly related to the change of decision boundary. This differs from the existing non-exemplar-based method, e.g., PASS [42], leveraging the prototype (class mean feature) to create a constraint to train on the new class data to maintain the old decision boundary which is changing dynamically in CIL. After each incremental step, if a class is introduced with unseen subclasses, we obtain one prototype (class mean feature) of that class and add it to the prototype bank, which is the same as the PASS [42] storing one prototype of each new class in CIL. The prototype is calculated by the feature extractor f_θ and thus it has the same input dimension as the classifier. At step t, we denote the relative distance distortion as:

$$l_{\text{dist}}\ (\alpha_t) = \sum_{i=0}^{t-1} \sum_{j \in N_p^i} \frac{1}{C} \cdot \left| \frac{\alpha_t \cdot G_{\phi_t'}\left(K_{i,j}\right)}{G_{\phi_{t-1}}\left(K_{i,j}\right)} \right|_1 , \tag{9}$$

where $G_\phi\left(K_{i,j}\right) \in \mathbb{R}^C$ and $|\cdot|_1$ denotes the L1 norm, $K_{i,j}$ denotes the prototype of class j in step i and N_p^i denotes the set of index of the stored prototype in step i. We do element-wise division here to measure the relative distortion of each class between old and new classifiers under different α_t. The distance is normalized by the size of the label space C. For the acquisition measurement, before training on current step's training data D_t^{train}, we randomly sample a held-out validation set D_t^{val} from D_t^{train} to measure the relative improvement of validation accuracy between $G_{\phi_{t-1}} + \alpha_t \cdot G_{\phi_t'}$ and $G_{\phi_{t-1}}$, denoted as $l_{val}(\alpha_t)$. The proper α_t is obtained by optimizing an objective function modeling the balance:

$$\alpha_t = \arg\max_{\alpha_t} l_\alpha = \arg\max_{\alpha_t} l_{val}\left(\alpha_t\right) - l_{dist}\left(\alpha_t\right), \tag{10}$$

which means we want more acquisition while also less forgetting. Note that in Stage-2 we do not update any classifier by backpropagation. Instead we fix both the old and new classifiers, $G_{\phi_{t-1}}$ and $G_{\phi_t'}$, and search for the proper α_t by solving Eqn. 10, shown in Fig. 3. The α_t can be readily searched by simple line search. More details and discussions are included in our supplementary.

4 Experiments

Datasets. We leverage the latest BREEDS datasets [28] in our experiments. BREEDS simulates the real-world subpopulation shifting based on the ImageNet [7], and it comprises four different datasets: Entity-13, Entity-30, Living-17, and Non-Living-26, with a total of 0.86 million (M) of images. The dataset configurations and statistics are all included in supplementary. However, BREEDS is not proposed for incremental subpopulation learning (ISL), so we need to further create the ISL-specific benchmark based on it. Since we focus on the incremental learner's performance in the sufficiently long run, hence in present work, our main testbeds are based on Entity-13 and Entity-30 from BREEDS as they have the most number of subclasses, i.e., totally 260 and 240 subclasses respectively, and more than 0.6M images. To the best of our knowledge, this is the first time to leverage such large-scale datasets to investigate the ISL.

Comparison Methods. A strict requirement of ISL is that no previous training images can be retained, thus it is rarely studied before. Moreover, very few methods are proposed tailored to the related Incremental Domain Learning (IDL); Methods like BOCL [31] for IDL needs to store and use old images during training on new instances, which can not satisfy the ISL requirements. Thus following the works benchmarked the IDL [12,32,31] and other settings, we choose several general and representative non-exemplar-based (NEB) methods [42,21] and benchmark them under BREEDS [28] for the first time. They include EWC [14], LwF [15], LwF-MC [25], MUC [17], LwM [8] and PASS [42]. Among them, EWC and LwF are widely used to benchmark various IL settings including IDL [12,32,31,33] and achieve comparable results to the state-of-the-art (SOTA). Methods like MUC and PASS were tested in CIL, but as stated in their papers [17,42], they are also *general* for different IL settings including ISL. PASS is the SOTA NEB method in CIL. We also compare the naive baselines, i.e., finetune the whole model ("Finetune All") and finetune only the last layer ("Finetune Last"). The joint training of all the data is the "Oracle".

Evaluation Metrics. We report the common metrics [42] in IL literature for evaluation, i.e., the average top-1 accuracy (%) on all the seen and unseen subclasses we learned for each class (includes the *base step*), denoted as "*All*", and the average forgetting $\mathbb{F}_i$ to measure the forgetting in previous steps. At step i, the forgetting score on step j is $\mathbf{f}_i^j = \max_{t\in 0,\dots,i-1}\left(a_{t,j} - a_{i,j}\right), \forall j < i$, where $a_{i,j}$ denotes the accuracy of step j after the training of step i. Then $\mathbb{F}_i$ is defined as $\mathbb{F}_i = \frac{1}{i}\sum_{j=0}^{i-1}\mathbf{f}_i^j$. We further define "*Unseen*" as the average test accuracy only on all the unseen subclasses and report it in Tab. 1 and 2 to show how well each method acquires the unseen subpopulation after incremental learning.

Experimental Design. Entity-30 and Entity-13 have 30 and 13 classes where each class has 8 and 20 subclasses respectively. We design 3 protocols for each dataset. In the *base step*, the training set of each class comprises data from 4 and 10 subclasses for Entity-30 and Entity-13 respectively, the same as BREEDS to simulate subpopulation shifting. Then we split the rest of 120 and 130 unseen subclasses in each dataset respectively to create different protocols. For Entity-30, we design protocols with 4, 8, 15 incremental steps: in each step, for 4 Steps setup, each class is introduced with 1 unseen subclass; for 8 and 15 Steps setups, we randomly choose 15 and 8 out of 30 classes respectively to introduce with 1 unseen subclass. For Entity-13, we design protocols with 5, 10, 13 incremental steps: in each step, for 5 and 10 Steps setups, we introduce 2 and 1 unseen subclasses for each class respectively; For 13 Steps setup, we randomly sample 10 out of 13 classes to introduce with 1 unseen subclass. These designs simulate two scenarios: (1) all the classes are updated with at least 1 unseen subclass; (2) only a part of classes are updated with unseen subclasses. We denote the former as **even update** and the latter as **uneven update**.

Implementation Details. We use ResNet-18 [10] for all methods as [42]. For a fair comparison, all methods are initialized with the same *base step model* and then start incremental learning. As the first benchmark for ISL, it is essential to compare different methods fairly. Therefore, we use the Continual Hyperparameter Framework (CHF) proposed by [6] to find the hyperparameters for comparison methods, and also use the same data augmentation as in BREEDS [28]

Table 1. Results on Entity-30 benchmark. Smaller $\mathbb{F}_i$ and larger *Unseen*/*All* is better. Before incremental learning, "*Unseen*" is 50.18 for all the methods.

	4 Steps (Even Update)			**8 Steps (Uneven Update)**			**15 Steps (Uneven Update)**		
Method	*Unseen*	*All*	$\mathbb{F}_4$	*Unseen*	*All*	$\mathbb{F}_8$	*Unseen*	*All*	$\mathbb{F}_{15}$
Oracle	88.03	87.63	-	88.03	87.63	-	88.03	87.63	-
Finetune All	53.72	48.08	47.75	26.45	23.08	73.86	14.68	13.77	84.49
Finetune Last	55.25	58.30	32.43	30.85	32.50	60.82	19.98	21.56	72.40
EWC [14]	56.17	54.10	40.69	30.50	29.00	66.94	22.20	23.68	74.03
LwF [15]	62.67	58.85	32.32	34.52	29.69	64.38	32.62	31.17	62.51
LwF-MC [25]	**68.28**	64.43	28.20	46.93	43.69	50.88	34.53	33.79	62.36
MUC [17]	62.98	59.59	29.45	36.17	31.83	61.49	34.15	32.54	60.65
LwM [8]	63.32	59.20	33.13	42.47	38.90	55.59	33.43	30.78	61.23
PASS [42]	64.50	69.37	21.79	48.85	54.99	40.50	32.13	39.75	58.27
Ours	64.73	**72.88**	**4.16**	**58.63**	**72.14**	**2.30**	**56.87**	**71.69**	**3.48**

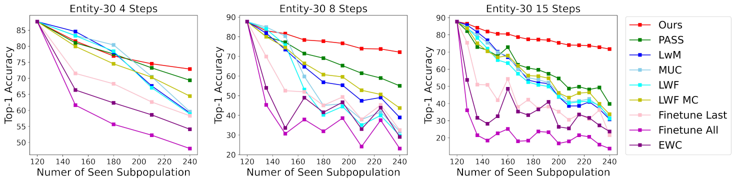

Fig. 4. Average top-1 test accuracy in each step under 3 protocols of Entity-30.

to train both the *base* and *incremental steps* consistently for all methods. All experimental details are in supplementary. The data augmentation comprises random resize crop, random horizontal flip, lighting, color jitter, etc. Note that such a heavy strategy is the same as the domain randomization (DR) method used in continual domain adaptation (CDA) [33] to achieve SOTA results.

4.1 Comparison with the state-of-the-art

From Tab. 1 and 2 we observe: when the incremental step is small and the update is even (the 4 Steps Entity-30 and 5 Steps Entity-13), all the NEB methods can improve their accuracy on the unseen subclasses ("*Unseen*") compared to themselves before incremental learning, reported in the captions of Tab. 1 and 2. However, when we compare the performance on all the subclasses ("*All*"), our method exceeds all compared methods with a large margin since those methods forget the learned subpopulations during ISL and thus lead to poor "*All*" performance when average on all the subclasses (as shown in Fig. 4 and 5). This demonstrates that most NEB methods can learn to recognize the unseen subpopulations in small steps but at the cost of forgetting the seen ones. When the incremental steps become large and **uneven update**, e.g., 15 and 13 Steps for Entity-30 and Entity-13, all the compared methods suffer from severe forgetting on the subpopulations learned in previous steps and perform significantly poorly. The baseline "Finetune Last" almost fails in 15 Steps Entity-30, though it may obtain comparable results to some NEB methods in small steps and even update. This shows it is hard to only fix the feature extractor to achieve excellent results in ISL in the long run. In contrast, our proposed method achieves small average

Table 2. Results on Entity-13 benchmark. Smaller $\mathbb{F}_i$ and larger *Unseen*/*All* is better. Before incremental learning, "*Unseen*" is 62.03 for all the methods.

	5 Steps (Even Update)			**10 Steps (Even Update)**			**13 Steps (Uneven Update)**		
Method	*Unseen*	*All*	$\mathbb{F}_5$	*Unseen*	*All*	$\mathbb{F}_{10}$	*Unseen*	*All*	$\mathbb{F}_{13}$
Oracle	90.61	90.46	-	90.61	90.46	-	90.61	90.46	-
Finetune All	61.54	59.16	37.79	51.55	50.88	46.76	41.98	41.72	56.97
Finetune Last	65.52	71.15	18.89	61.52	67.37	25.47	49.89	55.23	40.31
EWC [14]	63.85	63.48	32.99	55.63	57.31	36.53	47.51	48.54	50.49
LwF [15]	66.91	64.82	31.47	59.97	59.17	36.26	51.14	51.05	46.31
LwF-MC [25]	67.57	65.96	30.64	59.58	59.22	38.42	59.45	59.70	37.02
MUC [17]	67.51	65.88	30.00	62.17	61.98	31.45	53.58	52.89	43.74
LwM [8]	69.69	67.61	28.22	63.49	62.25	31.72	51.05	50.80	46.31
PASS [42]	**73.12**	75.44	16.73	65.63	68.51	26.55	50.48	52.49	43.76
Ours	72.02	**78.92**	**3.29**	**68.31**	**77.53**	**3.35**	**69.69**	**78.75**	**3.35**

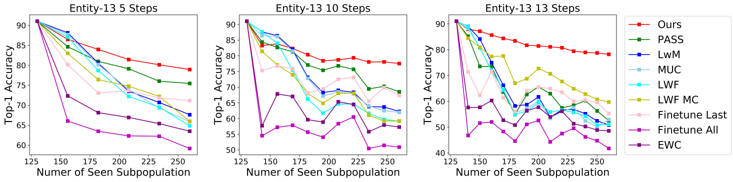

Fig. 5. Average top-1 test accuracy in each step under 3 protocols of Entity-13.

forgetting and great average accuracy even after 13 and 15 steps, outperforming the best existing method by 19.05% and 31.94% on "*All*" respectively. Interestingly, we also observe that our method can have smaller forgetting in longer steps (8 and 15 Steps Entity-30). This is due to the positive transfer in our method shown in Tab. 4, where the test accuracy of some steps can be improved after ISL and have no forgetting. The reason is our method can gradually learn unseen subpopulations and strike a much better balance between acquisition and forgetting than existing methods. The acquired knowledge from new unseen subpopulations sometimes could be helpful for better distinguishing the old seen subpopulations, which is essential for countering forgetting in ISL.

Further Discussions and Analyses. We further analyze the performance of existing methods in both ISL and other IL settings based on our empirical observation. We highlight our analyses below, and more details are in our supplementary: (1) *ISL provides new challenges for the representative NEB methods.* For instance, EWC and LWF can achieve comparable, or even SOTA results in the benchmark of IDL [32,12], NI [31] and CDA [33], and the data augmentation strategy is also the same as the DR method proposed in CDA [33] to achieve SOTA results. However, they still suffer largely from forgetting in ISL, especially under **uneven update**. This is caused by the differences between ISL and other settings since the unseen subpopulation may not be simulated by strong augmentation as verified in [28] or only changing the seen subpopulation's views or environments. Thus it is hard to acquire the unseen subpopulations without forgetting the seen ones. (2) All compared methods control the forgetting by one or several hyperparameters. Although some of them may perform well in early steps in Fig. 4 and 5, such a mechanism can not strike a balance in the long run.

Table 3. Ablation study by Entity-13 13 Steps setup

	Unseen	*All*	$\mathbb{F}_{13}$
Cross Entropy	49.89	55.23	40.31
ℓ_m	53.13	60.38	29.48
ℓ_m+ Random α_t	62.60	71.59	5.93
ℓ_m+ Fixed $\alpha_t = 1$	63.83	75.24	4.23
ℓ_m+ Obtained α_t (Ours)	**69.69**	**78.75**	**3.35**

Table 4. Positive Transfer. $\mathbf{f}_{15}^{8}$ means the forgetting score of 8-th step of overall 15 Step.

	Positive Transfer
15 Step Entity-30	$\mathbf{f}_{15}^{8}$=-0.5, $\mathbf{f}_{15}^{11}$=-2.5
8 Step Entity-30	$\mathbf{f}_{8}^{2}$=-0.8, $\mathbf{f}_{8}^{4}$=-3.73, $\mathbf{f}_{8}^{6}$=-4.4
10 Step Entity-13	$\mathbf{f}_{10}^{6}$=-1.5

4.2 Ablation Study and Analysis

To further explore the proposed method, we investigate the contribution of each model component. As our method is two-stage and also optimizes a new learning objective instead of cross entropy loss, thus we compared our method with: (1) "Cross Entropy": directly finetune the last layer by the cross entropy loss known as the "Finetune Last" in Tab. 1 and 2; (2) "ℓ_m": only finetune the last layer by our margin loss from Eqn. 3 (only Stage-1); (3) "ℓ_m + Random α_t": use both Stage-1 and 2 but update the model by a random α_t without using Eqn. 10; (4) "ℓ_m + Fixed $\alpha_t = 1$": the same as (3) except the α_t is fixed as 1 to equally weight the influence of acquisition and forgetting. We observe from Tab. 3: (1) The margin loss performs better than the cross entropy, confirming the formal discussion in Sec. 3.2. However, the margin loss can not completely defy the forgetting in the long run and its performance is still far from satisfactory. (2) The proposed Stage-2 further improves the performance of the margin loss, shown in both "ℓ_m + Random α_t" and "ℓ_m + Fixed $\alpha_t = 1$". However, without explicitly optimizing Eqn. 10 to obtain the proper α_t, their performance are inferior to "Ours" after the long run. This illustrates the importance of the proposed objective function to search for the proper α_t to achieve a remarkable balance over the long run for ISL. Besides we also find that: (1) the proposed forgetting proxy estimation has a strong statistical correlation with the actual performance drop of the seen subpopulations; (2) our method can robustly perform well for ISL under different sizes of the training dataset in the *base step* and different network structures, which relieves our concern of sharing the feature extractor for ISL. All the details and discussions of limitations are in our supplementary.

5 Conclusion

To alleviate the challenging subpopulation shifting issue, we explore incremental subpopulation learning (ISL) and propose a novel two-stage model to better balance forgetting and acquisition. We provide the first extensive benchmark of existing methods for ISL. Empirical results show that our method outperforms existing ones significantly under different and challenging protocols, which could be a promising baseline for ISL and enlighten future research.

Acknowledgements This work was supported in part by National Science Foundation grant IIS-1815561 and IIS-2007613.

References

1. Abdelsalam, M., Faramarzi, M., Sodhani, S., Chandar, S.: Iirc: Incremental implicitly-refined classification. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 11038–11047 (2021)
2. Ahn, H., Cha, S., Lee, D., Moon, T.: Uncertainty-based continual learning with adaptive regularization. In: Advances in Neural Information Processing Systems. pp. 4392–4402 (2019)
3. Ahn, H., Kwak, J., Lim, S., Bang, H., Kim, H., Moon, T.: Ss-il: Separated softmax for incremental learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 844–853 (October 2021)
4. et al., V.: Brain-inspired replay for continual learning with artificial neural networks. Nature Communications **11**(1), 1–14 (2020)
5. Aljundi, R., Chakravarty, P., Tuytelaars, T.: Expert gate: Lifelong learning with a network of experts. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
6. Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., Tuytelaars, T.: A continual learning survey: Defying forgetting in classification tasks. IEEE Transactions on Pattern Analysis and Machine Intelligence (2021)
7. Deng, J., Dong, W., Socher, R., Li, L.J., Li, K., Fei-Fei, L.: Imagenet: A large-scale hierarchical image database. In: 2009 IEEE Conference on Computer Vision and Pattern Recognition. pp. 248–255. IEEE (2009)
8. Dhar, P., Singh, R.V., Peng, K.C., Wu, Z., Chellappa, R.: Learning without memorizing. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5138–5146 (2019)
9. Frigyik, B.A., Srivastava, S., Gupta, M.R.: An introduction to functional derivatives. Dept. Electr. Eng., Univ. Washington, Seattle, WA, Tech. Rep **1** (2008)
10. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 770–778 (2016)
11. Hou, S., Pan, X., Loy, C.C., Wang, Z., Lin, D.: Learning a unified classifier incrementally via rebalancing. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 831–839 (2019)
12. Hsu, Y.C., Liu, Y.C., Ramasamy, A., Kira, Z.: Re-evaluating continual learning scenarios: A categorization and case for strong baselines. NeurIPS Continual learning Workshop (2018)
13. Kim, C.D., Jeong, J., Kim, G.: Imbalanced continual learning with partitioning reservoir sampling. In: European Conference on Computer Vision. pp. 411–428. Springer (2020)
14. Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A.A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al.: Overcoming catastrophic forgetting in neural networks. Proceedings of the National Academy of Sciences **114**(13), 3521–3526 (2017)
15. Li, Z., Hoiem, D.: Learning without forgetting. IEEE Transactions on Pattern Analysis and Machine Intelligence **40**(12), 2935–2947 (2017)
16. Liu, Y., Schiele, B., Sun, Q.: Rmm: Reinforced memory management for class-incremental learning. Advances in Neural Information Processing Systems **34** (2021)
17. Liu, Y., Parisot, S., Slabaugh, G., Jia, X., Leonardis, A., Tuytelaars, T.: More classifiers, less forgetting: A generic multi-classifier paradigm for incremental learning.

In: Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XXVI 16. pp. 699–716. Springer (2020)

18. Lomonaco, V., Maltoni, D.: Core50: a new dataset and benchmark for continuous object recognition. In: Conference on Robot Learning. pp. 17–26. PMLR (2017)
19. Lopez-Paz, D., Ranzato, M.: Gradient episodic memory for continual learning. In: Advances in Neural Information Processing Systems. pp. 6467–6476 (2017)
20. Maltoni, D., Lomonaco, V.: Continuous learning in single-incremental-task scenarios. Neural Networks **116**, 56–73 (2019)
21. Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D., van de Weijer, J.: Class-incremental learning: survey and performance evaluation on image classification. arXiv preprint arXiv:2010.15277 (2020)
22. Muhlbaier, M.D., Topalis, A., Polikar, R.: Learn ++. nc: Combining ensemble of classifiers with dynamically weighted consult-and-vote for efficient incremental learning of new classes. IEEE Transactions on Neural Networks **20**(1), 152–168 (2008)
23. Polikar, R., Upda, L., Upda, S.S., Honavar, V.: Learn++: An incremental learning algorithm for supervised neural networks. IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews) **31**(4), 497–508 (2001)
24. Prabhu, A., Torr, P.H., Dokania, P.K.: Gdumb: A simple approach that questions our progress in continual learning. In: European Conference on Computer Vision. pp. 524–540. Springer (2020)
25. Rebuffi, S.A., Kolesnikov, A., Sperl, G., Lampert, C.H.: icarl: Incremental classifier and representation learning. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) (July 2017)
26. Saberian, M., Vasconcelos, N.: Multiclass boosting: Margins, codewords, losses, and algorithms. Journal of Machine Learning Research **20**(137), 1–68 (2019), `http://jmlr.org/papers/v20/17-137.html`
27. Saberian, M.J., Vasconcelos, N.: Multiclass boosting: Theory and algorithms. In: Advances in Neural Information Processing Systems. pp. 2124–2132 (2011)
28. Santurkar, S., Tsipras, D., Madry, A.: {BREEDS}: Benchmarks for subpopulation shift. In: International Conference on Learning Representations (2021), `https://openreview.net/forum?id=mQPBmvyAuk`
29. Schapire, R.E., Freund, Y.: Boosting: Foundations and algorithms. Kybernetes (2013)
30. Shin, H., Lee, J.K., Kim, J., Kim, J.: Continual learning with deep generative replay. In: Advances in Neural Information Processing Systems. pp. 2990–2999 (2017)
31. Tao, X., Hong, X., Chang, X., Gong, Y.: Bi-objective continual learning: Learning 'new'while consolidating 'known'. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 34, pp. 5989–5996 (2020)
32. Van de Ven, G.M., Tolias, A.S.: Three scenarios for continual learning. NeurIPS - Continual Learning workshop (2018)
33. Volpi, R., Larlus, D., Rogez, G.: Continual adaptation of visual representations via domain randomization and meta-learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 4443–4453 (2021)
34. Wu, C., Herranz, L., Liu, X., van de Weijer, J., Raducanu, B., et al.: Memory replay gans: Learning to generate new categories without forgetting. In: Advances in Neural Information Processing Systems. pp. 5962–5972 (2018)
35. Wu, G., Gong, S., Li, P.: Striking a balance between stability and plasticity for class-incremental learning. In: Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV). pp. 1124–1133 (October 2021)

36. Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., Fu, Y.: Large scale incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 374–382 (2019)
37. Yan, S., Xie, J., He, X.: Der: Dynamically expandable representation for class incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 3014–3023 (2021)
38. Yoon, J., Yang, E., Lee, J., Hwang, S.J.: Lifelong learning with dynamically expandable networks. In: International Conference on Learning Representations (2018)
39. Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., Weijer, J.v.d.: Semantic drift compensation for class-incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 6982–6991 (2020)
40. Zenke, F., Poole, B., Ganguli, S.: Continual learning through synaptic intelligence. Proceedings of Machine Learning Research **70**, 3987 (2017)
41. Zhao, B., Xiao, X., Gan, G., Zhang, B., Xia, S.T.: Maintaining discrimination and fairness in class incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 13208–13217 (2020)
42. Zhu, F., Zhang, X.Y., Wang, C., Yin, F., Liu, C.L.: Prototype augmentation and self-supervision for incremental learning. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5871–5880 (2021)