A Tale of Two CILs: The Connections between Class Incremental Learning and Class Imbalanced Learning, and Beyond

Chen He[1,2], Ruiping Wang[1,2], Xilin Chen[1,2]
[1]Key Laboratory of Intelligent Information Processing of Chinese Academy of Sciences (CAS),
Institute of Computing Technology, CAS, Beijing, 100190, China
[2]University of Chinese Academy of Sciences, Beijing, 100049, China
chen.he@vipl.ict.ac.cn, {wangruiping, xlchen}@ict.ac.cn

Abstract

Catastrophic forgetting, the main challenge of Class Incremental Learning, is closely related to the classifier's bias due to imbalanced data, and most researchers resort to empirical techniques to remove the bias. Such anti-bias tricks share many ideas with the field of Class Imbalanced Learning, which encourages us to reflect on why these tricks work, and how we can design more principled solutions from a different perspective. In this paper, we comprehensively analyze the connections and seek possible collaborations between these two fields, i.e. Class Incremental Learning and Class Imbalanced Learning. Specifically, we first provide a panoramic view of recent bias correction tricks from the perspective of handling class imbalance. Then, we show that an adapted post-scaling technique which originates from Class Imbalanced Learning is on par with or even outperforms SOTA Class Incremental Learning method. Visualization via violin plots and polar charts further sheds light on how SOTA methods address the class imbalance problem from a more intuitive geometric perspective. These findings may encourage further infiltration between the two closely connected fields, but also raise concerns about whether it is correct that Class Incremental Learning degenerates into a class imbalance problem.

1. Introduction

Incremental Learning, a prerequisite ability for open world applications such as service robots to continually acquire new knowledge in the ever-changing environment, receives increasing attention from both academia and industry. Despite its importance in open world recognition [3], it remains rather difficult to design well-acknowledged academic settings that emulate the natural incremental scenarios, let alone devise algorithms to solve them. To facilitate academic research, three simplified incremental settings are proposed and accepted by researchers [23, 61, 41][1]: Task Incremental, Domain Incremental, and Class Incremental Learning. Among them, Class Incremental Learning is the most popular and promising one, and the reasons are two-fold: for one thing, it assumes no task boundaries and the classifier needs to recognize all seen classes, which is more realistic than Task Incremental Learning; for another, Class Incremental Learning is more challenging than the other settings based on the performance, and there is still a large gap between SOTAs and the performance upperbound, which leaves room for further improvement.

One notorious phenomenon in Class Incremental Learning is catastrophic forgetting [44], which implies that the model may completely forget old knowledge when learning new information. Such a forgetting phenomenon is usually reflected by the acute drops of accuracies on old classes, or the classifier's bias towards old classes[2]. Note that learning incrementally does not necessarily lead to catastrophic forgetting, since storing all historical data and training the model with them will incur little or no forgetting. The main cause of forgetting is attributed to the common assumption in Incremental Learning that the memory is limited, thus only a small portion of old samples can be stored, which makes it resemble a class imbalance problem. Without proper treatment, the model may suffer a lot from the imbalance and the performance might be largely degraded.

From a higher view, the battle against class imbalance in machine learning is a wider topic with a long history [25, 19], since the class imbalance or long-tail distribution is an ubiquitous problem [74, 63, 14]. Although important, techniques that handle class imbalance in the deep learning era are empirically designed and often seen as "tricks" especially in AI challenges [57]. However, the last few years have witnessed rising fields such as *Class*

[1]Strictly speaking, [41] uses New Instances (NI) to indicate DIL, Multi-Task-NC for TIL, and New Instances and Classes (NIC) resembles CIL.

[2]Here the word *bias* means that the decision boundary is closer to the centroid of the minority class, which is in line with [64].

Imbalanced Learning and *Long-Tailed Classification* which seek principled ways to tackle class imbalance of deep models. The resurgence of these fields encourages us to reflect upon the progress of Class Incremental Learning from a novel perspective, and more importantly to see if inspirations could be drawn from these fields in helping design more effective anti-forgetting techniques.

Motivated by these goals, in this paper we comprehensively analyze the connections and seek possible collaborations between these two closely related fields, i.e. Class Incremental Learning and Class Imbalanced Learning. Specifically, we first concretely clarify that the techniques in SOTA Class Incremental Learning methods share similar ideas with Class Imbalanced Learning (Sec. 3.1). Then, we show that an adapted post-scaling technique which originates in Class Imbalanced Learning (Sec. 3.2) can obtain on-par or even better result with Class Incremental Learning methods (Sec. 4.2). Finally, qualitative analyses help us understand why post-scaling works, and how it correlates with a SOTA method MDFCIL [71] from a geometric view (Sec. 4.3). Based on these findings, we provide our thoughts on further collaborations of these two learning paradigms and concerns over the recent progress of Class Incremental Learning (Sec. 4.4).

2. Related Works

Class Incremental Learning (CIL). Class Incremental Learning [53], a thriving subfield in Incremental Learning [59, 10], has been attracting increasing attention in the computer vision community. To alleviate catastrophic forgetting [44], the main challenge in this field, additional memory must be leveraged for memory replay. The memory generally falls into two categories: episodic memory which holds a small number of old exemplars [53, 5, 65, 71], and generative memory which stores generative models such as GANs or VAEs [58, 17, 66, 49]. Since generative memory usually needs longer training time and higher memory footprint, episodic memory based approaches are more promising. However, an underlying problem with episodic memory is that the ratio of the number of old exemplars to that of new samples might be very high, resulting in severe class imbalance. Thus, techniques to address class imbalance must be leveraged. Interestingly, the development of Class Incremental Learning in recent years is almost a history of finding solutions to address class imbalance [53, 5, 2, 65, 71]. In this paper we reflect upon these approaches from the view of Class Imbalanced Learning.

Class Imbalanced Learning (CIL). Class imbalance or long-tail distribution is ubiquitous, and the battle against it has a long history [26, 64, 4]. In recent years, we have witnessed a popular trend of independent fields such as *Class Imbalanced Learning* [29, 16] and *Long-Tailed Classification* [28, 72], where the former has a much longer history [73]. A major finding in Class Imbalanced Learning is that the learned classifier might be biased towards the minority class [64], since the variance in the minority class is often underestimated due to insufficient samples. According to [4], methods for addressing class imbalance can be divided into two main categories: *data-level methods* and *classifier-level methods*. Data-level methods operate on the training set and change the class distribution, and examples are under-sampling [26], over-sampling [39], SMOTE [7] etc. Classifier-level methods keep the training set intact and adjust the training or inference algorithms, and examples are cost-sensitive learning (a.k.a. re-weighting), post-scaling [36, 4] etc. Since Class Incremental Learning also exhibits a problem of biasing towards old classes, existing techniques in Class Imbalanced Learning can also be applied in Class Incremental Learning.

3. Connections between Two CILs

Class Incremental Learning (CIL) and Class Imbalanced Learning (CIL) have exactly the same acronym. For disambiguation, we use **CIncL** and **CImbL** to denote them respectively hereinafter. Apart from the similarity in their names literally, there are more connections between them from a technical point of view. In this section, we first elaborate on the relationship between the technical designs of SOTA CIncL methods and similar CImbL ideas, then show that CImbL techniques can be also applied in CIncL.

3.1. CIncL exhibits Class Imbalance

In the typical CIncL setting, a bounded memory to hold old exemplars is maintained [53]. At each incremental phase, the method has to train the model with a combination of old exemplars and new samples. Since the memory is limited, the number of old exemplars for each class is usually much lower than that of a new class, resulting in severe class imbalance. In the following paragraphs, we will elaborate on the relationship between anti-bias techniques in recent CIncL works and ideas in CImbL. The general pipeline for these methods is shown in Fig. 1.

iCaRL [53]. The core building block in handling class imbalance is the *Nearest Class Mean (NCM) classifier*, and we can understand its anti-imbalance property from two aspects: (1) From the perspective of non-parametric classifier, NCM is a distance-based classification method similar to k-NN. The difference is that k-NN also has a biasing problem [43, 40], however, NCM alleviates this problem by aggregating the information of an arbitrary number of samples into a single prototype, which resembles a *prototype generation* (a type of *under-sampling*) in nearest neighbor approaches [60]. Therefore, other classic *prototype generation* methods such as LVQ [31] could also be leveraged as alternatives; (2) NCM can be seen as a Bayesian method which assumes that each class obeys a multivariate Gaus-

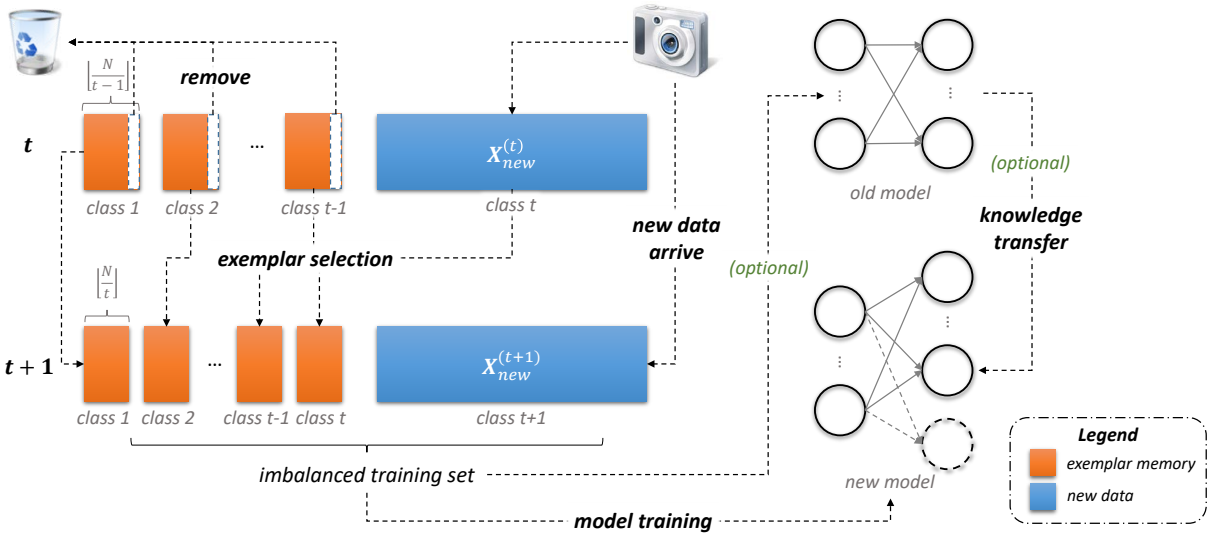

Figure 1. The general pipeline for many Class Incremental Learning methods from time t to $t+1$. On the left, it shows the exemplar management process, where the memory has a fixed budget of N evenly allocated for each old class. On the right, it depicts the process of model training given the imbalanced training set, where the knowledge transfer from the old model is optional for some methods [2, 52].

sian distribution with an *identical isotropic* covariance matrix (i.e. $\Sigma_c = \sigma^2 I$). The *identity* of the covariance matrices may remedy the class imbalance issue, and the *isotropy* further eases the computation. The readers could refer to the mathematical derivations in the supplementary material.

End-to-End Incremental Learning (EEIL) [5]. The technique to address class imbalance is the *balanced fine-tuning* in the 2nd stage as indicated by [5]. From a CImbL's view, the combination of regular training on the imbalanced dataset and balanced fine-tuning resembles a *two-phase training* strategy [15]. The main difference is that [15] adopts a balanced training in the 1st stage, because it wants most of the capacity (i.e. feature extractor) to account for the diversity in a balanced way for all classes. For the 2nd stage, it trains on imbalanced dataset with the feature extractor fixed, making the output layer reflect the natural frequencies of the classes in the data.

Large-Scale Incremental Learning (LSIL) [65]. LSIL might be one of the few works that leverage validation sets in CIncL. The idea is to learn a linear model to correct the bias of the output logits for the new classes, which is similar to a classic *probability calibration* method called *Platt scaling* [51]. The difference is that LSIL is a softmax version instead of sigmoid version. For more recent probability calibration methods, the readers could refer to [35, 12, 34].

Incremental Learning with Dual Memory (IL2M) [2]. IL2M stores the average confidence of the model at each incremental phase and the logits (before softmax) of each class when the corresponding class is first added for learning. At test time, it rectifies the scores based on the above-mentioned statistics if a test sample is predicted as new classes. Judging from the characteristics of IL2M, it also resembles a *probability calibration* method [51, 69, 70] except that the parameters are directly estimated rather than learned given the validation set.

Maintaining Discrimination and Fairness in Class Incremental Learning (MDFCIL) [71]. MDFCIL is motivated by the empirical findings [13] that the norms of the weights in the final fully connected layers are related to the numbers of training samples for each class. By normalizing the weights to have similar norms after training, the biases induced by different numbers of samples can be removed. Interestingly, a similar approach appears in a recent CImbL paper with the name *τ-normalized classifier* [28]. The underlying reason for different norms of the weights might be that they can reflect the complexity of the decision boundaries [48, 47]. Since more samples generally lead to more complex intra-variation, the decision boundaries might be more complicated, making the corresponding norms larger.

GDumb [52]. GDumb maintains a *greedy balancing sampler* which always holds a balanced training set for all classes, and uses these samples to train the model whenever needed. The *greedy balancing sampler* also resembles an *under-sampling* strategy in CImbL, which means that only a small portion of new class samples are kept. The downside is that a large number of new class samples are wasted.

So far we have elaborated on the characteristics of anti-bias techniques in recent CIncL methods, and a panoramic view is provided in Table 1.

3.2. CImbL avails CIncL

Commonly used CImbL techniques such as re-weighting and re-sampling (e.g. under-sampling, over-sampling) can be easily applied in CIncL. In this section, we are more in-

Table 1. Anti-bias techniques of recent Class Incremental Learning (CIncL) methods and their corresponding ideas in Class Imbalanced Learning (CImbL). Methods are ordered chronologically.

Method	**Anti-bias Technique(s)**	**Corresponding Idea in CImbL**		**Which Phase**
		Level	Technique	
iCaRL [53]	NCM classifier	Classifier	Prototype generation	Test
EEIL [5]	Balanced fine-tuning	Data	Two-phase training	Train
LSIL [65]	Bias correction	Classifier	Probability calibration	Train
IL2M [2]	Rescaling	Classifier	Probability calibration	Test
MDFCIL [71]	Weight aligning	Classifier	N/A	Test
GDumb [52]	Greedy balancing sampler	Data	Under-sampling	Train

terested in adapting a more theory-driven and simple way called *post-scaling*[3] to the field of CIncL.

Let us denote the feature extractor as f. The weight and bias in the fully connected layer are $\mathbf{W}$ and $\mathbf{b}$. Then, based on the assumption that neural networks estimate posterior probabilities if softmax cross-entropy is used [54], the posterior probability of class c when the model converges is:

$$\begin{aligned} p_{tr}(c|\mathbf{x}) &= \frac{\exp(\mathbf{W}_c^T f(\mathbf{x}) + b_c)}{\sum_i \exp(\mathbf{W}_i^T f(\mathbf{x}) + b_i)} \\ &= \frac{p_{tr}(\mathbf{x}|c)p_{tr}(c)}{\sum_i p_{tr}(\mathbf{x}|i)p_{tr}(i)} \end{aligned} \tag{1}$$

The second line in Eq. 1 is obtained via Bayesian theorem on $p_{tr}(c|\mathbf{x})$. The subscript *tr* stands for the *training set* (*ts* for the *test set*). It is natural to assume that the samples of the training or the test set are obtained by the same generation process, thus we have $p_{tr}(\mathbf{x}|c) = p_{ts}(\mathbf{x}|c)$. Consequently, the posterior probability of class c at test time is:

$$\begin{aligned} p_{ts}(c|\mathbf{x}) &= \frac{p_{ts}(\mathbf{x}|c)p_{ts}(c)}{\sum_i p_{ts}(\mathbf{x}|i)p_{ts}(i)} \\ &= \frac{p_{tr}(\mathbf{x}|c)p_{tr}(c) \times \frac{p_{ts}(c)}{p_{tr}(c)}}{\sum_i p_{tr}(\mathbf{x}|i)p_{tr}(i) \times \frac{p_{ts}(i)}{p_{tr}(i)}} \end{aligned} \tag{2}$$

By applying Eq. 1 into Eq. 2, the prediction function is:

$$\begin{aligned} p_{ts}(c|\mathbf{x}) &= \frac{\exp(\mathbf{W}_c^T f(\mathbf{x}) + b_c) \times \frac{p_{ts}(c)}{p_{tr}(c)}}{\sum_i \exp(\mathbf{W}_i^T f(\mathbf{x}) + b_i) \times \frac{p_{ts}(i)}{p_{tr}(i)}} \\ &= \frac{\exp\{\mathbf{W}_c^T f(\mathbf{x}) + b_c + \log(\frac{p_{ts}(c)}{p_{tr}(c)})\}}{\sum_i \exp\{\mathbf{W}_i^T f(\mathbf{x}) + b_i + \log(\frac{p_{ts}(i)}{p_{tr}(i)})\}} \end{aligned} \tag{3}$$

Eq. 3 indicates that we do not need to change the softmax function, but only need to add an extra bias term to the logit for each class. In our implementation, we introduce a non-learnable post-scaling layer at the end of the network, whose role is simply to add $\log(\frac{p_{ts}(c)}{p_{tr}(c)})$ for each class. In the equation above, $p_{ts}(c)$ can be set to be $\frac{1}{C}$ where C is the number of classes, since it is equiprobable that a sample belongs to each class at test time. $p_{tr}(c)$ is simply estimated by the ratio of the sample number of class c to the total number of samples in the training set. From the perspective of *prior shift* [32] (a type of *dataset shift* [46]) where the prior probabilities of source and target domain are different, the *post-scaling* method above tries to compensate for *prior shift* and let the classifier adapt well to the test data.

Note that there are assumptions that may influence the performance: (1) The estimation of $p_{tr}(c)$ might be incorrect, because the number of samples might not reliably reflect the prior probability. Inspired by [8], one may use the effective number of samples to estimate the prior. (2) Eq. 1 holds when the model converges (i.e. the cross entropy loss is low on the training set). Therefore, this method might not work well when the model underfits (e.g. less number of epochs); (3) $p_{tr}(\mathbf{x}|c) = p_{ts}(\mathbf{x}|c)$ might not hold due to insufficient samples, as noted by a recent CImbL paper [24].

4. Experiment

4.1. Experimental Setups

Datasets. We use CIFAR-100 [33] and Group ImageNet [55]. Group ImageNet is a 100-class ImageNet subset which covers 10 super-categories and each super-category has exactly 10 sub-categories. Also, it is down-sampled to 64×64 for faster evaluation. The details of Group ImageNet is shown in the supplementary material.

Evaluation Metrics. We use the top-1 accuracy in the final class increment as other CIL papers do.

Methods. As for CIncL, we compare Learning without Forgetting (LwF) [38], iCaRL [53], End-to-End Incremental Learning (EEIL) [5], Large Scale Incremental Learning (LSIL) [65], Maintaining Discrimination and Fairness in Class Incremental Learning (MDFCIL) [71], and GDumb [52]. Note that we remove exclusive techniques in these papers such as intense data augmentation and gradient noise in EEIL [5], because we want to fairly compare their

[3] "Post" means that it functions at test phase. "Scaling" means that the network outputs are multiplied by certain numbers. Although the adapted form might be not in line with the literal name, we still use *post-scaling* in this paper since the modification over the original version is tiny.

anti-imbalance techniques. As for CImbL-inspired methods, we compare random under-sampling, random over-sampling, SMOTE [7], ADASYN [18], cluster centroids that uses k-means and k-medoids, re-weighting, and the adapted post-scaling mentioned in Sec. 3.2.

Implementations. The codes are implemented via Tensorflow 2.1 [1] [4]. An Adam optimizer [30] is adopted for all experiments, and its learning rate is determined via a grid search. For CIFAR-100, we use a LeNet-like network and ResNet-34 [21]. For Group ImageNet, we use ResNet-18 [21] and MobileNetV2 [56]. The number of training epochs is set to 70, and the learning rate is multiplied by 0.1 at epoch 49 and 63. Random exemplar selection is adopted, since other specially designed selection strategies do not have substantial improvement over random selection [66, 6, 27]. As for the techniques in CImbL, most of them are provided by Python module *imblearn* [37]. More details can be found in the supplementary material.

4.2. Quantitative Results

The final accuracies of methods on these two datasets are summarized in Table 2. Note that the accuracies might not be precisely in line with other papers, since the goal here is to fairly compare the anti-bias techniques in these methods. From the results, it can be observed that more recent CIncL approaches generally yield better performance. It is also noteworthy that the bias correction techniques in these methods can boost the performance, and some improvements are very impressive judging from the accuracies before and after "/" in Table 2! It indicates that handling class imbalance is very necessary if one wants to largely boost the classification performance.

As for CIncL methods, **MDFCIL** [71] is the most effective one which outperforms others in most cases. **LwF** [38] is much lower than others, which is reasonable since it incorporates no anti-imbalance technique, making it vulnerable to class imbalance. **iCaRL** [53] is better than LwF, but there is still a large gap between iCaRL and SOTA methods. The reason might be that the prototype is generated by samples of the corresponding class only. The prototype can well represent the given class, but it lacks discriminative power and its superiority might mainly lie in handling adversarial attacks in open-set recognition [3, 67]. **EEIL** [5] performs better than iCaRL, because it trains the network in an end-to-end fashion, which might alleviate the incompatibility between the feature extractor and the classifier. The superiority of **LSIL** [65] over EEIL is not so clear, and the reason might be that some exemplars are not used to train the base model, but belong to the validation set in LSIL. **IL2M** [2] performs less satisfactory, and the main reason is that it lacks the distillation loss adopted in other approaches. In our implementation, we find that the distillation loss is important for the old model to transfer knowledge to the new model. **GDumb** [52] is much unsatisfactory because it transfers no knowledge from the previous model (i.e. training from scratch), and it discards a lot of samples in the *greedy balancing sampler*. The accuracy 1% for MobileNet is not a mistake, since we find that MobileNet is difficult to train for almost all base learning rates. Advanced techniques in GDumb (e.g. SGDR [42], cutmix [68] etc.) and more epochs (i.e. 256) in the original paper could give better results, but that is not the main focus of this paper.

As for CImbL-inspired techniques, **post-scaling** performs pretty well and outperforms MDFCIL in some cases. The other techniques (i.e. re-weighting and re-sampling) all perform less satisfactory, and such results are in line with [72] which assumes that re-sampling or re-weighting may damage the representation. It is also noteworthy that advanced over-sampling techniques such as **SMOTE** [7] and **ADASYN** [18] have no substantial improvement over the naive **random over-sampling**. The reason might be that we perform SMOTE or ADASYN in the image space rather than feature space. Since the generated samples may not exist in the real world, it might have a negative effect on the performance. Such a phenomenon can also be seen in the comparison between **random under-sampling** and **cluster centroids (k-means)**, since the samples generated by k-means might also be unrealistic. To resolve this problem, by using k-medoids instead of k-means, we can observe a marginal but consistent improvement since the samples generated by k-medoids all belong to the original dataset.

Apart from accuracies, we are also interested in the efficiency of these methods. To evaluate it, we recorded the average running time of each class incremental phase in these methods, and summarized them in Table 3. Note that the statistics might be inaccurate, since the running time is correlated with the CPU/GPU overload at that time. However, they can roughly reflect the efficiency of these methods. Among them, LSIL and SMOTE are most time-consuming. As for LSIL, the reason might be that it needs twice the number of epochs to learn the bias correction parameters. In practice, we notice that such a large number of epochs is unnecessary, and reducing it might yield similar results. As for SMOTE, it will continually select neighboring images to synthesize new ones, which will need much computation.

Judging from both performance and efficiency, we recommend that the practitioners should consider MDFCIL and post-scaling first for CIncL.

4.3. Further Analyses

So far, we find that the technique *post-scaling* in Sec. 3.2 and *weight aligning* in MDFCIL [71] are very effective. Here, we want to understand their characteristics more intuitively from a geometric view.

[4] The codes are at http://vipl.ict.ac.cn/resources/codes or https://github.com/TonyPod/Two-CILs

Table 2. Final accuracies of methods on different datasets. CIncL stands for Class Incremental Learning methods, and CImbL indicates Class Imbalanced Learning inspired methods. For methods with an extra bias correction stage, the accuracies before and after the bias correction are separated by a slash (/). Each result is obtained by averaging the accuracies under 5 class orders on CIFAR-100 and 3 class orders on Group ImageNet (excluding *lowerbound* and *upperbound*).

Method		**CIFAR-100**		**Group ImageNet**	
		LeNet	ResNet32	MobileNet	ResNet18
Lowerbound		8.88	9.08	9.26	9.36
CIncL	LwF† (*ECCV'16*) [38]	30.04	37.22	26.69	34.45
	iCaRL (*CVPR'17*) [53]	29.30/35.10	37.40/42.95	24.67/34.05	33.98/42.01
	EEIL (*ECCV'18*) [5]	31.15/37.61	38.34/43.27	31.74/39.98	35.93/46.13
	LSIL (*CVPR'19*) [65]	36.49/39.04	35.24/41.38	18.71/**39.49**	28.22/36.77
	IL2M (*ICCV'19*) [2]	27.61/26.71	34.22/35.27	24.27/27.38	30.58/36.23
	MDFCIL (*CVPR'20*) [71]	30.49/**39.94**	39.63/**47.96**	17.78/34.33	31.93/**47.19**
	GDumb (*ECCV'20*) [52]	21.09	19.41	1.00	19.97
CImbL	Re-weighting	33.29	37.23	31.60	38.08
	Random over-sampling	34.53	35.12	27.69	35.85
	SMOTE	33.85	35.71	27.61	34.95
	ADASYN	33.77	35.93	26.68	34.07
	Random under-sampling	31.06	35.13	25.95	37.28
	Cluster centroids (k-means)	28.23	31.85	21.01	32.70
	Cluster centroids (k-medoids)	31.32	35.80	26.65	37.64
	Post-scaling	31.58/**38.68**	38.92/**44.44**	30.55/**40.50**	37.09/**48.36**
Upperbound		58.98	66.85	68.82	72.74

Table 3. Average running times of methods for a single class incremental phase on different datasets (in seconds). The organization of the table is similar to Table 2. Each result is an average based on 5 class orders on CIFAR-100 and 3 class orders on Group ImageNet.

Method		**CIFAR-100**		**Group ImageNet**	
		LeNet	ResNet32	MobileNet	ResNet18
CIncL	LwF† (*ECCV'16*) [38]	329.20	522.96	1507.75	1204.74
	iCaRL (*CVPR'17*) [53]	441.82	546.44	1750.17	1421.89
	EEIL (*ECCV'18*) [5]	641.00	876.38	2655.51	2207.73
	LSIL (*CVPR'19*) [65]	2278.35	2453.36	5576.40	4565.65
	IL2M (*ICCV'19*) [2]	323.00	483.82	1634.51	1097.80
	MDFCIL (*CVPR'20*) [71]	334.19	520.64	1655.27	1266.25
	GDumb (*ECCV'20*) [52]	194.39	258.02	448.06	295.15
CImbL	Re-weighting	341.50	557.00	1535.39	1218.52
	Random over-sampling	1067.41	1954.07	8926.25	6530.99
	SMOTE	1207.14	2138.63	8012.42	6201.43
	ADASYN	1221.74	2005.56	1877.00	1550.95
	Random under-sampling	203.77	255.79	343.31	384.83
	Cluster centroids (k-means)	1083.24	1113.02	2629.24	2354.59
	Cluster centroids (k-medoids)	195.65	280.40	587.23	413.90
	Post-scaling	340.58	500.42	1434.07	1331.76

Post-scaling. The idea of post-scaling is to translate the decision boundary towards the majority class along the normal vector. To facilitate analysis, we only consider one old class (***min**ority class*) and one new class (***maj**ority class*), the decision boundary between them can be determined by assuming that their probabilities after softmax to be equal:

$$\frac{\exp(\mathbf{W}_{min}^T f(\mathbf{x}) + b_{min})}{\sum_i \exp(\mathbf{W}_i^T f(\mathbf{x}) + b_i)} = \frac{\exp(\mathbf{W}_{maj}^T f(\mathbf{x}) + b_{maj})}{\sum_i \exp(\mathbf{W}_i^T f(\mathbf{x}) + b_i)} \tag{4}$$

The definition of f, $\mathbf{W}$ and $\mathbf{b}$ is the same as Sec. 3.2. Obviously, the denominator can be canceled out. Also, since $\exp(\cdot)$ is monotonous, Eq. 4 can be further derived as:

$$(\mathbf{W}_{min}^T - \mathbf{W}_{maj}^T)f(\mathbf{x}) + (b_{min} - b_{maj}) = 0 \quad (5)$$

Eq. 5 implies that the decision boundary is linear with respect to the feature space. Consequently, we can project $f(\mathbf{x})$ onto the normal vector of the decision boundary, and its "distance" to the decision boundary can be obtained by[5]:

$$d(\mathbf{x}) = \frac{(\mathbf{W}_{min}^T - \mathbf{W}_{maj}^T)f(\mathbf{x}) + (b_{min} - b_{maj})}{\|\mathbf{W}_{min}^T - \mathbf{W}_{maj}^T\|} \quad (6)$$

Based on Eq. 6, we can obtain the distributions of $d(\mathbf{x})$ for samples of these two classes and draw two-dimension violin plots [22] (Fig. 2). By comparing *training all* and *test*, we can find that the variance of the minority class becomes smaller, because only a small number of exemplars are stored and the variance is underestimated. By comparing *training* and *test*, we can find that the decision boundary before post-scaling actually fits *training* pretty well, but it is not suitable for *test*. After post-scaling, the decision boundary moves slightly towards the majority class and can better separate the samples of these two classes.

Weight aligning. If there is no bias term in the final fully connected layer, then *weight aligning* tries to rotate the decision boundary towards the majority class around the origin. To understand it, we show the decision boundary in a two-dimensional polar space. The original boundary is:

$$\mathbf{W}_{min}^T f(\mathbf{x}) = \mathbf{W}_{maj}^T f(\mathbf{x}) \quad (7)$$

Using the property of inner product, Eq. 7 is derived as:

$$\|\mathbf{W}_{min}\| \cos(\theta^*) = \|\mathbf{W}_{maj}\| \cos(\beta - \theta^*) \quad (8)$$

In Eq. 8, β is the angle between $\mathbf{W}_{min}$ and $\mathbf{W}_{maj}$, and θ^* ($\theta^* \in [-\frac{\pi}{2}, \frac{\pi}{2}]$) is the angle between $\mathbf{W}_{min}$ and the decision boundary. θ^* can be solved via trigonometric formulas:

$$\theta^* = \arctan(\frac{\|\mathbf{W}_{min}\|}{\|\mathbf{W}_{maj}\| \sin(\beta)} - \cot(\beta)) \quad (9)$$

We can obtain the decision boundary after *weight aligning* in a similar way. Note that $\theta^* + k\pi (k \in \mathbb{Z})$ is also a decision boundary in the polar coordinate system due to periodicity. Then, we want to project a sample $f(\mathbf{x})$ onto the subspace spanned by $\mathbf{W}_{min}$ and $\mathbf{W}_{maj}$. Using Gram-Schmidt orthogonalization, we can construct the following two normal basis vectors:

$$\begin{aligned} \mathbf{e}_1 &= \frac{\mathbf{W}_{min}}{\|\mathbf{W}_{min}\|} \\ \mathbf{e}_2 &= \frac{\mathbf{W}_{maj} - (\mathbf{W}_{maj} \cdot \mathbf{e}_1)\mathbf{e}_1}{\|\mathbf{W}_{maj} - (\mathbf{W}_{maj} \cdot \mathbf{e}_1)\mathbf{e}_1\|} \end{aligned} \quad (10)$$

[5]The absolute value of Eq. 6 is the distance. We remove $|\cdot|$ to denote which side of the point lies with respect to the decision boundary.

Consequently, for each $f(\mathbf{x})$ we can get its projected vector represented by $\mathbf{e}_1$ and $\mathbf{e}_2$, and the angle in the polar space θ ($\theta \in [-\pi, \pi]$) can be determined by *atan2* provided by many Python packages such as *NumPy* [62].

$$\theta = atan2(f(\mathbf{x}) \cdot \mathbf{e}_2, f(\mathbf{x}) \cdot \mathbf{e}_1) \quad (11)$$

Based on Eq. 11, all samples can be represented as angles and we can show their distributions in a polar plot (Fig. 3). In Fig. 3b, It can be observed that the decision boundary can separate the old exemplars and new samples pretty well. However, the range of angles in the old class (i.e. minority class) is underestimated (Fig. 3a). By rotate the decision boundary anti-clockwise, the biasing phenomenon due to class imbalance can be alleviated, and the new decision boundary is much better (Fig. 3c).

Summary. Although we only analyze the characteristics of two classes, the *post-scaling* or *weight aligning* process on the multi-class classifier can be seen as a re-adjustment of all two-class decision boundaries. After that, the underestimated variances of old classes are compensated (i.e. those of new classes are shrinked), whether by altering the distances in *post-scaling* or angles in *weight aligning*.

4.4. Discussions

So far, we have shown that CIncL and CImbL are highly correlated. The **merit** is that techniques in CImbL can be applied in CIncL as well. Note that we have only visited a small portion of CImbL approaches, and there might be more principled and effective approaches left untouched. Generalized Low-Shot Learning (GLSL)[6] [11, 9, 50] which also exhibits a class imbalance problem could be another field we can draw inspirations from. From the results in Sec. 4.2, it can be found that *weight aligning* [71] in CIncL and *post-scaling* in CImbL are two effective methods with little computation overhead. There are many connections between these two methods: *Weight aligning* multiplies a scalar to the logits based on empirical findings, whereas *post-scaling* adds a scalar to the logits based on theoretical derivation; *Post-scaling* translates the decision boundary towards the majority class, whereas *weight aligning* rotates the decision boundary (Sec. 4.3). This is a small step towards bridging the gap between CIncL and CImbL, and we can foresee more collaborations between these two fields. Note that although we revolve around class incremental scenarios, there is still a need for techniques in CImbL for more flexible incremental learning settings [45, 20].

However, the **downside** is that CIncL seems to degenerate to a CImbL problem. If so, what is the exclusive problem left for CIncL? If the numbers of samples are equal for all classes, would recent CIncL methods still have large

[6]Also known as Generalized Few-Shot Learning (GFSL).

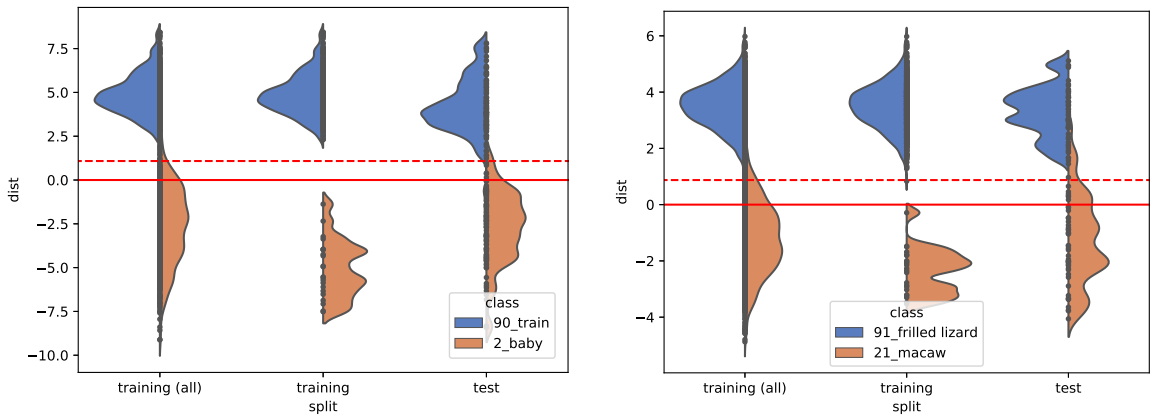

Figure 2. Violin plots of the *post-scaling* method in the final class increment on CIFAR-100 (left) and Group ImageNet (right). The y-axis represents $d(\mathbf{x})$. Along the x-axis, there are three different splits of the dataset: *training all* (all training samples, including the discarded ones), *training* (exemplars for the old class plus the new class samples), *test* (the test samples). The red solid line (i.e. $d(\mathbf{x}) = 0$) and the red dashed line are the decision boundary trained with the *training* split before and after *post-scaling* respectively. The two classes consist of an old class (in blue) and a new class (in orange). The number before the class name in the legend is the index in the class order.

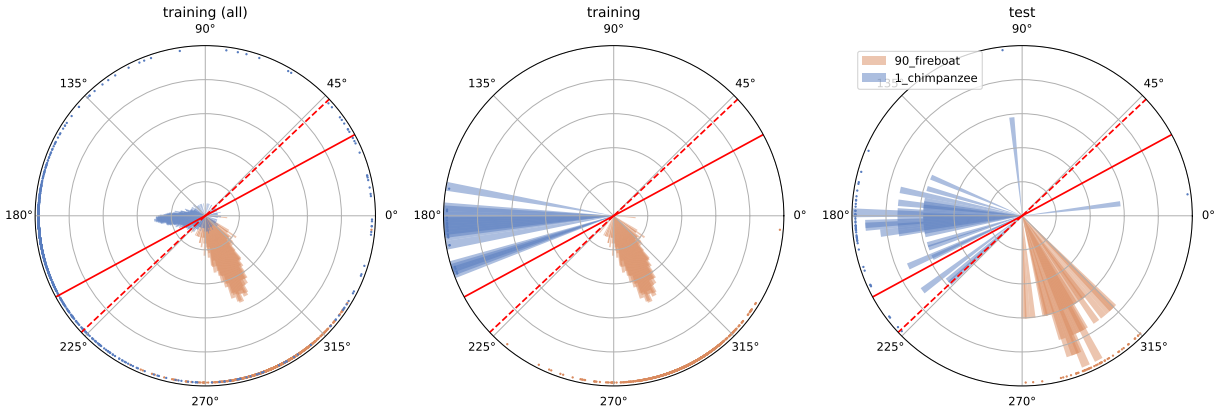

Figure 3. Polar charts of the *weight aligning* method in the final class increment on Group ImageNet. There are three sub-graphs indicating three different splits of the dataset similar to Fig. 2. Each sub-graph depicts the histograms of θ (Eq. 9) of the samples in the corresponding split of the dataset. Note that θ is in $[-\pi, \pi]$. To make it agree with the range of the angular axis $[0, 2\pi]$, we add π to θ. Since some detailed information might be not visible via histograms, θ of individual samples are also shown beside the largest circle (magnify for better view). The red solid line and the red dashed line are the decision boundary before and after *weight aligning* respectively. The two classes consist of an old class (in blue) and a new class (in orange). More results can be found in the supplementary material.

difference in performance? What if no old samples are discarded? Also, is it questionable that addressing class imbalance is equivalent to avoiding catastrophic forgetting? These questions are challenging and deserve more attention.

5. Conclusion

In this paper, we comprehensively analyze the connections between Class Incremental Learning (CIncL) and Class Imbalanced Learning (CImbL). Specifically, we show that existing techniques in recent CIncL methods share many ideas with CImbL. Also, by introducing common approaches that originate in CImbL, we find that a simple *post-scaling* technique achieves on-par or better performance than SOTA in CIncL with little memory usage. Visualization via violin charts or polar charts offer geometric views about how bias correction works, and shed lights on the underlying similarity between two effective methods that originate in two fields: *post-scaling* in CImbL and *weight aligning* in CIncL. Future works might be encouraging more collaborations between these two learning paradigms. We should also reflect upon what the exclusive problem left for CIncL is, and whether catastrophic forgetting and class imbalance are closely related.

Acknowledgements. This work is partially supported by Natural Science Foundation of China under contracts Nos. 61922080, U19B2036, 61772500, CAS Frontier Science Key Research Project No. QYZDJ-SSWJSC009, and Beijing Academy of Artificial Intelligence No. BAAI2020ZJ0201.

References

[1] Martín Abadi, Paul Barham, Jianmin Chen, Zhifeng Chen, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Geoffrey Irving, Michael Isard, et al. Tensorflow: A system for large-scale machine learning. In *12th USENIX Symposium on Operating Systems Design and Implementation (OSDI 16)*, pages 265–283, 2016.

[2] Eden Belouadah and Adrian Popescu. IL2M: Class incremental learning with dual memory. In *IEEE International Conference on Computer Vision (ICCV)*, pages 583–592, 2019.

[3] Abhijit Bendale and Terrance Boult. Towards open world recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1893–1902, 2015.

[4] Mateusz Buda, Atsuto Maki, and Maciej A Mazurowski. A systematic study of the class imbalance problem in convolutional neural networks. *Neural Networks*, 106:249–259, 2018.

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

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

[7] Nitesh V Chawla, Kevin W Bowyer, Lawrence O Hall, and W Philip Kegelmeyer. SMOTE: Synthetic minority oversampling technique. *Journal of Artificial Intelligence Research*, 16:321–357, 2002.

[8] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9268–9277, 2019.

[9] Hang Gao, Zheng Shou, Alireza Zareian, Hanwang Zhang, and Shih-Fu Chang. Low-shot learning via covariance-preserving adversarial augmentation networks. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 983–993, 2018.

[10] Alexander Gepperth and Barbara Hammer. Incremental learning algorithms and applications. 2016.

[11] Spyros Gidaris and Nikos Komodakis. Dynamic few-shot visual learning without forgetting. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4367–4375, 2018.

[12] Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning (ICML)*, pages 1321–1330. PMLR, 2017.

[13] Yandong Guo and Lei Zhang. One-shot face recognition by promoting underrepresented classes. *arXiv preprint arXiv:1707.05574*, 2017.

[14] Agrim Gupta, Piotr Dollar, and Ross Girshick. LVIS: A dataset for large vocabulary instance segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 5356–5364, 2019.

[15] Mohammad Havaei, Axel Davy, David Warde-Farley, Antoine Biard, Aaron Courville, Yoshua Bengio, Chris Pal, Pierre-Marc Jodoin, and Hugo Larochelle. Brain tumor segmentation with deep neural networks. *Medical Image Analysis*, 35:18–31, 2017.

[16] Munawar Hayat, Salman Khan, Syed Waqas Zamir, Jianbing Shen, and Ling Shao. Gaussian affinity for max-margin class imbalanced learning. In *IEEE International Conference on Computer Vision (ICCV)*, pages 6469–6479, 2019.

[17] Chen He, Ruiping Wang, Shiguang Shan, and Xilin Chen. Exemplar-supported generative reproduction for class incremental learning. In *British Machine Vision Conference.*, page 98, 2018.

[18] Haibo He, Yang Bai, Edwardo A Garcia, and Shutao Li. ADASYN: Adaptive synthetic sampling approach for imbalanced learning. In *2008 IEEE International Joint Conference on Neural Networks (IEEE World Congress on Computational Intelligence)*, pages 1322–1328. IEEE, 2008.

[19] Haibo He and Edwardo A Garcia. Learning from imbalanced data. *IEEE Transactions on Knowledge and Data Engineering*, 21(9):1263–1284, 2009.

[20] Jiangpeng He, Runyu Mao, Zeman Shao, and Fengqing Zhu. Incremental learning in online scenario. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13926–13935, 2020.

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

[22] Jerry L Hintze and Ray D Nelson. Violin plots: a box plot-density trace synergism. *The American Statistician*, 52(2):181–184, 1998.

[23] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*, 2018.

[24] Muhammad Abdullah Jamal, Matthew Brown, Ming-Hsuan Yang, Liqiang Wang, and Boqing Gong. Rethinking class-balanced methods for long-tailed visual recognition from a domain adaptation perspective. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7610–7619, 2020.

[25] Nathalie Japkowicz et al. Learning from imbalanced data sets: a comparison of various strategies. In *AAAI Workshop on Learning from Imbalanced Data Sets*, volume 68, pages 10–15. AAAI Press Menlo Park, CA, 2000.

[26] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent Data Analysis*, 6(5):429–449, 2002.

[27] Khurram Javed and Faisal Shafait. Revisiting distillation and incremental classifier learning. In *Asian Conference on Computer Vision (ACCV)*, pages 3–17. Springer, 2018.

[28] Bingyi Kang, Saining Xie, Marcus Rohrbach, Zhicheng Yan, Albert Gordo, Jiashi Feng, and Yannis Kalantidis. Decoupling representation and classifier for long-tailed recognition. In *International Conference on Learning Representations (ICLR)*, 2020.

[29] Salman Khan, Munawar Hayat, Syed Waqas Zamir, Jianbing Shen, and Ling Shao. Striking the right balance with uncertainty. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 103–112, 2019.

[30] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

[31] Teuvo Kohonen. The self-organizing map. *Proceedings of the IEEE*, 78(9):1464–1480, 1990.

[32] Wouter M Kouw and Marco Loog. An introduction to domain adaptation and transfer learning. *arXiv preprint arXiv:1812.11806*, 2018.

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

[34] Meelis Kull, Miquel Perello-Nieto, Markus Kängsepp, Hao Song, Peter Flach, et al. Beyond temperature scaling: Obtaining well-calibrated multiclass probabilities with dirichlet calibration. *arXiv preprint arXiv:1910.12656*, 2019.

[35] Meelis Kull, Telmo M Silva Filho, Peter Flach, et al. Beyond sigmoids: How to obtain well-calibrated probabilities from binary classifiers with beta calibration. *Electronic Journal of Statistics*, 11(2):5052–5080, 2017.

[36] Steve Lawrence, Ian Burns, Andrew Back, Ah Chung Tsoi, and C Lee Giles. Neural network classification and prior class probabilities. In *Neural Networks: Tricks of the Trade*, pages 299–313. Springer, 1998.

[37] Guillaume Lemaître, Fernando Nogueira, and Christos K Aridas. Imbalanced-learn: A python toolbox to tackle the curse of imbalanced datasets in machine learning. *The Journal of Machine Learning Research*, 18(1):559–563, 2017.

[38] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *European Conference on Computer Vision (ECCV)*, pages 614–629. Springer, 2016.

[39] Charles X Ling and Chenghui Li. Data mining for direct marketing: Problems and solutions. In *Proceedings of the Fourth International Conference on Knowledge Discovery and Data Mining*, pages 73–79, 1998.

[40] Wei Liu and Sanjay Chawla. Class confidence weighted kNN algorithms for imbalanced data sets. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 345–356. Springer, 2011.

[41] Vincenzo Lomonaco, Lorenzo Pellegrini, Pau Rodriguez, Massimo Caccia, Qi She, Yu Chen, Quentin Jodelet, Ruiping Wang, Zheda Mai, David Vazquez, et al. CVPR 2020 continual learning in computer vision competition: Approaches, results, current challenges and future directions. *arXiv preprint arXiv:2009.09929*, 2020.

[42] Ilya Loshchilov and Frank Hutter. SGDR: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016.

[43] Inderjeet Mani and I Zhang. kNN approach to unbalanced data distributions: a case study involving information extraction. In *Proceedings of Workshop on Learning from Imbalanced Datasets*, volume 126. ICML United States, 2003.

[44] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Elsevier, 1989.

[45] Fei Mi, Lingjing Kong, Tao Lin, Kaicheng Yu, and Boi Faltings. Generalized class incremental learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*, pages 240–241, 2020.

[46] Jose G Moreno-Torres, Troy Raeder, Rocío Alaiz-Rodríguez, Nitesh V Chawla, and Francisco Herrera. A unifying view on dataset shift in classification. *Pattern Recognition*, 45(1):521–530, 2012.

[47] Behnam Neyshabur, Srinadh Bhojanapalli, David McAllester, and Nathan Srebro. Exploring generalization in deep learning. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 5949–5958, 2017.

[48] Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. Norm-based capacity control in neural networks. In *Conference on Learning Theory*, pages 1376–1401. PMLR, 2015.

[49] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11321–11329, 2019.

[50] Zhimao Peng, Zechao Li, Junge Zhang, Yan Li, Guo-Jun Qi, and Jinhui Tang. Few-shot image recognition with knowledge transfer. In *IEEE International Conference on Computer Vision (ICCV)*, pages 441–449, 2019.

[51] John Platt et al. Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods. *Advances in Large Margin Classifiers*, 10(3):61–74, 1999.

[52] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *European Conference on Computer Vision (ECCV)*, pages 524–540. Springer, 2020.

[53] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. iCaRL: Incremental classifier and representation learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2001–2010, 2017.

[54] Michael D Richard and Richard P Lippmann. Neural network classifiers estimate bayesian a posteriori probabilities. *Neural Computation*, 3(4):461–483, 1991.

[55] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision*, 115(3):211–252, 2015.

[56] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. MobileNetV2: Inverted residuals and linear bottlenecks. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4510–4520, 2018.

[57] Li Shen, Zhouchen Lin, and Qingming Huang. Relay backpropagation for effective learning of deep convolutional neural networks. In *European Conference on Computer Vision (ECCV)*, pages 467–482. Springer, 2016.

[58] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 2990–2999, 2017.

[59] Ray J Solomonoff. A system for incremental learning based on algorithmic probability. In *Proceedings of the Sixth Israeli Conference on Artificial Intelligence, Computer Vision and Pattern Recognition*, pages 515–527, 1989.

[60] Isaac Triguero, Joaquín Derrac, Salvador Garcia, and Francisco Herrera. A taxonomy and experimental study on prototype generation for nearest neighbor classification. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(1):86–100, 2011.

[61] Gido M van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019.

[62] Stefan Van Der Walt, S Chris Colbert, and Gael Varoquaux. The numpy array: a structure for efficient numerical computation. *Computing in Science & Engineering*, 13(2):22–30, 2011.

[63] Grant Van Horn, Oisin Mac Aodha, Yang Song, Yin Cui, Chen Sun, Alex Shepard, Hartwig Adam, Pietro Perona, and Serge Belongie. The inaturalist species classification and detection dataset. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 8769–8778, 2018.

[64] Byron C Wallace, Kevin Small, Carla E Brodley, and Thomas A Trikalinos. Class imbalance, redux. In *IEEE International Conference on Data Mining (ICDM)*, pages 754–763. IEEE, 2011.

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

[66] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, Zhengyou Zhang, and Yun Fu. Incremental classifier learning with generative adversarial networks. *arXiv preprint arXiv:1802.00853*, 2018.

[67] Hong-Ming Yang, Xu-Yao Zhang, Fei Yin, and Cheng-Lin Liu. Robust classification with convolutional prototype learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3474–3482, 2018.

[68] Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *IEEE International Conference on Computer Vision (ICCV)*, pages 6023–6032, 2019.

[69] Bianca Zadrozny and Charles Elkan. Obtaining calibrated probability estimates from decision trees and naive bayesian classifiers. In *International Conference on Machine Learning (ICML)*, volume 1, pages 609–616. Citeseer, 2001.

[70] Bianca Zadrozny and Charles Elkan. Transforming classifier scores into accurate multiclass probability estimates. In *Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 694–699, 2002.

[71] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13208–13217, 2020.

[72] Boyan Zhou, Quan Cui, Xiu-Shen Wei, and Zhao-Min Chen. BBN: Bilateral-branch network with cumulative learning for long-tailed visual recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 9719–9728, 2020.

[73] Zhi-Hua Zhou and Xu-Ying Liu. Training cost-sensitive neural networks with methods addressing the class imbalance problem. *IEEE Transactions on Knowledge and Data Engineering*, 18(1):63–77, 2005.

[74] Xiangxin Zhu, Dragomir Anguelov, and Deva Ramanan. Capturing long-tail distributions of object subcategories. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 915–922, 2014.