Generalized Class Incremental Learning

Fei Mi* Lingjing Kong* Tao Lin Kaicheng Yu Boi Faltings

École Polytechnique Fédérale de Lausanne (EPFL)

{fei.mi, lingjing.kong, tao.lin, kaicheng.yu, boi.faltings}@epfl.ch

Abstract

Many real-world machine learning systems require the ability to continually learn new knowledge. Class incremental learning receives increasing attention recently as a solution towards this goal. However, existing methods often introduce some assumptions to simplify the problem setting, which rarely holds in real-world scenarios. In this paper, we formulate a Generalized Class Incremental Learning (GCIL) framework to systematically alleviate these restrictions, and introduce several novel realistic incremental learning scenarios. In addition, we propose a simple yet effective method, namely ReMix*, which combines Exemplar Replay (*ER*) and* Mixup *to deal with different challenges in realistic GCIL setups. We demonstrate on CIFAR-100 that* ReMix *outperforms the state-of-the-art methods in different GCIL setups by significant margins without introducing additional computation cost.*

1. Introduction

The ability to continually acquire and accumulate new knowledge is a hallmark of general intelligence. Many real-world machine learning applications require learning from data that arrive continually over time [8]. For example, a robot needs to continually learn new objects it has never seen before without forgetting the ones it has already seen. To this end, Incremental Learning, a.k.a Continual Learning or Lifelong Learning, that learns from data arriving sequentially receives increasing attention. A widely studied setting in this field is on image classification tasks, namely Class Incremental Learning (CIL) [21, 5, 13, 3], where the data of new classes arrive *phase* by *phase*[1].

In CIL, a set of new classes need to be learned in each phase, as depicted in Figure 1 (upper row). The following three assumptions often exist: (i) the number of classes across different phases is fixed; (ii) classes appearing in earlier phases will not appear in later phases again; (iii) training samples are well-balanced across different classes in each phase. However, these assumptions rarely hold in real-world applications. For example, in the Internet of Things (IoT) era, a deployed object recognition model needs to incrementally and periodically refine its model through data collected from different input devices (e.g. surveillance cameras) [23]. The number of different objects in an update phase is hardly balanced (e.g. a "truck" might appear more often than a "taxi"), and objects might *reappear* continuously (e.g. a "truck" might appear in multiple update cycles).

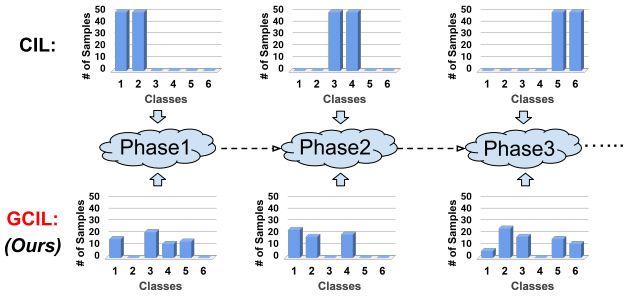

Figure 1: **Comparison between GCIL and CIL.** In each phase of CIL, the model observes a fixed number of balanced classes, and classes appeared in previous phases will not appear again. These restrictive assumptions are removed in our GCIL setting.

As shown in Figure 1 (lower row), we propose a framework, namely Generalized Class Incremental Learning (GCIL), that alleviates the limitations of CIL by allowing classes to appear in a realistic manner across multiple phases. Specifically, we characterize each phase with the following three quantities: the appearing class number, the appearing classes and the sample sizes of each appearing class. In our GCIL setting, these quantities are sampled from probabilistic distributions. Thus different realistic scenarios can be simulated by varying these distributions. Apart from the catastrophic forgetting challenge [20, 10] in previous CIL settings, we identify two other challenges, namely sample efficiency and imbalanced classes in real-

*Equal contribution

[1] [17, 18] refer to 'phase' as 'batch'. To avoid the confusion with the 'batch' in the model optimization stage, we use 'phase' instead.

istic GCIL settings. To this end, we propose a simple yet effective solution *ReMix* that combines Exemplar Replay(*ER* [21, 5]) and *Mixup* [26]. To the best of our knowledge, this is the first time that *Mixup* is adopted in incremental learning scenarios.

By simulating different realistic scenarios on CIFAR-100, we empirically show that (i) methods incorporating *ER* are superior to regularization methods; (ii) our proposed *ReMix* outperforms evaluated state-of-the-art methods by a margin of 5-10%, and it successfully deals with the two new challenges in GCIL. Altogether, our work is the first to generalize CIL to be realistic through a systematic probabilistic formulation, and the superior performance of *ReMix* can lead to interesting future explorations.

2. Related work

In general, two groups of incremental training protocols are considered in current class incremental learning (CIL) literature: (i) *Multi-epoch CIL*: new classes or patterns arrive phase by phase, and only data in the current phase are available for the model training. During training, data of each phase can be passed by multiple epochs [21, 5, 13, 24]; (ii) *Online CIL*: although training data still arrive sequentially, this setup only allows the model to be trained on each sample *once* [19, 6, 2].

The major challenge for multi-epoch CIL is catastrophic forgetting [20, 10], where optimization over classes of the current phase leads to performance degradation on classes in previous phases. Regularization [16, 14, 25, 4] and Exemplar Replay [21, 7, 5] are two major lines of research targeted at mitigating catastrophic forgetting. Regularization methods add specific regularization terms to consolidate knowledge from previous phases. In this direction, [21, 5, 24, 13, 27] adopt knowledge distillation [12] to penalize model logits changes on classes in previous phases. [14, 25, 1] measure the importance of each model parameter and penalize changes on parameters that are crucial to previous phases. Exemplar Replay methods store and replay past samples (a.k.a exemplars) selectively and periodically to prevent model forgetting classes or patterns in previous phases. As for exemplar selection, [21] adopt the *Herding* technique [22], which is based on the distance to the mean feature vector of each class and soon becomes popular [5, 24, 13, 27]. In order to maintain a feasible memory footprint, usually only a small number of exemplars are stored. This leads to the imbalanced class issue since the class sample size of the current phase is usually larger than that of the exemplars. To combat this imbalance, [5] fine-tune the output layer with a balanced subset of samples of all classes. [13, 27] propose to normalize parameters of the output layer; [24] uses another network to adjust the bias in the output layer. In this paper, we focus on the multi-epoch CIL setting and refer to it as CIL to avoid verbosity.

3. Generalized Class Incremental Learning

We firstly identify three key properties in GCIL, based on which we propose a probabilistic formulation.

3.1. Key Properties of GCIL

We denote the complete set of available classes as $\mathbb{S}$ with size n. The *sample sizes* (the number of samples) of different classes appearing in phase t are modeled as a random vector $\mathbf{C}_t \in \mathbb{R}^n$. Each entry $C_{t,i}$ is a random variable denoting the *sample size* of class i in this phase. The size of phase t is $n_t = ||\mathbf{C}_t||_1$. In the generalized form, $\mathbf{C}_t$ is generated from a phase-dependent distribution $\mathcal{H}(t)$

$$\mathbf{C}_t \sim \mathcal{H}(t)\,. \tag{1}$$

We summarize the following three properties that often hold in realistic incremental learning scenarios.

Property 1: The number of classes in a phase is not fixed. Suppose K_t is the number of classes in phase t, we have:

$$K_t = |\{i \in \mathbb{S} : C_{t,i} > 0\}| \sim \mathcal{K}(t)\,, \tag{2}$$

where $\mathcal{K}(t)$ is some phase-dependent distribution.

Property 2: Classes appearing in different phases could overlap. For two phases t and t', $t \neq t'$, we have:

$$P(\mathbf{C}_t \odot \mathbf{C}_{t'} \neq \mathbf{0}) > 0\,, \tag{3}$$

where $\odot$ denotes element-wise multiplication of two vectors with the same dimension.

Property 3: In one phase, sample sizes of different classes could be different. That is, for phase t, we have

$$\forall i, j \in \mathbb{N}, i \neq j, P(C_{t,i} \neq C_{t,j} | C_{t,i} \neq 0, C_{t,j} \neq 0) > 0\,. \tag{4}$$

However, these three properties are not satisfied in previous CIL setups, where old classes in earlier phases do not reappear, and a fixed number of new classes appear in each phase with balanced class sample sizes

3.2. Our GCIL Formulation

We consider a GCIL setting that satisfies the above three properties. A probabilistic formulation of $\mathcal{H}(t)$ can be formed through three steps:

$$\begin{aligned} K_t &\sim \mathcal{D}(t) \\ \mathbf{S}_t &\sim \mathcal{R}(\mathbf{W}_t^1, K_t) \\ \mathbf{C}_t &\sim \mathcal{M}(\mathbf{W}_t^2, \mathbf{S}_t) \end{aligned} \tag{5}$$

We explain these three steps separately below.

Class number K_t. The number of classes K_t to appear in phase t follows a phase-dependent discrete distribution

$\mathcal{D}(t)$. Therefore, K_t is a random quantity (c.f. a fixed constant in CIL setting) that satisfies *Property 1*. Different scenarios regarding the number of appearing classes in each phase can be simulated through different choices of $\mathcal{D}(t)$.

Appearing Classes $\mathbf{S}_t$. Classes appearing in phase t are modeled as a random vector $\mathbf{S}_t \in \mathbb{R}^n$. $\mathbf{S}_t$ is a binary indicator vector with ones corresponding to classes appearing in t. This vector is sampled from distribution $\mathcal{R}(\mathbf{W}_t^1, K_t)$ across phases such that *Property 2* is satisfied. Moreover, $\mathcal{R}$ depends on the class number K_t and a *class appearance weight vector* $\mathbf{W}_t^1 \in \mathbb{R}^n$. Each entry of $\mathbf{W}_t^1$ represents the appearing probability of the class in phase t. Classes with larger weights are more likely to appear in the phase. In Section 5, we choose sampling without replacement as a realization of $\mathcal{R}$.

Class sample sizes $\mathbf{C}_t$. The last step is to determine the sample size of each appearing class in $\mathbf{S}_t$, which is encoded as random vector $\mathbf{C}_t$. $\mathbf{C}_t$ follows a distribution $\mathcal{M}(\mathbf{W}_t^2, \mathbf{S}_t)$, which depends on the appearing class $\mathbf{S}_t$ and a *class sample size weight vector* $\mathbf{W}_t^2 \in \mathbb{R}^n$. $\mathbf{W}_t^2$ determines the sample size of each class appearing in phase t, and it can model different degrees of class imbalance within a phase. Therefor, *Property 3* is satisfied. We stress that $\mathbf{W}_t^2$ is intrinsically different from $\mathbf{W}_t^1$. For example, a class might appear frequently among different phases (i.e. with a large weight in $\mathbf{W}_t^1$) but it only appears with a small quantity per phase (i.e. with a small weight in $\mathbf{W}_t^2$). In Section 5, we choose multinomial distribution as a realization of $\mathcal{M}$.

With the above realistic GCIL formulation, two more challenges other than catastrophic forgetting need to be tackled. First, GCIL allows sample size of appearing classes to be much smaller than that in CIL to reflect potential data scarcity of some classes. Therefore, *sample efficiency* needs to be improved to learn from a limited amount of data. Second, GCIL allows classes to be imbalanced within a phase, therefore, the model needs to handle *imbalanced classes*.

4. Proposed Solution for GCIL – *ReMix*

We propose a simple yet effective method *ReMix* that combines Exemplar Replay and *Mixup*.

Exemplar Replay (*ER*) based on Herding. *ER* methods [21, 5, 13, 24] have shown great success in standard CIL settings to mitigate the catastrophic forgetting issue. *ER* stores a couple of exemplars for all experienced classes until the current phase. Exemplars are combined with data in the current phase to update the model in each phase. We adopt the *Herding* [22, 21] technique to select exemplars. For each class, *Herding* selects samples that best approximate the *average feature vector* over all training exemplars of this class till the current phase.

***Mixup*.** *ER* addresses catastrophic forgetting, but the issues of sample efficiency and imbalanced classes remain unsolved. Next, we introduce a data augmentation technique *Mixup* [26] as a complementary component on top of *ER* to address these two challenges.

The idea of *Mixup* is simple: it creates *virtual training samples* through a linear interpolation between raw training samples in order to learn smooth decision boundaries between all classes to improve the model generalization ability. Formally, a virtual training sample $(\tilde{x}, \tilde{y})$ is generated by a convex combination between a pair of raw samples (x_i, y_i) and (x_j, y_j) by:

$$\tilde{x} = \lambda x_i + (1-\lambda)x_j \, , \tilde{y} = \lambda y_i + (1-\lambda)y_j \, .$$

where $\lambda \sim Beta(\alpha, \alpha)$, with hyperparameter $\alpha \in (0, \infty)$.

***ReMix*.** We propose to use *Exemplar **Re**play* together with ***Mix**up*, referred to as *ReMix*, to deal with all three challenges of GCIL. In each incremental training phase, exemplars of different classes are selected using the *Herding* technique. Then *Mixup* is applied to the training mini-batches containing both samples in the current phase and exemplars as in *ER*. In Section 5.3, we show that *ReMix* significantly outperforms both *ER* and *Mixup* when they are used separately.

Three nice properties of *ReMix* are analyzed below. First, as a data augmentation method, *ReMix* generates virtual samples based on samples from both the current phase and the stored exemplars. Thus the limited number of exemplars in memory can be augmented to further mitigate the catastrophic forgetting challenge. Similarly, classes with insufficient samples can also be augmented to improve sample efficiency. Second, the regularization effect of *ReMix* helps to deal with imbalanced classes. It prevents the model from overfitting dominant classes in the current phase by smoothing decision boundaries among all classes. Last but not the least, *ReMix* can be easily applied to GCIL scenarios as it does not rely on any restrictive assumptions on the data distribution of incoming phases. Also, it introduces minimal computation overhead with neither extra training epochs nor extra data.

5. Experiments

In this section, we evaluate *ReMix* and a wide range of state-of-the-art methods in different GCIL setups.

5.1. Baseline Methods

- ***Finetune***: The model is updated with only data in the current phase without using exemplars.
- ***GEM*** [19]: For each update, current gradients are projected to a feasible region formed by exemplar gradients. A ring buffer stores 200 exemplars for each phase to keep a memory size equivalent to that in *ER*.

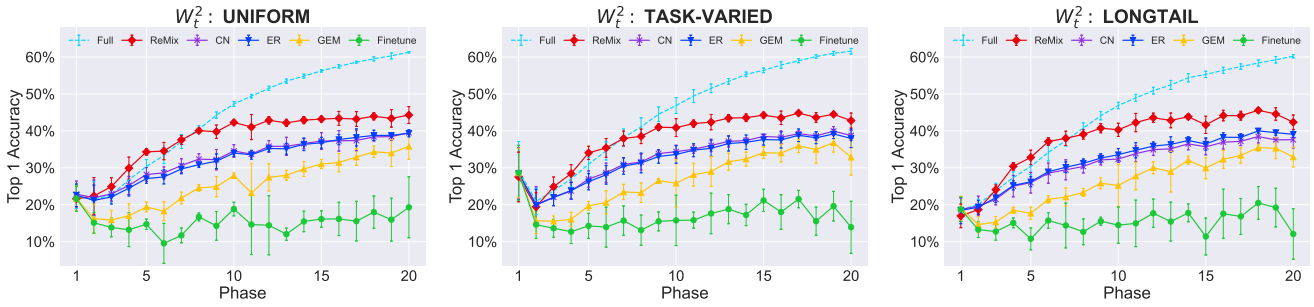

Figure 2: **Performances with varying $\mathbf{W}_t^2$ on *CIFAR-100*.** At each incremental training phase, mean top-1 accuracy and standard deviation averaged over 5 runs with different random seeds are plotted.

- ***ER*** [7, 21]: **E**xemplar **R**eplay updates the model with data in the current phase and exemplars in previous phases selected by *Herding*. For methods (*ER*, *CN*, *ReMix*) using *Herding*, 20 exemplars per class are stored.
- ***CN*** [13]: **C**osine **N**ormalization is applied to the output layer in *ER* to deal with imbalanced classes. Two other tricks used in [13] are not evaluated because they are not compatible with the GCIL setup.
- ***Full***: In each phase, the model is trained with data from the current phase and *all* previous phases. This is a common performance "upper bound" in CIL.

5.2. Dataset and Implementation Details

In our experiments, we use CIFAR-100 [15] dataset. 20 phases are tested with 1,000 images in each phase. At each incremental training phase, a 32-layer ResNet [11] is trained by stochastic gradient descent with 100 epochs. λ of *ReMix* is set to 1. The learning rate starts from 0.1 and is divided by 10 after 60 and 80 epochs; weight decay is 5e-4 and momentum is 0.9. Models are evaluated by *TOP-1 accuracy* on the balanced test set consisting of all classes that appeared so far.

We set $\mathcal{D}(t)$ as a uniform distribution $\mathcal{U}(1, 100)$, $\mathbf{W}_t^1$ as a uniform distribution over all classes. Three variations of $\mathbf{W}_t^2$ are tested: **UNIFORM**: $\mathbf{W}_t^2$ is a fixed uniform distribution overall all classes in $\mathbb{S}$. **TASK-VARIED**: $\mathbf{W}_t^2$ varies across different phases by adding independent Gaussian noises (0 mean and 20% of uniform class weight as standard deviation) to each class weight of UNIFORM. **LONGTAIL**: $\mathbf{W}_t^2$ is a fixed long-tailed distribution. The weight $W_{t,i}^2$ for class i in the long-tailed distribution is generated by an exponential function $W_{t,i}^2 = \mu^i$ [9]. Different μ's correspond to different degrees of class imbalance. In our setting, the largest weight is 5 times larger than the smallest.

5.3. Results on CIFAR-100

In Figure 2, we present the Top-1 accuracy of different methods at each phase under three GCIL setups. Several interesting observations can be noted: **First**, Exemplar Replay methods (*ER*, *CN*) based on Herding perform better than *GEM*. *CN* significantly outperforms *ER* in CIL setups [13, 24], nevertheless, it only achieves comparable performance to *ER* in realistic GCIL setups. **Second**, *ReMix* outperforms the state-of-the-art methods by large margins (5%-10%) in different GCIL setups. Specifically, *ReMix* shows multiple advantages: (i) better sample efficiency, as indicated by its superior performance over *Full* in early phases; and (ii) robust to imbalanced classes phenomenon under $\mathbf{W}_t^2$ = LONGTAIL. More detailed analyses of *ReMix* compared to *ER* are included in the Appendix.

ReMix	*ReMix-v1*	*ReMix-v2*	*Mixup*	*ER*
36.27%	34.52%	32.39%	15.93%	30.93%

Table 1: **Ablation study for *ReMix*.** Reported by Top-1 accuracy averaged over 20 phases when $\mathbf{W}_t^2$ =LONGTAIL.

Ablation Study for *ReMix*. In Table 1, three variants of *ReMix* are evaluated. The fact that *Mixup* (w/o exemplars) alone fails badly shows that exemplars are crucial for *ReMix*. In *ReMix-v1*, *Mixup* is only performed among exemplars, while data in the current phase are raw. In *ReMix-v2*, *Mixup* is only performed on data in the current phase, while exemplars are raw. Although *ReMix-v1* and *ReMix-v2* outperform *ER*, they are both inferior to *ReMix*. These experiments justify the importance of keeping exemplars and interpolating exemplars with data in the current phase.

6. Conclusion

In this paper, we revisit the oversimplified setup in the current class incremental learning research and propose a Generalized Class Incremental Learning (GCIL) framework. Moreover, we propose a simple yet effective method, *ReMix*, which consistently outperforms previous methods by significant margins across different scenarios. We hope our exploration of the realistic GCIL could motivate more research ideas in this direction.

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018. 2

[2] Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11254–11263, 2019. 2

[3] Eden Belouadah and Adrian Popescu. Il2m: Class incremental learning with dual memory. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 583–592, 2019. 1

[4] Yassine Benyahia, Kaicheng Yu, Kamil Bennani Smires, Martin Jaggi, Anthony C. Davison, Mathieu Salzmann, and Claudiu Musat. Overcoming multi-model forgetting. In Kamalika Chaudhuri and Ruslan Salakhutdinov, editors, *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 594–603, Long Beach, California, USA, 09–15 Jun 2019. PMLR. 2

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

[6] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations (ICLR)*, 2019. 2

[7] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. Continual learning with tiny episodic memories. *arXiv preprint arXiv:1902.10486*, 2019. 2, 4

[8] Zhiyuan Chen and Bing Liu. Lifelong machine learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 12(3):1–207, 2018. 1

[9] Yin Cui, Menglin Jia, Tsung-Yi Lin, Yang Song, and Serge Belongie. Class-balanced loss based on effective number of samples. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 9268–9277, 2019. 4

[10] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in Cognitive Sciences*, pages 128–135, 1999. 1, 2

[11] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778, 2016. 4

[12] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015. 2

[13] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Learning a unified classifier incrementally via rebalancing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 831–839, 2019. 1, 2, 3, 4

[14] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 2

[15] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 4

[16] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 2

[17] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. In *Conference on Robot Learning*, pages 17–26, 2017. 1

[18] Vincenzo Lomonaco, Davide Maltoni, and Lorenzo Pellegrini. Fine-grained continual learning. *arXiv preprint arXiv:1907.03799*, 2019. 1

[19] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017. 2, 3

[20] 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. 1, 2

[21] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2001–2010, 2017. 1, 2, 3, 4

[22] Max Welling. Herding dynamical weights to learn. In *Proceedings of the 26th International Conference on Machine Learning*, pages 1121–1128. ACM, 2009. 2, 3

[23] Zhenyu Wen, Renyu Yang, Peter Garraghan, Tao Lin, Jie Xu, and Michael Rovatsos. Fog orchestration for iot services: issues, challenges and directions. *IEEE Internet Computing*, 21(2):16–24, 2017. 1

[24] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning, 2019. 2, 3, 4

[25] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *Proceedings of the 34th International Conference on Machine Learning*, pages 3987–3995. JMLR. org, 2017. 2

[26] Hongyi Zhang, Moustapha Cisse, Yann N Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *International Conference on Learning Representations (International Conference on Learning Representations (ICLR))*, 2018. 2, 3

[27] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shutao Xia. Maintaining discrimination and fairness in class incremental learning. *arXiv preprint arXiv:1911.07053*, 2019. 2