Adapt Your Teacher: Improving Knowledge Distillation for Exemplar-free Continual Learning

Filip Szatkowski[*1,2], Mateusz Pyla[2,3,4], Marcin Przewięźlikowski[2,3,4],
Sebastian Cygert[2,5], Bartłomiej Twardowski[2,6,7], and Tomasz Trzciński[1,2,8]

[1]Warsaw University of Technology, [2]IDEAS NCBR, [3]Jagiellonian University, Faculty of Mathematics and Computer Science,
[4]Jagiellonian University, Doctoral School of Exact and Natural Sciences, [5]Gdańsk University of Technology,
[6]Autonomous University of Barcelona, [7]Computer Vision Center, [8]Tooploox

Abstract

In this work, we investigate exemplar-free class incremental learning (CIL) with knowledge distillation (KD) as a regularization strategy, aiming to prevent forgetting. KD-based methods are successfully used in CIL, but they often struggle to regularize the model without access to exemplars of the training data from previous tasks. Our analysis reveals that this issue originates from substantial representation shifts in the teacher network when dealing with out-of-distribution data. This causes large errors in the KD loss component, leading to performance degradation in CIL models. Inspired by recent test-time adaptation methods, we introduce Teacher Adaptation (TA), a method that concurrently updates the teacher and the main models during incremental training. Our method seamlessly integrates with KD-based CIL approaches and allows for consistent enhancement of their performance across multiple exemplar-free CIL benchmarks. The source code for our method is available at https://github.com/fszatkowski/cl-teacher-adaptation.

1. Introduction

Continual learning aims to create machine learning models capable of acquiring new knowledge and adapting to evolving data distributions over time. One of the most challenging continual learning scenarios is *class incremental learning* (CIL) [32,50], where the model is trained to classify objects incrementally from the sequence of tasks, without forgetting the previously learned ones.

A simple and effective method of reducing forgetting is by leveraging *exemplars* [6,21,37,41] of previously encountered training examples, *e.g.* by replaying them or using them for regularization. However, this approach presents challenges, particularly in terms of high storage needs and privacy concerns. These problems can affect edge devices, due to their limited storage capacity, and medical data, given their sensitive nature.

A common approach for exemplar-free CIL is knowledge distillation (KD), where the current model (student) is trained on the new data with a regularization term that minimizes the output difference with the previous model (teacher), which is kept frozen. This approach was introduced by LwF [29] and has been extended by many other methods. However,

---
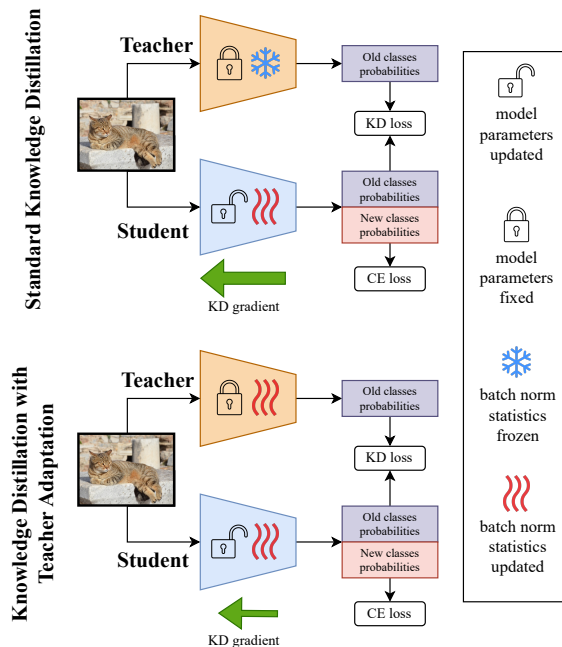[*]corresponding author, email: filip.szatkowski.dokt@pw.edu.pl

Figure 1. Enhancement of vanilla Knowledge Distillation approach used in Continual Learning with our method of Teacher Adaptation. When training the student model on the new task, we allow the teacher model to continuously update its batch normalization statistics, which reduces the divergence between the representations in both models. Our method leads to lower knowledge distillation loss and an overall more stable model.

most of these methods use exemplars, such as iCaRL [41], EEIL [8], LUCIR [18], PodNET [14], SSIL [2], or rely on external datasets [28,61].

Exemplar-free CIL still remains challenging [47] for KD methods due to the possibility of significant distribution drift in subsequent tasks. Such drift leads to large errors during training with KD loss, causing more undesired changes in the main model and harming the overall performance of the CIL training. This raises the question: *Can we adjust the teacher model to better transfer knowledge from earlier tasks?*

Motivated by the recent domain adaptation methods [45, 52], we examine the role of batch normalization statistics in CIL training. We conjecture that in standard KD methods, the KD loss between models with different normalization
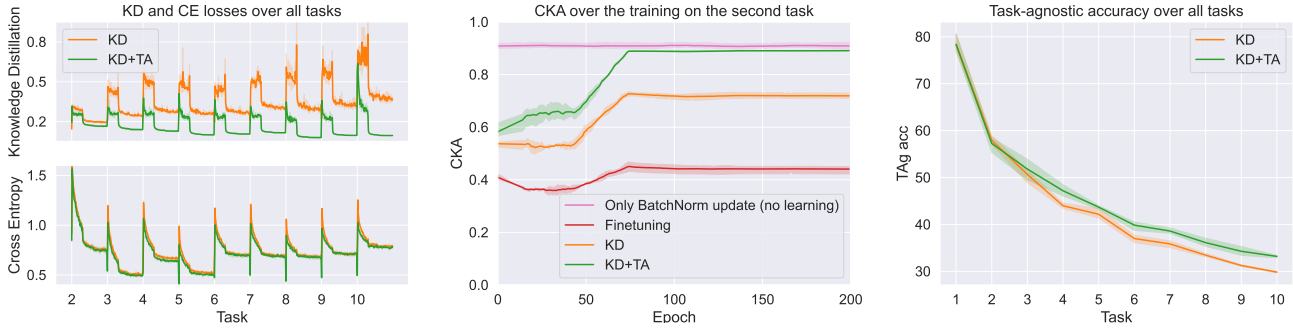
Figure 2. Applying our teacher adaptation (TA) method reduces knowledge distillation (KD) loss and improves stability throughout continual learning. (left) KD loss and cross-entropy (CE) loss of training the model with and without TA. Our method leads to more consistent representation, as visualized by the CKA [24] between the representations of the new data obtained in the teacher and student models while learning the second task (middle). KD with TA leads to better task-agnostic accuracy (right). We conduct the experiments on CIFAR100 split into 10 tasks.

statistics may introduce unwanted model updates due to the data distribution shifts. To avoid this, we propose to continuously adapt them to the new data for the teacher model while training the student.

We show that adapting the teacher's batch normalization statistics to the new task can significantly lower KD loss without affecting the CE loss, which reduces changes in the model's representations (Figure 2). We note that, while the idea of changing the teacher model was explored in the standard KD settings [30, 64], our approach is the first application of this idea to CIL scenario, where the teacher and the model are trained on non-overlapping data. Moreover, our method works differently by exploiting the batch normalization statistics. We apply our method on top of different distillation strategies in CIL and show consistent improvements across various settings.

In summary, we make the following contributions:

1. We revisit the KD-based class-incremental learning (CIL) framework and study the negative impact of regularization using out-of-distribution data. We are the first to highlight the need for adjusting the teacher model in an exemplar-free situation, where it is usually kept frozen.

2. We propose a simple yet highly effective technique called Teacher Adaptation (TA), that enhances KD for exemplar-free CIL.

3. Through extensive experiments, we demonstrate that TA can be seamlessly integrated with various KD approaches, leading to significant improvements over the baselines across a wide range of continual learning scenarios for various datasets. We show that those improvements hold even when using pretrained models or in the presence of substantial distributional shifts between consecutive domains.

2. Related works

Class Incremental Learning (CIL) [32, 50] is a subfield of continual learning, where the aim is to learn incrementally from a stream of tasks, without the task identifier. There exist several families of approaches to CIL:

Memory-based methods store either exemplars or features from the previous tasks in the buffer [6, 21, 37, 41] and use them during training the new task to consolidate previously learned knowledge. Those methods usually perform well, but their practical applications are limited due to privacy concerns and memory requirements that arise when storing the data. Architectural approaches focus on modifying the structure of the model, often by allocating certain parameters to corresponding tasks [53, 54]. Finally, regularization-based methods aim to preserve the knowledge in the network by imposing constraints on the changes in model weights [23] or activations during learning the new task [29]. Many CIL methods often also combine the above approaches [8, 42, 55], for instance using both memory and regularizaion [2, 29, 41].

Regularization methods for continual learning offer a way to prevent forgetting with constant memory usage and no privacy issues. There are two main types of regularization methods: (i) parameter regularization and (ii) functional regularization. The first type of methods regularizes the model weights, for example using the Fisher Information Matrix [23], synaptic saliency [59] or the gradient inspection [3]. On the other hand, functional regularization methods employ knowledge distillation (KD) techniques to regularize model activations. KD was originally proposed by Hinton et al. [17] to transfer the knowledge from a larger model to a smaller one. In CL, KD was first applied in Learning without Forgetting (LwF) [29], where the model is discouraged from drifting too far from the model from previous tasks. We describe KD methods in detail in Section 3.1.

Functional regularization has been widely used in

CL since the introduction of LwF [18, 37, 40, 41] and numerous variants of KD have been proposed. Particularly, SSIL [2] uses task-wise knowledge distillation, while POD-NET [14] regularizes using spatial-based distillation loss applied throughout the model.

Multi-level knowledge distillation [13] uses the current model to distill the knowledge from pruned snapshots of all previous models, while ANCL [22] distills simultaneously from the previous task model and the model learned specifically for the new task. Moreover, DMC [60] uses knowledge distillation on an auxiliary dataset to consolidate the knowledge from previous tasks.

However, most of those methods use a memory buffer and their performance depends on it heavily, which makes them impractical for exemplar-free settings. Recently, several works explored the idea of modifying the teacher model through meta-learning for better knowledge transfer in standard KD setting [30, 64]. Similarly, works on KD [16] suggest that updating the normalization statistics of the teacher model on the data used to train the student improves the performance. However, to our knowledge, our method is the first one that explores different approaches to teacher adaptation, such as updating the normalization statistics, in the context of CIL.

Batch Normalization (BN) [20] is widely used in deep learning models, but can be problematic for settings where the data distribution changes over time. Alternative approaches such as LayerNorm [5] or GroupNorm [57] do not rely on the batch-wise statistics, but directly replacing BN layers with them was shown to often decrease the performance of the models. Several domain adaptation methods achieve domain transfer through the use of normalization statistics [45, 52]. Recent work on efficient finetuning of large language models using only normalization layers [38] also suggests that the normalization layers play a crucial role in training deep neural networks. In CL, it was shown that BN can cause a discrepancy between the training and testing phases of BN, as the testing data is normalized using the statistics biased towards the current task, which results in higher forgetting of older tasks [44]. Several works have attempted to address this issue by CL-specific modifications to BN [9, 36]. However, those approaches are not suited for exemplar-free settings.

3. Method

In class-incremental learning setup, the model learns tasks sequentially. Each task contains several classes which are disjoint with the classes in other tasks. During training task t, we only have access to the data D^t from task t which contains images $\mathbf{x}_i \in X^t$ with class labels $y_i \in C^t$. Thus an incremental learning problem $\mathcal{T}$ with n tasks can be formulated as: $\mathcal{T} = \left\{ D^1, D^2, ..., D^t, ..., D^n \right\}$, where after training n tasks we evaluate the model on all classes $C^1 \cup$

$\ldots \cup C^n$, without knowing the task label at inference time (this is different than task-incremental learning, where task id can be used).

Below, we first introduce standard KD-based methods for exemplar-free CIL. Then we outline a problem of diverging batch normalization statistics between the teacher and student model caused by the shifts in training data between subsequent tasks. Finally, we propose to address this issue with a method that we call *Teacher Adaptation* - a simple, yet effective solution that allows the teacher model to continuously update its normalization statistics alongside the student when training on the new data. The method in comparison to standard LwF is presented in Figure 1.

3.1. Knowledge Distillation in Continual Learning

Knowledge distillation is one of the most popular techniques employed to reduce forgetting between subsequent tasks in incremental learning. Continual learning methods that use knowledge distillation save the model Θ_t (*teacher*) trained after each task t and use it during learning the model Θ_{t+1} (*student*) on new task $t + 1$. The learning objective for task $t + 1$ then becomes:

$$L = L_{CE} + \lambda L_{KD}, \tag{1}$$

where L_{CE} is the cross-entropy loss for classification on new data, L_{KD} is the knowledge distillation loss computed using Θ_t and Θ_{t+1}, and λ is the coefficient that controls the trade-off between stability and plasticity. The general formula for knowledge distillation loss can include either output from the final layer of the model [2, 29, 41], or also representations from intermediate model layers [12, 14]. In practice, most exemplar-free methods that use knowledge distillation compute knowledge distillation loss using only the final layer outputs, and various methods that use intermediate representations usually only perform well with exemplars [47].

Multiple variants of knowledge distillation loss were proposed for continual learning. In exemplar-free CIL, KD loss is usually computed with the logits y_o, $\hat{y_o}$ returned by the current and previous models respectively. Following Li et al. [29], we denote that the loss uses logits corresponding to previously seen classes with a subscript o. Ahn et al. [2] classify KD methods into general KD (GKD), which aggregates together logits belonging to the classes from all the previous tasks, and task-wise KD (TKD), which treats classes within each task separately.

GKD loss appears in several works [28, 56, 62] and usually uses cross-entropy:

$$\mathcal{L}_{GKD}(\mathbf{y}_o, \hat{\mathbf{y}}_o) = -\sum_{i=1}^{|C_t|} p_o^{(i)} \log \hat{p}_o^{(i)}, \tag{2}$$

where $p_o^{(i)}$ is the probability of the i-th class and $|C_t|$ is the number of classes learned by previous model Θ_t. Probabilities $p_o^{(i)}$, $\hat{p}_o^{(i)}$ are computed with temperature parameter T as follows:

$$p_o^{(i)} = \frac{e^{y_o/T}}{\sum_j e^{y_o/T}}, \quad \hat{p}_o^{(i)} = \frac{e^{\hat{y}_o/T}}{\sum_j e^{\hat{y}_o/T}} \tag{3}$$

Comparatively, TKD loss, which was also used in several works [2, 8, 29], sums of the separately computed losses for each task:

$$\mathcal{L}_{TKD}(\mathbf{y}_o, \hat{\mathbf{y}}_o) = \sum_{i=1}^{t} \mathcal{D}_{KL}(p_o^{(i)} \log \hat{p}_o^{(i)}), \tag{4}$$

where $\mathcal{D}_{KL}$ is Kullback–Leibler divergence and $p_o^{(i)}$, $\hat{p}_o^{(i)}$ are computed task-wise across the classes that belong to task i as in Equation (3).

Rebuffi et al. [41] proposed another distinct variant of KD for multi-class incremental learning, where the loss is computed element-wise for each class (MKD):

$$\mathcal{L}_{MKD}(\mathbf{y}_o, \hat{\mathbf{y}}_o) = -\sum_{i=1}^{|C_t|} \sigma(y_o^i) \log \sigma(\hat{y}_o^i), \tag{5}$$

where σ is a sigmoid function.

Additionally, a more recent KD-based method, Auxiliary Network Continual Learning (ANCL) [22], explores the idea of multi-teacher KD for continual learning. ANCL trains an auxiliary network trained only for the current task and combines standard GKD loss with the KD loss computed between outputs for the current task and outputs of this auxiliary network.

In this work, we investigate the aforementioned KD techniques (GKD, MKD, TKD, ANCL).

3.2. Teacher Adaptation

Most models used in class incremental learning for vision tasks are convolutional neural networks such as ResNet [15]. Those models typically use batch normalization layers and keep the parameters and statistics of those layers in the teacher model Θ_t fixed during learning Θ_{t+1}. However, with the changing distribution of the data in the new task, batch normalization statistics in the student and teacher models quickly diverge, which leads to higher KD loss. Gradient updates in this case not only regularize the model towards the stable performance on previous tasks but also compensate for the changes in normalization statistics, which needlessly overwrites the knowledge stored in the model and harms the learning process.

Inspired by test-time adaptation methods [52], we propose to reduce this negative interference with a simple method that we label *Teacher Adaptation (TA)*. Our method updates

batch normalization statistics of both models simultaneously on new data while learning the new task. As shown in Figure 2, it allows for significantly reduced KD loss over learning from sequential tasks in CIL, which improves the overall model stability. We provide additional analysis on why TA improves learning with KD in Section 4.4 and in the Appendix.

4. Experiments

4.1. Experimental setup

We conduct experiments on common continual learning benchmarks, such as CIFAR100 [26], TinyImageNet200 [1] and ImageNet-Subset [11]. We measure models' adaptability to large shifts in data distributions on DomainNet [35] dataset. Additionally, following FACIL [32] we construct fine-grained classification benchmark using Oxford Flowers [34], MIT Indoor Scenes [39], CUB-200-2011 Birds [51], Stanford Cars [25], FGVC Aircraft [31], and Stanford Actions [58]. Finally, we also introduce a corrupted CIFAR100 setting where data in every other task contains noise of varying severity, which allows us to measure the impact of TA under varying and controllable degrees of data shift.

We create CIL scenarios by splitting the classes in each dataset into disjoint tasks. We experiment with two particular types of settings: the first type of setting is built by splitting the classes in the dataset into tasks containing an equal number of classes, while the other simulates pretaining the network and uses half of the classes as a larger first task, with subsequent tasks composed of the evenly split remaining classes.

For all experiments, we use FACIL framework provided by Masana et al. [32]. For experiments on CIFAR100, we keep the class order from iCaRL [41] and we use ResNet32 [15]. For TinyImageNet, following [22], we rescale images to 32x32 pixels and also use ResNet32. For the other datasets, we use ResNet18 [15]. We always use the same hyperparameters for all variants within the single KD method unless stated otherwise, we report the results averaged over three runs with different random seeds.

In every setup, we train the network on each new task for 200 epochs with batch size 128. We use SGD optimizer without momentum or weight decay, with a learning rate scheduler proposed by Zhou et al. [63], where the initial learning rate of 0.1 is decreased 10x after 60th, 120th and 160th epoch. For experiments conducted on CIFAR100 and TinyImageNet200 in Table 1 we also employ a warmup phase [27] for the new classification head. In the Appendix, we provide the ablation study of the warmup with different benchmarks, alongside the details of its implementation and discussion on the method. Additionally, we provide the evaluation of our method with different model architectures and batch sizes.

| | | Equal split of classes | | | | Start from half of the classes | | | |
| | | 10 tasks | | 20 tasks | | 11 tasks | | 26 tasks | |
		$Acc_{Inc}\uparrow$	$Forg_{Inc}\downarrow$	$Acc_{Inc}\uparrow$	$Forg_{Inc}\downarrow$	$Acc_{Inc}\uparrow$	$Forg_{Inc}\downarrow$	$Acc_{Inc}\uparrow$	$Forg_{Inc}\downarrow$
a) CIFAR100	GKD	42.52±0.76	22.26±0.31	31.89±0.45	34.68±1.87	41.69±1.18	18.09±0.88	17.64±0.93	9.67±0.26
	+*ours*	**45.25±1.02**	**19.87±0.34**	**37.11±0.64**	**24.87±1.04**	**46.27±1.09**	**13.98±0.98**	**26.15±0.94**	**8.73±0.85**
	MKD	39.36±0.70	42.74±0.52	32.89±0.42	32.01±1.36	41.04±0.93	15.37±0.33	19.14±1.36	**8.76±0.35**
	+*ours*	**44.85±0.80**	**30.08±0.57**	**36.79±0.70**	**21.84±0.33**	**44.19±1.17**	**12.56±0.53**	**26.10±0.75**	9.17±0.20
	TKD	43.74±0.84	23.65±0.79	34.58±0.34	21.13±1.17	40.44±1.40	12.20±0.46	14.64±0.33	**6.02±0.54**
	+*ours*	**46.21±0.86**	**20.45±0.57**	**36.26±0.71**	**17.01±0.89**	**44.22±1.08**	**11.79±0.57**	**22.00±0.97**	9.36±0.68
	ANCL	43.15±0.49	32.78±1.52	34.32±0.41	36.74±1.38	45.16±0.32	20.21±0.30	21.84±1.33	11.79±0.51
	+*ours*	**46.73±0.20**	**26.86±1.17**	**38.48±0.94**	**27.99±1.20**	**48.25±0.33**	**16.11±0.19**	**29.67±1.13**	**10.98±0.56**
b) TinyImageNet200	GKD	32.12±0.57	21.21±1.31	25.75±0.65	27.55±1.81	34.32±1.96	11.41±2.80	23.30±1.59	12.63±2.21
	+*ours*	**33.90±0.78**	**17.70±1.48**	**27.85±0.90**	**21.65±1.43**	**37.50±1.84**	**10.14±1.10**	**28.82±1.95**	**11.44±0.61**
	MKD	31.04±1.00	16.90±0.75	25.22±0.88	25.80±2.46	**33.75±2.11**	10.30±1.63	23.42±1.95	10.41±1.97
	+*ours*	**32.22±0.95**	**11.94±1.07**	**27.39±1.33**	**16.74±1.52**	32.99±1.98	**6.42±0.80**	**25.75±1.74**	**8.35±0.82**
	TKD	33.15±0.45	21.05±0.39	27.29±0.76	23.20±2.46	37.31±1.36	11.83±1.31	23.94±2.14	10.20±1.09
	+*ours*	**34.58±0.96**	**17.27±0.54**	**28.71±1.06**	**17.37±1.78**	**38.41±1.52**	**9.16±0.62**	**28.10±1.97**	**9.56±0.65**
	ANCL	32.84±0.78	27.24±1.47	26.98±0.60	32.45±1.18	37.74±0.60	17.28±2.59	27.95±1.98	20.91±0.92
	+*ours*	**34.59±0.75**	**23.64±0.76**	**29.18±0.71**	**26.50±1.34**	**40.10±1.39**	**15.12±1.01**	**32.60±1.45**	**17.94±0.39**
c) ImageNet100	GKD	54.62±0.52	25.95±0.11	42.82±0.58	35.39±0.88	**57.94±0.90**	**14.47±0.83**	21.91±0.06	**9.29±0.69**
	+*ours*	**55.82±0.61**	**20.52±0.24**	**45.88±0.79**	**23.25±0.62**	57.18±0.45	17.24±0.39	**22.31±0.64**	11.28±0.98
	MKD	54.01±0.01	28.19±0.61	43.39±0.66	34.25±0.81	**56.18±0.90**	14.94±0.17	**26.07±0.29**	16.00±0.06
	+*ours*	**56.02±0.20**	**18.60±0.76**	**46.18±0.54**	**19.14±0.79**	52.05±0.24	**14.23±0.46**	22.25±0.13	**11.94±1.23**
	TKD	55.70±0.49	23.55±0.35	44.75±0.28	32.16±0.14	**54.72±0.86**	**10.16±0.34**	19.32±0.23	**9.67±0.61**
	+*ours*	**56.23±0.70**	**18.09±0.26**	**46.45±0.42**	**19.55±0.30**	53.85±0.39	13.15±0.16	**22.55±0.83**	9.96±0.28
	ANCL	55.81±0.41	27.13±0.50	44.94±0.63	34.24±1.14	**60.97±0.62**	**15.72±0.14**	29.19±0.76	16.12±0.27
	+*ours*	**57.41±0.22**	**21.54±0.50**	**47.49±0.35**	**22.61±0.40**	59.22±0.26	17.34±0.53	**29.64±0.50**	**14.67±0.84**

Table 1. Standard Knowledge Distillation (KD) techniques with and without our Teacher Adaptation (TA) method on different splits of a) CIFAR100, b) TinyImageNet200 and c) ImageNet100. TA is generally beneficial to the CIL process, and the improvements occur most consistently in scenarios with a long sequence of equally sized tasks, where the initial model learns a weaker feature extractor.

Evaluation metrics. The average incremental accuracy at task k is defined as $A_k = \frac{1}{k}\sum_{j=1}^{k} a_{k,j}$, where $a_{k,j} \in [0,1]$ be the accuracy of the j-th task ($j \leq k$) after training the network sequentially for k tasks [4]. Overall average incremental accuracy Acc_{Inc} is the mean value from all tasks. We also report *average forgetting* as defined in [10], while the $Forg_{Inc}$ is similarly the mean value from all tasks. We provide results with additional metrics such as final accuracy Acc_{Final} and final forgetting $Forg_{Final}$ in the Appendix.

4.2. Standard CIL benchmarks

We evaluate knowledge distillation approaches described in Section 3.1 on the standard CIL benchmarks CIFAR100, TinyImageNet200 and ImageNet100, using different class splits. We present the results in Table 1 a), b) and c) respectively. We also provide results for more settings and ablation

study of our method on those datasets in the Appendix.

In most settings, we observe that our method improves upon the baseline knowledge distillation. We notice that the improvement TA is generally more significant in settings with a larger number of tasks and an equal split of the classes. In settings with half the classes presented in the first task, the gains from TA are sometimes not that visible, as in this case the initial model already learns a good feature extractor, and the distribution of its normalization statistics after the first task is a better approximation of the statistics for the whole dataset. TA sometimes underperforms with MKD, which might be caused by the fact that the loss formula of MKD uses sigmoid function, and the differences between the probabilities for KD loss are insignificant if the values of logits are not small and centered around zero, which is not guaranteed without imposing additional learning constraints.

	6 tasks, 20 classes each		12 tasks, 10 classes each		24 tasks, 5 classes each	
	$Acc_{Inc} \uparrow$	$Forg_{Inc} \downarrow$	$Acc_{Inc} \uparrow$	$Forg_{Inc} \downarrow$	$Acc_{Inc} \uparrow$	$Forg_{Inc} \downarrow$
GKD	56.03±3.75	**30.23±0.72**	44.03±1.53	42.70±2.32	30.14±4.00	51.95±2.28
+*ours*	**57.24±1.79**	33.12±1.76	**50.80±1.27**	**37.58±3.01**	**42.86±1.82**	**39.01±1.55**
MKD	**60.12±1.59**	**26.42±1.60**	45.19±2.91	43.81±1.56	32.61±0.55	55.91±1.77
+*ours*	55.77±2.23	31.43±2.28	**51.16±1.84**	**34.89±2.60**	**45.49±1.82**	**34.48±0.89**
TKD	56.69±3.36	33.86±1.36	46.28±1.42	43.38±0.82	32.17±2.71	51.72±1.95
+*ours*	**57.99±1.88**	**33.79±1.80**	**51.66±1.63**	**35.43±2.75**	**43.95±2.60**	**33.28±1.86**

Table 2. Average task-agnostic accuracy and forgetting for KD-based CL methods on fine-grained classification datasets.

	6 tasks		6 tasks, pretrained		12 tasks		12 tasks, pretrained	
	$Acc_{Inc} \uparrow$	$Forg_{Inc} \downarrow$	$Acc_{Inc} \uparrow$	$Forg_{Inc} \downarrow$	$Acc_{Inc} \uparrow$	$Forg_{Inc} \downarrow$	$Acc_{Inc} \uparrow$	$Forg_{Inc} \downarrow$
GKD	18.63±0.27	**23.27±0.26**	43.27±0.10	36.83±0.88	14.45±0.25	**29.04±0.37**	35.98±0.96	43.00±1.66
+*ours*	**19.55±0.42**	27.22±0.21	**43.52±0.17**	**34.98±0.19**	**16.25±0.46**	33.03±0.19	**38.89±0.52**	**41.10±0.42**
TKD	19.12±0.26	**26.66±0.38**	42.42±0.10	**40.83±1.11**	16.31±0.55	37.32±0.87	38.15±0.22	42.28±0.50
+*ours*	**19.57±0.10**	29.96±0.05	**42.75±0.14**	41.79±0.65	**16.74±0.55**	**33.32±0.47**	**39.06±0.33**	**40.19±0.85**
MKD	**18.74±0.52**	**19.10±0.40**	**45.70±0.30**	**27.48±0.60**	13.45±0.53	**27.23±0.98**	**39.14±0.21**	36.53±1.00
+*ours*	18.04±0.16	22.70±0.25	42.91±0.04	29.43±0.04	**15.30±0.35**	28.71±0.26	37.84±0.35	**34.09±0.39**
ANCL	19.58±0.46	**25.63±0.20**	**42.90±0.84**	37.28±1.86	14.82±0.41	33.46±0.40	33.34±0.55	49.05±0.65
+*ours*	**20.34±0.40**	30.73±0.44	42.67±0.51	38.56±1.24	**17.19±0.06**	37.40±0.38	**35.81±0.18**	**43.84±0.23**

Table 3. Average task-agnostic accuracy and forgetting for KD and TA under significant semantic drift on DomainNet. We test scenarios with 6 tasks of 50 classes and 12 tasks of 25 classes, both when training from scratch and starting from pretrained model. Aside from MKD, TA generally leads to better results.

4.3. TA under severe distribution shifts

Motivated by the continual learning settings in which the data distribution changes significantly across the tasks, we conduct a series of experiments to empirically verify the benefits of our method.

4.3.1 Fine-grained classification datasets

We evaluate TA on fine-grained classification tasks using six datasets: Stanford Actions, FGVC Aircraft, Stanford Cars, CUB-200-2011 Birds, MIT Indoor Scenes and Oxford Flowers. We create CIL tasks by randomly sampling a subset of classes from each dataset, in the above-mentioned order. We sample the classes without replacement, and to obtain the settings with 12 or 24 tasks we repeat the procedure. For this set of experiments, we start from ResNet18 checkpoint pretrained on ImageNet.

We conduct experiments using splits of 24 tasks with 5 classes each, 12 tasks with 10 classes each, and 6 tasks with 20 classes each. We show the results in Table 2. Consistently with the results from Section 4.2, our method generally improves upon the base KD, with the improvements being more visible on the longer tasks.

We omit ANCL method from our analysis, as we were unable to obtain sufficiently good results with its official implementation. We provide the results of additional experiments conducted with reverse order of datasets and with the full datasets used as a single task in the Appendix.

4.3.2 Large domain shifts with DomainNet

To verify the effectiveness of teacher adaptation for continual learners under significant data distribution shifts, we use DomainNet [35] as our evaluation dataset. DomainNet consists of images from 6 domains and 345 classes. We select the first 50 classes and create each task from a different domain, resulting in more severe data drift between tasks in CIL. This allows us to measure how well the models can adapt to new data distributions. We use ResNet18 and compare two settings: training from scratch and from starting from the model pretrained on ImageNet. Table 3 shows the results of our experiments. Consistently with the results from previous Sections, we find that, aside from MKD, TA generally performs better than the baselines, and the differences are more visible when training on 12 tasks, where the model is exposed to more changes in the data distribution.
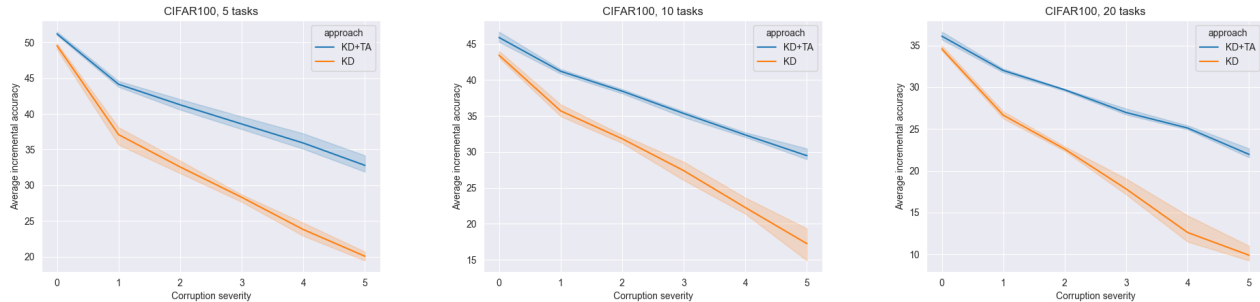
Figure 3. Average incremental accuracy for standard KD and our method of TA under varying strength of data shift on splits of CIFAR100. As the noise strengthens, the gap between TA and standard KD widens, indicating that our method leads to more robust learning in case of data shifts. We obtain data shifts by adding noise of varying strength to every other task, using the Gaussian noise and noise severity scale proposed by Michaelis et al. [33].

4.3.3 Varying the strength of the distribution shift

We create CIL settings with controllable levels of data distribution shift between subsequent tasks by corrupting every other task. We split CIFAR100 into 5, 10, and 20 tasks of equal size and add Gaussian noise to every other task, so that in subsequent tasks the data distribution changes from clean to noisy or vice versa. We obtain varying strength of distribution shift by using different levels of noise severity, following the methodology of Michaelis et al. [33].

We show the results of this experiment in Figure 3. We see that as the noise severity increases, the gap between standard KD and TA widens, indicating that our method is better suited to more challenging scenarios of learning under extreme data distribution shifts.

4.4. Detailed analysis

4.4.1 Alternatives to batch normalization

We conduct a series of ablation experiments on CIFAR100 split into 10 tasks to justify the validity of our method over other potential solutions for adaptation of batch normalization layers. The results of those experiments are shown in Table 4. We compare the following settings: 1) standard training with batch normalization statistics from the previous task fixed in the teacher model, but updated in the student model, 2) batch normalization layers removed, 3) batch normalization statistics fixed in both models after learning the first task, 4) batch normalization layers replaced with LayerNorm [5] or 5) GroupNorm [57] layers, and finally 6) our solution of Teacher Adaptation.

Fixing or removing BatchNorms leads to unstable training. This can be partially fixed by setting a high gradient clipping value or lowering the lambda parameter, but both solutions lead to much worse network performance. Different normalization layers enable stable training, but ultimately converge to much worse solutions than the network with BatchNorm. Our solution is the only one that improves over

$clip = 100$	$\lambda = 5$		$\lambda = 10$	
	$Acc_{Final} \uparrow$	$Acc_{Inc} \uparrow$	$Acc_{Final} \uparrow$	$Acc_{Inc} \uparrow$
1) **GKD**	25.47±0.57	41.59±0.32	27.96±0.34	42.28±0.67
2) **-BN**	0.33±1.15	2.01±2.67	0.33±1.15	2.85±3.81
3) **fix BN**	-	-	-	-
4) **-BN +LN**	21.94±0.95	34.7±0.48	22.76±1.05	34.48±0.15
5) **-BN +GN**	21.92±0.46	32.15±0.16	22.01±0.82	31.71±0.35
6) **+TA**	**31.39±0.17**	**44.98±0.38**	**31.85±0.10**	**44.06±0.69**

$clip = 1$	$\lambda = 5$		$\lambda = 10$	
	$Acc_{Final} \uparrow$	$Acc_{Inc} \uparrow$	$Acc_{Final} \uparrow$	$Acc_{Inc} \uparrow$
1) **GKD**	20.80±0.56	34.28±0.24	27.96±0.34	42.28±0.67
2) **-BN**	19.47±0.18	29.83±0.53	0.33±1.15	2.85±3.81
3) **fix BN**	20.21±0.31	32.07±0.20		-
4) **-BN +LN**	18.49±1.41	30.39±0.72	22.76±1.05	34.48±0.15
5) **-BN +GN**	16.17±0.89	32.15±0.16	15.73±1.01	25.07±1.10
6) **+TA**	**24.19±0.90**	**36.13±0.24**	**31.85±0.10**	**44.06±0.69**

Table 4. Results for different solutions to the problem of diverging batch normalization layers when using knowledge distillation in continual learning. We use GKD with different λ and gradient clipping values. We compare the baseline with variants without batch normalization layers, with batch normalization statistics fixed after the first task and with batch normalization layers replaced with LayerNorm or GroupNorm. "-" indicates that training crashes due to instability. TA is the only solution that improves upon the baseline.

different values of λ and does not require controlling the gradients by clipping the high values.

4.4.2 Alternative methods of teacher adaptation.

We study alternative methods of adapting the teacher model and try pretraining (P) or continuously training (CT) the teacher model. For pertaining, we train the teacher on the new data in isolation for a few epochs before the training of the main model. During continuous training, we update the teacher alongside the main model using the same batches

of new data. With both approaches, we set a lower learning rate for the teacher. We conduct those experiments by training either the full teacher model (FM) or only its batch normalization layers (BN). Finally, to isolate the impact of changing batch normalization statistics and training model parameters, we repeat all the experiments with fixed batch normalization statistics ($fix\ BN$).

We conduct our experiment on CIFAR100 split into 10 tasks and present the results in Table 5. Alternative solutions perform within the standard deviation of TA with tuned hyperparameters, but the values of the hyperparameters for those models (described in the Appendix) are also very small, indicating that the teacher model doesn't change much. Upon closer study, we find that it's mostly batch normalization statistics that change throughout the training. Therefore, knowing that other successful methods from test-time adaptation [52] use a similar approach, we continue with Teacher Adaptation based on batch normalization layers, as it does not require any hyperparameter tuning or additional pretraining epochs.

Method	$Acc_{Final}\uparrow$	$Acc_{Inc}\uparrow$	$Forg_{Final}\downarrow$	$Forg_{Inc}\downarrow$
Base	27.53±0.15	42.22±0.38	31.28±1.64	23.11±1.58
P-FM	31.54±0.67	43.46±0.72	24.18±1.17	20.80±1.51
+fix BN	28.02±0.60	42.33±0.53	29.91±1.27	22.66±0.95
P-BN	31.16±0.54	43.64±0.77	24.44±0.96	20.13±0.75
+fix BN	27.62±0.48	42.12±0.38	29.95±1.64	22.50±0.95
CT-FM	31.37±0.94	43.38±0.77	24.34±1.37	20.93±1.58
+fix BN	28.17±0.49	42.29±0.42	29.79±1.02	22.55±0.67
CT-BN	31.35±0.63	43.69±0.76	24.29±0.61	20.23±0.59
+fix BN	27.33±0.50	42.09±0.45	30.20±1.73	22.50±0.85
TA	**32.15±0.12**	**44.31±0.26**	**23.55±0.51**	**19.85±0.93**

Table 5. Ablation study of different ways to adapt the teacher model. Our method achieves the best results while requiring no additional hyperparameters. We try teacher adaptation during pretraining (P) and continuous training (CT). We train either full model (FM) or only batch normalization layers (BN). *fix BN* indicates fixed BN statistics.

5. Conclusions

We propose Teacher Adaptation, a simple yet effective method to improve the performance of knowledge distillation-based methods in exemplar-free class-incremental learning. Our method continuously updates the teacher network by adjusting batch normalization statistics during learning a new task both for the currently learning model and the teacher model saved after learning the previous tasks. This mitigates the changes in the model caused by knowledge distillation loss that arise as the current learner is continuously trying to compensate for the modified normalization statistics. We further improve the stability of

the model by introducing a warm-up phase at the beginning of the task, where a new classification head is trained in isolation before finetuning the whole model. The warm-up phase ensures that the initialization of the weights is not random in the initial phases of training, and reduces the gradient updates to the whole model. We conduct experiments with Teacher Adaptation on several class-incremental benchmarks and show that it consistently improves the results for different knowledge distillation-based methods in an exemplar-free setting. Moreover, our method can be easily added to the existing class-incremental learning solutions and induces only a slight computational overhead.

Discussion Since the introduction of Learning without Forgetting, KD-based methods have emerged as effective solutions to mitigate forgetting in CIL models. Several approaches, such as iCaRL, EEIL, BiC, LUCIR, and SSIL, have integrated KD with exemplars, which helps maintain a balanced discrepancy between the teacher and student models. In scenarios where a sufficient number of exemplars are available, teacher adaptation may not be required, as their presence in the training data mitigates the divergence between the normalization statistics of the subsequent tasks. Nevertheless, our research is dedicated exclusively to the exemplar-free setting, in which we investigate techniques that do not rely on storing exemplars. To the best of our knowledge, we are the first to propose the adaptation of the teacher model within the context of KD-based exemplar-free CIL.

Impact Our method focuses on exemplar-free scenarios, and therefore we alleviate the issues with storing potentially confidential, private, or sensitive data. However, we recognize that machine learning algorithms can be harmful if applied carelessly, and we encourage practitioners to carefully check training data and the models to ensure that the results of their work do not perpetuate biases or discriminate against any minority.

All our work was conducted using publicly available datasets and open-source code. To allow other researchers to build on our work and validate the results, we will share the code for the experiments in this paper on GitHub upon acceptance.

Acknowledgements

References

[1] Zoheb Abai and Nishad Rajmalwar. Densenet models for tiny imagenet classification. *arXiv preprint arXiv:1904.10429*, 2019. 4

[2] Hongjoon Ahn, Jihwan Kwak, Subin Lim, Hyeonsu Bang, Hyojun Kim, and Taesup Moon. Ss-il: Separated softmax for incremental learning. In *Proceedings of the IEEE/CVF International conference on computer vision*, pages 844–853, 2021. 1, 2, 3, 4

[3] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget, 2018. 2

[4] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *CVPR*, pages 3366–3375, 2017. 5

[5] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. Layer normalization. *arXiv preprint arXiv:1607.06450*, 2016. 3, 7

[6] Jihwan Bang, Heesu Kim, YoungJoon Yoo, Jung-Woo Ha, and Jonghyun Choi. Rainbow memory: Continual learning with a memory of diverse samples. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8218–8227, 2021. 1, 2

[7] Tudor Berariu, Wojciech Czarnecki, Soham De, Jorg Bornschein, Samuel Smith, Razvan Pascanu, and Claudia Clopath. A study on the plasticity of neural networks. *arXiv preprint arXiv:2106.00042*, 2021. 12

[8] 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, 4

[9] Sungmin Cha, Sungjun Cho, Dasol Hwang, Sunwon Hong, Moontae Lee, and Taesup Moon. Rebalancing batch normalization for exemplar-based class-incremental learning, 2023. 3

[10] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *ECCV*, pages 532–547, 2018. 5

[11] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Fei-Fei Li. Imagenet: A large-scale hierarchical image database. In *2009 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2009), 20-25 June 2009, Miami, Florida, USA*, pages 248–255, 2009. 4

[12] Prithviraj Dhar, Rajat Vikram Singh, Kuan-Chuan Peng, Ziyan Wu, and Rama Chellappa. Learning without memorizing. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 5138–5146, 2019. 3

[13] Fei Ding, Yin Yang, Hongxin Hu, Venkat Krovi, and Feng Luo. Multi-level knowledge distillation via knowledge alignment and correlation, 2021. 3

[14] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow,* UK, August 23–28, 2020, Proceedings, Part XX 16*, pages 86–102. Springer, 2020. 1, 3

[15] 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

[16] Byeongho Heo, Jeesoo Kim, Sangdoo Yun, Hyojin Park, Nojun Kwak, and Jin Young Choi. A comprehensive overhaul of feature distillation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1921–1930, 2019. 3

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

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

[19] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 12

[20] Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *International conference on machine learning*, pages 448–456. pmlr, 2015. 3

[21] Ahmet Iscen, Thomas Bird, Mathilde Caron, Alireza Fathi, and Cordelia Schmid. A memory transformer network for incremental learning, 2022. 1, 2

[22] Sanghwan Kim, Lorenzo Noci, Antonio Orvieto, and Thomas Hofmann. Achieving a better stability-plasticity trade-off via auxiliary networks in continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 11930–11939, 2023. 3, 4, 12

[23] 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

[24] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019. 2

[25] Jonathan Krause, Michael Stark, Jia Deng, and Li Fei-Fei. 3d object representations for fine-grained categorization. In *2013 IEEE International Conference on Computer Vision Workshops*, pages 554–561, 2013. 4

[26] Alex Krizhevsky. Learning multiple layers of features from tiny images. 2009. 4

[27] Ananya Kumar, Aditi Raghunathan, Robbie Matthew Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net, 2022. 4, 12

[28] Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee. Overcoming catastrophic forgetting with unlabeled data in the wild. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 312–321, 2019. 1, 3

[29] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 1, 2, 3, 4

[30] Xinge Ma, Jin Wang, Liang-Chih Yu, and Xuejie Zhang. Knowledge distillation with reptile meta-learning for pretrained language model compression. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 4907–4917, 2022. 2, 3

[31] S. Maji, J. Kannala, E. Rahtu, M. Blaschko, and A. Vedaldi. Fine-grained visual classification of aircraft. Technical report, 2013. 4

[32] Marc Masana, Xialei Liu, Bartłomiej Twardowski, Mikel Menta, Andrew D Bagdanov, and Joost van de Weijer. Class-incremental learning: survey and performance evaluation on image classification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2022. 1, 2, 4, 12

[33] Claudio Michaelis, Benjamin Mitzkus, Robert Geirhos, Evgenia Rusak, Oliver Bringmann, Alexander S. Ecker, Matthias Bethge, and Wieland Brendel. Benchmarking robustness in object detection: Autonomous driving when winter is coming. *arXiv preprint arXiv:1907.07484*, 2019. 7

[34] Maria-Elena Nilsback and Andrew Zisserman. Automated flower classification over a large number of classes. In *2008 Sixth Indian Conference on Computer Vision, Graphics & Image Processing*, pages 722–729, 2008. 4

[35] Xingchao Peng, Qinxun Bai, Xide Xia, Zijun Huang, Kate Saenko, and Bo Wang. Moment matching for multi-source domain adaptation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1406–1415, 2019. 4, 6

[36] Quang Pham, Chenghao Liu, and Steven Hoi. Continual normalization: Rethinking batch normalization for online continual learning. *arXiv preprint arXiv:2203.16102*, 2022. 3

[37] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II 16*, pages 524–540. Springer, 2020. 1, 2, 3

[38] Wang Qi, Yu-Ping Ruan, Yuan Zuo, and Taihao Li. Parameter-efficient tuning on layer normalization for pre-trained language models. *arXiv preprint arXiv:2211.08682*, 2022. 3

[39] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *2009 IEEE Conference on Computer Vision and Pattern Recognition*, pages 413–420, 2009. 4

[40] Sylvestre-Alvise Rebuffi, Hakan Bilen, and Andrea Vedaldi. Efficient parametrization of multi-domain deep neural networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 8119–8127, 2018. 3

[41] 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

[42] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. *Advances in Neural Information Processing Systems*, 32, 2019. 2

[43] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4510–4520, 2018. 12

[44] Shibani Santurkar, Dimitris Tsipras, Andrew Ilyas, and Aleksander Madry. How does batch normalization help optimization?, 2019. 3

[45] Steffen Schneider, Evgenia Rusak, Luisa Eck, Oliver Bringmann, Wieland Brendel, and Matthias Bethge. Improving robustness against common corruptions by covariate shift adaptation. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 11539–11551. Curran Associates, Inc., 2020. 1, 3

[46] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 12

[47] James Seale Smith, Junjiao Tian, Shaunak Halbe, Yen-Chang Hsu, and Zsolt Kira. A closer look at rehearsal-free continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2409–2419, 2023. 1, 3

[48] Leslie N Smith and Nicholay Topin. Super-convergence: Very fast training of neural networks using large learning rates. In *Artificial intelligence and machine learning for multi-domain operations applications*, volume 11006, pages 369–386. SPIE, 2019. 12

[49] Michael Sun, Ananya Kumar, Divyam Madaan, and Percy Liang. Improving representational continuity via continued pretraining. *arXiv preprint arXiv:2302.13289*, 2023. 12

[50] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 1, 2

[51] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The caltech-ucsd birds-200-2011 dataset. 2011. 4

[52] Dequan Wang, Evan Shelhamer, Shaoteng Liu, Bruno A. Olshausen, and Trevor Darrell. Tent: Fully test-time adaptation by entropy minimization. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. 1, 3, 4, 8

[53] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. In *Computer Vision–ECCV 2022: 17th European Conference, Tel Aviv, Israel, October 23–27, 2022, Proceedings, Part XXVI*, pages 631–648. Springer, 2022. 2

[54] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, pages 139–149, 2022. 2

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

[56] Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 374–382, 2019. 3

[57] Yuxin Wu and Kaiming He. Group normalization, 2018. 3, 7

[58] Bangpeng Yao, Xiaoye Jiang, Aditya Khosla, Andy Lai Lin, Leonidas Guibas, and Li Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *2011 International Conference on Computer Vision*, pages 1331–1338, 2011. 4

[59] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence, 2017. 2

[60] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry Heck, Heming Zhang, and C. C. Jay Kuo. Class-incremental learning via deep model consolidation, 2020. 3

[61] Junting Zhang, Jie Zhang, Shalini Ghosh, Dawei Li, Serafettin Tasci, Larry P. Heck, Heming Zhang, and C.-C. Jay Kuo. Class-incremental learning via deep model consolidation. In *IEEE Winter Conference on Applications of Computer Vision, WACV 2020, Snowmass Village, CO, USA, March 1-5, 2020*, pages 1120–1129. IEEE, 2020. 1

[62] Bowen Zhao, Xi Xiao, Guojun Gan, Bin Zhang, and Shu-Tao Xia. Maintaining discrimination and fairness in class incremental learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 13208–13217, 2020. 3

[63] Da-Wei Zhou, Fu-Yun Wang, Han-Jia Ye, and De-Chuan Zhan. Pycil: A python toolbox for class-incremental learning, 2023. 4

[64] Wangchunshu Zhou, Canwen Xu, and Julian J. McAuley. BERT learns to teach: Knowledge distillation with meta learning. In Smaranda Muresan, Preslav Nakov, and Aline Villavicencio, editors, *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 7037–7049. Association for Computational Linguistics, 2022. 2, 3