On the importance of cross-task features for class-incremental learning

Albin Soutif–Cormerais [1] **Marc Masana** [1 2] **Joost Van de Weijer** [1] **Bartłomiej Twardowski** [1]

Abstract

In class-incremental learning, an agent with limited resources needs to learn a sequence of classification tasks, forming an ever growing classification problem, with the constraint of not being able to access data from previous tasks. The main difference with task-incremental learning, where a task-ID is available at inference time, is that the learner also needs to perform cross-task discrimination, i.e. distinguish between classes that have not been seen together. Approaches to tackle this problem are numerous and mostly make use of an external memory (buffer) of non-negligible size. In this paper, we ablate the learning of cross-task features and study its influence on the performance of basic replay strategies used for class-IL. We also define a new forgetting measure for class-incremental learning, and see that forgetting is not the principal cause of low performance. Our experimental results show that future algorithms for class-incremental learning should not only prevent forgetting, but also aim to improve the quality of the cross-task features, and the knowledge transfer between tasks. This is especially important when tasks contain limited amount of data.

1. Introduction

In order to allow agents with limited resources to update their knowledge in an efficient manner without access to previously learned data, we need to solve the problem of incremental learning. This setting assumes that an agent is exposed to a sequence of tasks which have to be learned one at a time while avoiding catastrophic forgetting of previous tasks (French, 1999; Rebuffi et al., 2017; Thrun, 1996). In practice, this is justified due to the inconvenience or infeasibility of storing data from each task and having to train a new model each time on all data seen so far, which requires a lot of computation and memory. Incremental Learning can be divided between task-incremental learning (task-IL) (Delange et al., 2021) and class-incremental learning[1] (class-IL) (Masana et al., 2020a) depending on the availability or not of the task-ID at test time, respectively. In both cases, each task can be learned during the span of a training session.

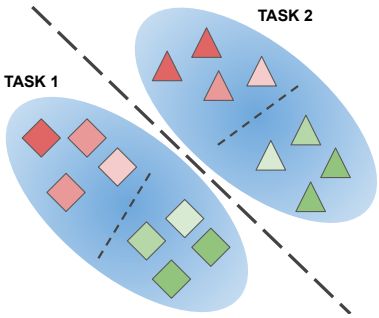

Figure 1: Fictive scenario illustrating the two types of features considered. In this scenario, the color is an intra-task feature, and the shape is a cross-task feature. The intra-task features are insufficient to solve the final 4-class problem.

A particular challenge of class-incremental learning is that classes in different tasks are never learned together but have to be discriminated from each other (no task-ID at test time). One way to tackle this is to store a small subset of instances of each class seen so far to later do rehearsal when learning new ones (Castro et al., 2018; Rebuffi et al., 2017). Therefore, the agent is capable of seeing classes from different tasks together over the course of training. Even though some class-IL methods do not make use of a memory buffer (Dhar et al., 2019b; Yu et al., 2020), these can currently not compete with methods using one (Masana et al., 2020a). Therefore, in this paper, we focus our attention on methods with a memory buffer. However, because of the memory restrictions of such rehearsal memory buffer, a class imbalance is introduced during the learning process. One of the side effects of this class imbalance is that the biases and the norm of weights in the classifier tend to be higher in the classes

[1]Computer Vision Center, Barcelona [2]ICG, Graz University of Technology and DES Silicon Austria Labs. Correspondence to: Albin Soutif–Cormerais <albin@cvc.uab.cat>.

Proceedings of the 38th International Conference on Machine Learning, PMLR 139, 2021. Copyright 2021 by the author(s).

[1]In class-IL and task-IL, data comes as a sequence of batches of classes which we refer to as tasks. In this article, we do not consider the scenario in which these batches would contain only one class.

from newer tasks (Hou et al., 2019; Masana et al., 2020a; Wu et al., 2019). This phenomenon is called task-recency bias and is tackled with different bias-correction strategies by recent approaches (Belouadah & Popescu, 2019; Hou et al., 2019; Wu et al., 2019). Having addressed the task-recency bias, there is still a large gap with respect to the upper-bound (joint training). In this paper, we investigate to what extend the training of better cross-task features can narrow this gap.

Another focus of this work is the study of the causes of performance drop in class-IL. As in other incremental learning settings, processing the sequence of tasks in class-IL leads to a dramatic drop in performance from earlier tasks to the latest ones. Many class-IL works associate *catastrophic forgetting*, which has been widely studied in the task-IL setting, to this performance drop. However, the metrics defined for task-IL are not well suited to the class-IL setting and can give misleading results when applied directly to the latter. In this work, we will define similar metrics adapted to class-IL, and use them to increase understanding of the main causes of performance drop when moving to that setting.

When learning in an incremental manner, we consider two types of features that can be learned by the feature extractor (see Fig. 1). A *cross-task discriminative feature* discriminates between classes that belong to different tasks, while an *intra-task discriminative feature* discriminates between classes from the same task. Furthermore, some features might appear during training that satisfy both types of discrimination. Given feature types, we postulate that replay in class-IL should fulfil multiple roles at the feature extractor level. First, to maintain previously learned intra-task discriminative features. Second, to create cross-task discriminative features capable of discriminating between classes not present in the same task. For instance, in Fig. 1, discriminating between classes of the same task only requires to learn color features (intra-task), while solving the cumulative tasks proposed in class-IL requires to also learn shape features (cross-task). Its third role is to enable knowledge transfer between the new task and previous tasks so that one task can benefit from the learning of another as it occurs when learning multiple tasks concurrently (Baxter, 2000). Failing to learn either type of feature would result in higher miss-classification rate. In this paper, we aim to study whether these two types of features are learned properly with replay, especially cross-task features.

Our contributions are summarised as follow:

- We compare two baselines using replay. One aims to learn cross-task features while the other does not. This helps us to assess what is the importance of cross-task features in the class-IL setting.
- We question the use of classic task-IL metrics in class-IL and propose new appropriate metrics. We observe that *catastrophic forgetting* is not the main cause of performance drop in class-IL.

In addition, we perform experiments allowing an increasing amount of memory to the class-incremental algorithms. These results suggest that even though there is no forgetting, there is room for improvements in task specific accuracy that can significantly improve cross-task discrimination without explicitly building features for it.

2. Related Work

Class-incremental learning approaches can be divided in three categories (Masana et al., 2020a): regularisation-based, bias-correction, and rehearsal-based, which are often combined to tackle multiple incremental learning challenges.

Regularisation-based approaches add a regularisation term to the loss which penalises changes in the weights (Aljundi et al., 2018; Chaudhry et al., 2018; Kirkpatrick et al., 2017) or activations (Jung et al., 2016; Li & Hoiem, 2017; Rannen et al., 2017; Zhang et al., 2020). This maintains the stability-plasticity trade-off between maintaining previous knowledge and learning new tasks. These approaches have been widely used in task-IL. Furthermore, they have also shown promising results in class-incremental learning, whether by pairing with rehearsal-based approaches (Castro et al., 2018; Hou et al., 2019; Rebuffi et al., 2017; Wu et al., 2019) or with other strategies such as attention (Dhar et al., 2019a).

Rehearsal has become the most commonly used approach in class-incremental learning, focusing on images or feature replay of previous tasks while learning a new one. Rehearsal-based approaches can be further divided into different subcategories. Pseudo-rehearsal methods replay generated images (Shin et al., 2017), thus avoiding the privacy issue of storing raw data. They also have the advantage of reducing storage memory and generating it on-the-fly instead. In MerGan (Wu et al., 2018), a conditional GAN is used to replay images, introducing the learning and storage of an advanced network instead of having to store exemplars. In Generative Feature Replay (Liu et al., 2020a), a GAN is used to replay smaller size features in to the classification layer, making the GAN learning easier at the cost of fixing the feature extractor, limiting the power of future learning. Some pseudo-rehearsal methods generate images by directly optimising in the image space using a loss that aims to prevent forgetting (Ji et al., 2020; Liu et al., 2020b), thus avoiding the use of an external generator. Classical rehearsal methods store exemplars and replay them along with the data from the task at hand. However, this introduces a data imbalance problem between the few exemplars available for previously learned classes and the large amount of data for the new ones. For that reason some approaches combine rehearsal

with bias correction (Belouadah & Popescu, 2019; Hou et al., 2019; Wu et al., 2019), which also tackles task-recency bias. In IL2M (Belouadah & Popescu, 2019), a dual-memory is proposed, storing both images and class-statistics, which are used to rectify the scores of past classes. Noticeably, most rehearsal approaches use a distillation loss term on the outputs of the model, while IL2M obtains similar results with a classic fine-tuning cross-entropy loss instead.

A more challenging setting is introduced with online learning, where no more than one pass is allowed on the current task data. Therefore, instead of incremental training sessions which can iterate over several epochs on the data, each sample is seen once unless stored in the reduced external exemplar memory. This setting was originally explored for task-IL, where the task-ID is known at test time (Chaudhry et al., 2019a; Lopez-Paz & Ranzato, 2017). In GEM (Lopez-Paz & Ranzato, 2017) and A-GEM (Chaudhry et al., 2019a), gradients are modified in order to avoid both increasing the loss on the exemplars and over-fitting on them. Eventually, it is observed in *tiny episodic memories* (Chaudhry et al., 2019b), that just replaying exemplars, even when only a few are available, efficiently prevents forgetting in the task-IL setting. Recently, some approaches have been applied to online class-IL (Aljundi et al., 2019a;b; Hayes et al., 2020). In MIR (Aljundi et al., 2019a), controlled sampling is performed to automatically rehearse samples from tasks currently undergoing the most forgetting. REMIND (Hayes et al., 2020) compresses features from the exemplars using learned quantization modules, which later are replayed.

In recent surveys (Belouadah et al., 2020; Masana et al., 2020a), inter-task confusion and bias correction are identified among the main challenges of class-IL. We argue that additional challenges are the ones of knowledge transfer between tasks, and the learning of cross-task features (in contrast with task-recency bias). Finally, GDumb (Prabhu et al., 2020) introduces a baseline which only uses exemplars from the buffer, obtaining comparable performance to the state-of-the-art in the online setting. The proposed method also works for class-IL since it only needs the exemplars present in the memory buffer, however, it achieves a lower performance. Similarly, our proposed baselines are also applicable to both online and offline class-IL.

3. Intra- and Cross-task training

3.1. Notation

We assume that the data comes in the form of the following task sequence:

$$\mathcal{T} = \{(C^1, D^1), (C^2, D^2), ...(C^n, D^n)\}, \qquad (1)$$

where n is the number of tasks, C^t is the set containing the classes of task t, with the constraint of non-overlapping classes between different tasks ($\forall\, 1 \leq i \leq n, i \neq j, C^i \cap C^j = \varnothing$). Let D^t be the dataset for task t, containing the labelled images. The learner is a neural network, function of its parameters θ and the input x, which we will denote by $f(x;\theta)$. We further split this network into a feature extractor $\Psi(x;\theta_\Psi)$ with weights θ_Ψ, and a classification layer $\nu(x;\theta_\nu)$ with weights θ_ν. The logits are obtained by applying the feature extractor first, then the classification layer, obtaining $f(x;\theta) = \nu(\Psi(x;\theta_\Psi);\theta_\nu)$. The upper-scripted θ^t denote the weights of the model after seeing task t. Additionally, we denote the current task as T, a batch of data as $\mathcal{B}$, and the upper-scripted subset containing data from task t as $\mathcal{B}^t = \{(x,y) \in \mathcal{B}^t, y \in C^t\}$. So that we can consider all classes and data that the network has seen so far, we define $C^k_\Sigma = \bigcup_{t=1}^k C^t$ and $D^k_\Sigma = \bigcup_{t=1}^k D^t$, where Σ stands for cumulative.

3.2. Two Replay Baselines

In order to better dissect the replay mechanism, we propose two different baselines to address class-IL. The two methods differ only in the loss they apply to the feature extractor. The first one explicitly tries to learn both cross-task and intra-task discriminative features, while the second one avoids learning cross-task discriminative features, restricting itself to learn only intra-task ones. They are defined as:

$$\mathcal{L}_{CE}(\mathcal{B};\theta) = \sum_{(x,y)\in\mathcal{B}} -\log \frac{\exp f(x;\theta)_y}{\sum_{c\in C^T_\Sigma} \exp f(x;\theta)_c}, \qquad (2)$$

$$\mathcal{L}_{CE\text{-}IT}(\mathcal{B};\theta) = \frac{1}{T}\sum_{t=1}^{T} \sum_{(x,y)\in\mathcal{B}^t} -\log \frac{\exp f(x;\theta)_y}{\sum_{c\in C^t} \exp f(x;\theta)_c}. \qquad (3)$$

$\mathcal{L}_{CE}$ (Eq. 2) is the classical cross-entropy over all classes seen so far, which seems natural to use whenever using exemplars. $\mathcal{L}_{CE\text{-}IT}$ (Eq. 3) is the sum of the cross-entropies of each separate tasks, as it would be done in multi-task training or task-incremental learning, this loss is only learning intra-task discriminative features ($CE\text{-}IT$ stands for cross-entropy intra-task). During training, $\mathcal{B}$ will be drawn from $D^t \cup \mathcal{M}$ where $\mathcal{M}$ is a memory buffer that retains a small number of samples per class.

To tackle task-recency bias that might arise from unbalanced training, and calibrate the classification heads on top of the feature extractor, we add an extra step to our proposed baseline algorithms. Taking inspiration from EEIL (Castro et al., 2018), we perform an additional balanced fine-tuning step after learning the current task, training the network on data only from the exemplars memory buffer, which contains a balanced number of training samples per class. We use $\mathcal{L}_{CE}$ on both cases for this step, but only back-propagating it through the classifier, leaving the feature extractor frozen with the previously learned knowledge.

Algorithm 1 Two step procedure

Input: task sequence $\mathcal{T}$, data D, mem. buffer $\mathcal{M}$, loss $\mathcal{L}$
for $t \in 1 \dots T$ **do**
  for $i \in 1 \dots N_{\text{epochs}}$ **do**
    for $\mathcal{B} \sim D^t \cup \mathcal{M}$ **do**
      $\theta' \leftarrow \text{SGD}(\mathcal{L}, \mathcal{B}, \theta)$
    end for
  end for
  $\mathcal{M} \leftarrow \text{FillMemory}(D^t, \mathcal{M})$
  for $i \in 1 \dots N_{\text{FTepochs}}$ **do**
    for $\mathcal{B}_{mem} \sim \mathcal{M}$ **do**
      $\theta''_\nu \leftarrow \text{SGD}(\mathcal{L}_{CE}, \mathcal{B}_{mem}, \theta')$
    end for
  end for
end for

This is a similar procedure as FT^{BAL} (Belouadah et al., 2020) with the difference of the feature extractor parameters being frozen while the balancing step takes place.

The detailed training procedure is explained in Algorithm 1, where $\mathcal{L}$ is the loss used during the first training step, and is replaced by $\mathcal{L}_{CE}$ or $\mathcal{L}_{CE\text{-}IT}$ depending on the baseline we are using. We will later refer to these baselines as FT^{BAL}_{+CTF} (using $\mathcal{L}_{CE}$) and $FT^{BAL}_{\text{-}CTF}$ (using $\mathcal{L}_{CE\text{-}IT}$) where $+CTF$ and $\text{-}CTF$ stand for with and without Cross Task Features respectively. Indeed, $FT^{BAL}_{\text{-}CTF}$ will emulate the learning of a feature extractor as it would be done in the task-incremental learning setting. This will allow us to evaluate how features learned similarly as in the task-IL setting perform when used in the class-IL scenario, without access to task id. This fine-tuning step can be seen as a sort of probing of the feature extractor, similarly to what is done in (Alain & Bengio, 2017), though here it is performed on a reduced dataset.

Naturally, we expect FT^{BAL}_{+CTF} to perform better in the class-IL setting since it is targeted to that setting. Our interest will rather focus on the gap between FT^{BAL}_{+CTF} and $FT^{BAL}_{\text{-}CTF}$. Studying this gap and its contribution to the larger gap between these methods and the upper bound for this setting will help us understand how improvements in task-incremental learning can benefit class-incremental learning.

Finally, we shortly recall the main challenges for task-IL since these are inherited by class-IL. In task-IL, since the task-ID is available at test time, it is not necessary to learn any cross-task features to obtain good performance on each specific tasks. This setting stills brings its share of challenges. Catastrophic forgetting (Goodfellow et al., 2014; Kirkpatrick et al., 2017) is one of the challenges that has been addressed the most in the literature. In task-IL, it is described as the drop in performance on previous tasks when the learning of a new task occurs. While catastrophic forgetting can be tough to solve in some settings and without access to old samples, when a memory buffer of reasonable size is used (as it is done in class-incremental learning replay methods), it becomes easier to avoid and many works have shown how to prevent it (Lopez-Paz & Ranzato, 2017; Chaudhry et al., 2019a). However, other challenges have been identified, among them, the one of knowledge transfer between tasks. Primarily discussed in the multi-task learning literature (Baxter, 2000), the transfer of knowledge between tasks enables the learner to boost its performance on separate tasks when it is learning them concurrently. In the task-incremental learning setting, this transfer between tasks has been referred to as backward- and forward- transfer in GEM (Lopez-Paz & Ranzato, 2017). Where backward denotes the positive influence on previous tasks performance that learning subsequent tasks could have, while forward denotes the positive influence learning previous tasks could have on the learning of new tasks.

3.3. On forgetting in class-incremental learning

Catastrophic forgetting (Goodfellow et al., 2014; Kirkpatrick et al., 2017) has been widely cited as being the main cause of performance drop in many continual learning settings. From task-IL works with two tasks similar to transfer learning (Kirkpatrick et al., 2017), the measurement of forgetting was defined as how much of the performance on the source task decreased when learning the target task. Initially, this measurement was very useful even when moving from two tasks to a sequence of them since each task was evaluated separately. It has been widely observed that training a neural network on a sequence of tasks without accessing data from previous tasks results in catastrophic forgetting of previously learned ones. Similar variations of that metric have also been proposed for both task-IL and class-IL (Chaudhry et al., 2018; Lee et al., 2019; Serra et al., 2018). However, we argue that they do not directly transfer to class-IL since, in that setting, the average accuracy is reported on an ever-more difficult problem, which is classifying between all C^n classes. For this reason, looking at the classic forgetting measure in class-IL it is not clear if it captures only the forgetting of previous tasks but also the increasing difficulty of the cumulative task.

Task-IL metrics: Consider that $a^t_k \in [0, 1]$ denotes the accuracy of task k after learning task t ($k \leq t$), then the *average accuracy* at task t is defined as $A_t = \frac{1}{t}\sum_{i=1}^{t} a^t_i$. *Forgetting* estimates how much the model forgot about previously learned task k at current task t and is defined as $f^t_k = \max_{i \in \{k,\dots,t-1\}} a^i_k - a^t_k$. As with accuracy, this measure can be averaged over all tasks learned so far: $F^t = \frac{1}{t-1}\sum_{i=1}^{t-1} f^t_i$ (Chaudhry et al., 2019a). We denote this version of forgetting as *classic forgetting*.

Class-IL metrics: To consider these metrics for the case of class-IL, we have to replace the notion of separate tasks that

Table 1: Average accuracy after learning all tasks for CIFAR-100 on ResNet-32 from scratch.

scenario	approach	growing memory size				
		20/cls	10/cls	5/cls	2/cls	max
10 tasks	FT^{BAL}_{+CTF}	40.55 ± 2.1	31.5 ± 3.0	19.1 ± 2.5	9.0 ± 0.69	66.8 ± 1.1
	FT^{BAL}_{-CTF}	34.8 ± 2.1	27.4 ± 1.5	20.1 ± 1.2	14.3 ± 0.5	56.6 ± 1.2
20 tasks	FT^{BAL}_{+CTF}	34.2 ± 1.9	26.6 ± 2.7	12.5 ± 1.6	4.4 ± 0.15	67.0 ± 1.5
	FT^{BAL}_{-CTF}	26.8 ± 2.6	20.1 ± 2.0	15.1 ± 1.7	9.2 ± 0.7	51.8 ± 2.3

is used in task-IL by the cumulative task that is composed of the concatenation of all tasks seen so far $\mathcal{T}^k_\Sigma = (C^k_\Sigma, D^k_\Sigma)$. When evaluating a network on the cumulative task k, we propose the following method to predict the label:

$$\hat{y}_k(x;\theta) = \arg\max_{c \in C^k_\Sigma} f(x;\theta)_c \,, \tag{4}$$

which restricts the output to the C^k_Σ classes. Note that the predictions can change when you chose another task $j \neq k$, which is desirable since it makes the task easier when k gets smaller, getting rid of the confusion between classic forgetting and the growing task difficulty discussed above. We further introduce the term *cumulative accuracy* of task k as:

$$b^t_k = \frac{1}{|D^k_\Sigma|} \sum_{x,y \in D^k_\Sigma} 1_{\{y\}}\big(\hat{y}_k(x;\theta^t)\big) \,, \tag{5}$$

which refers to the accuracy obtained on the cumulative task k by the model θ^t, and where $1_{\{y\}}$ is the indicator function which is 1 when the prediction is correct, and 0 otherwise. To summarise the performance of the continual learning approach after t tasks have been learned we simply report b^t_t, which is equivalent to the average accuracy classically reported in class-IL works. We then can analogously define *cumulative forgetting* about a previous cumulative task k at current task t as:

$$f^t_k = \max_{i \in \{1,\dots,t\}} b^i_k - b^t_k \,. \tag{6}$$

This measure can be averaged over all tasks learned so far: $F^t_\Sigma = \frac{1}{t-1}\sum_{i=1}^{t-1} f^t_i$. The above defined measurements are an adaptation of the ones from (Chaudhry et al., 2019a) to the setting of class-incremental learning, the main difference is that ours consider the incremental cumulative tasks instead of the separate ones.

4. Experimental Results

Our implementation is based on the FACIL framework (Masana et al., 2020a). We use the same data processing and scenario configurations. We extend it with our proposed baselines to compare with its implementation of state-of-the-art methods. More details are available in the supplementary material. Our code is available at `https://github.com/AlbinSou/cross_task_cil`.

Datasets: CIFAR-100 (Krizhevsky, 2009) contains 60k rgb images of size 32x32x3 divided in 100 classes, each having 500 training images and 100 test images. Since there is no defined validation set, we set apart 10% of the train set as validation, and keep it the same for all experiments. This is a common setting in class-IL (Belouadah & Popescu, 2019; Castro et al., 2018; Hou et al., 2019; Rebuffi et al., 2017; Wu et al., 2019). The classes are divided between 10 or 20 task splits to be learned incrementally, with a memory buffer that contains 20 exemplars per class for rehearsal (2,000 total). Exemplars are selected using the herding strategy, which has been shown to be slightly more robust than other strategies (Masana et al., 2020a). Data augmentation is applied via padding, random crop and random horizontal flips during training. Finally, task and class orderings (Masana et al., 2020b), are fixed to the commonly used iCaRL seed (Masana et al., 2020a; Rebuffi et al., 2017; Wu et al., 2019).

Imagenet (Russakovsky et al., 2015) contains 1,000 object classes with different number of samples per class. In our experiments, we use a reduced version composed of its 100 first classes, Imagenet-Subset (as in (Rebuffi et al., 2017)). Data is pre-processed using 224×224 random crop, normalization and random horizontal flip. We split this dataset into 25 tasks with a random class order fixed for all experiments. Exemplars are handled in the same way as the CIFAR-100 experiments.

Network: we use the commonly paired network for CIFAR-100 experiments: ResNet-32 (He et al., 2016), learned with Stochastic Gradient Descent with a patience scheme. More details about the hyper-parameters used during training can be found in the supplementary material. For Imagenet-Subset, we use ResNet-18, which allows for larger input size and provides more capacity.

Upper bounds: for each of the two baselines that use a limited amount of memory, we consider their respective upper bounds that have access to all data from previous tasks, namely $FT^{BAL}_{+CTF}(max)$ and $FT^{BAL}_{-CTF}(max)$.

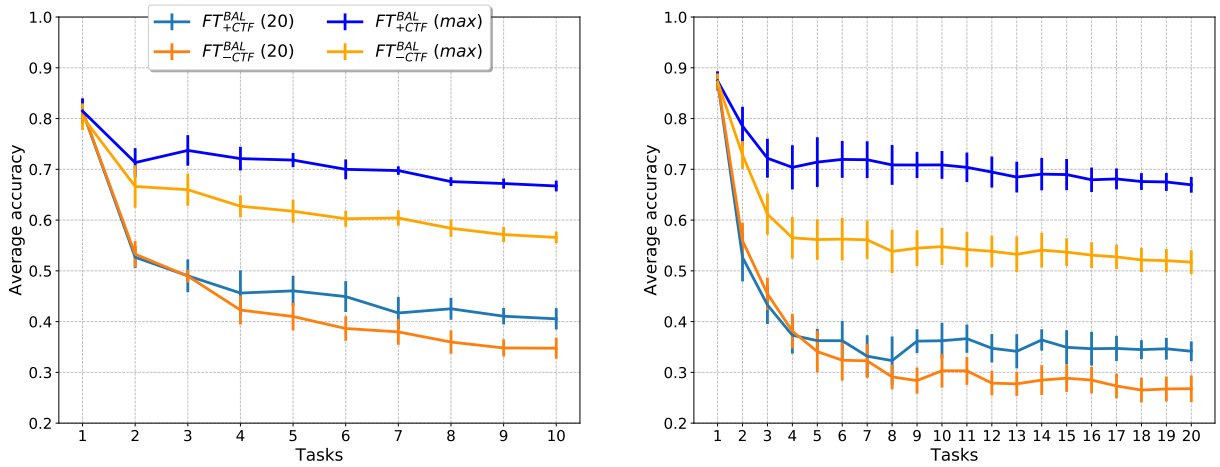

Figure 2: Average accuracies on CIFAR-100 splitted in 10 tasks (left) and 20 tasks (right) for FT^{BAL}_{+CTF} and FT^{BAL}_{-CTF} using 20 exemplars per class compared to their respective upper-bounds (500 exemplars per class). Mean and standard deviation over 10 runs are reported.

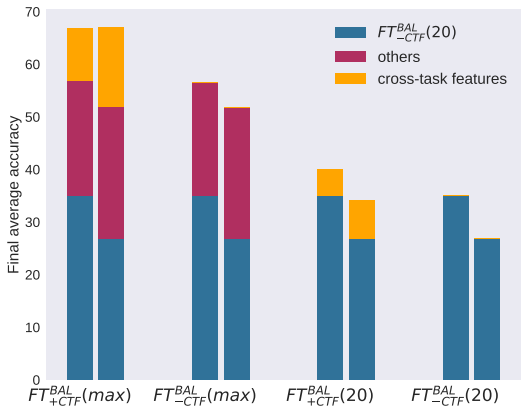

Figure 3: Final average accuracy obtained by each method and their respective upper bounds on CIFAR-100 splitted in 10 tasks (Left) and 20 tasks (Right). The part in red coined "others" can be obtained with better intra-task features, while the orange part is the additional gain obtained when learning cross-task features

Table 2: Average forgetting (classic) compared to newly defined cumulative forgetting measure on CIFAR100 10 and 20 tasks split for FT^{BAL}_{+CTF}. Cumulative forgetting shows no sign of performance drop due to a forgetting phenomenon.

memory size	10 Tasks classic	10 Tasks cumulative	20 Tasks classic	20 Tasks cumulative
(2/class)	81.6	4.0	86.0	6.4
(5/class)	69.5	1.6	74.1	1.1
(10/class)	56.0	0.9	49.5	0.7
(20/class)	28.8	0.4	39.9	0.6

4.1. Importance of cross-task features

Results for CIFAR-100 are shown in Table 1. For the 10 tasks split, using FT^{BAL}_{+CTF} performs better than FT^{BAL}_{-CTF} when using 10 or more exemplars per class, indicating that it is able to learn some cross-task features. For very low number of exemplars, we observe that FT^{BAL}_{+CTF} is performing poorly compared to FT^{BAL}_{-CTF}. The average accuracy gap after 10 tasks between the two methods is around $\sim 5\%$ when using 20 exemplars per class, which is the most commonly considered memory size for class-IL. As we move to the 20 tasks split, both methods suffer from a $\sim 5\%$ drop in performance. Through the comparison of these two losses with their respective upper bounds using the maximum of memory (see Fig. 2), we observe that the gap due to not learning cross-task features is of $\sim 10\%$, and jumps to $\sim 15\%$ when moving to 20 tasks. This gap is filled in part ($\sim 5\%$ for 10 tasks and $\sim 7\%$ for 20 tasks) by FT^{BAL}_{+CTF} when 20 growing memory per class is used. It means that the gap due to the learning of cross-task features is already filled by a half by FT^{BAL}_{+CTF}, the remaining performance gap is then due to something else, materialised by the difference between the upper bound for FT^{BAL}_{-CTF} (max) and FT^{BAL}_{-CTF} (20). We summarise these observations in Fig. 3.

On Imagenet-Subset (see Fig. 4), we observe the same conclusions. There is a consistent difference between the two baselines due to the additional learning of cross-task features. But this difference does not account for the main part of the performance gap, which is observed when comparing FT^{BAL}_{-CTF} (max) and FT^{BAL}_{-CTF} (20), and is not related to cross-task features.

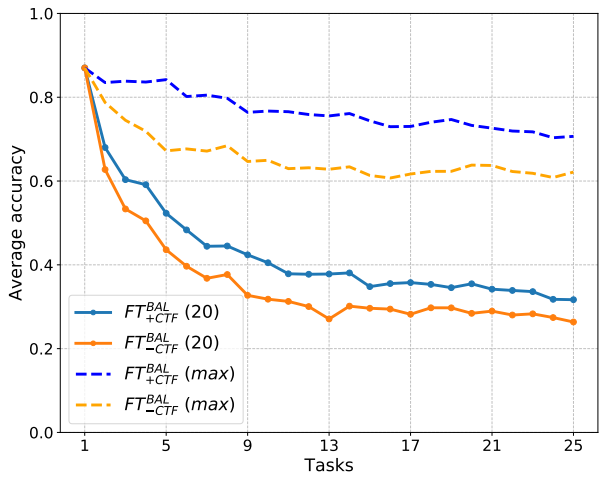

Figure 4: Average accuracies on Imagenet-Subset splitted in 25 tasks for FT^{BAL}_{+CTF} and FT^{BAL}_{-CTF} using 20 exemplars per class compared to their respective upper-bounds with maximum memory. As it is the case on cifar, the gap due to the learning of cross-task features is not predominant, and is partly filled by the use of FT^{BAL}_{+CTF}.

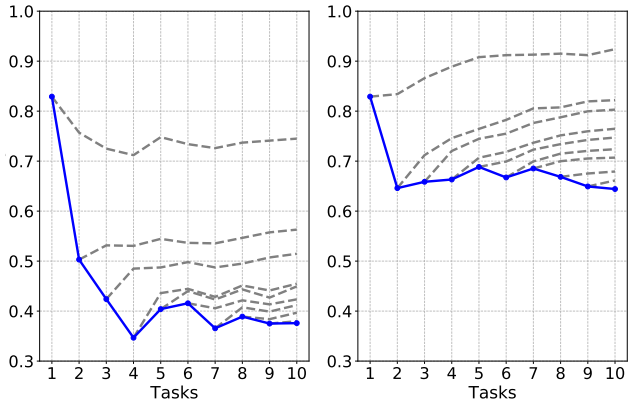

Figure 5: Cumulative accuracies b^t_k on CIFAR100 (10 tasks) for $FT^{BAL}_{+CTF}(20mem/cls)$ (Left) and $FT^{BAL}_{+CTF}(max)$ (Right). Grey dashed lines represent b^t_k for varying t and one fixed k per line. The blue dotted line represent b^t_t for varying t, which is the average accuracy.

4.2. Cumulative accuracy and cumulative forgetting

We have argued that the forgetting measure defined for task-IL cannot directly be applied to class-IL (as done in some papers). Instead the definitions should be adapted to consider cumulative accuracy of the tasks (see Eq. 5).

We apply the metrics defined in Sec. 3.3 to the experiments on CIFAR-100 (10 and 20 tasks). One insightful way to analyse the learning process of class-IL methods is provided in Fig. 5, we coin this a *cumulative accuracy graph*. It allows you to analyse the behaviour of the various cumulative tasks while the learning progresses. For example, the line starting from Task 2 indicates b^t_2. There, we can observe an initial positive transfer when learning task 3 and a subsequent planar behaviour where performance remains constant. The important fact to observe from this graph is that while the average accuracy decreases considerably over the course of training, the values of the cumulative accuracy b^t_k when fixing k are quite stable. The resulting cumulative forgetting measure is consequently very low, and only grows for the smaller memory sizes (see Table 2). This means that cumulative forgetting cannot explain on its own the decrease in average accuracy. This strongly contrasts with the values obtained using classic forgetting, which are higher and indicate that most of the performance drop is caused by forgetting. In conclusion, we argue that the classic forgetting method is inapplicable to class-IL and instead encourage the usage of cumulative forgetting. More cumulative accuracies graph for other class-IL methods are made available in the supplementary.

Finally, for the model using maximum memory (see Fig. 5), we observe that the cumulative accuracies grow over the course of training, which means the model gets better at previous cumulative tasks. This is showing that positive backward transfer is occurring: learning new tasks improves performance on older tasks.

4.3. Comparison to state-of-the-art

In Fig. 6, we display a comparison of the considered baselines to other state of the art methods like iCarl (Rebuffi et al., 2017), LUCIR (Hou et al., 2019), BiC (Wu et al., 2019) and EEIL (Castro et al., 2018). **FT-E** is another baseline which is similar to FT^{BAL}_{+CTF} but does not use a second fine-tuning stage. We observe that FT^{BAL}_{+CTF} and FT^{BAL}_{-CTF} perform comparatively to the latter methods in this setting, the comparison with **FT-E** shows the gain of the additional fine-tuning stage, which helps to get rid of the task-recency bias. The main purpose to include this comparison to state-of-the-art methods is to show that the baseline FT^{BAL}_{+CTF} gets competitive results which means that our analyses on the challenges for FT^{BAL}_{+CTF} could generalize to other state-of-the-art methods.

4.4. Discussion

We here explore the efficiency of cross-task features learning done by FT^{BAL}_{+CTF} as a function of the memory size. To do so, we use two additional metrics that focus on specific points of the learning. The task-aware accuracy $\mathcal{A}_{taw}$ is the accuracy of the model on the test data when it is provided with the task-id, averaged over all tasks the model has seen so far. It is commonly used in task-IL and aims to evaluate the task-specific performance across tasks. Since class-incremental learning comes with the additional challenge of discriminating across tasks, we additionally use a

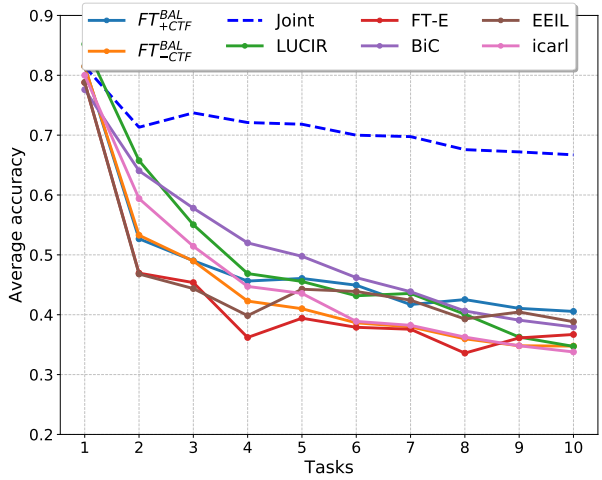

Figure 6: Comparison including multiple class incremental learning methods and the two baselines we use. CIFAR100 10 tasks, 20 growing memory per class. The baselines used perform comparatively to other state of the art methods.

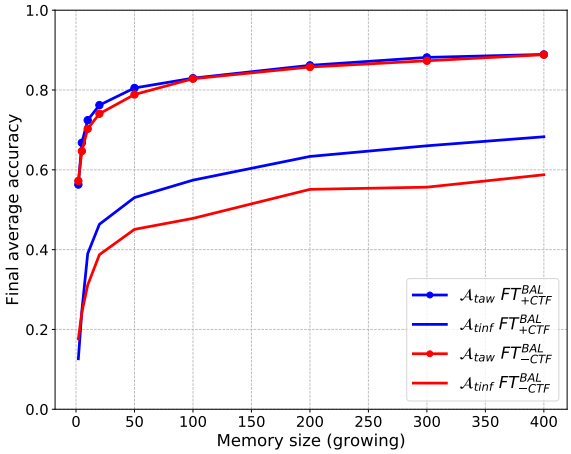

Figure 7: Task inference and Task aware accuracy obtained at the end of the task sequence for different memory sizes. CIFAR100 10 tasks. The gap in task inference between FT^{BAL}_{+CTF} and FT^{BAL}_{-CTF} stabilises after 50 exemplars per class. Additional performance gains can be obtained by increasing task-aware accuracy, which is correlated with task inference accuracy.

task-inference accuracy $\mathcal{A}_{tinf}$. The latter is computed by counting the number of predictions that fall into the correct task. We report the values of these two measurements with an increasing memory size in Fig. 7.

Although the use of FT^{BAL}_{+CTF}, explicitly learning cross-task features accounts for a part of the gap between 20 memory and the maximum amount, we observe that both task-aware accuracy and task inference accuracy grow when increasing the memory size. Since forgetting is eliminated as a possible cause (see Tab. 2), the improvement in task-aware accuracy when the memory increases is most likely explained by knowledge transfer between tasks. This happens for both FT^{BAL}_{+CTF} and FT^{BAL}_{-CTF} and indicates that there is a greater potential gain in task-inference by learning better quality task-specific features (For instance, through improved backward and forward transfer) rather than by learning better cross-task features. Indeed in this case, the learning of cross-task features is already maximum at the 50 exemplars mark (the gap between the red curve and blue curve does not increase anymore).

5. Conclusion and future directions

Through the ablation of one major component of class-incremental learning that makes it different from task-incremental learning, we attempted to link the challenges met by these two settings. By studying the impact that the absence of explicit cross-task features learning could have on popular benchmarks, we have shown that the following is already partly solved by the use of a simple replay procedure. While the learning of cross-task feature could still be improved, its influence on the final result may not be as decisive as we might think. Instead, the major part of the gap could be due to other sources. The lack of knowledge transfer studied in task-IL could be one of them. We have also highlighted that forgetting as defined for task-IL has to be carefully adapted for class-IL. Using our proposed cumulative forgetting measure, we observed that forgetting is not the main cause of performance drop in that setting.

We hope that future research can draw more links between task-IL and class-IL. While these two settings differ, we saw that they share similar challenges. In particular, we think that it could be interesting to tackle the task-IL setting using similar memory constraints as in class-IL, and aim for better knowledge transfer instead of zero forgetting. Indeed, enabling more knowledge transfer between tasks could directly be applied to class-incremental learning by learning classification heads on top of the learned feature extractor, similarly to what we did with FT^{BAL}_{-CTF}).

Another promising type of approaches to improve the quality of the learnt representation is learning with an unlabelled data stream concurrently to the current task, as done in (Lee et al., 2019). That way knowledge is transferred between the data stream and the current task instead of in-between tasks. However, this requires to have access to such a data stream with nice properties w.r.t the tasks at hand.

Acknowledgments

We acknowledge the support from Huawei Kirin Solution, and the Spanish project PID2019-104174GB-I00 (MINECO, Spain). and the CERCA Programme of Generalitat de Catalunya.

References

Alain, G. and Bengio, Y. Understanding intermediate layers using linear classifier probes. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Workshop Track Proceedings*. OpenReview.net, 2017.

Aljundi, R., Babiloni, F., Elhoseiny, M., Rohrbach, M., and Tuytelaars, T. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 139–154, 2018.

Aljundi, R., Belilovsky, E., Tuytelaars, T., Charlin, L., Caccia, M., Lin, M., and Page-Caccia, L. Online continual learning with maximal interfered retrieval. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019a.

Aljundi, R., Lin, M., Goujaud, B., and Bengio, Y. Gradient based sample selection for online continual learning. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché-Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019b.

Baxter, J. A model of inductive bias learning. *Journal of artificial intelligence research*, 12:149–198, 2000.

Belouadah, E. and Popescu, A. Il2m: Class incremental learning with dual memory. *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 583–592, 2019.

Belouadah, E., Popescu, A., and Kanellos, I. A comprehensive study of class incremental learning algorithms for visual tasks. *Neural Networks*, 2020.

Castro, F. M., Marín-Jiménez, M. J., Guil, N., Schmid, C., and Alahari, K. End-to-end incremental learning. In *Proceedings of the European conference on computer vision (ECCV)*, pp. 233–248, 2018.

Chaudhry, A., Dokania, P. K., Ajanthan, T., and Torr, P. H. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 532–547, 2018.

Chaudhry, A., Ranzato, M., Rohrbach, M., and Elhoseiny, M. Efficient lifelong learning with a-GEM. In *International Conference on Learning Representations*, 2019a.

Chaudhry, A., Rohrbach, M., Elhoseiny, M., Ajanthan, T., Dokania, P. K., Torr, P. H., and Ranzato, M. Continual learning with tiny episodic memories. 2019b.

Delange, M., Aljundi, R., Masana, M., Parisot, S., Jia, X., Leonardis, A., Slabaugh, G., and Tuytelaars, T. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

Dhar, P., Singh, R. V., Peng, K.-C., Wu, Z., and Chellappa, R. Learning without memorizing. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 5138–5146, 2019a.

Dhar, P., Singh, R. V., Peng, K.-C., Wu, Z., and Chellappa, R. Learning without memorizing. In *Conference on Computer Vision and Pattern Recognition*, 2019b.

French, R. M. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. An empirical investigation of catastrophic forgeting in gradient based neural networks. In *In Proceedings of International Conference on Learning Representations (ICLR*, 2014.

Hayes, T. L., Kafle, K., Shrestha, R., Acharya, M., and Kanan, C. Remind your neural network to prevent catastrophic forgetting. In *European Conference on Computer Vision*, pp. 466–483. Springer, 2020.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 770–778, 2016.

Hou, S., Pan, X., Loy, C. C., Wang, Z., and Lin, D. Learning a unified classifier incrementally via rebalancing. In *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

Ji, X., Henriques, J. F., Tuytelaars, T., and Vedaldi, A. Automatic recall machines: Internal replay, continual learning and the brain. *CoRR*, abs/2006.12323, 2020.

Jung, H., Ju, J., Jung, M., and Kim, J. Less-forgetting learning in deep neural networks. *arXiv preprint arXiv:1607.00122*, 2016.

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, 2009.

Lee, K., Lee, K., Shin, J., and Lee, H. Overcoming catastrophic forgetting with unlabeled data in the wild. In

Proceedings of the IEEE/CVF International Conference on Computer Vision, pp. 312–321, 2019.

Li, Z. and Hoiem, D. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017.

Liu, X., Wu, C., Menta, M., Herranz, L., Raducanu, B., Bagdanov, A. D., Jui, S., and de Weijer, J. v. Generative feature replay for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pp. 226–227, 2020a.

Liu, Y., Su, Y., Liu, A.-A., Schiele, B., and Sun, Q. Mnemonics training: Multi-class incremental learning without forgetting. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 12245–12254, 2020b.

Lopez-Paz, D. and Ranzato, M. A. Gradient episodic memory for continual learning. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc., 2017.

Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A. D., and van de Weijer, J. Class-incremental learning: survey and performance evaluation on image classification. *CoRR*, abs/2010.15277, 2020a.

Masana, M., Twardowski, B., and Van de Weijer, J. On class orderings for incremental learning. In *ICML Workshop on Continual Learning*, 2020b.

Prabhu, A., Torr, P., and Dokania, P. Gdumb: A simple approach that questions our progress in continual learning. In *The European Conference on Computer Vision (ECCV)*, August 2020.

Rannen, A., Aljundi, R., Blaschko, M. B., and Tuytelaars, T. Encoder based lifelong learning. In *Proceedings of the IEEE International Conference on Computer Vision*, pp. 1320–1328, 2017.

Rebuffi, S.-A., Kolesnikov, A., Sperl, G., and Lampert, C. H. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. doi: 10.1007/s11263-015-0816-y.

Serra, J., Suris, D., Miron, M., and Karatzoglou, A. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pp. 4548–4557. PMLR, 2018.

Shin, H., Lee, J. K., Kim, J., and Kim, J. Continual learning with deep generative replay. In Guyon, I., von Luxburg, U., Bengio, S., Wallach, H. M., Fergus, R., Vishwanathan, S. V. N., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pp. 2990–2999, 2017.

Thrun, S. Is learning the n-th thing any easier than learning the first? In Touretzky, D., Mozer, M. C., and Hasselmo, M. (eds.), *Advances in Neural Information Processing Systems*, volume 8. MIT Press, 1996.

Wu, C., Herranz, L., Liu, X., Wang, Y., van de Weijer, J., and Raducanu, B. Memory replay gans: learning to generate images from new categories without forgetting. In *Conference on Neural Information Processing Systems (NIPS)*, 2018.

Wu, Y., Chen, Y., Wang, L., Ye, Y., Liu, Z., Guo, Y., and Fu, Y. Large scale incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 374–382, 2019.

Yu, L., Twardowski, B., Liu, X., Herranz, L., Wang, K., Cheng, Y., Jui, S., and Weijer, J. v. d. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 6982–6991, 2020.

Zhang, J., Zhang, J., Ghosh, S., Li, D., Tasci, S., Heck, L., Zhang, H., and Kuo, C.-C. J. Class-incremental learning via deep model consolidation. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pp. 1131–1140, 2020.