Achieving a Better Stability-Plasticity Trade-off via Auxiliary Networks in Continual Learning

Sanghwan Kim Lorenzo Noci Antonio Orvieto Thomas Hofmann
ETH Zürich
Zürich, Switzerland
{sanghwan.kim, lorenzo.noci, antonio.orvieto, thomas.hofmann}@inf.ethz.ch

Abstract

In contrast to the natural capabilities of humans to learn new tasks in a sequential fashion, neural networks are known to suffer from catastrophic forgetting*, where the model's performances on old tasks drop dramatically after being optimized for a new task. Since then, the continual learning (CL) community has proposed several solutions aiming to equip the neural network with the ability to learn the current task (*plasticity*) while still achieving high accuracy on the previous tasks (*stability*). Despite remarkable improvements, the plasticity-stability trade-off is still far from being solved and its underlying mechanism is poorly understood. In this work, we propose* Auxiliary Network Continual Learning *(ANCL), a novel method that applies an additional auxiliary network which promotes plasticity to the continually learned model which mainly focuses on stability. More concretely, the proposed framework materializes in a regularizer that naturally interpolates between plasticity and stability, surpassing strong baselines on task incremental and class incremental scenarios. Through extensive analyses on ANCL solutions, we identify some essential principles beneath the stability-plasticity trade-off. The code implementation of our work is available at* https://github.com/kim-sanghwan/ANCL.

1. Introduction

The continual learning (CL) model aims to learn from current data while still maintaining the information from previous training data. The naive approach of continuously fine-tuning the model on sequential tasks, however, suffers from *catastrophic forgetting* [8, 21]. Catastrophic forgetting occurs in a gradient-based neural network because the updates made with the current task are likely to override the model weights that have been changed by the gradients from the old tasks.

Catastrophic forgetting can be understood in terms of *stability-plasticity dilemma* [22], one of the well-known challenges in continual learning. Specifically, the model not only has to generalize well on past data (*stability*) but also learn new concepts (*plasticity*). Focusing on stability will hinder the neural network from learning the new data, whereas too much plasticity will induce more forgetting of the previously learned weights. Therefore, CL model should strike a balance between stability and plasticity.

There are various ways to define the problem of CL. Generally speaking, it can be categorized into three scenarios [27] : *Task Incremental Learning* (TIL), *Domain Incremental Learning* (DIL), and *Class Incremental Learning* (CIL). In TIL, the model is informed about the task that needs to be solved; the task identity is given to the model during the training session and the test time. In DIL, the model is required to solve only one task at hands without the task identity. In CIL, the model should solve the task itself and infer the task identity. Since the model should discriminate all classes that have been seen so far, it is usually regarded as the hardest continual learning scenario. Our study performs extensive evaluations on TIL and CIL setting which will be further explained in Sec. 4.

Recently, several papers [19, 20, 28, 31] proposed the usage of an auxiliary network or an extra module that is solely trained on the current dataset, with the purpose of combining this additional structure with the previous network or module that has been continuously trained on the old datasets. For example, *Active Forgetting with synaptic Expansion-Convergence* (AFEC) [28] regularizes the weights relevant to the current task through a new set of network parameters called the expanded parameters based on weight regularization methods. The expanded parameters are solely optimized on the current task and are allowed to forget the previous ones. As a result, AFEC can reduce potential negative transfer by selectively merging the old parameters with the expanded parameters. The stability-plasticity balance in AFEC is adjusted via hyperparameters which scale the regularization terms for remembering the old tasks and learning the new tasks.

The authors of the above papers propose to mitigate the stability-plasticity dilemma by infusing plasticity through the auxiliary network or module (detailed explanation in Appendix A). However, a precise characterization of the interactive mechanism between the previous model and the auxiliary model is still missing in the literature. Therefore, in this paper, we first formalize the framework of CL that adopts the auxiliary network called *Auxiliary Network Continual Learning* (ANCL). Given this environment, we then investigate the stability-plasticity trade-off through various analyses from both a theoretical and empirical point of view.

Our main contributions can be summarized as follows:

- We propose the framework of *Auxiliary Network Continual Learning* (ANCL) that can naturally incorporate the auxiliary network into a variety of CL approaches as a plug-in method (Sec. 3.1).
- We empirically show that ANCL outperforms existing CL baselines on both CIFAR-100 [16] and Tiny ImageNet [17] (Sec. 4).
- Furthermore, we perform three analyses to investigate the stability-plasticity trade-off within ANCL (Sec. 5): *Weight Distance*, *Centered Kernel Alignment*, and *Mean Accuracy Landscape*.

2. Related Work

Continual learning approaches can be roughly categorized into weight regularization [2, 5, 14, 28], knowledge distillation [6, 13, 18, 31], memory replay [4, 26], bias correction [7, 12, 29, 32], and dynamic structure [1, 20, 30].

Weight Regularization Method: A standard way to alleviate catastrophic forgetting is to include a regularization term which binds the dynamics of each network's parameter to the corresponding parameter of the old network. For example, *Elastic Weight Consolidation* (EWC) [14] calculates the regularizer through the approximation of Fisher Information Matrix (FIM). *Memory Aware Synapses* (MAS) [2] proposes the regularizer which accumulates the changes of each parameter throughout the update history. Recently, [28] suggests a biologically inspired argument to propose Active Forgetting with synaptic Expansion-Convergence (AFEC) where an additional regularization term associated with expanded parameters (or auxiliary network) is added to the loss of EWC.

Knowledge Distillation Method: A separate line of work adopts knowledge distillation [3, 11] which was originally designed to train a more compact student network from a larger teacher network. In this way, the main network can emulate the activation or logit of the previous (or old) network while learning a new task. For instance, *Learning without Forgetting* (LwF) [18] proposes to learn the soft target generated by the old network while *less-forgetting learning* (LFL) [13] regularizes the difference between the activations of the main network and the old network. Based on LwF, *Learning without Memorizing* (LwM) [6] takes advantage of the attention of the previous network to train the current network. A recent distillation approach called *Deep Model Conolidation* (DMC) [31] proposes *double distillation loss* to resolve the asymmetric property of training between old and new classes using a new network (or auxiliary network) and an unlabeled auxiliary dataset.

Memory Replay Method: Unlike the previous methods, replay-based methods keep a part of the previous data (or exemplars) in a memory buffer. Then, a model is trained on the current dataset and the previous exemplars to prevent the forgetting of the previous tasks. *Incremental Classifier and Representation Learning* (iCaRL) [26] proposes the usage of the memory buffer derived from LwF [18]. Then, iCaRL calculates the mean feature representations for each class and selects the exemplars iteratively so that the mean of the exemplars is closer to the class mean in feature space, which is called *herding* sampling strategy. Another replay-based approach named *End-to-End Incremental Learning* (EEIL) [4] introduces an additional stage called *balanced training* to fine-tune the model on a balanced dataset. The balanced dataset consists of the equal number of exemplars from each class that have been seen so far.

Bias Correction Method: In memory replay methods, the network is trained on the highly unbalanced dataset composed of the few exemplars from the previous task and fresh new samples from the new ones. As a result, the network is biased towards the data of new tasks, and this can lead to distorted predictions of the model, which is called *task-recency bias*. To solve this problem, *Bias Correction* (BiC) [29] introduces a two-stage training where they perform the main training in the first stage and subsequently mitigate the bias through a linear transformation. Likewise, Weight Aligning (WA) [32] proposes two-stage training. The first stage is equal to that of BiC and they normalize the weight vectors of the new classes and the old classes to reduce the bias in the second stage. Another bias correction method called *Learning a Unified Classifier Incrementally via Rebalancing* (LUCIR) [12] alleviates task-recency bias by including three components into their training: cosine normalization, less-forget constraint, and inter-class. Built upon LUCIR, *Pooled Outputs Distillation Network* (POD-Net) [7] applies pooled out distillation loss and local similarity classifier.

Dynamic Structure Method: Dynamic structure approaches use masking for each task or expansion of the model to prevent forgetting and increase the model capacity to learn a new task. For instance, *Conditional Channel Gated Networks* (CCGN) [1] dynamically adds an extra convolutional layer whenever the model learns a new task

and it is only optimized for the new data. *Adaptive Aggregation Networks* (AANets) [20] expands a Residual Network (ResNet) [10] to have the two types of residual block at each residual level to balance stability and plasticity: a stable block that is trained on a first task and frozen afterward and a plastic block that is freely trained on a current task. Another dynamic structure method called *Dynamically Expandable Representation Learning* (DER) [30] suggests to expand a feature extractor. The new feature extractor is trained solely on the current dataset with channel level masking and the whole model is fine-tuned on balanced dataset.

3. Method

In this Section, we propose *Auxiliary Network Continual Learning* (ANCL), a framework which combines original *Continual Learning* (CL) approaches with an auxiliary network (Sec. 3.1). In addition, we explain the detailed training steps of ANCL (Sec. 3.2).

3.1. The Formulation of Auxiliary Network Continual Learning

ANCL applies the auxiliary network trained on the current task to the continually learned previous network to achieve a balance between stability and plasticity. Fig. 1 illustrates the conceptual difference between CL and ANCL, where CL can be any continual learning method that includes a regularizer that depends on the old network. Before training on the dataset D_t of task t, CL freezes and copies the previous continual model θ^{CL}_{t-1} that has been trained until task $t-1$ as the old network $\theta^*_{1:t-1}$. Then, the old network regularizes the main training through the regularization strength λ. We can formally define the loss of CL on task t as follows:

$$\mathcal{L}_{\text{CL}} = \mathcal{L}_{\text{t}}(\theta) + \Omega(\theta; \theta^*_{1:t-1}, \lambda), \tag{1}$$

where the first term denotes a task-specific loss with respect to main network weights $\theta \in \mathbb{R}^P$ and the second term represents the regularizer that binds the dynamic of the network parameters θ to the old network parameters $\theta^*_{1:t-1} \in \mathbb{R}^P$. $\lambda \in \mathbb{R}$ is the regularization strength which is usually selected by a grid search procedure. These two loss terms can be calculated on the current dataset D_t or on the combined dataset D^+_t (current dataset D_t + previous exemplars $P_{1:t-1}$) depending on the method to which it is applied. In classification problems, the task-specific loss becomes cross-entropy loss. The original CL approaches mainly focus on retaining the old knowledge obtained from the previous tasks by preventing large updates that would depart significantly from the old weights $\theta^*_{1:t-1}$. However, this might harmfully restrict the model's ability to learn the new knowledge, which will hinder the right balance between stability and plasticity.

Methods	$\Omega(\theta; \theta^*, \lambda)$
EWC [14]	$\frac{\lambda}{2}\sum_i F_i(\theta_i - \theta^*_i)^2$
MAS [2]	$\frac{\lambda}{2}\sum_i M_i(\theta_i - \theta^*_i)^2$
LwF [18]	$\lambda \sum_{c=1}^{C_{1:t}} -y^c(x;\theta^*)\log y^c(x;\theta)$
LFL [13]	$\lambda \lVert f(x;\theta) - f(x;\theta^*) \rVert_2^2$
iCaRL [26]	$\lambda \sum_{c=1}^{C_{1:t}} -y^c(x;\theta^*)\log y^c(x;\theta)$
BiC [29]	$\lambda \sum_{c=1}^{C_{1:t}} -y^c(x;\theta^*)\log y^c(x;\theta)$
LUCIR [12]	$\lambda(1 - \langle \bar{f}(x;\theta), \bar{f}(x;\theta^*) \rangle)$
PODNet [7]	$\lambda[\sum_{l=1}^{L-1} \mathcal{L}_{\text{POD-spatial}}(f_l(x;\theta), f_l(x;\theta^*)) + \mathcal{L}_{\text{POD-flat}}(f_L(x;\theta), f_L(x;\theta^*))]$

Table 1. The definition of $\Omega(\theta; \theta^*, \lambda)$ depends on different methods. The first four methods (EWC, MAS, LwF, and LFL) are calculated on the current dataset D_t while the last four methods (iCaRL, BiC, LUCIR, and PODNet) are measured on the combined dataset D^+_t with the memory buffer. Detailed explanation and loss function of each method can be found in Appendix B.

On the contrary, ANCL keeps two types of network to maintain this balance: (1) the auxiliary network θ^*_t, which is solely optimized on the current task t allowing for forgetting (*plasticity*) and (2) the old network $\theta^*_{1:t-1}$ that has been sequentially trained until task $t-1$ (*stability*). Then, both models are used to construct the regularizers in the following objective:

$$\mathcal{L}_{\text{ANCL}} = \mathcal{L}_{\text{t}}(\theta) + \Omega(\theta; \theta^*_{1:t-1}, \lambda) + \Omega(\theta; \theta^*_t, \lambda_a), \tag{2}$$

where the first two terms are the same as in Eq. (1) and the last term promotes the learning of the new task t based on the parameters of the auxiliary network $\theta^*_t \in \mathbb{R}^P$ and the regularization strength $\lambda_a \in \mathbb{R}$. Note that the new regularizer $\Omega(\theta; \theta^*_t, \lambda_a)$ is obtained in the same way as the original method, and thus we expect our model to naturally merge the old feature representation (or weight itself) with the new one. This is mathematically explained in Appendix E where we analyze and compare the gradient of CL and ANCL. Moreover, we initialize the auxiliary network with the old network parameters so that the auxiliary model is weakly biased toward the old model, thus facilitating the integration of the two models in the corresponding regularizers of Eq. 2.

In Tab. 1, we show how $\Omega(\theta; \theta^*, \lambda)$ materializes in selected CL methods, given the current network parameters θ, the reference network parameters θ^* (the old or auxiliary network), and the regularization strength λ. For example, the original CL loss of EWC can be expressed as follows by applying Tab. 1 to Eq. (1):

$$\mathcal{L}_{\text{EWC}} = \mathcal{L}_{\text{t}}(\theta) + \frac{\lambda}{2}\sum_i F_{1:t-1,i}(\theta_i - \theta^*_{1:t-1,i})^2 \tag{3}$$

where F_t is the approximation of the Fisher Information Matrix of the old network parameters $\theta^*_{1:t-1}$ and the reg-

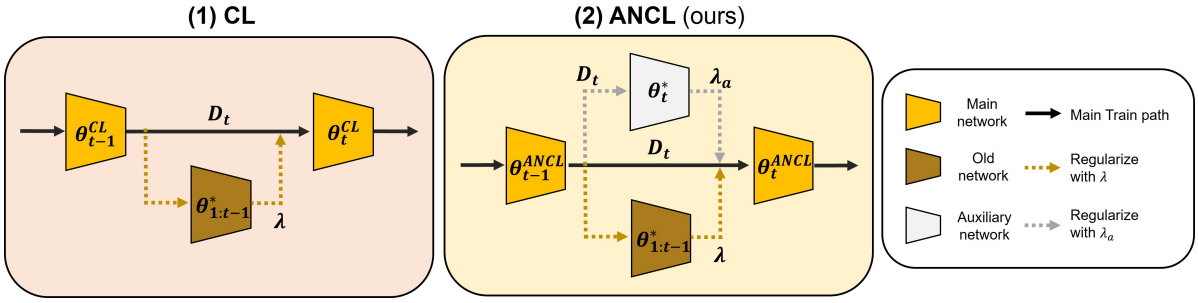

Figure 1. Conceptual comparison of *Continual Learning* (CL) and *Auxiliary Network Continual Learning* (ANCL) (ours) on task t. (1) CL: the previous weights θ^{CL}_{t-1} are frozen in the old network as $\theta^*_{1:t-1}$ and the old network regularizes the main training through λ. (2) ANCL: the auxiliary network initialized by θ^{ANCL}_{t-1} is trained on the dataset D_t and then frozen as θ^*_t. It regularizes the main training via λ_a in addition to the regularization of the old network.

ularization term calculates the difference between the network parameter θ_i $(i = 1, \ldots, P)$ and the corresponding old network parameter $\theta^*_{1:t-1,i}$. Next, if we apply ANCL to EWC to build the loss of the so-called *Auxiliary Network EWC* (A-EWC), we get:

$$\mathcal{L}_{\text{A-EWC}} = \mathcal{L}_{\text{EWC}} + \frac{\lambda_a}{2} \sum_i F_{t,i}(\theta_i - \theta^*_{t,i})^2 \qquad (4)$$

which adds the new regularizer built upon the auxiliary network parameter $\theta^*_{t,i}$. The application of ANCL to other methods in Tab. 1 can be found in Appendix C.

In ANCL, the auxiliary network accounts for plasticity while the old network stands for stability. Furthermore, both networks are equally reflected through the regularization term Ω, thus preventing bias toward either network. Adjusting both regularizers via λ and λ_a, ANCL is more likely to achieve a better stability-plasticity balance than CL, under proper hyperparameter tuning. How ANCL solutions appropriately weigh the old network and the auxiliary network is further investigated in Sec. 5. Furthermore, we mathematically analyze and compare the gradient of CL and ANCL losses in terms of the stability-plasticity trade-off in Appendix E.

Comparison with AFEC The auxiliary network of ANCL works similarly to the expanded parameter of AFEC with respect to adding an additional loss term, but ANCL uses a *method-dependent* regularizer compared to the *fixed and independent* regularizer of AFEC based on Fisher Information Matrix. In other words, while AFEC plugs in the same loss term calculated on the expanded parameter to every method, ANCL generates the loss term from the auxiliary network in the same way as the original CL where ANCL is applied. ANCL adopts two regularizers of the same type to equally represent stability and plasticity which is explicitly controlled by the scaling hyperparameters (λ and λ_a in Eq. (2)). If the two regularizers are of different types like in AFEC, each regularizer will change in different magnitude at every epoch. Consequently, it is less likely that the model will arrive at the best equilibrium. In Appendix D, we empirically show that ANCL outperforms AFEC.

3.2. Algorithm

Detailed training steps of our ANCL framework is summarized in Alg. 1. This is applicable to all ANCL methods if an appropriate ANCL loss is substituted in the algorithm. Given the training over total N tasks, Lines 3-4 shows the training of the main network weight θ with task-specific loss $\mathcal{L}_t$ on the dataset of task 1. Then, the optimal weight θ^* for task 1 is saved as the old weight $\theta^*_{1:1}$ in Line 5. On task $t(> 1)$, the auxiliary weight θ_t is initialized by the previous old weight $\theta^*_{1:t-1}$ and trained with task-specific loss $\mathcal{L}_t$ (Lines 7-9). In Line 10, the auxiliary weight θ^*_t is frozen and saved. Subsequently, the main network is trained with ANCL loss explained in Eq. (2) (Lines 11-12). The optimal main network on task t is frozen and saved as an old weight $\theta^*_{1:t-1}$ for the next loop (Line 13). If Lines 7-10 are skipped and "ANCL Loss (Eq. (2))" in Line 12 is replaced with "CL Loss (Eq. (1))", Alg. 1 becomes the original CL algorithm.

4. Experiment

Benchmark: CIFAR-100 [16] and Tiny ImageNet [17] are chosen to evaluate ANCL. CIFAR-100 contains 60,000 colored images from 100 classes with the size of 32×32. For task incremental scenario, CIFAR-100 is divided into 10 tasks of 10 classes each and 20 tasks of 5 classes each to construct two benchmarks: **(1) CIFAR-100/10** and **(2) CIFAR-100/20**. In addition, we build two more benchmarks for class incremental scenario: **(5) CIFAR-100/6** and **(6) CIFAR-100/11**. In these settings, 50 classes are learned at an initial phase and the rest classes are learned sequentially with 10 classes or 5 classes per phase after the initial one. Tiny ImageNet consists of 110,000 colored images

Algorithm 1: ANCL Algorithm

Input: Main network weight θ, Auxiliary network weight θ_t^*, Old network weight $\theta_{1:t-1}^*$, Hyperparameters λ, λ_a

Output: Optimal main network weight θ^*

for *task t = 1, 2, .., N* **do**
  if $t = 1$ **then**
    // Train main network
    for *epoch e = 1, 2, .., E* **do**
      Train θ with task-specific loss $\mathcal{L}_t$ to obtain θ^* on task 1
    // Save main network weight as old network weight
    Freeze and save θ^* as $\theta_{1:1}^*$
  else
    // Initialize auxiliary network
    θ_t = copy($\theta_{1:t-1}^*$)
    // Train auxiliary network
    for *epoch e = 1, 2, .., E* **do**
      Train θ_t with task-specific loss $\mathcal{L}_t$ to obtain θ_t^* on task t
    // Save auxiliary network weight
    Freeze and save θ_t^*
    // Train main network
    for *epoch e = 1, 2, .., E* **do**
      Train θ with ANCL Loss (Eq. (2)) to obtain θ^* on task t
    // Save main network weight as old network weight
    Freeze and save θ^* as $\theta_{1:t-1}^*$

(size 64×64) from 200 classes which are resized as 32×32 for both training and inference. We equally divide Tiny ImageNet into 10 and 20 tasks to build two benchmarks for task incremental scenario: **(3) TinyImagenet-200/10** and **(4) TinyImagenet-200/20**. For class incremental scenario, the model is trained on 100 classes at an initial phase and then trained continuously on 10 classes or 5 classes per phase after the initial one: **(7) TinyImagenet-200/11** and **(8) TinyImagenet-200/21**.

Architecture: We select Resnet32 [10] for all benchmarks which is commonly chosen in the literature of continual learning [7,12,26,29,32]. For task incremental scenario, multi-head layer is deployed instead of the last layer in Resnet32 to generate an output with a task identity. In class incremental scenario, single-head evaluation is adopted due to the absence of task identity during inference.

Implementation: The model is trained from scratch and every experiment is carried out 3 times with different seeds to generate averaged metrics. SGD optimizer with momentum 0.9 and batch size 128 is applied to all experiments. In task incremental learning, we evaluate our methods on a strict setting of continual learning where the previous data is not visited again. In class incremental learning, we relax the regularization of accessing previous data. 20 exemplars per class of the old training data are selected by herding sampling strategy and stored in the memory buffer (more details in Appendix F.1).

Gridsearch on Parameters: We conduct a comprehensive hyperparameter search for all methods and report the best scores for a fair comparison. We follow the way AFEC [28] performs the grid search on λ and λ_a. First, an extensive grid search is made on λ using the original CL loss and λ is fixed afterward. Then, we use ANCL loss to conduct the grid search of λ_a. Grid search result of λ and λ_a for all benchmarks can be found in Appendix F.4

Evaluation Metrics: In task incremental scenario, averaged accuracy (AAC) for T task is calculated after the training of all tasks. In class incremental scenario, averaged incremental accuracy ($AIAC$) is used instead:

$$AAC = \frac{1}{T}\sum_{i=1}^{T} A_{T,i}, \quad AIAC = \frac{1}{N+1}\sum_{i=0}^{N} A_i. \tag{5}$$

In AAC, $A_{j,k}$ is the test accuracy of task k after the continual learning of task j. In AIAC, A_i denotes the test accuracy of the classes seen so far at the ith phase for the benchmark consisting of $N+1$ phases including the initial one.

Baseline: Fine-tuning is the naive approach that a model is fine-tuned on each task (or each phase), which is regarded as a lowerbound and joint uses the whole dataset to train the model, which becomes an upperbound. In task incremental setting, we evaluate EWC [14], MAS [2], LwF [18], LFL [13], LwM [6], and DMC [31]. For a fair comparison, DMC is modified to only use the original dataset like other methods instead of an unlabeled auxiliary dataset. Then, we apply ANCL to the original CL approaches. In class incremental setting, we test EEIL [4], iCaRL [26], BiC [29], LUCIR [12], and PODNet [7] with their applications to ANCL.

Evaluation on Task Incremental Scenario: Tab. 2 shows that applying ANCL consistently gives an extra boost in accuracy by 1-3 % compared to naive CL and A-LwF achieves the best accuracy in all benchmarks. ANCL can be more compatible with specific methods than others. For example in benchmark (1), applying ANCL outperforms MAS baseline by 3.87 % while it improves LFL baseline only by 0.73 %. This is because ANCL is more effective when the two regularizers in Eq. (2) are well suited to each other and CL has less plasticity at the beginning. The detail accuracy for all tasks can be found in Appendix F.2.

Evaluation on Class Incremental Scenario: In Tab. 3, we can clearly see that ANCL surpasses CL baselines in all methods by 1-3 % including state-of-the-art (SOTA) methods such as BiC [29], LUCIR [12], and PODNet [7]. Similarly to Tab. 2, ANCL is more compatible with LUCIR and

Methods	CIFAR-100 (1)	CIFAR-100 (2)	Tiny ImageNet (3)	Tiny ImageNet (4)
Fine-tuning	$38.90_{\pm 1.59}$	$27.81_{\pm 0.80}$	$28.51_{\pm 0.75}$	$20.35_{\pm 1.70}$
Joint	$89.64_{\pm 0.37}$	$93.42_{\pm 0.27}$	$67.98_{\pm 1.15}$	$70.02_{\pm 2.63}$
LwM [6]	$78.46_{\pm 1.11}$	$78.27_{\pm 0.38}$	$59.04_{\pm 0.63}$	$59.78_{\pm 1.08}$
DMC [31]	$51.90_{\pm 0.91}$	$53.72_{\pm 1.11}$	$45.65_{\pm 0.15}$	$44.50_{\pm 0.73}$
EWC [14]	$58.13_{\pm 0.87}$	$60.03_{\pm 1.23}$	$50.10_{\pm 0.78}$	$52.53_{\pm 0.91}$
w/ ANCL (ours)	$60.86_{\pm 1.46}$	$62.47_{\pm 0.65}$	$52.49_{\pm 0.71}$	$53.86_{\pm 0.88}$
MAS [2]	$60.56_{\pm 0.82}$	$59.35_{\pm 1.09}$	$49.50_{\pm 1.18}$	$51.79_{\pm 0.51}$
w/ ANCL (ours)	$64.43_{\pm 1.17}$	$60.70_{\pm 1.11}$	$50.11_{\pm 1.09}$	$53.58_{\pm 0.73}$
LwF [18]	$78.87_{\pm 0.69}$	$76.96_{\pm 0.83}$	$59.04_{\pm 0.62}$	$62.09_{\pm 0.59}$
w/ ANCL (ours)	$79.42_{\pm 0.57}$	$79.99_{\pm 0.59}$	$60.96_{\pm 0.76}$	$63.79_{\pm 0.41}$
LFL [13]	$74.50_{\pm 0.57}$	$74.27_{\pm 0.72}$	$60.20_{\pm 0.66}$	$58.47_{\pm 0.95}$
w/ ANCL (ours)	$75.23_{\pm 0.67}$	$74.68_{\pm 1.04}$	$61.32_{\pm 0.68}$	$58.98_{\pm 0.74}$

Table 2. The averaged accuracy (%) on the benchmarks (1)-(4). Reported metrics are averaged over 3 runs (averaged accuracy $\pm$ standard error). ANCL methods are colored gray.

Methods	CIFAR-100 (5)	CIFAR-100 (6)	Tiny ImageNet (7)	Tiny ImageNet (8)
Fine-tuning	$45.78_{\pm 0.90}$	$43.57_{\pm 1.33}$	$27.44_{\pm 0.85}$	$24.18_{\pm 0.98}$
Joint	$67.84_{\pm 1.35}$	$66.40_{\pm 0.86}$	$46.85_{\pm 0.74}$	$46.02_{\pm 0.55}$
EEIL [4]	$49.81_{\pm 1.12}$	$48.65_{\pm 0.94}$	$28.68_{\pm 0.93}$	$28.00_{\pm 0.73}$
iCaRL [26]	$58.05_{\pm 0.94}$	$57.11_{\pm 0.77}$	$39.04_{\pm 0.61}$	$37.90_{\pm 0.98}$
w/ ANCL (ours)	$61.22_{\pm 0.88}$	$59.13_{\pm 0.68}$	$41.46_{\pm 0.85}$	$39.91_{\pm 1.02}$
BiC [29]	$56.74_{\pm 1.33}$	$55.73_{\pm 1.21}$	$40.56_{\pm 0.44}$	$39.21_{\pm 0.69}$
w/ ANCL (ours)	$58.32_{\pm 1.27}$	$58.23_{\pm 1.44}$	$42.61_{\pm 0.65}$	$40.56_{\pm 0.51}$
LUCIR [12]	$56.06_{\pm 0.45}$	$57.91_{\pm 0.57}$	$35.17_{\pm 0.58}$	$30.02_{\pm 0.13}$
w/ ANCL (ours)	$60.20_{\pm 0.78}$	$60.04_{\pm 0.80}$	$37.89_{\pm 0.74}$	$31.65_{\pm 0.25}$
PODNet [7]	$61.80_{\pm 0.77}$	$59.22_{\pm 0.93}$	$40.28_{\pm 0.36}$	$38.50_{\pm 0.49}$
w/ ANCL (ours)	$63.15_{\pm 0.62}$	$60.44_{\pm 0.67}$	$41.11_{\pm 0.23}$	$40.11_{\pm 0.64}$

Table 3. The averaged incremental accuracy (%) on the benchmarks (5)-(8). Reported metrics are averaged over 3 runs (averaged accuracy $\pm$ standard error). ANCL methods are colored gray.

iCaRL compared to others thereby A-iCaRL being able to compete with or even outperform the stronger baseline of PODNet. We also plot how each method's accuracy at each phase changes and report the final accuracy in Appendix F.3.

5. Stability-Plasticity Trade-off Analysis

In this chapter, we perform three analyses on (1) CIFAR-100/10 to study how the stability-plasticity dilemma is solved through ANCL: *Weight Distance*, *Centered Kernel Alignment*, and *Mean Accuracy Landscape*. For simplification, λ is first selected by grid search using CL loss on current task $t = 2$ and then fixed. Then, ANCL solutions with different λ_a are compared in various analyses. A training regime similar to the one in [23] is adopted for a fair comparison, which is explained in detail in Appendix G.1.

5.1. Weight Distance

If the parameters change less, it is reasonable to expect that less forgetting will occur. According to [24], forgetting $\mathcal{F}_1$ on task 1 is bounded using Taylor expansion of the loss as follows:

$$\mathcal{F}_1 = \mathcal{L}_1(\hat{\theta}_2) - \mathcal{L}_1(\hat{\theta}_1) \tag{6}$$

$$\approx \frac{1}{2}(\hat{\theta}_2 - \hat{\theta}_1)^T \nabla^2 \mathcal{L}_1(\hat{\theta}_1)(\hat{\theta}_2 - \hat{\theta}_1) \tag{7}$$

$$\leq \frac{1}{2}\lambda_1^{max} \|\hat{\theta}_2 - \hat{\theta}_1\|_2^2 \tag{8}$$

where $\mathcal{L}_1$ is the empirical loss on task 1 and $\nabla^2 \mathcal{L}_1(\hat{\theta}_1)$ is the Hessian for $\mathcal{L}_1$ at $\hat{\theta}_1$. λ_1^{max} is the maximum eigenvalue of $\nabla^2 \mathcal{L}_1(\hat{\theta}_1)$. Above inequality implies that the bound of forgetting $\mathcal{F}_1$ is determined by the norm of the difference between two weights near the minima of task 1 loss.

On task t, we measure the weight distance (WD) from

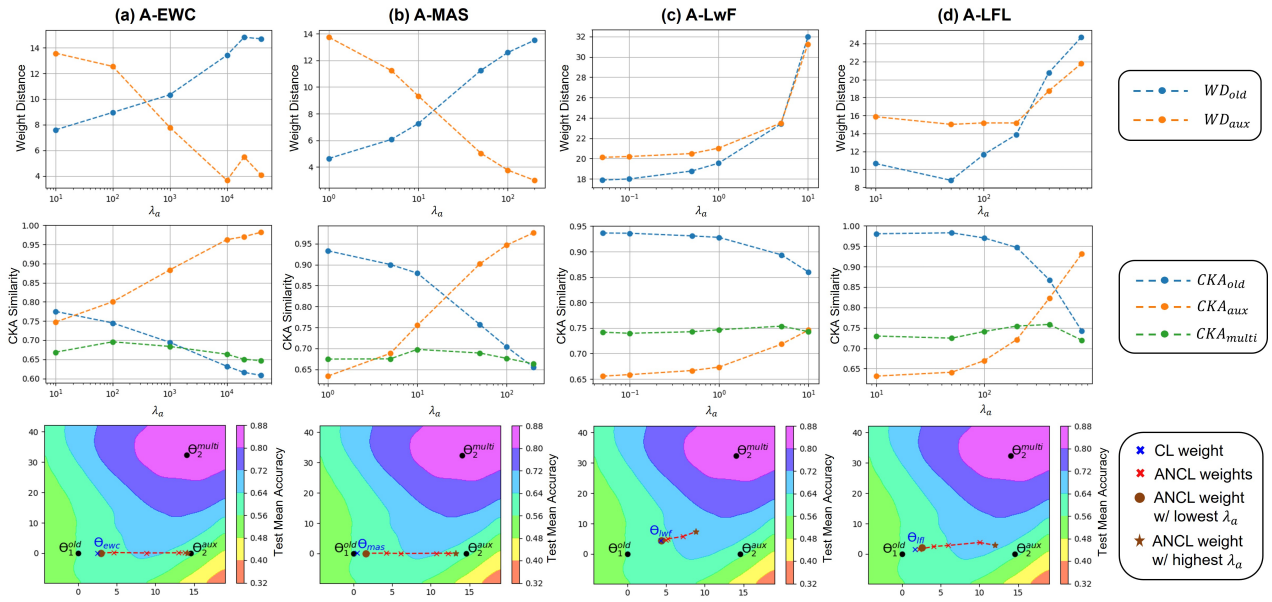

Figure 2. Anaylsis figures on (1) CIFAR-100/10: weight distance (top row), centered kernel alignment (middle row), and mean accuracy landscape (bottom row). The set of λ_a for each ANCL is as follows (λ is fixed): (a) A-EWC ($\lambda = 10000$) - $\lambda_a \in$ [10, 100, 1000, 10000, 20000, 40000], (b) A-MAS ($\lambda = 50$) - $\lambda_a \in$ [1, 5, 10, 50, 100, 200], (c) A-LwF ($\lambda = 10$) - $\lambda_a \in$ [0.05, 0.1, 0.5, 1, 5, 10] and (d) A-LFL ($\lambda = 400$) - $\lambda_a \in$ [10, 50, 100, 200, 400, 800].

the weights of the ANCL models θ_t^{ANCL} to the weights of the old model θ_{t-1}^{old} and the auxiliary model θ_t^{aux} respectively:

$$WD_{old} = \|\theta_t^{ANCL} - \theta_{t-1}^{old}\|_2, \tag{9}$$

$$WD_{aux} = \|\theta_t^{ANCL} - \theta_t^{aux}\|_2. \tag{10}$$

WD analysis is shown in the top row of Fig. 2. We calculate WD with different λ_a which directly adjusts the stability-plasticity trade-off while λ is fixed. The model parameters remain close to the old parameters when λ_a is small, which can be seen on the left side of all WD figures. For A-EWC and A-MAS, WD_{aux} decreases and WD_{old} increases as λ_a becomes larger. This result implies a direct interpolation between the old and auxiliary networks, which is consistent with the analysis of the ANCL gradient in Appendix E. For A-LwF and A-LFL, WD_{aux} becomes relatively smaller than WD_{old} with increasing λ_a but WD_{old} and WD_{aux} are both growing. Unlike EWC and MAS which directly regularize the weights itself, LwF and LFL have more flexibility to remember the previous knowledge by utilizing loss terms based on activations or logits. Therefore, for the distillation approaches, the model weights tends to move relatively closer to the auxiliary weights with increasing λ_a but not directly toward it like EWC or MAS. The difference between the regularization and distillation CL methods and the effect of λ_a on the stability-plasticity trade-off is studied further in the following analyses.

5.2. Centered Kernel Alignment

Centered Kernel Alignment (CKA) [15] measures the similarity of two-layer representations on the same set of data. Given N data and p neurons, the layer activation matrices $R_1 \in \mathbb{R}^{N\times p}$ and $R_2 \in \mathbb{R}^{N\times p}$ are generated by two layers from two independent networks. Then, CKA is defined as:

$$CKA(R_1, R_2) = \frac{HSIC(R_1, R_2)}{\sqrt{HSIC(R_1, R_1)}\sqrt{HSIC(R_2, R_2)}} \tag{11}$$

where $HSIC$ stands for Hilbert-Schmidt Independence Criterion [9]. We use linear $HSIC$ to implement CKA. It is well known that lower layers have relatively higher CKA scores than deeper layers and deeper layers generally contribute to forgetting [25]. In this analysis, we measure three CKA similarity:

$$CKA_{old} = \frac{1}{L}\sum_{l=1}^{L} CKA(R_{t,l}^{ANCL}, R_{t-1,l}^{old}), \tag{12}$$

$$CKA_{aux} = \frac{1}{L}\sum_{l=1}^{L} CKA(R_{t,l}^{ANCL}, R_{t,l}^{aux}), \tag{13}$$

$$CKA_{multi} = \frac{1}{L}\sum_{l=1}^{L} CKA(R_{t,l}^{ANCL}, R_{t,l}^{multi}). \tag{14}$$

where CKA is calculated and averaged over the set of layers $\{1, \ldots, L\}$ in Resnet32. Resnet32 consists of 1 initial convolution layer and 3 residual blocks. In order to measure the output similarity of two networks, we select 10 convolution layers in the last residual block of Resnet32 as our set. R_t^{ANCL}, R_{t-1}^{old} and R_t^{aux} are the activation matrices of the ANCL network, the old network, and the auxiliary network, respectively. R_t^{multi} is the activation output of the multitask model trained on the entire dataset $D_{1:t}$ until the task t. If CKA_{multi} is high, the model generates layer activations similar to those of the multitask model. Then, the model is highly likely to perform well on all tasks like the multitask model, which is the main goal of continual learning.

The middle row of Fig. 2 shows three CKA similarities with different λ_a. In all methods, increasing λ_a results in higher CKA_{aux} and lower CKA_{old}, which can be interpreted to mean that the representations of the ANCL network become more similar to that of the auxiliary network and less similar to that of the old network. We can clearly see that the stability-plasticity trade-off is controlled by λ_a through the interaction between the old and auxiliary networks. On the other hand, if CKA_{multi} reaches the highest score at specific λ_a, that model is highly likely to have the best trade-off. For example, (b) A-MAS and (d) A-LFL achieve the highest CKA_{multi} at $\lambda_a = 10$ and $\lambda_a = 400$ respectively. In general, CKA_{multi} of the distillation methods is higher than that of the regularization methods, which corresponds to the results in Tab. 2 where the distillation methods achieved a higher averaged accuracy compared to the regularization methods.

5.3. Mean Accuracy Landscape

Lastly, we visualize mean accuracy landscape of task 1 and 2 in weight vector space following [23] (details in Appendix G.3). θ_1^{old}, θ_2^{aux}, and θ_2^{multi} are used to build two-dimensional subspace denoting the weights of the old network, the auxiliary network and the multitask network, respectively. Multitask network is trained on whole dataset $D_{1:2}$ until task 2 and thus θ_2^{multi} is located in the highest contour indicating the highest mean accuracy. We project CL (blue) and ANCL (red) weight vectors on the subspace to see how ANCL parameters are shifted on the accuracy landscape with different λ_a. ANCL weights with the lowest λ_a are denoted as a brown circle and λ_a increases following the red dot line. Finally, the red dot line reaches a brown star which indicates ANCL weights with the highest λ_a.

In A-EWC and A-MAS, it is clearly observed that λ_a adjusts the interpolation between the CL weights θ_{CL} and the auxiliary weights θ_2^{aux}. The large λ_a drifts the ANCL weights θ_{ANCL} directly toward θ_2^{aux} and the ANCL with sufficiently small λ_a converges to CL methods. At the interpolation of the old weights θ_1^{old} and the auxiliary weights θ_2^{aux}, the ANCL weight achieves higher mean accuracy located in the higher contour. Similarly in A-LwF and A-LFL, θ_{ANCL} with the lowest λ_a starts near θ_{CL} and tends to move toward the region between θ_1^{old} and θ_2^{aux}. As the distillation methods have more flexibility to retain the previous knowledge, the weights of A-LwF and A-LFL do not directly move toward θ_2^{aux} like those of A-EWC and A-MAS. Because of its flexibility, ANCL with distillation methods can deviate from the interpolation line and climb to the higher contour of mean accuracy. As a result, the best trade-off is made at somewhere between θ_1^{old} and θ_2^{aux}. Again, the mean accuracy landscape figures show the projection of weight in the two-dimensional subspace built by three weights (θ_1^{old}, θ_2^{aux}, and θ_2^{multi}). Therefore, it approximates the relative positions of CL and ANCL weights but does not reflect the exact positions of them in the weight space.

As a result, three analyses strongly support the notion that ANCL is able to achieve a better stability-plasticity trade-off where CKA_{multi} and mean accuracy are the highest. The trade-off is mainly adjusted by the ratio between λ and λ_a. ANCL with high λ_a infuses more plasticity into the model, while ANCL with low λ_a seeks more stability. These results coincides with the analysis of ANCL in Appendix E where the solutions of A-EWC and A-MAS indicate the explicit interpolation between the old and auxiliary weights and the gradients of A-LwF and A-LFL derive the activation (or logit) of the main network toward the interpolated activation (or logit) between the old and auxiliary networks.

6. Conclusion

In our paper, we propose a novel framework called ANCL to pursue the proper balance between stability and plasticity inspired by the recent works [19,20,28,31] adopting an auxiliary network. Our method outperforms the original baselines, including SOTA methods on CIFAR-100 [16] and Tiny ImageNet [17]. To investigate the underlying mechanism of ANCL, we extensively conduct analyses and confirm that the balance is resolved via the interpolation between the old and auxiliary weights. In summary, our work provides a deeper understanding of the interaction between the old network and the auxiliary network, which is the key to recent research on continual learning.

Although ANCL can achieve better stability-plasticity trade-off compare to CL, it should be supported by enough hyperparameter search of λ and λ_a. Therefore, extra computational burdens are required to search appropriate hyperparameters for each method, and results can be variant depending on the scope of grid search. In the future, we will investigate a better way to find these hyperparameters such as in data-driven fashion or inside the optimization process.

References

[1] Davide Abati, Jakub Tomczak, Tijmen Blankevoort, Simone Calderara, Rita Cucchiara, and Babak Ehteshami Bejnordi. Conditional channel gated networks for task-aware continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3931–3940, 2020. 2

[2] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 139–154, 2018. 2, 3, 5, 6

[3] Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541, 2006. 2

[4] 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. 2, 5, 6

[5] Arslan Chaudhry, Puneet K Dokania, Thalaiyasingam Ajanthan, and Philip HS Torr. Riemannian walk for incremental learning: Understanding forgetting and intransigence. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 532–547, 2018. 2

[6] 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. 2, 5, 6

[7] Arthur Douillard, Matthieu Cord, Charles Ollion, Thomas Robert, and Eduardo Valle. Podnet: Pooled outputs distillation for small-tasks incremental learning. In *European Conference on Computer Vision*, pages 86–102. Springer, 2020. 2, 3, 5, 6

[8] Ian J Goodfellow, Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio. An empirical investigation of catastrophic forgetting in gradient-based neural networks. *arXiv preprint arXiv:1312.6211*, 2013. 1

[9] Arthur Gretton, Olivier Bousquet, Alex Smola, and Bernhard Schölkopf. Measuring statistical dependence with hilbert-schmidt norms. In *International conference on algorithmic learning theory*, pages 63–77. Springer, 2005. 7

[10] 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. 3, 5

[11] Geoffrey Hinton, Oriol Vinyals, Jeff Dean, et al. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2(7), 2015. 2

[12] 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. 2, 3, 5, 6

[13] Heechul Jung, Jeongwoo Ju, Minju Jung, and Junmo Kim. Less-forgetting learning in deep neural networks. *arXiv preprint arXiv:1607.00122*, 2016. 2, 3, 5, 6

[14] James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526, 2017. 2, 3, 5, 6

[15] 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. 7

[16] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. 2009. 2, 4, 8

[17] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 7(7):3, 2015. 2, 4, 8

[18] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 2, 3, 5, 6

[19] Guoliang Lin, Hanlu Chu, and Hanjiang Lai. Towards better plasticity-stability trade-off in incremental learning: A simple linear connector. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 89–98, 2022. 1, 8

[20] Yaoyao Liu, Bernt Schiele, and Qianru Sun. Adaptive aggregation networks for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2544–2553, 2021. 1, 2, 3, 8

[21] Michael McCloskey and Neal J Cohen. Catastrophic interference in connectionist networks: The sequential learning problem. In *Psychology of learning and motivation*, volume 24, pages 109–165. Elsevier, 1989. 1

[22] Martial Mermillod, Aurélia Bugaiska, and Patrick Bonin. The stability-plasticity dilemma: Investigating the continuum from catastrophic forgetting to age-limited learning effects, 2013. 1

[23] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. Linear mode connectivity in multitask and continual learning. *arXiv preprint arXiv:2010.04495*, 2020. 6, 8

[24] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. *Advances in Neural Information Processing Systems*, 33:7308–7320, 2020. 6

[25] Vinay V Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. *arXiv preprint arXiv:2007.07400*, 2020. 7

[26] 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. 2, 3, 5, 6

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

[28] Liyuan Wang, Mingtian Zhang, Zhongfan Jia, Qian Li, Chenglong Bao, Kaisheng Ma, Jun Zhu, and Yi Zhong. Afec: Active forgetting of negative transfer in continual learning. *Advances in Neural Information Processing Systems*, 34:22379–22391, 2021. 1, 2, 5, 8

[29] 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. 2, 3, 5, 6

[30] Shipeng Yan, Jiangwei Xie, and Xuming He. Der: Dynamically expandable representation for class incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3014–3023, 2021. 2, 3

[31] 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. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1131–1140, 2020. 1, 2, 5, 6, 8

[32] 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. 2, 5