# A Closer Look at Rehearsal-Free Continual Learning*

James Seale Smith[1], Junjiao Tian[1], Shaunak Halbe[1], Yen-Chang Hsu[2], Zsolt Kira[1]
[1]Georgia Institute of Technology, [2]Samsung Research America

## Abstract

*Continual learning is a setting where machine learning models learn novel concepts from continuously shifting training data, while simultaneously avoiding degradation of knowledge on previously seen classes which may disappear from the training data for extended periods of time (a phenomenon known as the catastrophic forgetting problem). Current approaches for continual learning of a single expanding task (aka class-incremental continual learning) require extensive rehearsal of previously seen data to avoid this degradation of knowledge. Unfortunately, rehearsal comes at a cost to memory, and it may also violate data-privacy. Instead, we explore combining knowledge distillation and parameter regularization in new ways to achieve strong continual learning performance without rehearsal. Specifically, we take a deep dive into common continual learning techniques: prediction distillation, feature distillation, L2 parameter regularization, and EWC parameter regularization. We first disprove the common assumption that parameter regularization techniques fail for rehearsal-free continual learning of a single, expanding task. Next, we explore how to leverage knowledge from a pre-trained model in rehearsal-free continual learning and find that vanilla L2 parameter regularization outperforms EWC parameter regularization and feature distillation. Finally, we explore the recently popular ImageNet-R benchmark, and show that L2 parameter regularization implemented in self-attention blocks of a ViT transformer outperforms recent popular prompting for continual learning methods.*

## 1. Introduction

Deep learning models for machine learning applications are typically trained offline on a large, static dataset. The model is then deployed to the real world with assumptions that the distribution of data it will encounter matches the distribution of data it was trained on. Unfortunately, this assumption does not hold for many applications because the model will encounter a natural distribution shift in the target data over time. These shifts lead to performance degradation, requiring that the model be replaced.

One way to replace a model is to collect additional training data, combine this new training data with the old training data, and then retrain the model from scratch. While this will guarantee high model performance, it is not practical for large-scale applications which may require long training times for the model. This may lead to *high financial [24] and environmental [32] costs* after numerous replacements. Instead, the preferred way is to update the model in the most efficient[1] manner possible. The simplest way to update the model is to train it on only the new training data. However, this leads to a phenomenon known as *catastrophic forgetting* [42], where the model overwrites previously acquired knowledge when learning the new data. This results in drastic performance degradation, or “forgetting”, over the previously learned training data distribution.

The study of catastrophic forgetting is referred to as **continual learning**. In this setting, a model sequentially learns from new task data while avoiding the catastrophic forgetting of previously seen data. This task data typically contain *semantic* distribution shifts (e.g., we encounter new object classes) rather than *covariate* distribution shifts [2] (e.g., we encounter new lighting or background conditions). The goal of the continual learning problem is to find the most *efficient* training strategy to update models which are sequentially trained on these task sequences. Strategies are typically evaluated on metrics such as task performance (e.g., classification accuracy for a classification problem), computational efficiency (e.g., training time), and memory efficiency (e.g., number of parameters stored).

In this paper, we focus on continual learning over a single, expanding classification head. This is different from the multi-task continual learning setting, known as *task-incremental* continual learning, where we learn separate classification heads for each task (and the task label is provided during inference) [23]. Unfortunately, SOTA methods for continual learning without task labels require that a subset of the training data be stored or generated to mix

*This material is based upon work supported by the National Science Foundation under Grant No. 2239292.

[1] W.r.t. to computation and/or memory, depending on the application.

[2] We note here that covariate distribution shifts have been studied in recent continual learning works [10,31], but this is not the focus of our paper. For more discussion on this comparison, the reader is referred to [58].

in with future task data, a strategy referred to as **rehearsal**. Many applications are unable to store this data because they work with *private user data that cannot be stored* long term. For example, some companies will collect user data to update the models in the short term (hours to day) but this data could have a timestamp and need deleting.

In this paper, we take a closer look at *rehearsal-free* strategies for continual learning which do not store training data. Rather than propose a new method, we offer an interesting and impactful new perspective building on existing strategies. Specifically, we start by asking the question: *what type of regularization (parameter-space or prediction-space) is best for rehearsal-free continual learning?* We provide analysis into *how these methods forget* from a feature-drift perspective, and show that parameter regularization is most effective at reducing forgetting in the feature encoder while prediction distillation *using multi-class sigmoid instead of softmax* is most effective for reducing forgetting and bias in the classifier head.

Unfortunately, we show that the gap between rehearsal and rehearsal-free methods remains large. We conjecture that pre-training may help close this gap, leading us to our next question: *what type of regularization (parameter-space or prediction-space) can best leverage a pre-trained model for rehearsal-free continual learning?* We surprisingly find that, while L2 regularization has low accuracy when the model is randomly initialized from scratch, it actually performs best in this pre-training setting and beats out more sophisticated methods, including recent prompting for continual learning methods [54, 60, 61].

Finally, we show that a simple method derived from our findings can even outperform rehearsal-based methods on a standard continual learning benchmark. *In summary, we make the following findings and contributions:*

1. We provide a closer look into rehearsal-free continual learning with best practices, identifying that forgetting largely happens in the later layers. The most effective mitigation is through regularizing the final predictions when pre-training is not available.
2. We extend the above investigations to the scenario where pre-training is available and find that regularizing parameters is more effective than regularizing predictions, pointing out the efficacy of methods can shift dramatically with continual learning problem settings.
3. We achieve SOTA results in the rehearsal-free setting and even outperform recent SOTA prompting for continual learning methods [54, 60, 61].

## 2. Background and Related Work

**Continual Learning**: Continual learning approaches can be organized into a few broad categories which are all useful depending on the problem setting and constraints. One group of approaches expand a model's architecture as new tasks are encountered; these are highly effective for applications where a model growing with tasks is practical [14,34,37,40,51]. We do not consider these methods because the model parameters grow with the number of tasks, but acknowledge that our contributions could be incorporated into these approaches.

Another approach is to regularize the model with respect to past task knowledge while training the new task. This can either be done by regularizing the model in the weight space (i.e., penalize changes to model parameters) [2, 13, 29, 55, 66] or the prediction space (i.e., penalize changes to model predictions) [1, 6, 21, 33, 36]. Regularizing knowledge in the prediction space is done using *knowledge distillation* [20] and it has been found to perform better than model regularization based methods for continual learning when task labels are not given [35, 57].

Rehearsal with stored data [3–5, 7, 8, 15, 16, 22, 28, 39, 47–49, 59] or samples from a generative model [26, 27, 43, 52, 56] is highly effective when storing training data or training/saving a generative model is possible. Unfortunately for many machine learning applications, long-term storage of training data will violate data-privacy, as well as incurring a large memory cost. With respect to the generative model, this training process is much more computationally and memory intensive compared to a classification model and additionally may violate data legality concerns because using a generative model increases the chance of memorizing potentially sensitive data [41]. This motivates us to work on the important setting of *rehearsal-free* approaches to mitigate catastrophic forgetting.

**Online Rehearsal-Free Continual Learning**: Other works have looked at rehearsal-free continual learning from an online "streaming" learning perspective using a frozen, pre-trained model [17, 38]. While these works focus on efficient online learning from a fixed, frozen feature space, we instead analyze non-frozen models which are allowed to train "to convergence" on task data (as is common for offline continual learning [63]). Therefore, our setting is very different from these works.

**Prototype-Based Approaches for Continual Learning**: Prototypes can be leveraged for continual learning as a means to avoid catastrophic forgetting without storing data. Recent methods learn a feature space for prototypes with approaches such as learning an embedding network [65] or leveraging strong augmentations for self-supervised learning [62, 68]. While learning prototypes in an embedding network [65] can better mitigate forgetting compared to cross-entropy classification, we avoid such approaches because training an embedding network with metric learning can often be a hard challenge [68]. While leveraging strong self-supervision to augment data and prototypes can achieve SOTA performance for rehearsal-free continual learning [62, 68], it is not clear if the performance increase is due to mitigating forgetting versus having generally better features due to an expanded dataset of strong data augmentations [11]. Additionally, these approaches *require* a large first-task to learn a strong initial feature space (which is not always valid). In summary, while these advanced strategies perform well in the absence of stored data, we instead offer our work as a different perspective on simple, existing, widely-adopted strategies rather than a complex, SOTA method which requires additional assumptions (e.g., having a large first task).

**Rehearsal-Free Continual Learning**: Recent works learn prompts within a frozen, pre-trained transformer model for continual learning [54, 60, 61]. While effective, this approach assumes that the data within the continual learning sequence can be separated with a pre-trained encoder; because this assumption is often invalid, it is still strongly desired to understand how fine-tuning based approaches forget in the rehearsal-free setting. Other works propose producing images for rehearsal using deep-model inversion [9, 53, 64]. While these methods perform well compared to generative modeling approaches and simply rehearsal from a small number of stored images, we argue that these methods have similar risks to generative approaches. Specifically, model-inversion is a slow process associated with high computational costs in the continual learning setting [53] and inverting images from a trained model can also violate the same data-privacy concerns [25]. This motivates us to ask: "how can we *entirely eliminate rehearsal including stored, trained, or inverted training data?*"

## 3. Preliminaries

**Continual Learning**: In continual learning, a model is shown labeled data corresponding to $M$ semantic object classes $c_1, c_2, \dots, c_M$ over a series of $N$ tasks corresponding to non-overlapping subsets of classes. We use the notation $\mathcal{T}_n$ to denote the set of classes introduced in task $n$, with $|\mathcal{T}_n|$ denoting the number of object classes in task $n$. Each class appears in only a single task, and the goal is to incrementally learn to classify new object classes as they are introduced while retaining performance on previously learned classes. To describe our inference model, we denote $\theta_{i,n}$ as the model $\theta$ at time $i$ that has been trained with the classes from task $n$. For example, $\theta_{n,1:n}$ refers to the model trained during task $n$ and the linear classification heads associated with all tasks up to and including class $n$. We drop the second index when describing the model trained during task $n$ with all linear classification heads (for example, $\theta_n$).

In this paper, we deal with the *class-incremental continual learning setting* rather than the *task-incremental continual learning* setting. Class-incremental continual learning is challenging because the learner must support classification across all classes seen up to task $n$ [23] (i.e., *no task labels are provided to the learner during inference*). Task-incremental continual learning is a simpler *multi-task* setting where the task labels are given during both training and inference.

## 4. Rehearsal-Free Regularization

When training on a new task $n$, the key to mitigating forgetting is to transfer knowledge from a "checkpoint" model, $\theta_{n-1}$ (which is copied and frozen at the end of task $n-1$), into the model being updated, $\theta_n$. In this section, we first review three classic ways to transfer knowledge in continual learning which can be described as "rehearsal-free". These approaches are visualized in Figure 1, and we encourage the reader to refer back to Figure 1 throughout reading this section. We then argue that one of these methods, prediction distillation, is more important for transferring knowledge from a model's *classifier*, whereas the other two methods, parameter regularization and feature distillation, are more important for transferring knowledge from a model's *feature encoder*. We will use this section as a foundation to motivate and understand the findings presented in Section 5.

### 4.1. Parameter Space Regularization

One of the earliest approaches for continual learning, EWC, proposed regularizing the model in the *model parameter space* [29]. At a high level, this approach searches for a solution in each new task that lies within the weight space of solutions to the previous tasks. This is done by calculating the L2 distance between each model parameter in $\theta_{n-1}$ and each model parameter in $\theta_n$, or:

$$\mathcal{L}_{ewc} = \sum_{j=1}^{N_{params}} F_{n-1}^{jj} \left(\theta_n^j - \theta_{n-1}^j\right)^2 \tag{1}$$

where $F_{n-1}^{jj}$ is the $j^{th}$ diagonal element of the $n-1^{th}$ Fisher information matrix $F_{n-1}$, which is calculated using the data and loss function in task $n-1$. We refer to this approach as **EWC** throughout this paper. Observe that if $F$ is given as the identify matrix, $\mathcal{L}_{ewc}$ simply becomes L2 regularization between the model parameters. We will analyze this approach as well and refer to it as simply **L2** for the rest of this paper. A strong advantage of L2 regularization versus EWC regularization is that L2 regularization can be applied in the absence of an importance-weighting matrix (e.g., L2 can be applied in the first task of a continual learning sequence in the presence of pre-training to retain the pre-trained knowledge). While the original work shows that using the identify matrix for $F$ hurts performance, we will show later that L2 can actually outperform EWC under certain continual learning settings.

### 4.2. Feature Space Regularization

Another approach for continual learning is to leverage *knowledge distillation* from $\theta_{n-1}$ to regularize the learning

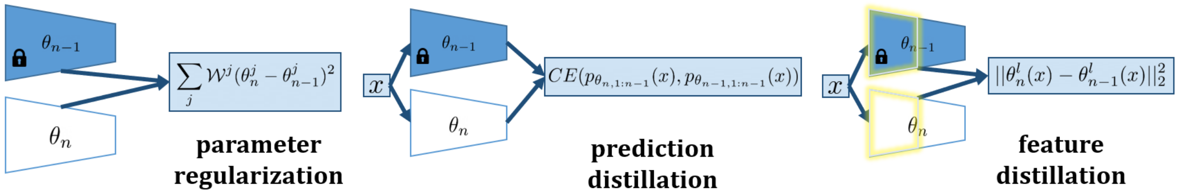

$\theta_{n-1}$
$\theta_n$
$\sum_j \mathcal{W}^j(\theta_n^j - \theta_{n-1}^j)^2$
parameter regularization
$x$
$\theta_{n-1}$
$\theta_n$
$CE(p_{\theta_{n,1:n-1}}(x), p_{\theta_{n-1,1:n-1}}(x))$
prediction distillation
$x$
$\theta_{n-1}$
$\theta_n$
$||\theta_n^l(x) - \theta_{n-1}^l(x)||_2^2$
feature distillation

Figure 1. **Three ways to transfer knowledge from a checkpoint model in continual learning**. Parameter regularization penalizes changes in the *model parameter space*. This can be weighted by the Fisher Information Matrix (e.g., EWC [29]) or with no weighting (e.g., L2 regularization). Prediction distillation (e.g., LwF [36]) penalizes features in the *model prediction space* with respect to some data $x$, whereas feature distillation penalizes features in the *model intermediate feature space* with respect to some data $x$. In the case where $x$ is strictly from the new task, both distillation methods are "rehearsal-free".

of $\theta_n$. This was first introduced for continual learning in the Learning without Forgetting (LwF) [36] method as knowledge distillation *in the prediction space*. We will refer to this as **PredKD** throughout the paper.

Let us denote $p_\theta(y \mid x)$ as the predicted class distribution produced by model $\theta$ for input $x$. Using this notation, the loss function for PredKD is defined as:

$$\mathcal{L}_{PredKD} = CE(p_{\theta_{n,1:n-1}}(x), p_{\theta_{n-1,1:n-1}}(x)) \quad (2)$$

where $CE$ is the standard cross-entropy loss. Knowledge can also be distilled in the *feature space* instead of the *prediction space*. The intuition here is to directly align the models' feature space so that the feature space does not drift far from the previous checkpoint solution. We will refer to this as **FeatureKD** throughout the paper with a loss function given as:

$$\mathcal{L}_{FeatKD} = ||\theta_n^l(x) - \theta_{n-1}^l(x)||_2^2 \quad (3)$$

Notice that, since we do not generate class predictions $p_\theta(y \mid x)$ at the intermediate feature space, we instead minimize the squared error.

### 4.3. Task-Bias

Another continual learning phenomena that exists in the absence of task labels during inference is *task bias* towards recent task data. This is typically mitigated with solutions relying on rehearsal data [1, 63]. Since we cannot reduce task bias with rehearsal data in our setting, we borrow from the rehearsal-free continual learning method LWF.MC [47] and use *sigmoid binary cross-entropy classification loss* (referred to as BCE) instead of the typical *softmax cross-entropy classification loss* (we note here that this is nearly equivalent to using the "labels trick" from [67]). The intuition here is that *softmax classification without rehearsal data results in a strong bias against the previously seen classes* because minimizing this loss reduces the magnitude of the old classes' corresponding logit outputs. We will show that the BCE classifier boosts the methods EWC, L2, and FeatKD into competitive SOTA approaches, despite having been previously reported to "fail" in the continual learning setting when task labels are not present [23].

## 5. Experiments

In this section, we take a closer look at EWC, L2, PredKD, and FeatKD in the rehearsal-free continual learning setting. We analyze performance of these four losses in addition to both i) a naive model trained with classification loss only (referred to as *naive*) and ii) an upper bound model trained with the joint training data from all tasks (referred to as *upper-bound*). We first provide benchmark results on the CIFAR-100 dataset [30] which contains 100 classes of 32x32x3 images. We train with a 18-layer ResNet [18] for 250 epochs using Adam optimization; the learning rate is set to 1e-3 and is reduced by 10 after 100, 150, and 200 epochs. We use a weight decay of 0.0002 and batch size of 128. Importantly, we do not tune our hyperparameters (i.e., the loss weights) on the full task set because tuning hyperparameters with hold out data from all tasks may violate the principle of continual learning that states each task is visited only once [58]. Instead, we tuned our hyperparameters (including the loss weight for each approach) (using a half-decade linear sweep from $1e-3$ to $1e2$) on a small task sequence of each dataset.

**Evaluation Metrics:** We evaluate methods using final accuracy, or the accuracy with respect to all past classes after having seen all $N$ tasks (referred to as $A_{N,1:N}$). Specifically, we have:

$$A_{i,n} = \frac{1}{|\mathcal{D}_n^{test}|} \sum_{(x,y)\in\mathcal{D}_n^{test}} \mathbf{1}(\hat{y}(x,\theta_{i,n}) = y \mid \hat{y} \in \mathcal{T}_n) \quad (4)$$

Note that $A_{i,n}$ gives the local task accuracy (i.e., inference in *task-incremental* learning where the task label is given, used to calculate local forgetting $F_N^L$ below) and $A_{i,1:n}$ gives the global task accuracy (i.e., the accuracy when the task label is unknown, used to calculate global forgetting $F_N^G$ below). For the final task accuracy in our results, we will denote $A_{N,1:N}$ as simply $A_{1:N}$. We also measure: (I) global forgetting, or the measurement of average decrease in performance on task $n$ with respect to the *global* task where no task label is given (referred to as $F_N^G$); and (II) local forgetting, or the measurement of average decrease in

Table 1. **Ablation results (%) on 10 task CIFAR-100.** $A_{1:N}$ gives the final task accuracy, $F_N^G$ gives the average *global* forgetting, and $F_L^G$ gives the average *local* forgetting. *BCE* refers to binary cross-entropy loss whereas *Soft* refers to softmax cross-entropy loss. We report the mean over 3 trials.

(a) parameter regularization

| Method | $A_{1:N}$ ($\uparrow$) | $F_N^G$ ($\downarrow$) | $F_N^L$ ($\downarrow$) |
|---|---|---|---|
| Upper-Bound | 56.2 | 0.0 | 0.0 |
| PredKD+EWC (BCE) | **22.7** | $\mathbf{-0.7}$ | **7.8** |
| EWC (BCE) | 7.7 | 64.0 | 58.5 |
| PredKD+EWC (Soft) | 7.3 | 44.4 | 56.8 |
| EWC (Soft) | 7.3 | 52.7 | 57.8 |

(b) feature regularization

| Method | $A_{1:N}$ ($\uparrow$) | $F_N^G$ ($\downarrow$) | $F_N^L$ ($\downarrow$) |
|---|---|---|---|
| Upper-Bound | 56.2 | 0.0 | 0.0 |
| PredK +FeatKD (BCE) | **19.1** | **7.0** | **26.5** |
| FeatKD (BCE) | 8.2 | 66.5 | 58.2 |
| PredKD+FeatKD (Soft) | 8.2 | 54.2 | 60.5 |
| FeatKD (Soft) | 8.5 | 63.9 | 61.8 |

Table 2. **Results (%) on 10 task CIFAR-100** using BCE classification. $A_{1:N}$ gives the final task accuracy, $F_N^G$ gives the average *global* forgetting, and $F_L^G$ gives the average *local* forgetting. We report the mean over 3 trials.

| Method | $A_{1:N}$ ($\uparrow$) | $F_N^G$ ($\downarrow$) | $F_N^L$ ($\downarrow$) |
|---|---|---|---|
| Upper-Bound | 56.2 | 0.0 | 0.0 |
| Naive | 8.6 | 71.0 | 63.4 |
| PredKD | **25.2** | 3.2 | 27.2 |
| PredKD + FeatKD | 19.1 | 7.0 | 26.5 |
| PredKD + EWC | 22.7 | $\mathbf{-0.7}$ | **7.8** |
| PredKD + L2 | 21.6 | 1.6 | 15.4 |

performance on task $n$ with respect to the *local* task where the task index is given (referred to as $F_N^L$). Global forgetting is taken from [33] and given as:

$$F_N^G = \frac{1}{N-1} \sum_{i=2}^{N} \sum_{n=1}^{i-1} \frac{|\mathcal{T}_n|}{|\mathcal{T}_{1:i}|} (R_{n,n} - R_{i,n}) \tag{5}$$

where:

$$R_{i,n} = \frac{1}{|\mathcal{D}_n^{test}|} \sum_{(x,y)\in\mathcal{D}_n^{test}} \mathbf{1}(\hat{y}(x, \theta_{i,1:n}) = y) \tag{6}$$

and local forgetting is taken from [39] and given as:

$$F_N^L = \frac{1}{N-1} \sum_{n=1}^{N-1} (A_{N,n} - A_{n,n}) \tag{7}$$

## 5.1. Rehearsal-Free Continual Learning

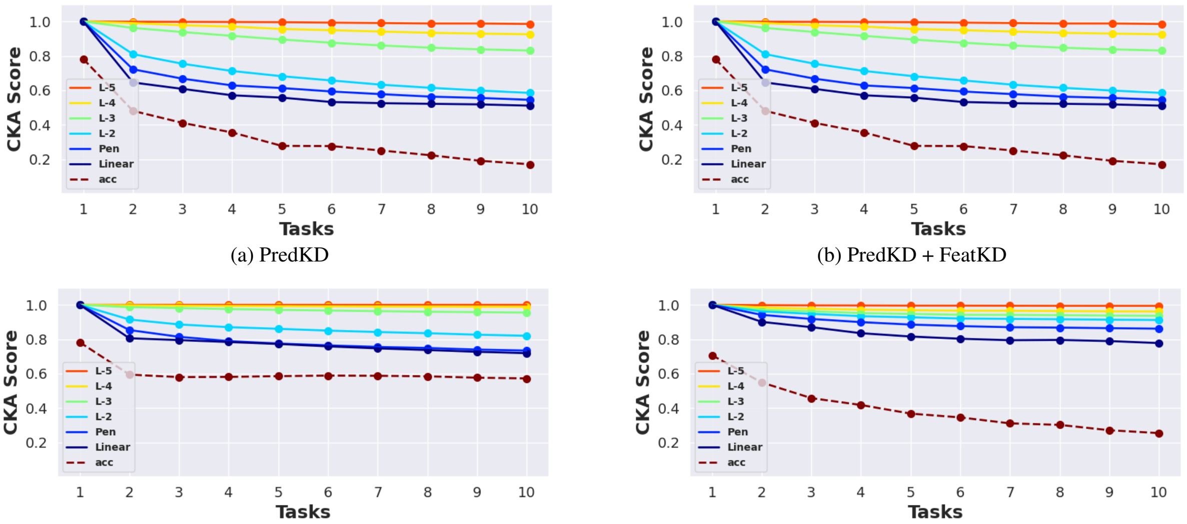
We start by analyzing performance on a 10 task sequence from CIFAR-100. Here, our model is shown 10 different tasks derived of 10 classes per task from the CIFAR-100 dataset. We use loss weights of $\{1e1, 5e{-}1, 1, 5\}$ for EWC, L2, PredKD, and FeatKD. Our first finding is that **PredKD and BCE are foundational for rehearsal-free continual learning**. In Table 1b, we tease apart two approaches which mitigate "feature drift": EWC and FeatKD[3]. For the two sides of this table, the top rows refer to the feature-drift method (EWC or FeatKD) using BCE when combined with PredKD. Below, we ablate the two methods separately, showing performance when i) PredKD is removed, ii) BCE is replaced with softmax classification, and iii) when PredKD is removed *and* BCE is replaced with softmax classification.

The bottom row demonstrates that vanilla EWC and FeatKD fail for continual learning (poor $A_{1:N}$) yet do reasonably well in mitigating local forgetting $F_N^L$ when a softmax PredKD is added (i.e., they perform well for task-incremental learning where the task label is given) [23]. The deeper finding here is that both EWC and FeatKD perform well at regularizing the *feature drift* yet fail at regularizing/debiasing the *classifier head*. As motivated in the prior section, we see that a significant jump in performance is achieved when combining *both BCE and PredKD*.

In order to closer examine the effects of parameter regularization and feature distillation on catastrophic forgetting, we consider the following approaches i) PredKD[4], ii) PredKD + FeatKD, iii) PredKD + EWC, and iv) PredKD + L2. Specifically, we want to understand *where forgetting is occurring* in these methods. We borrow the practice from [45] and look at the centered kernel alignment (CKA) similarity between feature representations over time for different layers in the model (higher is better). In Figure 2, we see the CKA similarity score plotted for each layer in each model across tasks for the task-1 data. The x-axis at task $n$ gives the CKA similarity score between features evaluated on task-1 holdout data from $\theta_1$ versus the features generated on task-1 holdout data from $\theta_n$. We calculate the CKA at the following layers: *Linear*, or the output of the linear layer; *pen*, or the output of the penultimate layer; and *L-2,L-3,L-4*, or the outputs at the second-to-last, third-to-last, and fourth-to-last layers. *Acc* refers to the accuracy $A_{n,1:n}$, where $n$ is the task number (i.e., x axis). We can interpret these scores with the full results in Table 2. For this experiment, we see that PredKD converges to the highest final accuracy, while PredKD + EWC has the lowest forgetting. Why is this? When we look at Figure 2, we see an *trade-*

[3] Here, we leave out L2 given that it is a special case of EWC.

[4] Notice that this is equivalent to the LwF.MC method from [47].



(a) PredKD

(b) PredKD + FeatKD

(c) PredKD + EWC

(d) PredKD + L2

Figure 2. **CKA Analysis on task-1 forgetting** for continual learning on CIFAR-100 for 10 tasks. *Linear* refers to the output of the linear layer, *pen* refers to the output of the penultimate layer, and *L-2..4* refers to the outputs at second-to-last, third-to-last, etc. layers. *Acc* refers to $A_{n,1:n}$, where $n$ is the task number (i.e., x axis).

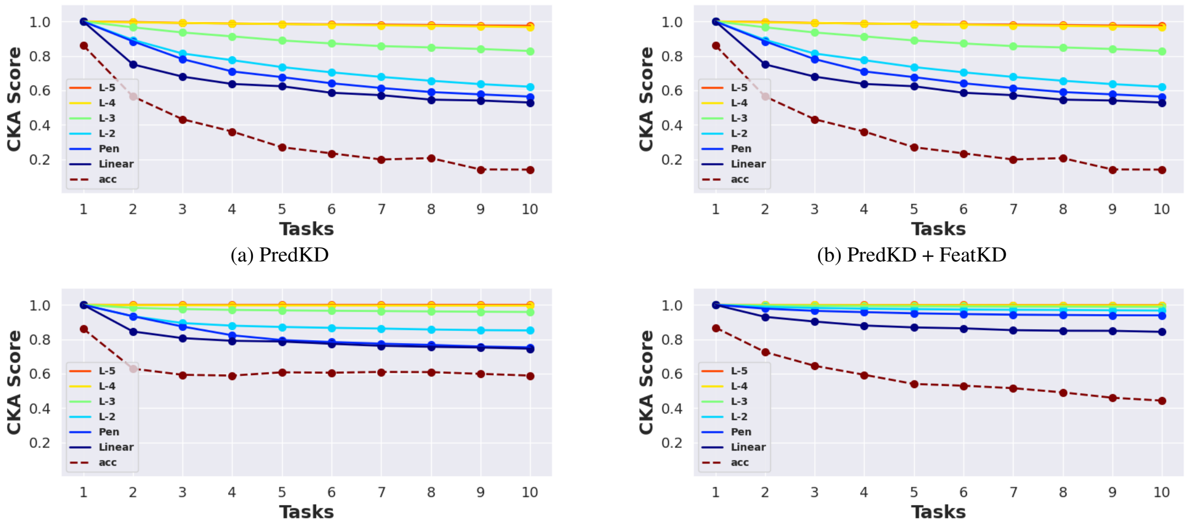
Table 3. **Results (%) on 10 task CIFAR-100 *leveraging pre-training*** and BCE classification. $A_{1:N}$ gives the final task accuracy, $F_N^G$ gives the average *global* forgetting, and $F_L^G$ gives the average *local* forgetting. We report the mean over 3 trials.

| Method | $A_{1:N}$ (↑) | $F_N^G$ (↓) | $F_N^L$ (↓) |
|---|---|---|---|
| Upper-Bound | 56.2 | 0.0 | 0.0 |
| Naive | 8.5 | 75.6 | 67.6 |
| PredKD | 26.6 | 3.9 | 34.3 |
| PredKD + FeatKD | 23.5 | 7.5 | 25.3 |
| PredKD + EWC | 31.1 | **−0.4** | **12.2** |
| PredKD + L2 | **35.6** | 1.0 | 15.0 |

*off* between retaining task 1 similarity across all layers versus final accuracy. Specifically, adding the parameter regularization losses (EWC and L2) induce *low forgetting* but at the cost of *low plasticity* (i.e., the ability to learn a new task). One surprising finding (which extends to the rest of this paper) is that FeatKD reduces the final accuracy without gaining any improvements in forgetting. *In summary, the main takeaways from this section are that:* **1) PredKD and BCE create a strong baseline for rehearsal-free continual learning** and **2) Parameter regularization in addition to this baseline reduces forgetting, but at the expense of low plasticity and therefore low final accuracy**.

## 5.2. How to Leverage Pre-Trained Models

Because the gap between SOTA and the upper bound for rehearsal-free continual learning remains large, we explore leveraging pre-trained models. Specifically, we ask the question: *what type of regularization (parameter-space or prediction-space) can best leverage model pre-training (from an auxiliary dataset) for rehearsal-free continual learning?* We note here that our work differs from [46] in that we *analyze the effect of regularization on forgetting in pre-trained models* rather than *show that pre-trained models are more robust to forgetting*. We repeat our experiments from the previous section, but this time our model is initialized with ImageNet [50] pre-training. We use loss weights of $\{1e2, 1e-1, 5, 5\}$ for EWC, L2, PredKD, and FeatKD. The main results are found in Table 3. **Surprisingly, we found the order of performance between PredKD, PredKD + EWC, and PredKD + L2 have been reversed!** While the forgetting metrics are not significantly affected, we see that $A_{1:N}$ has remained the same for PredKD but largely increased for EWC and L2. These results are reasonable after considering the following: with pre-training, less plasticity is needed because the model (features) are already useful for new tasks; thus, methods which achieved low forgetting at a cost of low performance on new tasks in the no pre-training scenario now have the "cost" removed. For further validation of this perspective, we notice that the CKA similarity scores presented in Figure 3 have not changed much from Figure 2, yet the differential in performance on new tasks, presented in Figure 4, is strikingly large. That is, pre-training does not seem to affect "forgetting" for these methods but rather enhances the ability to learn new tasks without forgetting. *In summary, the main takeaways from this section are that:* **1) L2 with PredKD outperforms EWC with PredKD in the presence of pre-training, and both of these approaches far outperform PredKD without parameter regularization** and **2) For parameter regularization**



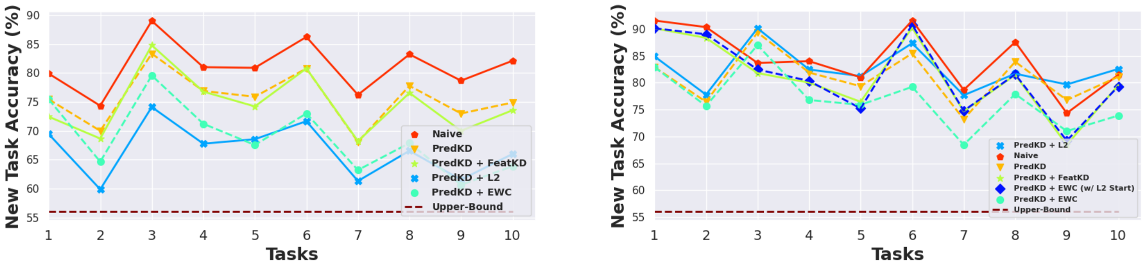
(a) PredKD

(b) PredKD + FeatKD

(c) PredKD + EWC

(d) PredKD + L2

Figure 3. **CKA Analysis on task-1 forgetting** for continual learning on CIFAR-100 for 10 tasks with **model pretraining** on ImageNet1k. *Linear* refers to the output of the linear layer, *pen* refers to the output of the penultimate layer, and *L-2..4* refers to the outputs at second-to-last, third-to-last, etc. layers. *Acc* refers to $A_{n,1:n}$, where $n$ is the task number (i.e., x axis).



(a) no pre-training

(b) ImageNet1k pre-training

Figure 4. **Most recent task accuracy versus tasks seen**. Here, the accuracy at task $n$ is the *local task accuracy* $A_{n,n}$ (as opposed to $A_{n,1:n}$), which we use to represent the ability to learn each new tasks. These plots demonstrate that the methods with ImageNet1k pre-training can achieve higher performance on new tasks (i.e. they require less plasticity to adapt to new tasks) compared to the methods with no pre-training.

**methods, pre-training greatly improves the ability to learn new tasks but has little effect on forgetting.**

### 5.3. Context with Current Literature - ResNet

We compare our results with SOTA methods on the exemplar-free continual learning setting on CIFAR-100 using the saem 18-layer ResNet backbone. The difference between our setting of *rehearsal-free* continual learning and this setting of *exemplar-free* continual learning is that, while neither store images for rehearsal, the latter setting often synthesizes these images. As discussed in Section 2, creating synthetic images with model inversion is a computationally expensive procedure and may still violate data-privacy concerns. We make our comparison in Table 4, showing final accuracy for the methods discusses in this paper. We find that **1) Pre-training can outperform SOTA rehearsal methods from synthetic data**, and **2) Pre-training can even outperform simple rehearsals methods that store a 2000 image coreset of data**.

### 5.4. Context with Current Literature - ViT

Next, motivated by our findings in the previous sections, we ask: *can parameter regularization outperform prompting for continual learning methods?* [54, 60, 61]. Specifically, we conjecture that, given a fair implementation and comparison, which targets modifying only the same spot of the ViT model as prompting methods, parameter regularization might outperform prompting for these benchmarks.

We benchmark using ImageNet-R [19,60] which is composed of 200 object classes with a wide collection of image styles, including cartoon, graffiti, and hard examples from the original ImageNet dataset [50]. This benchmark is

Table 4. **Results (%) for continual learning on CIFAR-100 on 10 tasks for different types of rehearsal and pre-training.** $A_{1:N}$ gives the final task accuracy.

| Method | Rehearsal | Pre-train | $A_{1:N}$ (↑) |
|---|---|---|---|
| Upper-Bound | None | None | 56.2 |
| Naive | None | None | 8.8 |
| PredKD | None | None | 24.6 |
| PredKD + FeatKD | None | None | 12.4 |
| PredKD + EWC | None | None | 23.3 |
| PredKD + L2 | None | None | 21.5 |
| PredKD | None | ImNet | 24.9 |
| PredKD + FeatKD | None | ImNet | 21.7 |
| PredKD + EWC | None | ImNet | 32.5 |
| PredKD + L2 | None | ImNet | **34.4** |
| DGR [18] | Gen. | None | 8.1 |
| DeepInversion [64] | Synth. | None | 10.9 |
| ABD [53] | Synth. | None | 33.7 |
| Rehearsal | 2k IMG | None | 24.0 |
| LwF [36] | 2k IMG | None | 27.4 |

attractive because the distribution of training data has significant distance to the pre-training data (ImageNet), thus providing a fair and challenging problem setting. We use the exact same experiment setting as the recent CODA-Prompt [54] paper. We implement our method and all baselines in PyTorch [44] using the ViT-B/16 backbone [12] pre-trained on ImageNet-1K [50]. We compare to the following methods (the same rehearsal-free comparisons of CODA-Prompt): Learning without Forgetting (LwF) [36], Learning to Prompt (L2P) [61], a modified version of L2P (L2P++) [54], and DualPrompt [60]. Additionally, we report the upper bound (UB) performance and performance for a neural network trained only on classification loss using the new task training data (we refer to this as FT).

We freeze most of the backbone and only fine-tune the QKV projection matrices of self-attention blocks throughout the ViT model. The intuition is that we are modifying the same modules as the prompting methods, but using classic continual learning methods that fine-tune with regularization rather than add prompts. We use loss weights of $\{1e3, 1\}$ for EWC and L2, respectively. Importantly, we use the same classification head as L2P, DualPrompt, and CODA-Prompt, and additionally compare to a FT variant, FT++, which uses the same classifier as the prompting methods and suffers from less forgetting. For additional details, we refer the reader to the CODA-Prompt [54] paper.

In Table 5, we benchmark against the popular and recent rehearsal-free continual learning methods. *We found that L2 achieves a high state-of-the-art in this setting.* Compared to the prompting methods L2P, DualPrompt, and the recent CODA-Prompt, L2 has clear and significant improvements, whereas EWC has poor performance. Our intuition is that L2 is much stronger given it begins regularization in task 1 (rather than task 2, such as EWC), and regularizes not only for past tasks but also future tasks by encouraging the model parameters to stay close to rich initial pre-training state. *In summary, the main takeaway from this experiment is that* **fine-tuning with L2 parameter regularization in the QKV projection matrices of ViT self-attention blocks outperforms prompting for continual learning methods.**

Table 5. **Results (%) on ImageNet-R** for 10 tasks (20 classes per task, 3 trials). $A_{1:N}$ gives the final task accuracy and $F_N^G$ gives the average *global* forgetting. We report mean % stdev over 5 trials.

| Method | $A_{1:N}$ (↑) | $F_N^G$ (↓) |
|---|---|---|
| UB | 77.13 | - |
| FT | $10.12 \pm 0.51$ | $25.69 \pm 0.23$ |
| FT++ | $48.93 \pm 1.15$ | $9.81 \pm 0.31$ |
| LwF.MC | $66.73 \pm 1.25$ | $3.52 \pm 0.39$ |
| L2P | $69.29 \pm 0.73$ | $2.03 \pm 0.19$ |
| L2P++ | $71.66 \pm 0.64$ | $1.78 \pm 0.16$ |
| DualPrompt | $71.32 \pm 0.62$ | $1.71 \pm 0.24$ |
| CODA-P (small) | $73.93 \pm 0.49$ | $1.60 \pm 0.20$ |
| CODA-P | $75.45 \pm 0.56$ | $1.64 \pm 0.10$ |
| EWC | $64.66 \pm 2.04$ | $\mathbf{1.55 \pm 0.25}$ |
| **L2** | $\mathbf{76.06 \pm 0.65}$ | $1.68 \pm 0.16$ |

## 6. Conclusions

In this work, we take a closer look at several popular continual learning strategies in the setting of *rehearsal-free continual learning*. This setting reflects machine-learning applications which cannot store or generate past-seen training data due to privacy concerns or memory constraints. We first show that parameter regularization techniques such as L2 and EWC can succeed in the rehearsal-free continual learning setting if softmax is removed from the classification head. Then, we compare parameter regularization, feature distillation, and prediction distillation on a 10-task continual learning benchmark. We find that with a randomly initialized model, parameter regularization methods achieves low forgetting but at the cost of low accuracy. When we initialize the model with pre-trained weights, we find that parameter regularization injects both low forgetting *and* high accuracy. Surprisingly, we found that *L2 regularization outperforms EWC in the pre-trained model scenario*. To validate these findings, we demonstrate that L2 parameter regularization implemented in a ViT transformer outperforms recently popular prompting for continual learning methods. In conclusion, our study has provided valuable insights into the efficacy of different types of regularization for continual learning and highlighted the potential of regularization in rehearsal-free settings.

# References

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

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

[3] Rahaf Aljundi, Eugene Belilovsky, Tinne Tuytelaars, Laurent Charlin, Massimo Caccia, Min Lin, and Lucas Page-Caccia. Online continual learning with maximal interfered retrieval. In *Advances in Neural Information Processing Systems*, pages 11849–11860, 2019. 2

[4] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. In *Advances in Neural Information Processing Systems*, pages 11816–11825, 2019. 2

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

[6] 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

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

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

[9] Yoojin Choi, Mostafa El-Khamy, and Jungwon Lee. Dual-teacher class-incremental learning with data-free generative replay. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3543–3552, 2021. 3

[10] Nikhil Churamani, Ozgur Kara, and Hatice Gunes. Domain-incremental continual learning for mitigating bias in facial expression and action unit recognition. *arXiv preprint arXiv:2103.08637*, 2021. 1

[11] Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 702–703, 2020. 3

[12] Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, et al. An image is worth 16x16 words: Transformers for image recognition at scale. *arXiv preprint arXiv:2010.11929*, 2020. 8

[13] Sayna Ebrahimi, Mohamed Elhoseiny, Trevor Darrell, and Marcus Rohrbach. Uncertainty-guided continual learning with bayesian neural networks. *arXiv preprint arXiv:1906.02425*, 2019. 2

[14] Sayna Ebrahimi, Franziska Meier, Roberto Calandra, Trevor Darrell, and Marcus Rohrbach. Adversarial continual learning. *arXiv preprint arXiv:2003.09553*, 2020. 2

[15] Alexander Gepperth and Cem Karaoguz. Incremental learning with self-organizing maps. *2017 12th International Workshop on Self-Organizing Maps and Learning Vector Quantization, Clustering and Data Visualization (WSOM)*, pages 1–8, 2017. 2

[16] Tyler L Hayes, Nathan D Cahill, and Christopher Kanan. Memory efficient experience replay for streaming learning. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 9769–9776. IEEE, 2019. 2

[17] Tyler L Hayes and Christopher Kanan. Lifelong machine learning with deep streaming linear discriminant analysis. *arXiv preprint arXiv:1909.01520*, 2019. 2

[18] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016. 4, 8

[19] Dan Hendrycks, Steven Basart, Norman Mu, Saurav Kadavath, Frank Wang, Evan Dorundo, Rahul Desai, Tyler Zhu, Samyak Parajuli, Mike Guo, Dawn Song, Jacob Steinhardt, and Justin Gilmer. The many faces of robustness: A critical analysis of out-of-distribution generalization. *ICCV*, 2021. 7

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

[21] Saihui Hou, Xinyu Pan, Chen Change Loy, Zilei Wang, and Dahua Lin. Lifelong learning via progressive distillation and retrospection. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 437–452, 2018. 2

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

[23] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, and Zsolt Kira. Re-evaluating continual learning scenarios: A categorization and case for strong baselines. *arXiv preprint arXiv:1810.12488*, 2018. 1, 3, 4, 5

[24] Daniel Justus, John Brennan, Stephen Bonner, and Andrew Stephen McGough. Predicting the computational cost of deep learning models. In *2018 IEEE international conference on big data (Big Data)*, pages 3873–3882. IEEE, 2018. 1

[25] Georgios Kaissis, Alexander Ziller, Jonathan Passerat-Palmbach, Théo Ryffel, Dmitrii Usynin, Andrew Trask, Ionésio Lima, Jason Mancuso, Friederike Jungmann, Marc-Matthias Steinborn, et al. End-to-end privacy preserving deep learning on multi-institutional medical imaging. *Nature Machine Intelligence*, 3(6):473–484, 2021. 3

[26] Nitin Kamra, Umang Gupta, and Yan Liu. Deep generative dual memory network for continual learning. *arXiv preprint arXiv:1710.10368*, 2017. 2

[27] Ronald Kemker and Christopher Kanan. Fearnet: Brain-inspired model for incremental learning. *International Conference on Learning Representations (ICLR)*, 2018. 2

[28] Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. *AAAI Conference on Artificial Intelligence*, 2018. 2

[29] 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*, 2017. 2, 3, 4

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

[31] Jogendra Nath Kundu, Rahul Mysore Venkatesh, Naveen Venkat, Ambareesh Revanur, and R Venkatesh Babu. Class-incremental domain adaptation. In *European Conference on Computer Vision*, pages 53–69. Springer, 2020. 1

[32] Alexandre Lacoste, Alexandra Luccioni, Victor Schmidt, and Thomas Dandres. Quantifying the carbon emissions of machine learning. *arXiv preprint arXiv:1910.09700*, 2019. 1

[33] Kibok Lee, Kimin Lee, Jinwoo Shin, and Honglak Lee. Overcoming catastrophic forgetting with unlabeled data in the wild. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 312–321, 2019. 2, 5

[34] Soochan Lee, Junsoo Ha, Dongsu Zhang, and Gunhee Kim. A neural dirichlet process mixture model for task-free continual learning. *arXiv preprint arXiv:2001.00689*, 2020. 2

[35] Timothée Lesort, Hugo Caselles-Dupré, Michael Garcia-Ortiz, Andrei Stoian, and David Filliat. Generative models from the perspective of continual learning. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019. 2

[36] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE transactions on pattern analysis and machine intelligence*, 40(12):2935–2947, 2017. 2, 4, 8

[37] Vincenzo Lomonaco and Davide Maltoni. Core50: a new dataset and benchmark for continuous object recognition. *arXiv preprint arXiv:1705.03550*, 2017. 2

[38] Vincenzo Lomonaco, Davide Maltoni, and Lorenzo Pellegrini. Rehearsal-free continual learning over small non-iid batches. In *CVPR Workshops*, pages 989–998, 2020. 2

[39] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, pages 6470–6479, USA, 2017. Curran Associates Inc. 2, 5

[40] Davide Maltoni and Vincenzo Lomonaco. Continuous learning in single-incremental-task scenarios. *Neural Networks*, 116:56–73, 2019. 2

[41] Vaishnavh Nagarajan, Colin Raffel, and Ian J Goodfellow. Theoretical insights into memorization in gans. In *Neural Information Processing Systems Workshop*, 2018. 2

[42] Cuong V Nguyen, Alessandro Achille, Michael Lam, Tal Hassner, Vijay Mahadevan, and Stefano Soatto. Toward understanding catastrophic forgetting in continual learning. *arXiv preprint arXiv:1908.01091*, 2019. 1

[43] Oleksiy Ostapenko, Mihai Puscas, Tassilo Klein, Patrick Jahnichen, and Moin Nabi. Learning to remember: A synaptic plasticity driven framework for continual learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 11321–11329, 2019. 2

[44] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *Advances in neural information processing systems*, 32, 2019. 8

[45] Vinay V Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. *arXiv preprint arXiv:2007.07400*, 2020. 5

[46] Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*, 2021. 6

[47] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H. Lampert. icarl: Incremental classifier and representation learning. In *2017 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR'17, pages 5533–5542, 2017. 2, 4, 5

[48] Anthony Robins. Catastrophic forgetting, rehearsal and pseudorehearsal. *Connection Science*, 7(2):123–146, 1995. 2

[49] David Rolnick, Arun Ahuja, Jonathan Schwarz, Timothy Lillicrap, and Gregory Wayne. Experience replay for continual learning. In *Advances in Neural Information Processing Systems*, pages 348–358, 2019. 2

[50] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *International journal of computer vision*, 115(3):211–252, 2015. 6, 7, 8

[51] Andrei A Rusu, Neil C Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks. *arXiv preprint arXiv:1606.04671*, 2016. 2

[52] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 2990–2999. Curran Associates, Inc., 2017. 2

[53] James Smith, Yen-Chang Hsu, Jonathan Balloch, Yilin Shen, Hongxia Jin, and Zsolt Kira. Always be dreaming: A new approach for data-free class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 9374–9384, October 2021. 3, 8

[54] James Seale Smith, Leonid Karlinsky, Vyshnavi Gutta, Paola Cascante-Bonilla, Donghyun Kim, Assaf Arbelle, Rameswar Panda, Rogerio Feris, and Zsolt Kira. Coda-prompt: Continual decomposed attention-based prompting for rehearsal-free

continual learning. *arXiv preprint arXiv:2211.13218*, 2022. 2, 3, 7, 8

[55] Michalis K Titsias, Jonathan Schwarz, Alexander G de G Matthews, Razvan Pascanu, and Yee Whye Teh. Functional regularisation for continual learning with gaussian processes. In *International Conference on Learning Representations*, 2019. 2

[56] Gido M van de Ven, Hava T Siegelmann, and Andreas S Tolias. Brain-inspired replay for continual learning with artificial neural networks. *Nature communications*, 11(1):1–14, 2020. 2

[57] Gido M van de Ven and Andreas S Tolias. Generative replay with feedback connections as a general strategy for continual learning. *arXiv preprint arXiv:1809.10635*, 2018. 2

[58] Gido M Van de Ven and Andreas S Tolias. Three scenarios for continual learning. *arXiv preprint arXiv:1904.07734*, 2019. 1, 4

[59] Johannes von Oswald, Christian Henning, João Sacramento, and Benjamin F Grewe. Continual learning with hypernetworks. *arXiv preprint arXiv:1906.00695*, 2019. 2

[60] Zifeng Wang, Zizhao Zhang, Sayna Ebrahimi, Ruoxi Sun, Han Zhang, Chen-Yu Lee, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, et al. Dualprompt: Complementary prompting for rehearsal-free continual learning. *arXiv preprint arXiv:2204.04799*, 2022. 2, 3, 7, 8

[61] Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149, 2022. 2, 3, 7, 8

[62] Guile Wu, Shaogang Gong, and Pan Li. Striking a balance between stability and plasticity for class-incremental learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1124–1133, 2021. 2

[63] 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, 4

[64] Hongxu Yin, Pavlo Molchanov, Jose M Alvarez, Zhizhong Li, Arun Mallya, Derek Hoiem, Niraj K Jha, and Jan Kautz. Dreaming to distill: Data-free knowledge transfer via deepinversion. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8715–8724, 2020. 3, 8

[65] Lu Yu, Bartlomiej Twardowski, Xialei Liu, Luis Herranz, Kai Wang, Yongmei Cheng, Shangling Jui, and Joost van de Weijer. Semantic drift compensation for class-incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6982–6991, 2020. 2

[66] Friedemann Zenke, Ben Poole, and Surya Ganguli. Continual learning through synaptic intelligence. In *International Conference on Machine Learning*, 2017. 2

[67] Chen Zeno, Itay Golan, Elad Hoffer, and Daniel Soudry. Task agnostic continual learning using online variational bayes. *arXiv preprint arXiv:1803.10123*, 2018. 4

[68] Fei Zhu, Xu-Yao Zhang, Chuang Wang, Fei Yin, and Cheng-Lin Liu. Prototype augmentation and self-supervision for incremental learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 5871–5880, 2021. 2