This CVPR paper is the Open Access version, provided by the Computer Vision Foundation.
Except for this watermark, it is identical to the accepted version;
the final published version of the proceedings is available on IEEE Xplore.

Probing Representation Forgetting in Supervised and Unsupervised Continual Learning

MohammadReza Davari[1,2] * Nader Asadi[1,2] * Sudhir Mudur[1]
Rahaf Aljundi [3] Eugene Belilovsky[1,2]
[1] Concordia University [2] Mila – Quebec AI Institute [3] Toyota Motor Europe

Abstract

Continual Learning (CL) research typically focuses on tackling the phenomenon of catastrophic forgetting in neural networks. Catastrophic forgetting is associated with an abrupt loss of knowledge previously learned by a model when the task, or more broadly the data distribution, being trained on changes. In supervised learning problems this forgetting, resulting from a change in the model's representation, is typically measured or observed by evaluating the decrease in old task performance. However, a model's representation can change without losing knowledge about prior tasks. In this work we consider the concept of representation forgetting, observed by using the difference in performance of an optimal linear classifier before and after a new task is introduced. Using this tool we revisit a number of standard continual learning benchmarks and observe that, through this lens, model representations trained without any explicit control for forgetting often experience small representation forgetting and can sometimes be comparable to methods which explicitly control for forgetting, especially in longer task sequences. We also show that representation forgetting can lead to new insights on the effect of model capacity and loss function used in continual learning. Based on our results, we show that a simple yet competitive approach is to learn representations continually with standard supervised contrastive learning while constructing prototypes of class samples when queried on old samples.[1]

1. Introduction

Continual Learning (CL) is concerned with methods for learners to manage changing training data distributions. The goal is to acquire new knowledge from new data distributions while not forgetting previous knowledge. A common scenario is CL in the classification setting, where the class labels presented to the learner change over time. In this scenario, a phenomenon known as catastrophic forgetting has been observed [11,31]. This phenomenon is often described as a loss of knowledge about previously seen data and is observed in the classification setting as a decrease in accuracy.

Deep learning has been traditionally motivated as an approach, which can automatically learn representations [4], forgoing the need to design handcrafted features. Indeed representation learning is at the core of deep learning methods in supervised and unsupervised settings [10]. In the case of many practical scenarios we may not simply be interested in the final performance of the model, but also the usefulness of the learned features for various downstream tasks [42]. Although a model's representation may change, sometimes drastically at task boundaries [5], this does not necessarily imply a loss of useful information and may instead correspond to a simple transformation. For example, consider a standard multi-head CL setting, where each task shares a representation and only differs through task specific "heads". A permutation of the features leading into the classification heads leads to total catastrophic forgetting as measured by standard approaches as the task heads no longer match with the representations. However, this does not correspond to a loss of knowledge about the data, nor a less useful representation. Indeed recent works have highlighted the importance of fast remembering versus catastrophic forgetting [14, 17], a looser continual learning requirement where in the task performance may decrease but the agent is able to recover rapidly upon observing a few samples from the previous task. In this light, maintaining a useful representation, which facilitates rapid recovery, is as important as maintaining high performance for the task.

CL envisions having learners operate over long time horizons while continually maintaining old knowledge and integrating new knowledge. Hence, in addition to directly measuring the performance on previous tasks using the last layer classifiers, it is sensible to consider the usefulness of

*Equal contribution, name order randomized.

This research was partially funded by NSERC Discovery Grant RGPIN-2021-04104 and RGPIN-2019-05729. We acknowledge resources provided by Compute Canada and Calcul Quebec. Correspondence to: m_davari@live.concordia.ca, nader.asadi@concordia.ca, eugene.belilovsky@concordia.ca

[1] The code to reproduce our results is publicly available at: https://github.com/rezazzr/Probing-Representation-Forgetting

their representations for previous tasks. In this paper we highlight that traditional approaches of evaluating forgetting are unable to properly disambiguate trivial changes in the features (e.g. permutation) from abrupt losses of useful representations. We instead use optimal Linear Probes (LP), commonly used to study unsupervised representations [8] and intermediate layer representations [36, 48], to evaluate CL algorithms and their effectiveness.

We revisit several CL settings and benchmarks and measure forgetting using LP. Our focus is particularly on re-evaluating finetuning approaches that do not apply explicit control for the non-iid nature of continual learning. We observe that in many commonly studied cases of catastrophic forgetting, the representations under naive finetuning approaches, undergo minimal forgetting, without losing critical task information.

Our major contributions in this work are as follows. First we bring three new significant insights obtained and demonstrated through extensive experimental analysis:

1. In a number of CL settings the observed accuracy can be a misleading metric for studying forgetting, particularly when compared to finetuning approaches
2. Naive training with SupCon [21] or SimCLR (in the unsupervised case) have advantageous properties for continual learning, particularly in longer sequences.
3. By using LP based evaluation, forgetting is clearly decreased for wider and deeper models, which is not seen that clearly from earlier observed accuracy.

Secondly, we suggest a simple approach to facilitate fast remembering, which does not require using a large memory during training; it relies only on a small memory combined with SupCon based finetuning.

2. Related Work

The design of CL methods is often focused on mitigating the catastrophic forgetting phenomenon, with aspects such as maximization of forward and backward transfer between tasks taken as secondary [27]. One class of methods focuses on bypassing this problem by growing architectures over time as new tasks arrive [2,25,41,43]. Under the fixed architecture setting, one can identify two main categories. The first category of methods rely on storing and re-using samples from the previous history while learning new ones; this includes approaches such as GEM [27] and ER [7]. The second category of methods encode the knowledge of the previous tasks in a prior that is used to regularize the training of the new task [1, 22, 26, 34, 49]. A classic method in this vein is Learning without Forgetting (LwF) [26], which mitigates forgetting by a regularization term that distills knowledge [18] from the earlier tasks. The network representations from earlier stages are recorded, and are used during training for a new task to regularize the objective by distilling knowledge from the earlier state of the network. Similarly, Elastic Weight Consolidation (EWC) [20] preserves the knowledge of the past tasks through a quadratic penalty on the network parameters important to the earlier tasks. The importance of the parameters is approximated via the diagonal of the Fisher information matrix [32]. The scale of the importance matrix, λ, determines the network's preference towards preserving old representations or acquiring new ones for the current task. In Sec. 4.1 we examine the effectiveness of these approaches in mitigating representation forgetting.

Recent works on elucidating the nature of catastrophic forgetting have examined the influence of task sequence [33], network architecture [3], and change in representation similarity [38]. Our work is related in spirit to [38] as we pursue measuring how much forgetting has occurred on the learned representation and we additionally study this for intermediate representations. One significant difference is that in [38], the authors use linear CKA (centred Kernel Alignment) [23] to measure the similarity between intermediate representations influenced by forgetting, while in our work we measure how much forgetting has occurred on the representations using LP.

Several recent works have also studied the behavior of networks with increasing model capacity. In [39] the authors examine several common architectures under the task incremental setting and demonstrate that pre-training is essential to combat forgetting and to achieve high performance on all tasks. They conclude that training only with larger models yields no benefit for continual learning. Our analysis revisit this setting and take a closer look at how representation forgetting is affected with increase in model width and depth.

Several works [40] have focused on modifying the last layer of a classification network to make more effective use of the representation for prior tasks. This indirectly highlights the fact that the last layer can be modified to yield better performance on prior tasks. Particularly [30, 40] use a buffer of old examples at training time to improve learning and then use them at evaluation time to construct a class mean prototype. This allows for more effective use of the representations of the network. These works consider settings where the CL methods are used to control forgetting, while we also emphasize that naive continuation of training under task shift can yield strong representations. Our work can also be seen as both a way to explain and to motivate the need for such approaches.

Self-supervised learning (SSL) is becoming increasingly popular in visual representation learning. Some of the best performing methods rely on contrastive learning [8, 15]. These methods have been recently evaluated in a limited continual learning setting [19] where a sequence was trained on non-iid unsupervised streaming data and then applied in transfer learning settings on multiple datasets.

However, forgetting was not evaluated. In contrast our work, which also uses a SSL loss, focuses on the LP evaluation and the study of representation forgetting with respect to previously seen distributions. Contrastive methods are also often used in the supervised setting, for example, the recently popular SupCon loss [21]. In [5] and [30] the use of SupCon is proposed in the online class-incremental setting in combination with experience replay. Our work too considers SupCon as one of the supervised representation learning approaches. However distinct from the other works we consider it in the offline task-incremental setting. We do not look at its use in combination with replay or other approaches, but study the effect of standard finetuning with SupCon loss, distinct from [5], where it is used to facilitate separation of contrasts between old and new classes, specific to the class incremental setting.

3. Background and Methods

In this section we review the key tools used in our analysis including linear probes, centered kernel alignment, and contrastive loss functions. Finally we discuss how the nearest mean of exemplars approach can be used in the context of non-rehearsal based methods, such as finetuning with SupCon, as a simple continual learning method that also facilitates rapid remembering.

3.1. Linear Probes for Representation Forgetting

Following the work in SSL [8] and in the analysis of intermediate representations [48] we evaluate the adequacy of representations by an optimal linear classifier using training data from the original task. A linear classifier is trained on top of the frozen activations of the base network given the training instances of a certain dataset. The test set accuracy obtained by using LP on that dataset is used as a proxy to measure the quality of the representations. The difference in performance of the LP before and after a new task is introduced, acts as a surrogate measure to the amount of forgetting observed by the representations and is referred to as representation forgetting.

Formally, for a given model f_{θ_i} computed from time step i of a task sequence, we compute its optimal classifier $W_i^* = \arg\min_{W_i} \mathcal{L}(W_i f_{\theta_i}(X_i), Y_i)$, where $\mathcal{L}$ is the objective function, and X_i and Y_i correspond to the data from task i. To assess representation forgetting between θ_a and a model at a later point in the sequence, say θ_b, we evaluate $T(W_a f_{\theta_a}(X_a), Y_a) - T(W_b f_{\theta_b}(X_a), Y_a)$ where T is the task performance (*e.g.* accuracy).

3.2. Centered Kernel Alignment

CKA [23] is a recent popular approach to compare representations. It has been commonly used to compare representations across depth as well as across models from different tasks in the CL settings [38]. Given a dataset comprised of m samples, and their representations X and Y, with features n_x and n_y respectively, *i.e.* $X \in \mathbb{R}^{m \times n_x}$ and $Y \in \mathbb{R}^{m \times n_y}$, the, typically used, linear CKA between X and Y is given as $\frac{\|Y^T X\|_F^2}{\|X^T X\|_F^2 \|Y^T Y\|_F^2}$. This similarity metric has the advantage of being invariant to scaling and orthogonal transformations. However, being a simple linear alignment comparison it is not invariant to general classes of invertible transformations. Furthermore, relative comparisons of CKA metrics are challenging to interpret compared to task performance degradation. CKA has been used in [38] to compare the intermediate representations of a model in consecutive task increments t and $t{+}1$. Ramasesh *et al.* [38] proxy the CKA similarity between the intermediate representations of the model f_{θ_t} and $f_{\theta_{t+1}}$ to measure representation forgetting. Thus, under this paradigm, a high value of CKA similarity is interpreted as minimal representation forgetting. One limitation of this metric for representation forgetting is its inability to distinguish between positive and negative backward transfer (see Sec. 4.4). This is addressed when measuring representation forgetting via LP.

3.3. Supervised and Unsupervised Contrastive Loss

Contrastive loss functions have become increasingly popular in representation learning, particularly visual representation learning. They have led to large advances in unsupervised learning [8, 15]. As well they are becoming a popular alternative to cross-entropy (CE) in the supervised setting [12, 21], referred to as SupCon. Given a representation f_θ, often consisting of a primary network and a projection, the SupCon loss for a minibatch X is given by:

$$\sum_{\mathbf{x}_i \in \mathbf{X}} \frac{-1}{|P(i)|} \sum_{\mathbf{x}_p \in P(i)} \log \frac{\mathtt{sim}\big(f_\theta(\mathbf{x}_p), f_\theta(\mathbf{x}_i)\big)}{\sum_{\mathbf{x}_a \in \mathbf{X}/x_i} \mathtt{sim}\big(f_\theta(\mathbf{x}_a), f_\theta(\mathbf{x}_i)\big)}$$

Where $\mathtt{sim}(a,b) = \exp(\frac{a^T b}{\tau \|a\| \|b\|})$ and $P(i)$ represents the same class samples as x_i from the minibatch, and the denominator is taken over all other samples. Note that we consider SupCon in the naive setting, thus all minibatches are from the current task in our evaluations of SupCon. Similar to this, in the unsupervised setting the SimCLR loss [8] is given by:

$$-\sum_{\mathbf{x}_i \in \mathbf{X}} \log \frac{\mathtt{sim}\big(f_\theta(\mathbf{x}_p(i)), f_\theta(\mathbf{x}_i)\big)}{\sum_{\mathbf{x}_n \in \mathbf{A}(i)} \mathtt{sim}\big(f_\theta(\mathbf{x}_n), f_\theta(\mathbf{x}_i)\big)}$$

Where $A(i)$ corresponds to all minibatch examples and their data augmentations except x_i, and $x_p(i)$ represents an augmented version of x_i.

3.4. Exemplars and Fast Remembering

Many CL methods utilize buffers of exemplars [40] that are constantly updated. Typically, these samples are used repeatedly to train the model [7]. In [30, 40], the samples in the memory are also used to continuously estimate a class

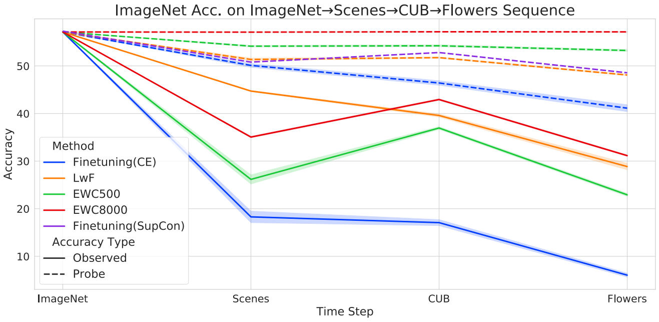

Figure 1. Performance on ImageNet during the sequence (ImageNet→Scenes→CUB→Flowers) using ResNet18. We observe that although observed accuracy heavily degrades the LP accuracy, in finetuning does not decay as drastically and can rival LP accuracy of methods such as LwF and EWC. Moreover, we observe that the LP Accuracy of SupCon training, which has no control for forgetting, outperforms the LwF, a method designed for CL. Note EWC with $\lambda = 8k$ is the best performing method in terms of LP and observed acc., however it does not perform well on the current task (see Tab. 1).

mean for old samples using the new representation. This is then used to construct a nearest mean of exemplars (NME) classifier, which can be seen as a fast way to construct a strong classifier that requires only a small amount of data. Instead of relying on the same exemplars during training and inference, one can use a small set of exemplars from the task distribution at the end of a task sequence to construct an NME based classifier in combination with a non-CL specific representation learning method such as SupCon [21]. Specifically the learner maintains a class mean for any class it has encountered. These class means are updated either by using a stored set of samples that is only used at inference or upon encountering an old task again, obtaining a small set of new samples to facilitate fast remembering. Notably, unlike the prior work on NME classifiers in continual learning, we don't suggest using exemplars as a rehearsal memory during the training, but as a method for fast remembering of class means for old tasks during evaluation time. This has the advantage of not increasing the computational complexity of training and not needing additional overhead in storing or retrieving samples, except at the end of a task sequence or upon re-encountering a task. Specifically, it can facilitate rapid remembering; in situations without prior data stored. Upon encountering a new task a model with minimal representation forgetting can rapidly adapt by updating just its class means.

4. Experiments

We perform evaluations in several CL scenarios, focusing on the task-incremental setting. The evaluations are based on LP and observed accuracy. Observed accuracy refers to the standard accuracy used in the CL literature. Specifically we measure observed accuracy, A_{ij}, as the accuracy of the model after step i on the test data of task j.

	Method	Acc. Scenes	Acc. CUB	Acc. Flowers
🟦	FT (CE)	$56.9\% \pm 1.1$	$54.5\% \pm 2.6$	$89.3\% \pm 1.1$
🟧	LwF	$57.6\% \pm 1.5$	$43.1\% \pm 2.9$	$85.3\% \pm 0.5$
🟩	$\text{EWC}_{\lambda:0.5k}$	$52.5\% \pm 1.1$	$47.8\% \pm 2.5$	$85.9\% \pm 1.6$
🟥	$\text{EWC}_{\lambda:8k}$	$42.1\% \pm 1.5$	$38.3\% \pm 0.9$	$79.1\% \pm 1.0$
🟪	FT (SupCon)	$57.1\% \pm 1.2$	$50.4\% \pm 1.0$	$85.3\% \pm 0.9$

Table 1. Observed accuracy of the current task in the sequence ImageNet→Scenes→CUB→Flowers using ResNet architecture. Although $\text{EWC}_{\lambda:8k}$ attains relatively poor performance on the current task, it achieves the highest LP and observed accuracy for the previously seen tasks (see Fig. 1). Moreover, the SupCon training achieves comparably high accuracy on the current task (even surpassing CE on Scenes) while suffering from relatively small representation forgetting (see Fig. 1).

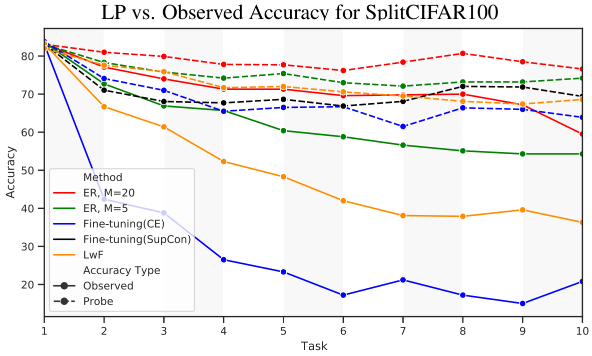

Figure 2. 10-Task SplitCIFAR100. We compare observed acc. and linear probe acc. Naively finetuning with CE does poorly if using the observed accuracy. However using the LP based evaluation we observe that performance gap to other methods is lower. Furthermore when finetuning is performed instead with the SupCon loss function LP performance can rival that of LwF.

Similarly, the average observed accuracy at the end of the sequence is $\frac{1}{T}\sum_{t \in T} A_{T,t}$ as used in *e.g.* [26]. Similarly we can measure the LP accuracy for each step i and task j as well as the average LP accuracy.

Datasets We use an ImageNet transfer setting based on [26], a common SplitCIFAR100 [24] setting (split into 10 tasks), and reproduce the SplitCIFAR10 (split into 2 tasks) setting from [38]. Finally, to evaluate in very long task sequence regimes, we use a downsampled version of the entire ImageNet dataset (ImageNet32 [9]) split into 200 tasks of 5 classes each. For the ImageNet transfer setting, we use a sequence consisting of the standard ImageNet (LSVRC 2012 subset) [42], MIT Scenes [37] for indoor scenes classification (5,360 samples over 67 classes), Caltech-UCSD Birds (CUB) [47] for classification of bird species (6,033 samples over 200 classes), and Oxford Flow-

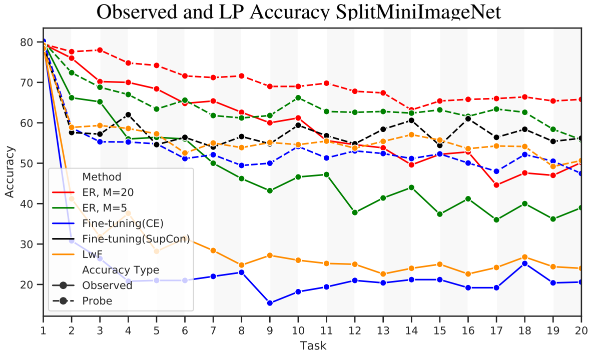

Figure 3. 20-Task SplitMiniImageNet. We compare observed accuracy and Linear probe accuracy for Task 1 data. Naively finetuning with CE as well as the LwF method does poorly if using the observed accuracy. However using the LP based evaluation we see that performance gap to other methods is less significant. LwF performs similar to finetuning with SupCon. For this Longer task sequence ER with large buffer, performance decays towards the end of the sequence, while SupCon stays flat.

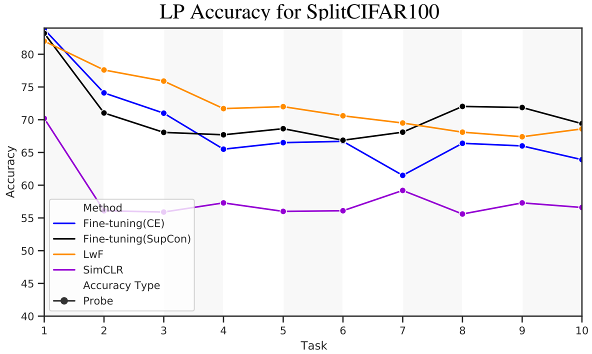

Figure 4. SplitCIFAR100 comparison of unsupervised Linear Probe accuracies on task 1 with supervised finetuning CE and SupCon as well as LwF. We observe that LwF and CE based finetuning can decay over time, while the unsupervised learning (SimCLR) has an initial drop and stays relatively flat.

ers [35] for flower classification (2,040 samples over 102 classes). The use of this sequence allows us to complement the standard long task sequence benchmarks with a more realistic and diverse larger scale sequence. Optimization hyperparameters for training are detailed in the Appendix.

Methods Compared Our work focuses on evaluating naive finetuning based approaches using CE and SupCon [21] loss functions, as well as a set of representative CL methods. For regularization-based baselines, we consider two of the most popular methods, which do not utilize memory of any past samples: LwF [26] and EWC [20]. For rehearsal-based baselines, which continuously store past samples we focus on Experience Replay (ER). Indeed a number of recent works have illustrated that ER, particularly with increase in buffer size, is a strong baseline [7, 38] and rivals or exceeds other rehearsal based methods such as iCaRL [40] and GEM [27]. Hence we use ER with both a small buffer, $M = 5$ samples per class, and a relatively large buffer, $M = 20$ samples per class.

4.1. Observed vs LP accuracy

In this section we study the observed vs LP accuracy for various task sequences and methods, in both supervised and unsupervised settings.

ImageNet Transfer We consider models trained on the large ImageNet data and subsequently applied to different tasks in the sequence. We take the setting of [26], which considers the ImageNet [42] transfer to various datasets, in particular CUB [46], and Scenes [37]. We extend this setting by including Flowers [35] in the task sequence. To reduce the computation of experiments we do random resize crops to 64×64 and utilize ResNet-18 for these experiments. Additional results further confirming our observations with larger crop size are given in the Appendix. As mentioned earlier, in addition to LwF, we also examine EWC [20] under two conditions: (a) large λ value ($8k$), so the network is inclined to preserve the knowledge important to the previous tasks, and (b) small λ value ($0.5k$), so the network is encouraged to perform competitively on the current task.

We report observed accuracy and LP accuracy on ImageNet validation set as the model is trained on the task sequence ImageNet $\rightarrow$ Scenes $\rightarrow$ CUB $\rightarrow$ Flowers (see Fig. 1). We also report the observed accuracy on the current task in Tab. 1. Our evaluation reveals that although the forgetting in terms of the traditional measure is high for finetuning compared to LwF (as shown in [26]), the LP accuracy of these methods suggest less drastic forgetting. Furthermore, the LP performance across finetuning and other methods is not as drastically different as their respective observed accuracies are. We see that the LP accuracy of SupCon based finetuning, which has no explicit control for forgetting, outperforms LwF, a method specifically designed for CL. It also closely tracks the performance of EWC$_{\lambda:0.5k}$, while outperforming on the current task performance. Indeed, as we can see in Tab. 1, SupCon training achieves comparably high accuracy on the current task (even surpassing CE finetuning on Scenes) with relatively small representation forgetting (see Fig. 1).

SplitCIFAR100 and SplitMiniImageNet We now consider the SplitCIFAR100 with 10 tasks of 10 classes each and the 20 task SplitMiniImageNet setting. We show in Fig. 2 and Fig. 3 the performance on the first task throughout the sequence for both settings. Similar to the previous case we see: for finetuning with CE the LP based evaluation shows much milder forgetting than observed accuracy. When finetuning with SupCon, LP performance drops initially but then stays relatively flat and even increases, suggesting that positive backward transfer is occurring. Over-

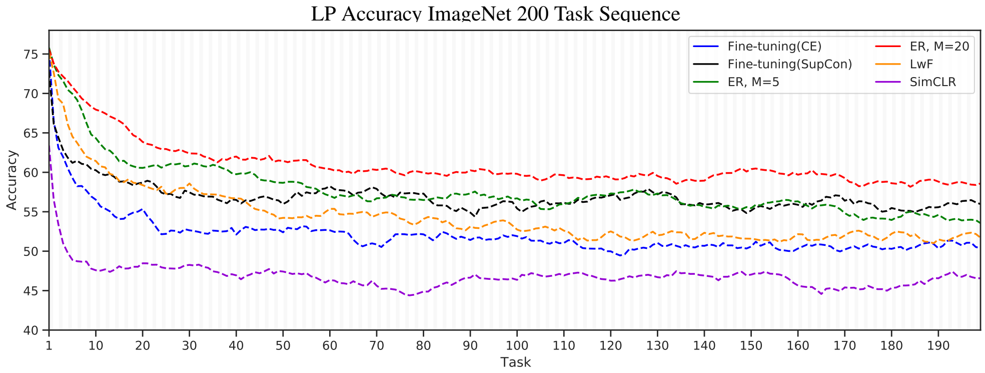

Figure 5. 200-Task ImageNet32. We compare Linear probe accuracy for Tasks 1 data over the whole sequence. As the model observes the later tasks of the sequence, the performance of finetuning with CE reaches LwF, and finetuning with SupCon outperforms ER with 5 samples per class.

time, SupCon outperforms the LwF [26] approach without any specific CL based control in both task sequences. It also obtains performance that becomes close to ER with 5 samples per class without access to any previous data. For the longer MiniImageNet task sequence we see that over time even the strong ER based baselines which train on old data demonstrate a reduced LP performance, while finetuning baselines remain relatively flat and even increase in the case of SupCon. This suggests that in very long sequences, finetuning baselines can be competitive to the more computationally expensive CL methods.

Utilizing SplitCIFAR100 we also consider the unsupervised representation learning case where linear probes follow naturally as a common evaluation setting. The literature on evaluation of continual learning methods in the unsupervised setting is limited. Hence we directly compare unsupervised and supervised approaches in their representation learning ability when presented with the same task sequences. We focus on the SimCLR loss and evaluate LP performance in comparison to other methods in Fig. 4. We can see in Fig. 4 that the initial LP performance is lower for SimCLR compared to the supervised losses. This is natural as it does not have access to the labeled data. Despite the higher starting accuracy, LwF and finetuning with CE show a decay that continues in the first several tasks following task 1. On the other hand SimCLR decays at the first step but then remains nearly flat over the rest of the sequence, showing a strong resistance to representation forgetting after this inital drop. However SupCon, which utilizes a loss similar to SimCLR in the supervised setting, shows the best of both worlds, has an initial drop and then illustrates gradual backward transfer properties.

200 Task Sequence - SplitImageNet32 We now consider a much longer sequence than typically studied in the literature to allow us to observe whether the trends we have seen so far continue to hold. Using Imagenet32 we construct 200 tasks of 5 classes each. Fig. 5 shows the performance on the first task throughout the whole sequence. We see that in a very long sequence of tasks, the previous trends are kept. Specifically, we see that as the model reaches the later stages of the sequence, finetuning with CE reaches LwF, and finetuning with SupCon outperforms the competitive baseline of ER with a small buffer *without* access to buffer data. Furthermore, we observe as in the previous section that SimCLR performance stays flat. In the supplementary material we also demonstrate that this pattern is not limited to the first task but is maintained for other tasks along the sequence.

4.2. Effects of Increased Model Capacity

Next, we use linear probes to evaluate the effect of increased model width and depth. Recently [39] has suggested that increased model size must be strictly combined with pre-training in order to get increased robustness to catastrophic forgetting. We revisit this in the context of both wider and deeper models on a SplitCIFAR100 sequence of 10 tasks with 10 classes each. Table 2 shows the results using a Resnet18 with a much wider model (128 vs. 32) and then a much deeper model (101 layers). We report both the LP accuracy of task 1 at the end of the sequence and the average of LP accuracies for all the tasks at the end of the sequence.

First, we can see that as in the other cases, the LP accuracy of finetuning is higher than observed accuracy, suggesting that forgetting is less catastrophic than what is indicated by observed accuracy. Secondly, we see that finetuning evaluated using the observed accuracy is particularly deceptive in revealing how the model representations changes with increasing capacity. The observed and accuracy of task 1 are relatively close despite increasing capacity (wider or deeper) while the corresponding LP accuracies show substantial gaps. Using observed accuracy one would conclude that increasing width and capacity of the model without applying any CL specific method does not reduce forgetting. This is consistent with the observations of [39], which evaluates only on observed accuracy. However, if we observe the LP accuracy, it reveals a more clear picture of what oc-

		Task 1	Obs. Acc. Task 1	Task 1 LP	LP Acc. All	Avg. Obs.
		Acc.	at T=10	T=10	T=10	Acc.
Finetuning(CE)	RN18, Width=32	82.2	20.8	64.8	70.8	35.5
	RN18, Width=128	83.3	21.2	70.5	74.2	36.8
	RN101, Width=32	82.9	19.8	67.9	72.4	35.9
ER-M5	RN18, Width=32	82.7	52.1	74.2	75.7	54.8
	RN18, Width=128	83.6	54.8	75.6	77.3	55.4
	RN101, Width=32	83	50.9	74.5	76.1	51.6
ER-M20	RN18, Width=32	82.4	61.3	76	76.4	65.2
	RN18, Width=128	83.2	63.5	78.8	80.1	67
	RN101, Width=32	82.9	60.7	77.1	77.5	63.9
LwF	RN18, Width=32	82.1	36.2	70.1	73.4	47.7
	RN18, Width=128	83.9	37.7	74.8	76.7	49.1
	RN101, Width=32	82.5	35.5	71.0	74.6	46.3

Table 2. Final Accuracy of 10 task SplitCIFAR100 sequence with variable width and depth in the offline setting. M indicates the number of samples per task used in the ER buffer. We observe that simple finetuning and LwF baselines show large forgetting, which do not improve significantly with width or depth. On the other hand, the LP evaluation reveals that representation quality for finetuning and LwF becomes closer to strong CL methods that store samples and also use more compute such as ER. Furthermore, LP evaluations reveal LwF (which does not store prior data) with wider models can rival ER

		Task 1	Obs. Acc. Task 1	Task 1 LP	LP Acc. All	Avg. Obs.
		Acc.	at T=10	T=10	T=10	Acc.
Finetuning(CE)	RN18, Width=32	18.6	12.2	39.8	36.4	22.3
	RN18, Width=128	19.4	12.7	42.3	41.7	19.8
	RN101, Width=32	14.6	11.8	28.2	29.4	14.5
ER-M5	RN18, Width=32	18.8	27.3	36.0	40.1	33.8
	RN18, Width=128	19.5	28.9	54.7	47.9	31.6
	RN101, Width=32	15.0	24.7	37.1	30.4	24.3
ER-M20	RN18, Width=32	18.4	32.0	46.8	43.5	34.7
	RN18, Width=128	20.0	31.8	51.2	50.7	32.5
	RN101, Width=32	14.5	25.4	36.5	33.9	24.3
LwF	RN18, Width=32	18.5	13.4	29.5	36.0	22.7
	RN18, Width=128	19.7	18.3	34.6	39.1	22.1
	RN101, Width=32	14.8	11.1	25.4	22.8	16.8

Table 3. Final Accuracy of 10 task SplitCIFAR100 sequence in the Online Setting. LP evaluations show that width substantially improves online representation learning, while observed Avg Accuracies suggest it decreases. Increasing depth on the other hand appears to be less effective in the online setting.

curs at the representation level, suggesting that larger models can indeed reduce forgetting even when trained from scratch without explicit control of forgetting. Moreover, we see that at the representation level as model capacity increases, naive finetuning becomes much closer in performance to costly (and under privacy constraints unusable) CL methods such as ER, which require more compute and memory.

In comparing depth and width we also see some key distinctions - increasing width appears to help more than increasing depth. For ER we also see that increasing depth yields a lower observed accuracy, while the LP evaluation suggests the representations are similar. Similarly, in Tab. 3 we report the results for the online task-incremental setting [7, 27]. In this setting, momentum tends to be detrimental to performance, thus we use a fixed learning rate of 0.01 with no momentum. We see similar behavior to the previous case, the larger models can end up appearing to do worse if we consider observed accuracy, but perform better using LP evaluation. Wider models appear to do particularly well in the online setting while deeper models have degraded LP accuracy in this setting. Finally we see that LwF which is a regularization method performs poorly in this setting. Indeed regularization based methods do poorly in the online setting. This suggests that amongst methods without access to a replay buffer, finetuning may provide the best representation learning.

4.3. Low-Cost Remembering with SupCon

The observed low representation forgetting properties of finetuning with SupCon loss suggest if we can approximate a classifier using it's representation it would allow for low cost remembering upon encountering a previously observed task. We thus evaluate the use of the NME in combination with SupCon. As discussed in Sec 3.4 such an approach allows a simpler alternative to ER methods and moreover facilitates fast remembering not relying on a buffer and repeatedly training the model with old samples. We use the SplitCIFAR100 dataset to compare against several CL specific methods such as LwF and ER in Tab. 4. We use a memory with $M = 5$ samples per class for this. We chose the exemplars at random to simulate re-encountering an old

Method	Obs Acc. Task 1 at T=10	Avg. Obs. Acc.
Finetuning(CE)	20.8%	35.5%
ER-M5	52.1%	54.8%
ER-M20	61.3%	65.2%
LwF	36.2%	47.7%
Finetune(SupCon) + NME-M5	48.0%	53.9%

Table 4. Final Accuracy of 10-task SplitCIFAR100 sequence comparing only the observed accuracy and SupCon+NME. Supcon+NME gives superior performance to CL specific methods such as LwF and nearly matches the performance of ER with a similar memory size while not needing access to the memory during task training.

task. We observe that just applying the simple approximation with a small number of samples allows for a rapid recovery of the performance with the finetuning approach alone, exceeding the performance of LwF on overall accuracy and task 1 accuracy. The overall performance is close to that of ER with the same memory size and slightly below the ER performance on task 1 at the end of the sequence. On the other hand ER requires samples to be available during the entire training sequence, requires the addition of extra algorithmic elements specifically to control forgetting, and uses substantially more compute ($\approx 2\times$ that of the finetuning step in this case).

4.4. Depth-wise Probes and Comparison to CKA

We consider a 2-task SplitCIFAR10 setting from [38]. We use the same models and training procedures and subsequently evaluate forgetting. In Tab. 5, we study the shift in representations of each block of the network by measuring the performance of LP on task 1 before and after training the network on task 2.

First we see that the observed accuracy decreases from 85% to 63%, suggesting large degradation in performance and large forgetting. However, following the optimal classifier evaluation protocol the accuracy degradation is seen to be only 5.7%, without any CL method applied to control forgetting. This suggests that the representations are still highly useful for Task 1 despite training on Task 2. Second, similar to [38] we note that the forgetting is concentrated at the top layers. Indeed early layers in the network experience almost no representation forgetting and in some cases improve their usefulness with regards to Task 1. Ramasesh *et al.*'s [38] analysis also showed forgetting occurring in early layers to a lower degree than in higher layers and suggested that forgetting is extreme in the upper layer representations. Specifically, the authors measured linear CKA [23] performance between layers (given in Tab. 5) showing that this similarity metric dropped progressively from close to 1 to 0.2 for both ResNet and VGG models. However, our evaluation suggests that forgetting does not exist in lower layers and also the loss in information is less catastrophic at higher layers than suggested by [38].

ResNet: Network Acc. on T-1 after T-2 training: 63.64%				
Block	LP Acc. Post T-1	LP Acc. Post T-2	Δ Acc.	CKA*
B-0	63.54%	64.62%	+1.08%	0.97
B-1	68.24%	69.50%	+1.26%	0.93
B-2	71.62%	71.34%	−0.28%	0.88
B-3	77.64%	76.52%	−1.12%	0.78
B-4	80.06%	78.98%	−1.08%	0.31
B-5	85.82%	80.10%	−5.72%	0.22
VGG: Network Acc. on T-1 after T-2 training: 57.88%				
B-0	67.94%	66.86%	−1.08%	0.95
B-1	73.60%	72.52%	−1.08%	0.93
B-2	78.58%	75.68%	−2.90%	0.85
B-3	81.54%	75.48%	−6.06%	0.66

Table 5. Representation forgetting of Task 1 measured via optimal linear probes (LP) on ResNet and VGG. The Accuracy degradation of LP trained on activations of stages (blocks of convolutions) before and after observing Task 2 suggests that the representations are still highly useful for Task 1 despite training on Task 2. *Note CKA results are taken from [38] for comparison.

5. Conclusion

We have highlighted the importance of evaluating representations and not just task accuracy in CL. Our results suggest a) representation forgetting under naive finetuning in supervised settings is not as catastrophic as other metrics suggest b) We demonstrate that without evaluation of features the effects of model size on forgetting and representation learning will be misinterpreted. c) We show that the self-supervised SimCLR loss and supervised SupCon loss have lesser representation forgetting in long tasks sequences, maintaining or increasing performance on early tasks. These results open up potential new directions for approaches in continual learning. One such direction of using memories that are not available to the learner during training is evaluated with promising initial results.

Limitations and Future Work Our work focuses on comparing linear probe performance as a proxy of knowledge retained from past tasks. However, task performance may not be the only criteria to fully evaluate knowledge retention related to past data. Another limitation of our work is that it currently focuses on the task-incremental setting and does not consider the important class-incremental setting, a subject for future studies. Finally though our work studies a diverse task sequence ImageNet → Scenes → CUB → Flowers, to fully understand the behavior of representation forgetting, results over more distant tasks may be needed (e.g. ImageNet → Sketch Images [13, 44])

References

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

[2] Rahaf Aljundi, Punarjay Chakravarty, and Tinne Tuytelaars. Expert gate: Lifelong learning with a network of experts. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 2

[3] Gaurav Arora, Afshin Rahimi, and Timothy Baldwin. Does an LSTM forget more than a CNN? an empirical study of catastrophic forgetting in NLP. In *Proceedings of the The 17th Annual Workshop of the Australasian Language Technology Association*, pages 77–86, Sydney, Australia, 4–6 Dec. 2019. Australasian Language Technology Association. 2

[4] Yoshua Bengio, Aaron Courville, and Pascal Vincent. Representation learning: A review and new perspectives. *IEEE transactions on pattern analysis and machine intelligence*, 35(8):1798–1828, 2013. 1

[5] Lucas Caccia, Rahaf Aljundi, Tinne Tuytelaars, Joelle Pineau, and Eugene Belilovsky. Reducing representation drift in online continual learning. *arXiv preprint arXiv:2104.05025*, 2021. 1, 3

[6] Rich Caruana, Steve Lawrence, and Lee Giles. Overfitting in neural nets: Backpropagation, conjugate gradient, and early stopping. *Advances in neural information processing systems*, pages 402–408, 2001. 11

[7] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and Marc'Aurelio Ranzato. Continual learning with tiny episodic memories. *arXiv preprint arXiv:1902.10486*, 2019. 2, 3, 5, 7

[8] Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. *arXiv preprint arXiv:2002.05709*, 2020. 2, 3

[9] Patryk Chrabaszcz, Ilya Loshchilov, and Frank Hutter. A downsampled variant of imagenet as an alternative to the cifar datasets. *arXiv preprint arXiv:1707.08819*, 2017. 4

[10] Ian Goodfellow, Yoshua Bengio, and Aaron Courville. Deep learning. Book in preparation for MIT Press, 2016. 1

[11] 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

[12] Florian Graf, Christoph Hofer, Marc Niethammer, and Roland Kwitt. Dissecting supervised constrastive learning. In *International Conference on Machine Learning*, pages 3821–3830. PMLR, 2021. 3

[13] David Ha and Douglas Eck. A neural representation of sketch drawings. *arXiv preprint arXiv:1704.03477*, 2017. 8

[14] Raia Hadsell, Dushyant Rao, Andrei A Rusu, and Razvan Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in cognitive sciences*, 2020. 1

[15] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. *arXiv preprint arXiv:1911.05722*, 2019. 2, 3

[16] 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. 11

[17] Xu He, Jakub Sygnowski, Alexandre Galashov, Andrei A Rusu, Yee Whye Teh, and Razvan Pascanu. Task agnostic continual learning via meta learning. *arXiv preprint arXiv:1906.05201*, 2019. 1

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

[19] Dapeng Hu, Qizhengqiu Lu, Lanqing Hong, Hailin Hu, Yifan Zhang, Zhenguo Li, Alfred Shen, and Jiashi Feng. How well self-supervised pre-training performs with streaming data? *arXiv preprint arXiv:2104.12081*, 2021. 2

[20] Ferenc Huszár. On quadratic penalties in elastic weight consolidation. *arXiv preprint arXiv:1712.03847*, 2017. 2, 5

[21] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. Supervised contrastive learning. In H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 18661–18673. Curran Associates, Inc., 2020. 2, 3, 4, 5

[22] 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. *arXiv preprint arXiv:1612.00796*, 2016. 2

[23] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019. 2, 3, 8

[24] Alex Krizhevsky, Geoffrey Hinton, et al. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009. 4

[25] Xilai Li, Yingbo Zhou, Tianfu Wu, Richard Socher, and Caiming Xiong. Learn to grow: A continual structure learning framework for overcoming catastrophic forgetting. In *International Conference on Machine Learning*, pages 3925–3934. PMLR, 2019. 2

[26] Zhizhong Li and Derek Hoiem. Learning without forgetting. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(12):2935–2947, 2017. 2, 4, 5, 6, 11, 12, 13

[27] David Lopez-Paz et al. Gradient episodic memory for continual learning. In *Advances in Neural Information Processing Systems*, pages 6467–6476, 2017. 2, 5, 7

[28] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. 11

[29] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017. 11

[30] Zheda Mai, Ruiwen Li, Hyunwoo Kim, and Scott Sanner. Supervised contrastive replay: Revisiting the nearest class mean classifier in online class-incremental continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3589–3599, 2021. 2, 3

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

[32] In Jae Myung. Tutorial on maximum likelihood estimation. *Journal of mathematical Psychology*, 47(1):90–100, 2003. 2

[33] 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. 2

[34] Cuong V Nguyen, Yingzhen Li, Thang D Bui, and Richard E Turner. Variational continual learning. *arXiv preprint arXiv:1710.10628*, 2017. 2

[35] M-E. Nilsback and A. Zisserman. Automated flower classification over a large number of classes. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image Processing*, Dec 2008. 5

[36] Edouard Oyallon. Building a regular decision boundary with deep networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5106–5114, 2017. 2

[37] Ariadna Quattoni and Antonio Torralba. Recognizing indoor scenes. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 413–420. IEEE, 2009. 4, 5, 11

[38] Vinay V Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. *arXiv preprint arXiv:2007.07400*, 2020. 2, 3, 4, 5, 8, 11

[39] Vinay Venkatesh Ramasesh, Aitor Lewkowycz, and Ethan Dyer. Effect of scale on catastrophic forgetting in neural networks. In *International Conference on Learning Representations*, 2022. 2, 6

[40] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proc. CVPR*, 2017. 2, 3, 5

[41] Amir Rosenfeld and John K Tsotsos. Incremental learning through deep adaptation. *IEEE transactions on pattern analysis and machine intelligence*, 42(3):651–663, 2018. 2

[42] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3):211–252, 2015. 1, 4, 5, 11

[43] 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

[44] Patsorn Sangkloy, Nathan Burnell, Cusuh Ham, and James Hays. The sketchy database: learning to retrieve badly drawn bunnies. *ACM Transactions on Graphics (TOG)*, 35(4):1–12, 2016. 8

[45] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014. 11

[46] Catherine Wah, Steve Branson, Peter Welinder, Pietro Perona, and Serge Belongie. The Caltech-UCSD Birds-200-2011 Dataset. Technical Report CNS-TR-2011-001, California Institute of Technology, 2011. 5, 11

[47] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010. 4

[48] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014. 2, 3

[49] Friedemann Zenke, Ben Poole, and Surya Ganguli. Improved multitask learning through synaptic intelligence. In *Proceedings of the International Conference on Machine Learning (ICML)*, 2017. 2