Rehearsal revealed:
The limits and merits of revisiting samples in continual learning

Eli Verwimp*
KU Leuven
eli.verwimp@kuleuven.be

Matthias De Lange*
KU Leuven
matthias.delange@kuleuven.be

Tinne Tuytelaars
KU Leuven
tinne.tuytelaars@kuleuven.be

Abstract

Learning from non-stationary data streams and overcoming catastrophic forgetting still poses a serious challenge for machine learning research. Rather than aiming to improve state-of-the-art, in this work we provide insight into the limits and merits of rehearsal, one of continual learning's most established methods. We hypothesize that models trained sequentially with rehearsal tend to stay in the same low-loss region after a task has finished, but are at risk of overfitting on its sample memory, hence harming generalization. We provide both conceptual and strong empirical evidence on three benchmarks for both behaviors, bringing novel insights into the dynamics of rehearsal and continual learning in general. Finally, we interpret important continual learning works in the light of our findings, allowing for a deeper understanding of their successes.[1]

1. Introduction

Recent advances of neural networks have shown promising results by surpassing human capabilities in a wide range of tasks [36, 29, 35]. However, these tasks are typically highly confined and remain static after deployment. This stems from a major limitation in neural network optimization, namely the assumption of independent and identically distributed (iid) training and testing distributions. When the iid assumption is not satisfied during learning, neural networks are prone to catastrophic forgetting [12], causing them to completely forget previously acquired knowledge.

Continual or lifelong learning strives to overcome this static nature of neural networks with a wide range of mechanisms [10], among which *rehearsal* has shown promising results [22, 28, 7, 9]. Rehearsal aims to approximate the observed input distributions over time and later resamples from this approximation to avoid forgetting. Although there are various ways to use the input distribution approximation

---

*Authors contributed equally.

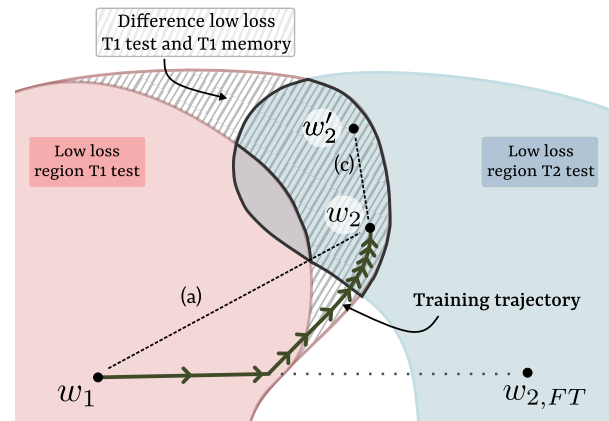[1]Code: https://github.com/Mattdl/RehearsalRevealed

Figure 1: Illustration of our findings, visualized as the loss values in parameter space. When using rehearsal to train task 2 ($T2$) after training task 1 ($T1$), i.e. from w_1 to w_2, the learning trajectory will first move towards a finetuned minimum $w_{2,FT}$ but deflects at a high-loss ridge of $T1$'s low-loss region. The crux is w_2 ending up in the striped area where low loss for $T1$ memory is in contrast to the higher loss observed for $T1$ test data. Additionally, we show empirically that linear low loss paths (a) and (c) exist between w_1 and w_2, and between two models trained with different rehearsal memories w_2 and w_2'.

(Section 2), this study focuses and refers to *rehearsal* in its most direct form, i.e. by sampling the input distribution in a limited rehearsal memory from which samples are revisited in later training batches.

Despite the wide use of rehearsal, due to its simplicity and effectiveness, fundamental analysis of why it works and what its limitations are is lacking in literature. Furthermore, we believe that insights into its internal workings might deepen our understanding of the catastrophic forgetting phenomenon in general. In this work, we make an initial attempt from the perspective of loss landscapes.

Fundamental open questions. Motivated by recent advances in continual learning literature, we define two fun-

damental open questions. Early work in rehearsal raised concerns about overfitting to the rehearsal memory as a consequence of repeated optimization on this limited set of data [22]. Following work [7] confirms this as the replay memory becomes perfectly memorized by the model, but also finds rehearsal to remain effective in terms of generalization. This leads to two open questions. First, "Why does rehearsal work even though overfitting on the rehearsal memory occurs?". Second, "How does overfitting on the rehearsal memory influence generalization?".

Motivation. To formalize these inquiries for this study, we formulate two main hypotheses motivated by advances in prior work. Firstly, recent work empirically finds continual learning minima of individual tasks to be linearly connected through a low-loss path with the multitask solution, when starting from the same initialization [25]. Multitask learning simultaneously learns multiple tasks, which rehearsal ultimately aims to approximate in the continual learning regime. Therefore, we hypothesize the rehearsal solution resides in the same low-loss region as the original task and the multitask solutions. Note that we refer *the same* low-loss region and not just any region with similar loss. Secondly, large models are able to completely memorize small sets of data, such as rehearsal memories, without any generalization capabilities [40]. Therefore we hypothesize that overfitting does harm generalization after all.

Contributions. Our two hypotheses are formalized as:

1. **Hypothesis 1:** Rehearsal is effective as its solution tends to reside in *the same* low-loss region as the task minimum from which learning is initiated.

2. **Hypothesis 2:** Rehearsal is suboptimal as it tends to overfit on the rehearsal memory, consequently harming generalization.

Section 4 provides extensive empirical evidence on MNIST, CIFAR, and Mini-Imagenet data sequences to test both hypotheses. Our findings can be summarized as follows:

1. The results in Section 4.1 unanimously support Hypothesis 1. Even after learning longer sequences of up to 5 tasks with rehearsal, the new minimum is found in the same low-loss region as for the first task. This suggests the existence of overlapping low-loss regions between the task's input distribution and its approximation by the rehearsal memory.

2. As suggested in [22, 7], Section 4.2 confirms overfitting on the rehearsal memory. However, this overfitting by itself is insufficient to explain harming generalization, as following Hypothesis 1 there exists an overlapping low-loss region with the one of the task's input distribution. Our empirical evidence finds clues

that internal rehearsal dynamics draw the learning trajectory near a high-loss ridge of the rehearsal memory. Additionally, near the high-loss ridge the approximation by the rehearsal memory for the task's input distribution loss deteriorates. Therefore, we can confirm the suboptimality of rehearsal in Hypothesis 2, but generalization is harmed by the combination of overfitting, internal rehearsal dynamics, and the low-quality approximation of the rehearsal memory near its high-loss ridges.

Furthermore, Section 4.3 provides additional evidence with a simple heuristic to withdraw from the rehearsal memory high-loss ridge, showing promising results compared to standard rehearsal. Additionally, Section 4.4 conceptually analyses the internal rehearsal dynamics w.r.t. our results. We believe these findings bring new insights into ultimately understanding catastrophic forgetting and proposed methods in literature. In Section 5 we discuss recent salient works in the light of our findings, namely GEM [22], MIR [2], and GDumb [27].

2. Related Work

Continual learning. Learning from non-stationary data streams has been well studied in literature. Following [10], we divide the main approaches to avoid catastrophic forgetting into three main families: regularization-based, parameter isolation, and replay methods. *Regularization-based methods* put regularization constraints on the parameter space when learning future tasks to preserve acquired knowledge. By using knowledge distillation [14], Li et al. [20] insist on staying close to the model distribution from before learning on the new data. Other regularization methods [17, 1, 39] capture parameter importance based on second-order approximations in task minima. When learning a new task, the squared $l2$-distance to the previous task minimum is minimized, weighed by importance. *Parameter isolation methods* allocate subsets of the model parameters to specific tasks. The allocation ranges from pruning based heuristics [24] and learnable masks [23, 32] to instantiating new subnetworks per task [30].

Replay methods in continual learning first compress the observed data and later use this approximation to compensate for the non-iid nature of the data stream. One way, called *rehearsal* is by storing a subset of representative samples or exemplars in a constrained rehearsal memory [22, 28]. Another way, called *pseudo-rehearsal*, learns the input distribution with a generative model [33, 34, 31]. Although promising results for generative models have been reported in confined setups[37], learning both the encoder and decoder in a continual fashion can be cumbersome. In comparison, sampling is a more computationally efficient

process to approximate the input distribution, avoiding optimization of an additional decoder.

Regarding the rehearsal memory, different approaches to exploit the exemplars have been explored. A first straightforward approach is Experience Replay (ER) [7] adding a batch of exemplars to each new batch of data. The union of these is then optimized for the same objective. ER is the plainest form of rehearsal, resulting in their terminology being used interchangeably. The exemplars can also be used for knowledge distillation [28] or both distillation and the original objective [5]. In the latter two works, the exemplars are stored based on optimally representing the feature mean, which is an exhaustive operation and requires recalculating the class means at task boundaries for nearest-mean classification [28]. Therefore, De Lange et al. [9] propose an online alternative with continually representative class means in the evolving embedding space. A second approach to exploit the exemplars is by using them to compose constraints in the gradient space [22]. The gradient of the new task data is confined to point in the same direction as the task-specific gradients, which are calculated based on the exemplars. Whenever the gradient fails to satisfy the constraints, it is projected to the least-squares solution on the constraining polyhedral convex cone. The gradient space has also been proposed to select samples for storage in the ER rehearsal memory [3]. As it is infeasible to repeatedly rehearse the entire rehearsal memory, a retrieval strategy is required to select the exemplars for ER. Random retrieval has been widely adopted [7, 6, 9, 8] while an alternative evaluates the potentially highest increase in the loss [2]. Recent work has shown that storing balanced greedy subsets of the data stream attains major increases in performance when learning offline with the rehearsal memory [27]. This indicates the difficulty of learning in a continual fashion compared to offline learning and suggests the undiscovered potential of fully exploiting the exemplars when learning continually. We discuss the retrieval and storage strategies for the rehearsal memory in more detail in Section 3.2.

3. Rehearsal in Continual Learning

Before considering our analysis, we outline the continual learning setup. Next, Section 3.1 provides an interpretation of the loss functions in the parameter space in continual learning and compares this to the iid setting. Finally, Section 3.2 formally introduces rehearsal and the approximations made during learning.

Typically, machine learning models rely on sampling from a stationary data distribution for both the learner and evaluator. In continual learning, the learner distribution can change over time [9]. Specifically in this work, we consider changes in the learner distribution $\mathcal{D}$ at discrete time steps t. This setup is also referred to as the class-incremental setup, defined as a sequence of N task sets D_t. Each set D_t consists of a set of samples $(\mathbf{x}_i, y_i)$ drawn from $\mathcal{D}_t$, with input data $\mathbf{x}_i$ and class label y_i. $T1$ and $T2$ refer to the first two tasks, with future tasks referenced similarly. The evaluator's test set D_{eval} is constant and contains samples drawn from all $\mathcal{D}_t$, sampled mutually exclusive w.r.t. the learner's set D_t.

3.1. Stochastic optimization in Continual Learning

Standard in machine learning is to optimize parameters w of a predicting function f_w w.r.t. the loss function $\mathcal{L}$. Ideally, the risk of the entire data generating distribution is minimized for f_w. However, this distribution is unknown and we only have the training set available to learn from, which we use to minimize the empirical risk R:

$$R(D, w) = \frac{1}{|D|} \sum_{(\mathbf{x}_i, y_i) \in D} \mathcal{L}(f_w(\mathbf{x}_i), y_i) \qquad (1)$$

The most straightforward way to optimize is by taking steps in the direction of steepest descent, indicated by the negative gradient g of the empirical risk. However, to maintain scalability for large datasets the stochastic gradient $\tilde{g}$ is calculated on a mini batch B from the full dataset. This approximation provably converges, as long as the expected stochastic gradient equals the full set's gradient [4]. The expected value of $\tilde{g}$ equals

$$\mathbb{E}[\tilde{g}] = \sum_{(\mathbf{x}_i, y_i) \in D} p_i \, \nabla \mathcal{L}(f_w(\mathbf{x}_i), y_i) \qquad (2)$$

with $p_i = S(d_i)$ the sampling probability for sample $d_i \in D$. For a uniform sampling distribution, the expectation of $\tilde{g}$ becomes equal to g.

However, in continual learning p_i is non-uniform. In our setting, it is only non-zero and uniformly distributed over samples belonging to the current task. Reformulating Eq. 2 for task datasets $D_{1..N}$ and current task set D_c reduces to

$$\mathbb{E}[\tilde{g}] = \sum_{D_t} \sum_{(\mathbf{x}_i, y_i) \in D_t} p_i \, \nabla \mathcal{L}(f_w(\mathbf{x}_i), y_i)$$
$$= \sum_{(\mathbf{x}_i, y_i) \in D_c} p_i \, \nabla \mathcal{L}(f_w(\mathbf{x}_i), y_i) \qquad (3)$$

As a consequence, when training on task set D_c, optimization disregards the other task distributions and calculates the gradients solely from the perspective of the current task loss function. This is the key issue resulting in catastrophic forgetting of previous tasks in neural networks.

3.2. Learning with rehearsal

Continual learning systems typically have fixed memory budgets and preferably do not grow with the number of tasks. Consequently, this requires making a trade-off with

rehearsal methods approximating the input distribution instead of storing all observed data. The operational memory $\mathcal{M}$ defines the additional memory requirements during learning of the continual learner [9]. Rehearsal implements $\mathcal{M}$ as a fixed rehearsal memory, with the memory for task distribution $\mathcal{D}_t$ indicated by $\mathcal{M}_t$. The expected gradient in Eq. 3 then sums over the union of both the current task set D_c and the stored samples in $\mathcal{M}$. This union could also be constructed with a generative model for pseudo-rehearsal as discussed in Section 2, but the scope of this study is limited to rehearsal by sampling.

Rehearsal is confined by two main constraints. Firstly, the limited rehearsal memory size $|\mathcal{M}|$ to approximate the input distribution, and secondly, the limited mini batch size $|B|$ in SGD to construct near iid mini batches that approximate Eq. 2. Selecting samples from the input distribution to store in $\mathcal{M}$ is handled by the *storage policy*, whereas samples for mini batch B are selected based on sampling distribution S, defined by the *retrieval policy*. Algorithm 1 summarizes rehearsal for a single mini batch B. In the following, we assume a random task-balanced storage policy dividing full memory $\mathcal{M}$ equally over all seen tasks. The retrieval policy's sampling distribution is uniform over $\mathcal{M}$.

Algorithm 1 Continual learning with Rehearsal.
1: **function** REHEARSALBATCH($B, \mathcal{M}$)
2: $\tilde{B} \leftarrow$ RETRIEVALPOLICY($\mathcal{M}$)   ▷ Retrieve exemplars
3: $w \leftarrow SGD\left(B \cup \tilde{B}, w\right)$ ▷ Optimize objective for union
4: STORAGEPOLICY($\mathcal{M}, B$)   ▷ Update rehearsal memory

---

4. Revealing Rehearsal: Analysis

Although rehearsal has been widely adopted in continual learning literature, the proposed approaches are often based on heuristics. In contrast, in this section we attempt to gain fundamental insights from the perspective of the loss landscape in parameter space which is key to progress our understanding of catastrophic forgetting in general. In the following, we define w_1 as the minimum obtained by the model after learning until convergence for the first task ($T1$) with task distribution $\mathcal{D}_1$. Subsequent models w_i are learned via rehearsal as defined in Section 3.2, e.g. w_2 is the solution for $T2$ initialized from w_1 and using samples from $T1$ stored in $\mathcal{M}$ during learning. We define a low-loss region as a connected part of the parameter space where the loss value stays below a small value. Any set of parameters in such a region will perform nearly equally well for the same data subset. Such regions are an important part of the parameters space because they are linked to better generalization capabilities of a model [15, 16].

Our two hypotheses are formulated in Section 1. We scrutinize the hypotheses in Section 4.1 and Section 4.2 re-

spectively with strong empirical evidence on three continual learning benchmarks and discuss the consequences with relation to hypotheses in previous work. Using these insights, we exemplify a simple baseline in Section 4.3 further confirming our hypotheses. Section 4.4 conceptually analyses the observed rehearsal dynamics. Finally, Section 4.5 compares our observations with prior work.

Datasets. We use three datasets for the rehearsal analysis: Split-MNIST, Split-CIFAR10, and Split-miniImagenet. Split-MNIST divides the original MNIST [19] database into 5 tasks with each 2 classes. Split-CIFAR10 uses the CIFAR10 dataset [18], and is similarly split into five two-label tasks. Split-miniImagenet is constructed by dividing miniImagenet [38], a 100-class subset of Imagenet, into 20 tasks with 5 labels each. For the analyses, only the first five tasks (i.e. 25 classes) are used in both training and evaluation. Unless mentioned otherwise, the MNIST and CIFAR10 sets are trained online, which implies that except for the data in memory, all data is seen only once. Split-miniImagenet uses 10 epochs per task since it is a much harder dataset. For brevity, we refer to the setups as MNIST, CIFAR10, and Mini-Imagenet.

Architectures and optimization. MNIST is trained on a fully-connected network, with two hidden layers of 400 nodes each. Both CIFAR10 and Mini-Imagenet are trained on a reduced Resnet18 [13], introduced by Lopez-Paz et al. [22]. All networks are trained with a shared final layer, referred to as *shared head*. This is opposed to the easier task-incremental setting which uses a different head per task [10]. The memory sizes for the three datasets are respectively 50, 100, and 100 samples per task for MNIST, CIFAR10, and Mini-Imagenet. All models are optimized using vanilla SGD with a cross-entropy loss. Each mini batch during training consists of 10 new and 10 memory samples, except for the first task which only has 10 new samples. Our code uses the Avalanche framework [21] to enhance reproducibility. We summarize all details in Appendix.

4.1. Hypothesis 1: Empirical evidence

To test Hypothesis 1, we need to quantify whether the model before and after learning a new task remains in the same low-loss region. For clarity, we start with an example of two subsequent tasks: after learning the first task (w_1), we can either learn $T2$ using rehearsal ending up in w_2, or finetune for $T2$ without rehearsal to end up in a $T2$ minimum ($w_{2,FT}$) while typically catastrophically forgetting $T1$. Figure 2 shows a two-dimensional projection in the parameter space on the plane defined by these three models, for CIFAR10 and Mini-Imagenet (see Appendix for MNIST results). The projection of the learning trajectory in parameter space in Figure 2 illustrates the large steps initially moving towards the $T2$ minimum $w_{2,FT}$, to then bend with smaller steps along the high-loss contour lines of $T1$'s
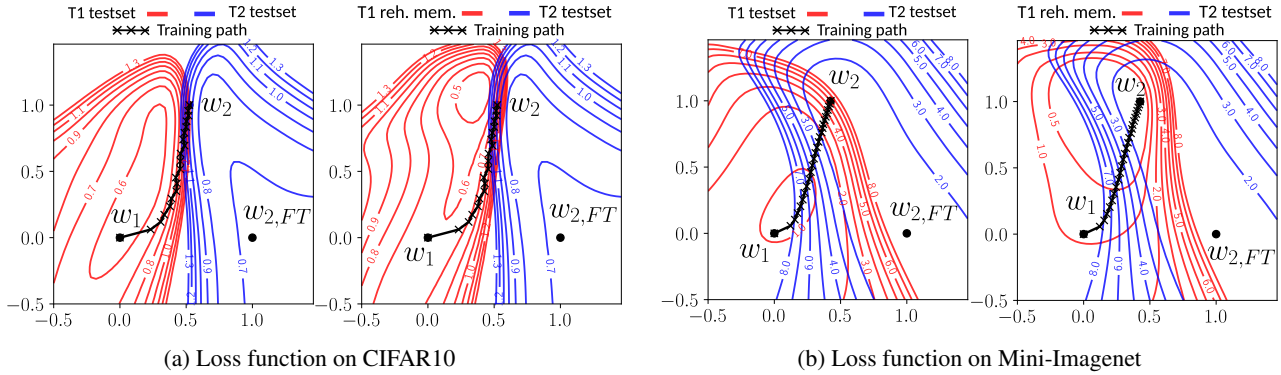
(a) Loss function on CIFAR10　　　　　　　　(b) Loss function on Mini-Imagenet

Figure 2: Projection of learning trajectories in parameter space on the plane of w_1, w_2 and $w_{2,FT}$. For the same $T2$ loss (blue), the loss for $T1$ (red) is calculated in two different ways. *(a) and (b) left*: $T1$ loss for the vast test set. *(a) and (b) right*: $T1$ loss for the limited rehearsal memory. Even in the 2D planes, overfitting to the rehearsal memory loss is clear. See Appendix for details.
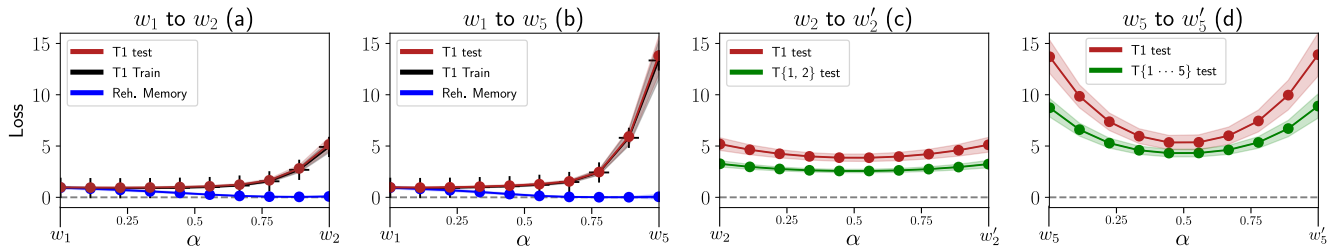

Figure 3: Avg. loss and standard deviation on linear paths between the models declared in Figure 1, trained on Mini-Imagenet and sampled 100 times for different model initializations and memory populations. A path from w_i to w_j is calculated as $(1-\alpha)w_i + \alpha w_j$. *(a) and (b)*: Loss on the linear path between the model after learning $T1$ (w_1) and the model after learning with rehearsal on $T2$ and $T5$ respectively. *(c) and (d)*: Loss of the path between two models learned with different memory populations, after $T2$ and $T5$ respectively. Red is the loss on the T1 testset, green the average loss of all tasks up to $T2$ and $T5$ respectively. Results contained no outliers with a higher loss on the path compared to the loss of the models.

rehearsal memory loss landscape. These findings indicate w_1 and w_2 remain in the same low-loss region, while w_2 is drawn to the high-loss ridge of this region.

To further support these findings, we analyze the loss in Figure 3 both after the second task (w_2) and after a sequence of 5 tasks (w_5) of Mini-Imagenet. Focusing on the loss values of $T1$ on the linear path from w_1 to w_2 (Figure 3a) and from w_1 to w_5 (Figure 3b), we observe monotonically increasing loss values for $T1$, hence indicating the compared models are in or on the edge of the same loss basin. The results are averaged over 100 runs with both different initialization and rehearsal memory population. We refer to Appendix for CIFAR10 and MNIST results, having the same trends as for Mini-Imagenet. Conform to literature, rehearsal is effective in alleviating catastrophic forgetting, as it significantly improves results compared to plain finetuning with 60%, 9% and 7% gain in average accuracy over all tasks for the increasingly difficult MNIST, CIFAR10, and Mini-Imagenet benchmarks.

Another question that arises within this analysis is: *"Do these findings still hold for different rehearsal memory populations and how are their solutions connected?"* To answer this, we extend the previous analysis by considering two alternative memory populations resulting in w_2' and w_5'. Figure 3 plots the loss values for the linear interpolation between the models w_2 and w_2', and w_5 and w_5'. As there is no increase in the loss value, we can conclude that for different random memory populations the resulting models all reside in the same low-loss basin as w_1. Figure 1 summarizes the setup for the $T2$ paths (a) and (c) in Figure 3, with paths (b) and (d) defined analogously after 5 tasks.

4.2. Hypothesis 2: Empirical evidence

Testing Hypothesis 2 requires an experiment showing overfitting on the rehearsal memory with deteriorating generalization as overfitting arises. We expand the initial two-task experiment for Hypothesis 1 by analyzing the optimization trajectory in the loss landscape of both the re-
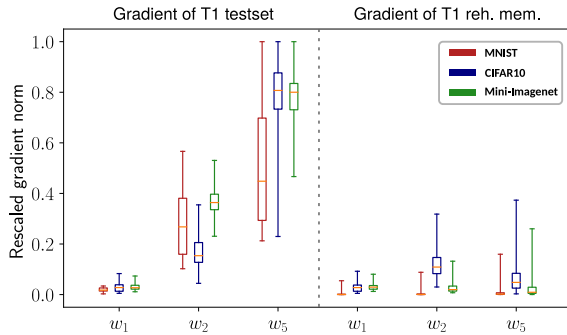
Figure 4: Comparison of rescaled gradient $l2$-norms of the $T1$ testset (left) and rehearsal memory (right) after learning $T1$ (w_1), $T2$ (w_2) and $T5$ (w_5), averaged over 100 seeds. Boxplot whiskers indicate *max* and *min* values. Results are rescaled by the maximal norm value per dataset.

hearsal memory and the test dataset. Starting with the loss landscape for the rehearsal memory of $T1$, Figure 2a (right) and Figure 2b (right) depict the rehearsal minimum w_2 to be in a low-loss region next to a high-loss ridge. However, in perspective of the loss landscape computed on the full test set in Figure 2a (left) and Figure 2b (left), w_2 resides *on* the high-loss ridge. This indicates that overfitting on the loss landscape for the rehearsal memory is harmful for generalization, especially as rehearsal draws the w_2 solution towards the high-loss ridge.

We confirm these findings further for longer sequences of tasks and observe for Mini-Imagenet in Figure 3a and Figure 3b that overfitting occurs for the rehearsal memory as indicated by the zero loss for w_2 and w_5. Moreover, these results suggest the harmfulness for generalization as w_2 and w_5 locate on the high-loss ridge for the $T1$ test dataset.

Furthermore, we find the rehearsal memory's loss to provide a poor approximation of $T1$'s high-loss ridge of the full training data. In Figure 3a and b the loss on the linear paths from w_1 to w_2 and w_5 is similar near w_1 but increases significantly near w_2 and w_5 for the $T1$ training data, while the rehearsal memory overfits. This is problematic, because rehearsal can only observe the rehearsal memory's view of the loss landscape. The rehearsal dynamics draw the solution near a high-loss ridge for the rehearsal memory and this is where the high-loss ridge is being poorly approximated. Therefore, instead of ending up *near* the high-loss ridge in perspective of the rehearsal memory, the solution in reality resides *on* the high-loss ridge for the training data, which consequently also harms generalization.

Additionally, we analyze the gradient norms for the w_1, w_2 and w_5 minima for all three benchmarks in Figure 4. By quantifying the gradient norms for both the test dataset and the rehearsal memory for $T1$, we can confirm how throughout learning the sequence of tasks, the solution resides in

a low-loss region for the rehearsal memory. However, the gradient norms on the test dataset increase from w_1 to w_2 and significantly increase further as more tasks are learned (w_5). This indicates that as the number of learned tasks increases, more overfitting occurs on the rehearsal memory and generalization deteriorates.

4.3. High-loss ridge aversion

The previous two sections report strong empirical evidence on three benchmarks to support our two hypotheses. In this section, we conduct an additional experiment relying on these insights to avoid the overfitting towards the high-loss ridges of the rehearsal memory.

Considering two tasks, from Hypothesis 1 we can derive that the w_2 minimum resulting from rehearsal with $\mathcal{M}_1$ resides in the same low-loss region as w_1. It follows that the low-loss region of the rehearsal memory $\mathcal{M}_1$ at least has an overlap with the low-loss region of $\mathcal{D}_1$. Additionally, Hypothesis 2 shows w_2 overfits on the rehearsal memory $\mathcal{M}_1$, with our empirical evidence in Section 4.2 illustrating the solution being drawn close to a high-loss ridge of $\mathcal{M}_1$.

Therefore, we test a simple heuristic to withdraw from the high-loss ridges, by isolating training on the rehearsal memory for n updates after converging to the new task minimum w_2. As this gives incentive for an inward movement in the $\mathcal{M}_1$ low-loss region, we also expect inward movement for the overlapping $T1$ low-loss region. This would reduce overfitting on the rehearsal memory and hence improve generalization. Table 1 shows the results for different memory sizes $|\mathcal{M}|$ for sequences of 5 tasks. For MNIST with the smallest rehearsal memory of 100 exemplars, we observe a significant improvement in generalization of 9.4%, reducing to 1.6% margin for 10 times more exemplars. These observations are aligned with our expectations, as smaller rehearsal memories suffer more from overfitting and therefore gain more by withdrawing from the high-loss ridge. The margin of increased generalization declines for the more difficult CIFAR10 and Mini-Imagenet sequences. Nonetheless, both sequences attain a respective margin of 0.7% and 4.1% for a small memory of 100 samples. To clearly demonstrate the effect of high-loss ridge aversion, all benchmarks train for 10 epochs per task to encourage overfitting on the rehearsal memory. For the two more challenging setups Stable-SGD [26] was used to attain wider minima.

4.4. Conceptual analysis of rehearsal dynamics

The results in Section 4.1 show that during rehearsal, the learning trajectory is initially drawn to a low-loss region of the new task before deflecting towards a minimum near the rehearsal memory's high-loss ridge. This can be explained by the relation between the loss values and the gradient magnitude. Following [11], the gradient of the loss $\nabla \mathcal{L}$ can

Experiment	Rehearsal memory size $|\mathcal{M}|$		
	100	*500*	*1000*
MNIST			
ER	72.9 ± 1.8	87.1 ± 0.6	90.2 ± 0.5
ER-step	82.3 ± 2.2 (n=20)	90.0 ± 0.4 (n=10)	91.8 ± 0.3 (n=10)
CIFAR10			
ER (stable)	46.7 ± 0.8	65.1 ± 1.0	70.7 ± 1.7
ER-step (stable)	47.4 ± 1.9 (n=10)	66.2 ± 1.1 (n=10)	70.8 ± 1.0 (n=1)
Mini-Imagenet			
ER (stable)	25.7 ± 1.7	39.7 ± 1.0	46.6 ± 1.1
ER-step (stable)	29.8 ± 2.0 (n=20)	42.3 ± 1.1 (n=10)	47.4 ± 1.4 (n=10)

Table 1: Avg. acc. for D_{eval} after learning a sequence of 5 tasks for different rehearsal memory sizes $|\mathcal{M}|$. *ER-step*: After attaining each task minimum, n update steps are performed solely on rehearsal memory, compared to *ER* (ER-step with $n = 0$). Stable indicates using Stable-SGD [26]. We report standard deviation over 5 initializations.

be decomposed using the chain rule into the gradient of the model ∇f_w w.r.t. its parameters w and the derivative of the loss $\mathcal{L}'$ w.r.t. the output of f_w:

$$\nabla \mathcal{L}(f_w(\mathbf{x}), y) = \nabla f_w(\mathbf{x}) \, \mathcal{L}'(y, f_w(\mathbf{x})) . \qquad (4)$$

For the standard cross-entropy loss with softmax $p(\cdot)$, the derivatives w.r.t. the elements of the output $\hat{\mathbf{y}} = f_w(\mathbf{x})$ become $\mathcal{L}' = (p(\hat{y}_c) - 1)$ for $\hat{y}_c$ corresponding to the ground-truth and $\mathcal{L}' = p(\hat{y}_i)$ for the other output elements $\hat{y}_i, \forall i \neq c$. This means that the gradients will be smaller if the output is closer to the ground truth one-hot vector. As shown in Eq. 3, the final gradient is the average of the gradients on the individual tasks. At w_1, the exemplars of $T1$ will have close to zero loss, while the new samples of $T2$ typically start at a high loss. Therefore, the gradient direction and magnitude are initially dominated by $T2$. However, as observed in Figure 2, the loss for $T1$ increases until both task losses are balanced, and the trajectory continues on loss contours of similar magnitude.

4.5. Comparison to prior evidence

In this section, the empirical findings supporting our hypotheses are compared to empirical evidence found in prior work. Concerns about overfitting on the rehearsal memory have been expressed in prior studies [22]. This was later criticized in up-following work [7], with their hypothesis formulated as *"although direct training on the examples in $\mathcal{M}_1$ (in addition to those coming from [T2]) does indeed lead to strong memorization of $\mathcal{M}_1$ [...], such training is still overall beneficial in terms of generalization on the original task $T1$ because the joint learning with the examples of the current task $T2$ acts as a strong, albeit implicit and data-dependent, regularizer for $T1$."* Their empirical evidence is based on a minimal MNIST rotation experiment where two subsequent tasks are considered for different lev-

els of relatedness, i.e. Task 1 $(0°)$ is compared to Task 2 for different degrees of rotation $(20°, 40°, 60°)$. For all levels of relatedness, they find training $T2$ with a small $T1$ rehearsal memory $\mathcal{M}_1$ remains beneficial for the $T1$ performance.

With the empirical findings in our study, we are able to explain *why* learning with a small rehearsal memory remains beneficial for generalization. However, we also indicate how overfitting in rehearsal can become harmful for generalization, hence providing counter-evidence for their hypothesis. Furthermore, our findings are explanatory of how learning $T2$ in their experiments can exhibit regularizing effects when $T2$ is highly related to $T1$.

Firstly, we address why learning with a small rehearsal can still generalize. In Section 4.1 we show that the low-loss regions of $T1$ and its rehearsal memory $\mathcal{M}_1$ intersect after learning $T1$. Our findings suggest that learning $T2$ with rehearsal remains in the same low-loss region of $\mathcal{M}_1$. This indicates that although overfitting can happen, a generalizing solution can be found because of the overlap between both regions.

Secondly, we address how overfitting in rehearsal can become harmful for generalization. The experiments in Section 4.2 clearly exhibit overfitting on the rehearsal memory. This finding by itself does not explain how overfitting becomes harmful for generalization, as we just indicated there exists a generalizing solution in the overlapping low-loss region for $T1$ and $\mathcal{M}_1$. However, Section 4.2 finds the rehearsal solution ending up near a high-loss ridge, where the approximation of the loss-landscape of $\mathcal{M}_1$ for $\mathcal{D}_1$ deteriorates. Therefore, our results suggest it is the dynamic of both overfitting and the rehearsal solution ending up near a high-loss ridge harming generalization.

As a consequence, thirdly, for dissimilar tasks $T2$ has no regularization effects for $T1$, and on the contrary, deteriorates performance by pulling w_2 towards the $\mathcal{M}_1$ high-loss ridge.

5. Revisiting state-of-the-art

In this section, we will interpret the successes and results of state-of-the-art rehearsal methods in continual learning in the light of the insights in Section 4. Although progress has been made in the field, a universal and fundamental understanding of these findings is yet to emerge. In the following, we make an initial effort in the perspective of loss landscapes. Discussion of other state-of-the-art works can be found in Appendix.

GEM [22] constrains the gradient of new samples such that there is no loss increase on the samples in the memory. By strictly adhering to the gradient constraint, GEM ensures that the model will reside in the same low-loss basin of the first task. However, the constraints also may encourage overfitting on the rehearsal memory as even within the low-loss basin no slight increase on the loss is allowed.
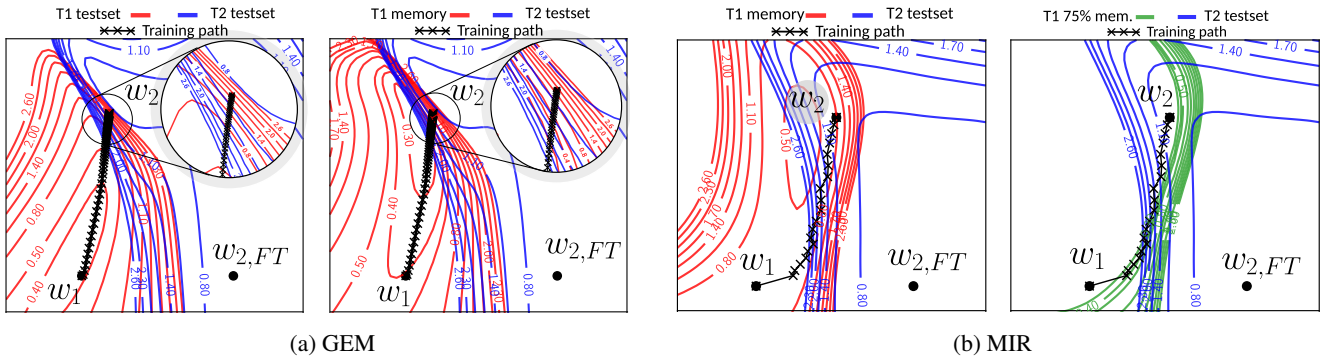
Figure 5: CIFAR10 learning trajectories of other rehearsal methods GEM [22] (a) and MIR [2] (b) in the parameter space.

As reported in [26], we observed GEM to diverge in the class-incremental settings (except for MNIST). We hypothesize this is due to the violation of the first order assumption, as GEM constraints the gradient direction but *not the magnitude*, which can increase significantly when learning new tasks. To facilitate convergence, we impose small fixed update steps by normalizing the gradients to a small norm (0.1) while training multiple epochs rather than one. Figure 5a confirms for CIFAR10 that initially the first order assumption holds as T1's memory loss declines. However, when converging near w_2, the loss rises so quickly that the first order approximation stops holding (even with the small norm), allowing a slight increase on the ridge. In contrast, rehearsal allows increases in the memory's loss as long as it comes with an equal or larger loss decrease in the new task (proof in Appendix). This makes wider exploration of the low-loss basin possible compared to a model trained with GEM. This might be an indication for the results in favor of rehearsal in [7, 2, 9, 3].

MIR [2] defines a retrieval policy for rehearsal by sampling exemplars from $\mathcal{M}$ with the highest increase in loss, measured by a tentative update of the model. Section 4.2 indicates that overfitting occurs on the rehearsal memory. The MIR retrieval policy increases the sampling probability based on the per-sample loss increase. The loss landscape in Figure 2 visualizes the high-loss ridge for the *average* over all rehearsal memory samples of the first task. MIR is more likely to select individual samples with the closest high-loss ridge when drawn to the $T2$ low-loss region. This possibly keeps the w_2 solution further from the average rehearsal memory high-loss ridge, although with the downside of majorly reselecting the same subset of $\mathcal{M}$, increasing the risk of further overfitting. Figure 5b left and right compare the loss of T1's entire memory compared to the samples constituting 75% of the retrieval frequency mass, i.e. the most sampled exemplars by MIR constituting about 10% of the rehearsal memory. As discussed, the loss on these samples (green on the right) increases faster (than red on the left) near the high-loss ridge, pushing the model away from the ridge of T1's testset (not shown due to limited space).

GDumb [27] is a controversial work questioning the progress made in continual learning by proposing a new baseline. Their Greedy Sampler and Dumb Learner (GDumb) greedily stores samples balanced over the observed classes and at inference learns a new model from scratch with the rehearsal memory. This simple baseline outperforms a vast range of continual learning methods. For the rehearsal methods, our findings show that the rehearsal minimum resides in the same low-loss region as $T1$. As more tasks are learned, the overlap of the low-loss regions can only decline for the union of all tasks. Therefore, it becomes gradually harder to find a solution for all tasks in the sequence. In contrast, GDumb overcomes this limitation by learning a model from scratch, enabling to find a joint minimum for all rehearsal memories. However, training is repeated for each task from scratch and does not build on the knowledge base acquired in the model learned on previous tasks. Hence GDumb is limited to indicating the rehearsal memory's offline potential, but doesn't allow measuring transfer learning dynamics, which are key for continual learning.

6. Conclusion

This work investigated the open fundamental questions of *why rehearsal works even though overfitting on the rehearsal memory occurs*, and *how this overfitting influences generalization*. To answer these questions, we formalized two hypotheses in the introduction, which were confirmed by comprehensive empirical evidence on three common benchmarks. Our observations suggest that rehearsal prevents a sequential model from leaving the first found low-loss region, and that this model is susceptible to overfitting towards the edge of the rehearsal memory's low-loss region, harming generalization. These findings enhance our understanding in both rehearsal and continual learning dynamics. We hope to encourage further research in this direction.

Acknowledgements

References

[1] Rahaf Aljundi, Francesca Babiloni, Mohamed Elhoseiny, Marcus Rohrbach, and Tinne Tuytelaars. Memory aware synapses: Learning what (not) to forget. In *ECCV*, pages 139–154, 2018. 2

[2] Rahaf Aljundi, Lucas Caccia, Eugene Belilovsky, Massimo Caccia, Min Lin, Laurent Charlin, and Tinne Tuytelaars. Online continual learning with maximally interfered retrieval. *Proceedings NeurIPS 2019*, 32, 2019. 2, 3, 8

[3] Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *NeurIPS*, pages 11816–11825, 2019. 3, 8

[4] Léon Bottou. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*, pages 177–186. Springer, 2010. 3

[5] 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. 3

[6] Arslan Chaudhry, Marc'Aurelio Ranzato, Marcus Rohrbach, and Mohamed Elhoseiny. Efficient lifelong learning with a-gem. In *International Conference on Learning Representations*, 2018. 3

[7] Arslan Chaudhry, Marcus Rohrbach, Mohamed Elhoseiny, Thalaiyasingam Ajanthan, Puneet K Dokania, Philip HS Torr, and M Ranzato. Continual learning with tiny episodic memories. 2019. 1, 2, 3, 7, 8

[8] Aristotelis Chrysakis and Marie-Francine Moens. Online continual learning from imbalanced data. In *International Conference on Machine Learning*, pages 1952–1961. PMLR, 2020. 3

[9] Matthias De Lange and Tinne Tuytelaars. Continual prototype evolution: Learning online from non-stationary data streams. In *Proceedings of the IEEE International Conference on Computer Vision*, 2021. 1, 3, 4, 8

[10] M. Delange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pages 1–1, 2021. 1, 2, 4

[11] Mehrdad Farajtabar, Navid Azizan, Alex Mott, and Ang Li. Orthogonal gradient descent for continual learning. In *International Conference on Artificial Intelligence and Statistics*, pages 3762–3773. PMLR, 2020. 6

[12] Robert M French. Catastrophic forgetting in connectionist networks. *Trends in cognitive sciences*, 3(4):128–135, 1999. 1

[13] 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

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

[15] Pavel Izmailov, Dmitrii Podoprikhin, Timur Garipov, Dmitry Vetrov, and Andrew Gordon Wilson. Averaging weights leads to wider optima and better generalization. In *34th Conference on Uncertainty in Artificial Intelligence 2018, UAI 2018*, pages 876–885. Association For Uncertainty in Artificial Intelligence (AUAI), 2018. 4

[16] Nitish Shirish Keskar, Dheevatsa Mudigere, Jorge Nocedal, Mikhail Smelyanskiy, and Ping Tak Peter Tang. On large-batch training for deep learning: Generalization gap and sharp minima. *ICLR*, 2017. 4

[17] 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. *PNAS*, page 201611835, 2017. 2

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

[19] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2324, 1998. 4

[20] Zhizhong Li and Derek Hoiem. Learning without forgetting. In *ECCV*, pages 614–629. Springer, 2016. 2

[21] Vincenzo Lomonaco, Lorenzo Pellegrini, Andrea Cossu, Antonio Carta, Gabriele Graffieti, Tyler L. Hayes, Matthias De Lange, Marc Masana, Jary Pomponi, Gido van de Ven, Martin Mundt, Qi She, Keiland Cooper, Jeremy Forest, Eden Belouadah, Simone Calderara, German I. Parisi, Fabio Cuzzolin, Andreas Tolias, Simone Scardapane, Luca Antiga, Subutai Amhad, Adrian Popescu, Christopher Kanan, Joost van de Weijer, Tinne Tuytelaars, Davide Bacciu, and Davide Maltoni. Avalanche: an end-to-end library for continual learning, 2021. 4

[22] David Lopez-Paz and Marc'Aurelio Ranzato. Gradient episodic memory for continual learning. In *Advances in neural information processing systems*, pages 6467–6476, 2017. 1, 2, 3, 4, 7, 8

[23] Arun Mallya, Dillon Davis, and Svetlana Lazebnik. Piggyback: Adapting a single network to multiple tasks by learning to mask weights. In *ECCV*, pages 67–82, 2018. 2

[24] Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *CVPR*, pages 7765–7773, 2018. 2

[25] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Dilan Gorur, Razvan Pascanu, and Hassan Ghasemzadeh. Linear mode connectivity in multitask and continual learning. In *International Conference on Learning Representations*, 2021. 2

[26] Seyed Iman Mirzadeh, Mehrdad Farajtabar, Razvan Pascanu, and Hassan Ghasemzadeh. Understanding the role of training regimes in continual learning. *arXiv preprint arXiv:2006.06958*, 2020. 6, 7, 8

[27] Ameya Prabhu, Philip HS Torr, and Puneet K Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *European Conference on Computer Vision*, pages 524–540. Springer, 2020. 2, 3, 8

[28] Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *CVPR*, pages 2001–2010, 2017. 1, 2, 3

[29] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, San-jeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, et al. Imagenet large scale visual recognition challenge. *IJCV*, 115(3):211–252, 2015. 1

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

[31] Ari Seff, Alex Beatson, Daniel Suo, and Han Liu. Contin-ual learning in generative adversarial nets. *arXiv preprint arXiv:1705.08395*, 2017. 2

[32] Joan Serra, Didac Suris, Marius Miron, and Alexandros Karatzoglou. Overcoming catastrophic forgetting with hard attention to the task. In *International Conference on Machine Learning*, pages 4548–4557. PMLR, 2018. 2

[33] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *NeurIPS*, pages 2994–3003, 2017. 2

[34] Hanul Shin, Jung Kwon Lee, Jaehong Kim, and Jiwon Kim. Continual learning with deep generative replay. In *Proceed-ings of the 31st International Conference on Neural Infor-mation Processing Systems*, pages 2994–3003, 2017. 2

[35] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrit-twieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484–489, 2016. 1

[36] David Silver, Thomas Hubert, Julian Schrittwieser, Ioan-nis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and go through self-play. *Science*, 362(6419):1140–1144, 2018. 1

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

[38] Oriol Vinyals, Charles Blundell, Timothy Lillicrap, Koray Kavukcuoglu, and Daan Wierstra. Matching networks for one shot learning. In *Proceedings of the 30th Interna-tional Conference on Neural Information Processing Sys-tems*, pages 3637–3645, 2016. 4

[39] Friedemann Zenke, Ben Poole, and Surya Ganguli. Contin-ual learning through synaptic intelligence. In *ICML*, pages 3987–3995. JMLR. org, 2017. 2

[40] Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learn-ing requires rethinking generalization. *arXiv preprint arXiv:1611.03530*, 2016. 2