CONTINUALLY LEARNING REPRESENTATIONS AT SCALE

Alexandre Galashov, Jovana Mitrovic, Dhruva Tirumala, Yee Whye Teh
DeepMind
{agalashov,mitrovic,dhruvat,ywteh}@deepmind.com

Timothy Nguyen, Arslan Chaudhry, Razvan Pascanu
DeepMind
{timothycnguyen,arslanch,razp}@deepmind.com

ABSTRACT

Many widely used continual learning benchmarks follow a protocol that starts from an untrained, randomly initialized model that needs to sequentially learn a number of incoming tasks. To maximize interpretability of the results and to keep experiment length under control, often these tasks are formed from well-known medium to large size datasets such as CIFAR or ImageNet. Recently, however, large-scale pretrained representations, also referred to as foundation models, have achieved significant success across a wide range of traditional vision and language problems. Furthermore, the availability of these pretrained models and their use as starting point for training can be seen as a paradigm shift from the classical end-to-end learning. This raises the question: *How does this paradigm shift influence continual learning research?* We attempt an answer, by firstly showing that many existing benchmarks are ill-equipped in this setting. The use of foundation model leads to state-of-art results on several existing and commonly used image classification continual learning benchmarks, from split CIFAR-100 to split ImageNet. Additionally, there is at best a small gap between keeping the representations frozen versus tuning them. While this is indicative of the overlap between pretraining distribution and the benchmark distribution, it also shows that these benchmarks can not be used to explore how to continually learn the underlying representations. Secondly, we examine what differentiates continually learning from scratch versus when relying on pretrained models, where the representation is learned under a different objective. We highlight that this brings about new challenges and research questions that cannot be studied in the sanitised scenario of learning from scratch explored so far.

1 INTRODUCTION

Deep Learning has made significant progress in the last decades, leading to impressive results across multiple topics, from reinforcement learning to language, vision and structured problems like protein folding. Multiple factors played a crucial role in these success stories, among them being the emergence of powerful neural network architectures and the reliance on gradient based learning algorithms. The latter, however, requires IID observations sampled from a stationary distribution. While the IID assumption had proven powerful, it has been argued that it rarely reflects the situation faced in practice. The distribution a deployed system needs to interact with changes continuously. For example a language model trained pre-2020 might not be able to generate language about *COVID* meaningfully. One can deal with distributional drift by retraining the model regularly, which often means retraining the model from scratch, without making use of the previously trained variants.

Continual Learning has focused on how to efficiently train neural networks in non-stationary settings (e.g. Hadsell et al., 2020; Parisi et al., 2019). Answers to this question can lead to systems that continuously adapt, even after deployment, or at least provide an efficient mechanism for adapting the systems offline. The formulation of the continual learning problem often assumes a piece-wise stationary distribution, where each stationary piece can be identified as a task. A solution to this problem is meant to learn these tasks sequentially, achieving a few goals, ranging from preserving performance on previous tasks to maximizing forward and backward transfer and controlling the amount of memory and compute used (Hadsell et al., 2020; Schwarz et al., 2018; Rusu et al., 2016; Lopez-Paz & Ranzato, 2017; Nguyen et al., 2018). Additional questions, like predicting boundaries between tasks or dealing with scenarios where we can not assume piece-wise stationarity have also been studied extensively within the community (Zeno et al., 2018; Aljundi et al., 2019; 2018).

Most of the research done in continual learning follows the successful recipe applied generally in deep learning of relying on established benchmarks to track progress. This allows to fix, sometimes implicitly, some of the moving pieces of the problem definition, for example the type of distributional shift, and to fairly compare methods. In order to improve interpretability of results, and for simplicity and maintaining fast turn around of experiments, most continual learning benchmarks tend to be built from well established medium to large sized IID counterparts. For example, some of the most widely used benchmarks are permuted-MNIST (Goodfellow et al., 2013) built from the well known MNIST dataset, Split-CIFAR-100 (Mallya & Lazebnik, 2018) made out of the CIFAR-100 dataset (Krizhevsky, 2009), split ImageNet (Mirzadeh et al., 2022), etc. This choice led to a fast growth of the field, with significant progress made on specific aspects of the problem. (Parisi et al., 2019; Hadsell et al., 2020) provide surveys of the field, where a few open foundational questions about the continual learning problem formulation are also discussed.

On the other hand, in the last few of years we have witnessed a proliferation of systems relying on adapting large-scale *pretrained* representations, sometimes referred to as *foundational models* (e.g. Devlin et al., 2018; Brown et al., 2020b; Bommasani et al., 2021; Alayrac et al., 2022), which have excellent generalization abilities. For example, BERT (Devlin et al., 2018), DALLE (Ramesh et al., 2021), GPT-3 (Brown et al., 2020a), CLIP (Radford et al., 2021b), ALIGN (Jia et al., 2021), SimVLM (Wang et al., 2021) or Flamingo (Alayrac et al., 2022) significantly increased the state-of-art performances across many difficult benchmarks in computer vision and language modelling. We regard the differentiation between *pretrained models* and *foundation models* to be in the scale of the pretraining stage, inline with existing literature. This differentiation can be problematic, as scale can be relative concept. What is considered large scale now might not be so in the near future. However, the the concept of foundation models is meant to highlight the robustness of these pretrained models and representations, that had led to a change in training paradigm within the community, moving away from end-to-end learning. In the vision context, there are at least two important contributing factors that are needed in explaining the emergence of these systems: a) the scale and breadth of the training data (billions of datapoints gathered from the internet spanning many different domains) and b) a paradigm shift in representation learning towards self-supervision and contrastive learning. Specifically contrastive losses seem to allow for learning robust, transferable and task-agnostic representations which can outperform their supervised counterparts (Mitrovic et al., 2020; Radford et al., 2021a).

This change in protocol has significant implications for the field, where it has become more ubiquitous to rely on composing pretrained artifacts in order to build larger systems. These artifacts, the foundational models, are seen as general purpose and stable representations able to capture the structure of a certain type of data, e.g. generic vision module or a generic language module. The focus of our work is to explore what this paradigm shift might imply for the continual learning problem. In particular, given the reliance of continual learning benchmarks on sequentializing older and well established datasets, how should we study the use of foundational models in continual learning? In Section 2, we present findings showing that pretrained models with simple finetuning strategies achieve SOTA performance on existing continual learning benchmarks. In light of these results, we propose to focus on continual learning at pretraining level (continual representation learning) rather than seeing continual learning as a problem when adapting to downstream tasks. We argue that large scale deployed systems need to address this problem in order to reduce the computational cost of keeping the system up to date while maintaining robust and generally useful representation. We further argue that we need new benchmarks focusing on these aspects that can highlight in a tractable way the different issues and research questions imposed by this setting. We study these questions in Section 3.

2 Foundational Models on typical Continual Learning benchmark

In this section, we demonstrate the performance of some foundational models on a number of commonly used continual learning benchmarks. For this purpose, we use very wide spread benchmarks, Split-CIFAR-100 (Mallya & Lazebnik, 2018), Split-TinyImageNet (Delange et al., 2021) and Split-ImageNet1k (Mirzadeh et al., 2022). We examine the task incremental setting, where the tasks are coming with known boundaries (or task identifiers), as well as the class incremental one, where data for new classes are introduced incrementally to the system.

Throughout the section, we assume a pretrained model backbone $g(x;\theta)$ with parameters θ. The output is given as:

$$f(x;\theta,\phi_i) = g(x;\theta)^T\phi_i, \tag{1}$$

where ϕ_i are (task-specific) output layer parameters. In task-incremental setting, index i is a task identifier. In the class-incremental setting, the prediction does not depend on i, i.e., $\phi_i = \phi$. The continual learning problem consists in sequentially learning both θ and ϕ_i, which we call **Full finetuning**. In case where θ is frozen, it reduces to a linear problem, which is called **Linear evaluation** (Zhai et al., 2019).

We employ standard continual learning metrics to measure performance. We denote by a_k^t - the accuracy on the task k at time t, $k = 1, \dots, K$, where $K > 0$ is the total number of tasks. We denote by T_k the end of the task k and the

Table 1: **Linear evaluation**. Task incremental learning on Split CIFAR-100 with 10 and 20 tasks. All pretrained models use the ImageNet 1k dataset for pretraining. We show average accuracy from Eq. 2. In **bold** we denote the method with highest performance. See Appendix for more details.

Method	Split CIFAR-100 10 tasks	Split CIFAR-100 20 tasks
Best Reported	63.3% (Saha et al., 2021)	73.70% (Mirzadeh et al., 2022)
No pretraining	31.33 %	42.25 %
Multi-task	68.8%	79.58%
Foundational models:		
Supervised	69.34%	78.1 %
BYOL	73.49%	81.96 %
ReLICv2	**79.48%**	86.16%
SemPPL	79.04 %	**86.56**%

Method	Tiny ImageNet 10 tasks	Split-ImageNet1k, 10 tasks
Best Reported	48.17% (Delange et al., 2021)	66.10 % (Mirzadeh et al., 2022)
No pretraining	25.38%	9.1 %
Multi-task	61.43 %	—
foundational Models:		
Supervised	74.68 %	**85.8** %
BYOL	76.75 %	76.17 %
ReLICv2	81.85 %	75.6 %
SemPPL	**82.42** %	78.87 %

Table 2: **Linear evaluation**. Task incremental learning on Split Tiny ImageNet 10 tasks and Split-ImageNet1k 60 tasks. All pretrained models use the ImageNet 1k dataset for pretraining. We show average accuracy from Eq. 2. In **bold** we denote the method with highest performance. See Appendix for more details.

total training time by T. Average accuracy, or the average accuracy at the end of the experiment is given by $\mathcal{A}_{acc}$. Learning accuracy, or the average accuracy at the end of each task is given by $\mathcal{L}_{acc}$. This quantity denotes how well each task is learned. Finally, we denote by forgetting $\mathcal{F}$, the difference between final accuracy at the end of the task and the accuracy at the end of the training:

$$\mathcal{A}_{acc} = \frac{1}{K}\sum_{k=1}^{K} a_k^T, \qquad \mathcal{L}_{acc} = \frac{1}{K}\sum_{k=1}^{K} a_k^{T_k}, \qquad \mathcal{F} = \frac{1}{K}\sum_{k=1}^{K}(a_k^{T_k} - a_k^T). \tag{2}$$

For each benchmark, we take foundational models pretrained on ImageNet-1k dataset (Russakovsky et al., 2015) which gives initialisation for parameters θ in Eq. 1. In particular, we compare supervised pretraining as well as a number of self-supervised methdos, such as BYOL (Grill et al., 2020), ReLICv2 (Tomasev et al., 2022) and SemPPL (Bošnjak et al., 2023). For each benchmark, we report the best reported SOTA from external papers.

2.1 Task Incremental benchmarks

In the task incremental learning setting, each task $\mathcal{T}_i$ corresponds to a dataset $\mathcal{D}_i = \{(x_k^i, y_k^i)\}_{k=1}^{M_i}$, where $M_i > 0$ is the number of points in the dataset, and i is a known task identifier. The objective of continual learning in this setting is to learn both task specific head parameters ϕ_i as well as shared backbone parameters θ, see equation 1. In our case, we use a pretrained model for the backbone parameters θ.

As concrete benchmarks, we consider Split-CIFAR-100 with 10 tasks split (see Saha et al. (2021) for SOTA) as well as with 20 tasks split (see Mirzadeh et al. (2022) for SOTA). Moreover, we consider split TinyImageNet benchmarks with 10 tasks split (see Delange et al. (2021) for SOTA) and split-ImageNet1k benchmark with 60 tasks (see Mirzadeh et al. (2022) for SOTA).

Linear evaluation. We first study the quality of representation of foundational models on these benchmarks. Concretely, we freeze the foundational model backbone parameters θ and learn the task specific head parameters ϕ_i which is a single linear layer. While this does not constitute a continual learning problem (since the heads are learned independently and the backbone is frozen), this will tell us how generic the representation learned by the foundational model is. This could be a valid approach in practice when we need to adapt a foundational model to a specific task.

Tables 1 and 2 demonstrate that pretrained models can outperform most continual learning methods across benchmark when simply freezing the representation and learning a specific output layer for each task. Moreover, we generally see that models pretrained via SSL methods perform better than supervised, except for *Split-ImageNet1k*. We hypothesise that this is due to the fact that Split-ImageNet1k benchmark is much closer in distribution space than others, which favors supervised pretraining that typically leads to less general representations. While it may not be surprising that using ImageNet-1k pre-trained models on standard continual learning benchmarks leads to strong performance, it does reflect the problem of using these benchmarks for continual learning research. This becomes even more problematic due to the fact that ImgeNet-1k pre-trained models are widespread and readily available to practitioners. However, if the distributions are sufficiently different, one could expect to require methods that will continually update the backbone parameters θ as well, in addition to the head (see section 2.3). Additionally, we would argue that instead of simplifying or changing the pretraining protocol, it would be more lucrative to focus on building richer benchmarks where linear evaluation fails. This is in particular due to how computationally expensive it could be to build these pretrained artifacts. And using small amounts of data for pretraining can lead to qualitatively different kind of dynamics, that might not be representative to what we tend to observe from large scale models.

Full finetuning. In order to get a full picture of existing benchmarks we also study the effect of full finetuning. For all the experiments, we used a backbone pretrained on ImageNet-1k with ReLICv2 as it seemed to provide good results with Linear Evaluation. Moreover, we explore using different learning rates for the head compared to the backbone, highlighting the different time scales imposed by not doing end-to-end learning. Concretely, we rescaled the gradients of the backbone by $\beta \in (0, 1]$, with $\beta = 1$ corresponding to ordinary finetuning. For simplicity of presentation, we finetuned all the parameters and used linear head. We do not expect conclusions in this section change if we study different subsets of finetuned parameters like in (Shysheya et al., 2022) and different heads like in (Panos et al., 2023). For more details, see Appendix.

As was observed in the literature (Fini et al., 2022), continual learning with Self-Supervised Learning (SSL) losses could outperform supervised variants. Motivated by this observation, we considered 3 options for finetuning the backbone of the foundational model. **Supervised** finetuning corresponds to using supervised objective for the backbone. **SSL** finetuning corresponds to using loss from ReLICv2 method pretrained on ImageNet-1k data. We use this SSL method because the foundational model was pretrained with the same objective. Finally, we also considered **Supervised + SSL** finetuning, where we simply sum two losses, Supervised and SSL. The head parameters in all cases were finetuned using supervised loss.

The results for full finetuning are given in Table 3. We observe that in some cases it is beneficial to finetune the representation in addition to the head parameters (note average accuracies relative to linear evaluation). Moreover, for $\beta = 1.0$ we find that SSL outperforms Supervised learning which is consistent with previous findings Fini et al. (2022). Interestingly, supervised finetuning provides the best performance at $\beta = 0.1$. That means that it is beneficial to treat task-specific head learning vs representation learning as two separate processes having their own learning rate scaling. Moreover, it suggests that supervised finetuning works well if we want to adapt the representation to a down-stream tasks. Intuitively, it is what we should expect because supervised finetuning provides the strongest signal related to a specific downstream task. Finally, for $\beta = 1.0$, we found that using both Supervised and SSL finetuning provides the best performance. In cases when we have additional task-specific unlabelled data, we expect this method to provide the best option.

Finetuning Method	Split CIFAR-100 10 tasks			Split CIFAR-100 20 tasks		
	Average Accuracy	Learning Accuracy	Forgetting	Average Accuracy	Learning Accuracy	Forgetting
Linear evaluation	79.48 %	79.48 %	0 %	86.16%	86.16%	0 %
Results with $\beta = 1.0$						
Supervised:	75.57 %	87.93 %	13.73 %	77.97 %	92.20 %	14.98 %
Self-supervised (SSL):	76.78 %	79.71 %	3.26 %	83.23 %	87.93 %	4.95 %
Supervised + SSL:	**81.78** %	87.44 %	6.29 %	**87.00** %	92.26 %	5.54 %
Results with $\beta = 0.1$						
Supervised:	**83.31** %	85.97 %	2.96 %	**86.74** %	92.78 %	6.36 %
Self-supervised (SSL):	75.55 %	78.25 %	3.00 %	83.02 %	86.15 %	3.29 %
Supervised + SSL:	79.50 %	85.63 %	6.81 %	86.69 %	92.43 %	6.04 %

Table 3: **Fine-tuning** backbone on task incremental learning Split CIFAR-100 with 10 and 20 tasks using ReLICv2 pretrained model. In **bold** font we denote the best finetuning method in terms of average accuracy. In red font we denote the method with highest forgetting. See Appendix for more details.

2.2 Class Incremental Learning

In this setting, we assume a sequence of tasks $\mathcal{T}_1, \ldots, \mathcal{T}_n$ such that each subsequent task contains new unseen classes and there are no explicit task boundaries. The setting was studied in multiple works such as (Rebuffi et al., 2017; Wu et al., 2019; Prabhu et al., 2020). This becomes a challenging problem even for the linear classifier, since the softmax layer may push down the probabilities for the classes which are no longer observed in the batches. We consider class-incremental Split-CIFAR100 as in (Wu et al., 2019) with 10 and 20 tasks, where each task has disjoint classes.

Similar to the previous section, we use pretrained foundational models and learn a linear layer on top of them. Since even learning linear layer on top of a frozen backbone may exhibit forgetting, we employ a simple replay strategy of sampling a small batch of data from each of the previous tasks, for each batch update. The results are given in Table 4. We observe that without replay we already achieve strong results, which are underperforming compared to the best reported results (Thengane et al., 2022; Yan et al., 2021). With a simple replay strategy, we immediately start to outperform the best reported method and as well as the reported upper bound (taken from (Wu et al., 2019)) , which corresponds to learning on the union of all the tasks together. The fact that we outperform the upper bound is not surprising due to the use of pretrained models. We see again that using pretrained models provides a large benefit.

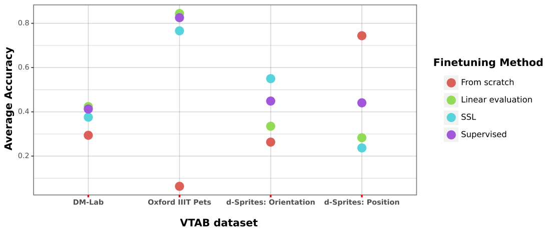
Method	Split CIFAR-100 10 tasks			Split CIFAR-100 20 tasks		
-	Average Accuracy	Learning Accuracy	Forgetting	Average Accuracy	Learning Accuracy	Forgetting
Best Reported	66.72% (Thengane et al., 2022)			69% (Yan et al., 2021)		
Upper Bound (Wu et al., 2019)	69.5%			69.5%		
Foundational Models:						
BYOL	39.85 %	62.17 %	24.8%	45.57 %	66.45 %	21.96%
Supervised	43.04%	53.69%	11.9%	45.7 %	54.83 %	9.6%
ReLICv2	46.88%	68.47%	24.04%	47.75%	68.43%	21.74%
SemPPL	**51.02** %	**66.74**%	17.45 %	**50.9** %	**64.38** %	14.17
foundational Models + replay:						
BYOL	62.69 %	66.49 %	1.2 %	69.29 %	78.27 %	9.42 %
Supervised	59.54%	59.76 %	0.2 %	68.73 %	72.96 %	4.45 %
ReLICv2	69.56%	73.26 %	3.92 %	74.84%	82.08 %	7.63%
SemPPL	**69.56** %	**73.28** %	4.22 %	**75.19** %	**83.13** %	8.44

Table 4: **Class incremental Learning.**. All foundational models use the ImageNet 1k dataset for pretraining. In **bold** we denote the best performing method. More details are given in Appendix.

2.3 Out-of-distribution finetuning

As argued above, the linear evaluation and finetuning could fail if the down-stream tasks are sufficiently out-of-distribution compared to the pretrained dataset. To validate it, we finetuned ReLICv2 model pretrained on ImageNet-1k on 4 VTAB (Zhai et al., 2019) tasks: Oxford IIIT Pets, DM-Lab, d-sprites position, d-sprites orientation. We used linear evaluation as well as supervised and SSL full finetuning strategies and compared it to learning from scratch on these tasks. The results are shown in Figure 1. While it seems that using pretrained representation is usually beneficial, sometimes (task **d-sprites position**) learning from scratch performs better. This suggests that pretrained representation might not always be useful for every down-stream tasks. Continually updating these representations to remain useful is important as the distribution changes, and, as in the case of **d-sprites position**, dealing with interference from misaligned representations can be necessary.

Figure 1: **Out-of-distribution finetuning experiments on VTAB**. On the Y-axis we report the average accuracy after finetuning ReLiCv2 pretrained model. X-axis indicates the task. Color indicates the fientuning method. For more details see Appendix.



2.4 Conclusion

We showed that using pre-trained models with simple finetuning strategies outperforms state-of-the-art continual learning methods on most of the benchmarks. Moreover, we found that overall the best finetuning method remains supervised. This raises a question about the usefulness of these benchmarks in continual learning research. Our findings and conclusions here resonate with recent work (Janson et al., 2022). However these results only study continual learning at the finetuning stage given an already pre-trained model. In the next section we advocate for studying the continual learning at the pre-training stage instead.

We limited the investigation in this Section to task-incremental and class-incremental settings. Other settings such as domain-incremental (Van de Ven & Tolias, 2019) and Online Continual Learning (OCL) (Mai et al., 2022; Cai et al., 2021) could also have been the object of our study in this section. However, we do not believe that these settings, in the context of Section 2, would offer additional insights and due to time constraints have decided not to pursue it.

3 Benchmarks for continually learning representations

When dealing with pretraining models, multiple questions are relevant, as for example the ones studied in concurrent works (Ostapenko et al., 2022; Cossu et al., 2022). In this work we focus on how to maintain representations relevant by continually learning them. In particular, we assume a sequence of datasets $\mathcal{D}_1, \dots, \mathcal{D}_N$ which are given to the model sequentially. Additionally, we assume a sequence of sets of down-stream tasks $\mathcal{P}_1, \dots, \mathcal{P}_N$, where each set is composed of an arbitrary number of tasks, $\mathcal{P}_i = \{\mathcal{T}_1, \dots, \mathcal{T}_{k_i}\}$. Our ultimate goal is to learn sequentially on datasets $\mathcal{D}_1, \dots, \mathcal{D}_N$ such that underlying representation after pretraining on $\mathcal{D}_i$ is useful for any down-stream task from $\bigcup\limits_{j \leq i} P_j$.
To evaluate the quality of the representation, we finetune independently on any of the relevant downstream tasks, though the downstream tasks do not directly alter the representation of the backbone for other or future downstream tasks. This separation allows for the downstream tasks to act as probes without interfering with the continual learning of the representation. This allows us to study the representation learning part of the problem, also leading to new ways of thinking about typical continual learning phenomena such as catastrophic forgetting.

3.1 First step towards learning sequentially at pretraining

The main difficulty in building such a benchmark consists in the choice of datasets $\mathcal{D}_i$. They need to be sufficiently large-scale while inducing meaningful distributional drifts that can affect both past and future downstream tasks while maintaining the computational tractability of the benchmark. We propose a simple variant of such a benchmark.

We operate on 2 datasets: $\mathcal{D}_A$ and $\mathcal{D}_B$, seen sequentially by the learning algorithm. On top of that, we assume that we have a single set of downstream tasks $\{\mathcal{T}_1, \dots, \mathcal{T}_K\}$, such that some of these tasks are associated with $\mathcal{D}_A$ and $\mathcal{D}_B$. For example, $\mathcal{D}_A$ contains images of animals and $\mathcal{D}_B$ contains artifacts, according to ImageNet classification of labels. Then downstream tasks that focus on classification of natural images, would be more in-domain for $\mathcal{D}_A$, while downstream tasks that focus on structured images are more aligned to $\mathcal{D}_B$. This allows us to test whether we manage to preserve what is useful in the representation for both scenarios in the finetuning process of the representation.

Formally, we assume to have pretrained representation F_A on the dataset $\mathcal{D}_A$ and the aim is to finetune it dataset $\mathcal{D}_B$, such that it preserves performance on downstream tasks associated with $\mathcal{D}_A$ and improves on downstream tasks associated with $\mathcal{D}_B$. We study three strategies to achieve this goal. First, we ignore the pretrained representation F_A and re-train **from scratch** on the union of $\mathcal{D}_A \cup \mathcal{D}_B$ (we will refer to this approach as **from scratch**). This is commonly used strategy in practice due to its simplicity and good results. However, this strategy is wasteful since it essentially ignores the compute spent on pre-training of F_A. Second, we study a method which starts from the representation initialized from F_A which is then finetuned on the union of $\mathcal{D}_A \cup \mathcal{D}_B$. In the literature (Ash & Adams, 2020), this strategy is called **warmstarting**. Unlike learning from scratch, warmstarting allows to leverage pre-trained model to speed up finetuning on new data. It was shown (Ash & Adams, 2020) that warmstarting leads to sub-optimal performance due to loss of plasticity. We, however, do not observe this phenomenon in our experiments and find that warmstarting generally leads to better than from scratch performance. Finally, we study **continual finetuning** strategy, where the representation is initialized from F_A and is then finetuned on $\mathcal{D}_B$. For all the finetuning strategies, we report the results as function of compute spent at funetuning in order to take into account the computational constraints. In addition to that, we study impact of learning objectives on finetuning performance of these different strategies.

3.2 Continual pretraining of ImageNet

In this section, we present an empirical study of continual pretraining on ImageNet. We propose the following two scenarios. In the first scenario, **ImageNet: 100 to 900**, the dataset $\mathcal{D}_A$ consists of the first 100 classes of ImageNet and the dataset $\mathcal{D}_B$ of 900 subsequent classes. In the second scenario **ImageNet: Animals to Artifacts**, the dataset $\mathcal{D}_A$ corresponds to the first 398 classes which could be categorized as **animals**, whereas the dataset $\mathcal{D}_B$ corresponds to the subsequent classes which can be categorized as **artifacts**.

In both scenarios, we pretrain model F_A on the dataset $\mathcal{D}_A$ for $N = 90$ epochs using supervised objective or $N = 1000$ epochs using SSL objective. Then, we study 3 different strategies of updating the pretrained representation on the new dataset $\mathcal{D}_B$ mentioned in Section 3.1. When finetuning representation, we use the same learning objective as the one used during pre-training of F_A. We study the performance of finetuned representation through two angles: *in-distribution* and *out-of-distribution*. *In-distribution* performance is captured by measuring ImageNet test set accuracy. This metric, however, has limited interpretability since it does not reflect how useful the representation is for down-stream tasks. *Out-of-distribution* performance, on the other hand, is studied by taking sequentially finetuned representation on dataset $\mathcal{D}_B$ and performing supervised finetuning on down-stream tasks represented by VTAB (Zhai et al., 2019). This performance is the most informative since it reflects how foundational models are used down-stream.

The results for *in-distribution* performance are given in Figure 2. The first immediate observation is that warmstarting works consistently better than learning from scratch. This finding is contrary to what was observed before (Ash & Adams, 2020). The gap between training from scratch and warm starting closes as finetuning compute budget increases, which is intuitively expected. Next, we observe that continual finetuning is always worse than learning from scratch or warmstarting. This is not surprising, because the ImageNet test set includes data relevant for both, $\mathcal{D}_A$ and $\mathcal{D}_B$, and continual finetuning exhibits forgetting of the dataset $\mathcal{D}_A$.

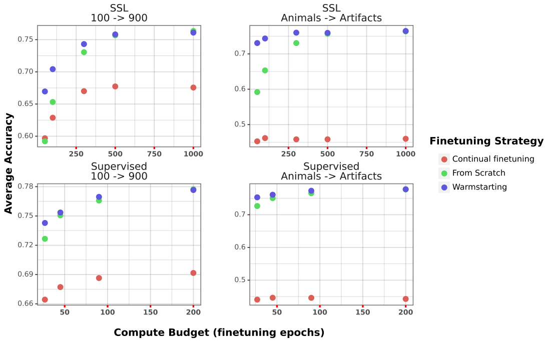


Figure 2: **Continual pretraining of ImageNet**. The rows correspond to the pretraining/finetuning objective (first row is SSL pretraining, second row is supervised). The columns correspond to the scenarios (left column is 100 to 1000, right column is animals to artifacts). X-axis corresponds to the compute budget at finetuning time. Y-axis corresponds to the average accuracy on ImageNet test set. For more experimental details, see Appendix.

We now test the different approaches to continually learn representations (from scratch, warmstarting, continual finetuning) by finetuning them VTAB and reporting $\frac{A_m - A_s}{A_s}$, where A_m is the accuracy of the method m, and A_s is the accuracy of learning on each of the VTAB task from scratch. When this quantity is positive, the model performs better than learning from scratch. The goal is to maximize this value.

We report performance on different subsets of VTAB benchmark: **All**, **Natural**, **Specialized**, **Structured**. We consider a variant where each task has only 1000 data points in the training set. These subsets can be found in the original VTAB paper (Zhai et al., 2019). The results for **Imagenet: 100 to 900** experiment are given in Figure 3, and the results for **ImageNet: Animals to Artifacts** experiment are given in Figure 4. We observe that on **specialized** and **structured** subsets, all the methods which used pretraining in any form, performed similarly (see the Y-axis scale). This is most likely due to the fact that most of these tasks are out-of-distribution with respect to ImageNet. We see most of the difference on **Natural** subset. In both cases, we observe that all the methods outperform learning from scratch, however as we increase the amount of compute, the gap with learning from scratch starts closing. Next, we see that for SSL finetuning methods (first row), there is a monotonic increase in performance when more compute is provided. For supervised finetuning methods, the trends is down-wards and the performance degrades when more compute is spent. This indicates a potential over-fitting of supervised finetuning at pretraining. Moreover, we observe that supevised finetuning outperforms SSL for smaller compute and underperforms for larger compute budgets. It indicates that *supervised finetuning is a more computationally efficient way of adapting to data*, though leads to less robust representations, while SSL acts as a regularizer leading to robust representations. Moreover, we observe that warmstarting provides generally a good method and outperforms both continual finetuning and learning from scratch. Interestingly, the continual finetuning method provides quite a good performance compared to learning from scratch, but seems to under-perform when learning from scratch have seen more compute. Finally, continual finetuning performs well for **ImageNet: 100 to 900** scenario. This is due to the case that training on 900 ImageNet classes already provides a strong representation useful for down-stream tasks. These indicate that there are trade-offs in terms of compute and memory which can be exploited, but warm-starting seems robust across scenarios.

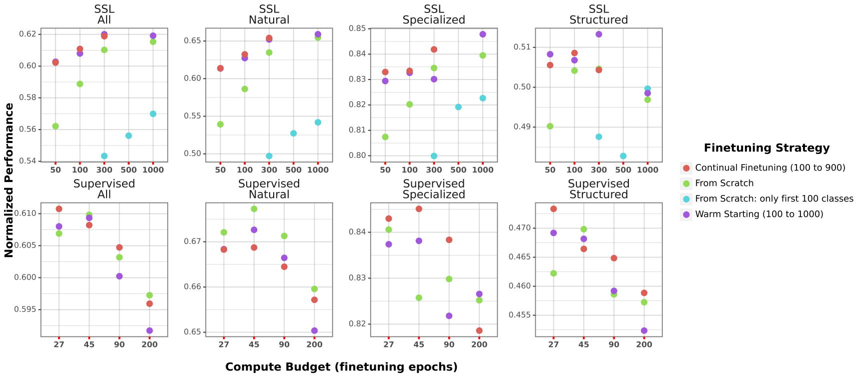


Figure 3: **Continual pretraining on first 100 and then next 900 classes of ImageNet**. The first row indicates the SSL objective, whereas the second row indicates supervised one. Different columns correspond to different datasets categories from VTAB. Each dot corresponds to per-dataset finetuning of a pretrained representation which was finetuned for a given on X-axis compute budget. The Y-axis indicates the relative normalized performance with respect to training from scratch on each of VTAB tasks. We also report performance of the model trained on $\mathcal{D}_A$ only (light blue dots). For more experimental details, see Appendix.

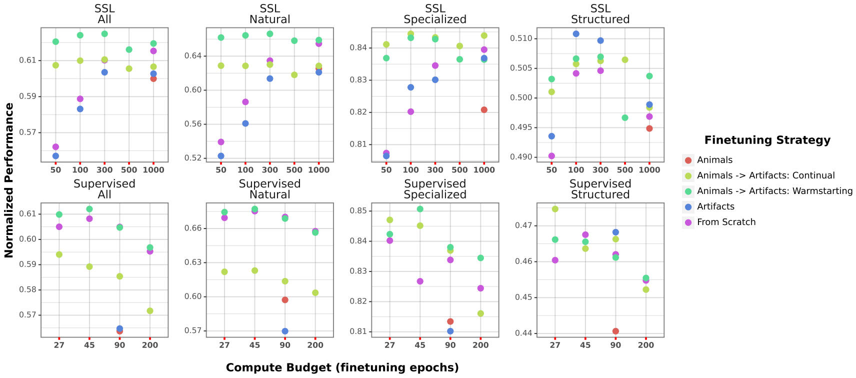


Figure 4: **Continual pretraining on Animals and then Artifacts classes of ImageNet**. The first row indicates the SSL objective, whereas the second row indicates supervised one. Different columns correspond to different datasets categories from VTAB. Each dot corresponds to per-dataset finetuning of a pretrained representation which was finetuned for a given on X-axis compute budget. The Y-axis indicates the relative normalized performance with respect to training from scratch on each of VTAB tasks. On top of finetuning methods, we also report performance for models trained on D_A – animals, and for models trained on $\mathcal{D}_B$ – artifacts. For more experimental details, see Appendix.

In Figure 5, we take a closer look on **ImageNet: Animals to Artifacts** experiment and present results on PETS and SUN397 tasks. Task PETS is more in-distribution with respect to animals subset, i.e. $\mathcal{D}_A$, whereas SUN397 is more in-distribution with respect to artifacts subset, i.e. $\mathcal{D}_B$. We observe that continual finetuning provides more efficient learning on SUN397, but loses performance on PETS. On the other hand, the warmstarting method provides generally a good trade-off in performance: it learns slightly worse than continual learning on SUN397, but it does not lose performance on PETS. Overall, warm-starting is robust across different scenarios and could be a good way to continually adapt foundational models.

3.3 Summary

Empirical results show that warm starting seems to be generally a good strategy when doing continual pretraining, though as the number of pretraining datasets increases one might need to re-weight the frequency at which the system

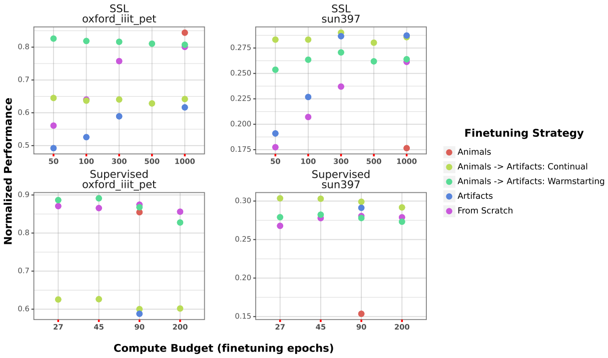


Figure 5: **Continual pretraining on Animals and then Artifacts classes of ImageNet**. Evaluation on Oxford IIIT Pets (closer to Animals) task and SUN397 (closer to Artifacts). The first row indicates the SSL objective, whereas the second row indicates supervised one. The first column indicates performance on IIIT Pets task. The second column indicates the performance on Sun 397 task. Each dot corresponds to per-dataset finetuning of a pretrained representation which was fine-tuned for a given on X-axis compute budget. The Y-axis indicates the relative normalized performance with respect to training from scratch on each of the tasks.

sees old and new data. Contrary to (Ash & Adams, 2020), we did not observe the the issues with loss of plasticity, and particularly due to the decoupling of the representation learning from the classifier, loss of plasticity might not be an issue that needs to be addressed in the near future. The results also indicate a significant performance gap between continual finetuning, warm-starting and learning from scratch. Continual finetuning learns efficiently on recent data but loses performance on the past. Moreover, it seems that learning objective has similar efficiency trade-offs. These issues suggest that the continual learning problem is far from being solved. This opens opportunity for future research.

4 Related Work

Continual learning is a fast growing field, with several surveys (e.g. Hadsell et al., 2020; Parisi et al., 2019), to which we direct the reader for a detailed breakdown of the field. Focusing specifically on our work,similar question has been raised in concurrent and recently published works Ostapenko et al. (2022); Cossu et al. (2022). There are however a few differences in how we approach this question. Ostapenko et al. (2022) starts by showing that on existing benchmarks starting from a foundational model leads to strong results. They argue that foundational models can be seen as a tool for making continual learning less expensive. In particular they propose to use some frozen representation and focus on learning continually the output layer, allowing for example the use of non-parametric methods to achieve this. The work rests on the assumption that we can rely on learning a sufficiently generic representation in a non-continual way that can be kept frozen for all future adaptation of the system. By relying on this assumption, certain challenges of the problem can be easier addressed and new opportunities can emerge. In contrast, in our work we take the position that such representation might not exist and focus of how the representation itself can continuously be adapted. In addition we propose a more thorough examination of the impact of foundational models on existing benchmarks.

Cossu et al. (2022) introduces the concept of *continual pretraining* that is aligned with the question of tuning representations, which we look at in this work as well. They focus on the impact of finetuning the existing architecture on a new data, exploring the question on language and the Core50 vision dataset. In contrast to this work we additionally try to contrast the existing continual learning frameworks to this new paradigm, highlighting what are new phenomena that emerges. We also look at the impact of using different objectives when continually learning and look at computation-performance trade-offs.

5 Conclusion

Recently the emerging of large scale pretrained models led to a paradigm shift from end-to-end learning to pretraining and finetuning. In this work, we explore the implications of this paradigm shift on continual learning. In particular, we look at the impact of foundation models on existing benchmarks in section 2 and focus on the question of continually learning representations in section 3. We argue that as these large systems needs to interact with an ever-changing world, new concepts emerge that need to be properly integrated in the representation of the systems. We believe fitting the representation to the current state of the world might be the right approach to solve this problem. However how to continually update representations, particularly in a computational efficient way, is still an open research question.

In our work we first show that existing continual learning benchmarks are not suitable for exploring these questions, as they tend not to be sufficiently large scale, nor do they tend to be sufficiently different from the data used typically for pretraining these large scale models. We propose a typical layout for the kind of benchmarks we might need to investigate continual representation learning, and take a first step in this direction constructing a limited scenario consisting on a single step of updating the pretrained representation.

We highlight that the separation of representation learning from the downstream tasks, leads to new research questions and opens the door to new approaches for continual learning. Specifically, it imposes two different time scales, the representation learning vs the downstream tasks, and the use of different objectives (e.g. contrastive learning versus supervised signal). It also requires rethinking certain concepts or metrics widely used in the community, like forgetting.

Overall, we see our work, among other concurrent works, as pushing the community towards a reformulation of the continual learning problem that takes into account overall trends in the larger community.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katie Millican, Malcolm Reynolds, et al. Flamingo: a visual language model for few-shot learning. *arXiv preprint arXiv:2204.14198*, 2022.

Rahaf Aljundi, Klaas Kelchtermans, and Tinne Tuytelaars. Task-free continual learning. *arXiv preprint arXiv:1812.03596*, 2018.

Rahaf Aljundi, Min Lin, Baptiste Goujaud, and Yoshua Bengio. Gradient based sample selection for online continual learning. *Advances in neural information processing systems*, 32, 2019.

Jordan Ash and Ryan P Adams. On warm-starting neural network training. *Advances in neural information processing systems*, 33:3884–3894, 2020.

Rishi Bommasani, Drew A Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, et al. On the opportunities and risks of foundation models. *arXiv preprint arXiv:2108.07258*, 2021.

Matko Bošnjak, Pierre H Richemond, Nenad Tomasev, Florian Strub, Jacob C Walker, Felix Hill, Lars Holger Buesing, Razvan Pascanu, Charles Blundell, and Jovana Mitrovic. Semppl: Predicting pseudo-labels for better contrastive representations. *arXiv preprint arXiv:2301.05158*, 2023.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020a.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020b.

Zhipeng Cai, Ozan Sener, and Vladlen Koltun. Online continual learning with natural distribution shifts: An empirical study with visual data. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 8281–8290, 2021.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. A simple framework for contrastive learning of visual representations. In *Proc. of International conference on machine learning (ICML)*, 2020.

Andrea Cossu, Tinne Tuytelaars, Antonio Carta, Lucia Passaro, Vincenzo Lomonaco, and Davide Bacciu. Continual Pre-Training Mitigates Forgetting in Language and Vision. 2022. URL `https://arxiv.org/abs/2205.09357v1`.

Matthias Delange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Greg Slabaugh, and Tinne Tuytelaars. A continual learning survey: Defying forgetting in classification tasks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021. doi: 10.1109/tpami.2021.3057446. URL `https://doi.org/10.1109%2Ftpami.2021.3057446`.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Enrico Fini, Victor G Turrisi Da Costa, Xavier Alameda-Pineda, Elisa Ricci, Karteek Alahari, and Julien Mairal. Self-supervised models are continual learners. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 9621–9630, 2022.

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.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. Bootstrap your own latent - a new approach to self-supervised learning. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin (eds.), *Advances in Neural Information Processing Systems*, volume 33, pp. 21271–21284. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf.

Raia Hadsell, Dushyant Rao, Andrei A. Rusu, and Razvan Pascanu. Embracing change: Continual learning in deep neural networks. *Trends in Cognitive Sciences*, 24(12):1028–1040, 2020. ISSN 1364-6613. doi: https://doi.org/10.1016/j.tics.2020.09.004. URL https://www.sciencedirect.com/science/article/pii/S1364661320302199.

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*, pp. 770–778, 2016.

Paul Janson, Wenxuan Zhang, Rahaf Aljundi, and Mohamed Elhoseiny. A simple baseline that questions the use of pretrained-models in continual learning. *arXiv preprint arXiv:2210.04428*, 2022.

Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. Scaling up visual and vision-language representation learning with noisy text supervision. In *International Conference on Machine Learning*, pp. 4904–4916. PMLR, 2021.

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

David Lopez-Paz and Marc-Aurelio Ranzato. Gradient episodic memory for continual learning. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 6467–6476. Curran Associates, Inc., 2017. URL http://papers.nips.cc/paper/7225-gradient-episodic-memory-for-continual-learning.pdf.

Zheda Mai, Ruiwen Li, Jihwan Jeong, David Quispe, Hyunwoo Kim, and Scott Sanner. Online continual learning in image classification: An empirical survey. *Neurocomputing*, 469:28–51, 2022.

Arun Mallya and Svetlana Lazebnik. Packnet: Adding multiple tasks to a single network by iterative pruning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 7765–7773, 2018.

Seyed Iman Mirzadeh, Arslan Chaudhry, Dong Yin, Timothy Nguyen, Razvan Pascanu, Dilan Gorur, and Mehrdad Farajtabar. Architecture matters in continual learning, 2022. URL https://arxiv.org/abs/2202.00275.

Jovana Mitrovic, Brian McWilliams, Jacob Walker, Lars Buesing, and Charles Blundell. Representation learning via invariant causal mechanisms. *arXiv preprint arXiv:2010.07922*, 2020.

Cuong V. Nguyen, Yingzhen Li, Thang D. Bui, and Richard E. Turner. Variational continual learning. In *International Conference on Learning Representations*, 2018. URL https://arxiv.org/abs/1710.10628.

Oleksiy Ostapenko, Timothee Lesort, Pau Rodríguez, Md Rifat Arefin, Arthur Douillard, Irina Rish, and Laurent Charlin. Foundational models for continual learning: An empirical study of latent replay, 2022. URL https://arxiv.org/abs/2205.00329.

Aristeidis Panos, Yuriko Kobe, Daniel Olmeda Reino, Rahaf Aljundi, and Richard E Turner. First session adaptation: A strong replay-free baseline for class-incremental learning. *arXiv preprint arXiv:2303.13199*, 2023.

German I Parisi, Ronald Kemker, Jose L Part, Christopher Kanan, and Stefan Wermter. Continual lifelong learning with neural networks: A review. *Neural networks*, 113:54–71, 2019.

Ameya Prabhu, Philip H. S. Torr, and Puneet K. Dokania. Gdumb: A simple approach that questions our progress in continual learning. In *Computer Vision – ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part II*, pp. 524–540, Berlin, Heidelberg, 2020. Springer-Verlag. ISBN 978-3-030-58535-8. doi: 10.1007/978-3-030-58536-5_31. URL https://doi.org/10.1007/978-3-030-58536-5_31.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8748–8763. PMLR, 18–24 Jul 2021a. URL https://proceedings.mlr.press/v139/radford21a.html.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pp. 8748–8763. PMLR, 2021b.

Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International Conference on Machine Learning*, pp. 8821–8831. PMLR, 2021.

Sylvestre-Alvise Rebuffi, Alexander Kolesnikov, Georg Sperl, and Christoph H Lampert. icarl: Incremental classifier and representation learning. In *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 2001–2010, 2017.

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. doi: 10.1007/s11263-015-0816-y.

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.

Gobinda Saha, Isha Garg, and Kaushik Roy. Gradient projection memory for continual learning. 2021. doi: 10.48550/ARXIV.2103.09762. URL `https://arxiv.org/abs/2103.09762`.

Jonathan Schwarz, Wojciech M. Czarnecki, Jelena Luketina, Agnieszka Grabska-Barwinska, Yee Whye Teh, Razvan Pascanu, and Raia Hadsell. Progress & compress: A scalable framework for continual learning. *ICML*, abs/1805.06370, 2018.

Aliaksandra Shysheya, John Bronskill, Massimiliano Patacchiola, Sebastian Nowozin, and Richard E Turner. Fit: Parameter efficient few-shot transfer learning for personalized and federated image classification. *arXiv preprint arXiv:2206.08671*, 2022.

Vishal Thengane, Salman Khan, Munawar Hayat, and Fahad Khan. Clip model is an efficient continual learner. *arXiv preprint arXiv:2210.03114*, 2022.

Nenad Tomasev, Ioana Bica, Brian McWilliams, Lars Buesing, Razvan Pascanu, Charles Blundell, and Jovana Mitrovic. Pushing the limits of self-supervised resnets: Can we outperform supervised learning without labels on imagenet? *arXiv preprint arXiv:2201.05119*, 2022.

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

Zirui Wang, Jiahui Yu, Adams Wei Yu, Zihang Dai, Yulia Tsvetkov, and Yuan Cao. Simvlm: Simple visual language model pretraining with weak supervision. *arXiv preprint arXiv:2108.10904*, 2021.

Yue Wu, Yinpeng Chen, Lijuan Wang, Yuancheng Ye, Zicheng Liu, Yandong Guo, and Yun Fu. Large scale incremental learning, 2019. URL `https://arxiv.org/abs/1905.13260`.

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

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

Xiaohua Zhai, Joan Puigcerver, Alexander Kolesnikov, Pierre Ruyssen, Carlos Riquelme, Mario Lucic, Josip Djolonga, Andre Susano Pinto, Maxim Neumann, Alexey Dosovitskiy, et al. A large-scale study of representation learning with the visual task adaptation benchmark. *arXiv preprint arXiv:1910.04867*, 2019.

A Appendix

A.1 Task-incremental and class incremental experiments

This section covers the Section 2 in the paper. For all the experiments in Section 2, we pretrained ResNet-50 (He et al., 2016) model on ImageNet-1k (Russakovsky et al., 2015) with different learning objectives. For supervised pretraining on ImageNet-1k, we followed the protocol formulated in (Chen et al., 2020), which does 90 epochs of learning on ImageNet-1k. Moreover, we also pretrained models via self-supervised objectives for 1000 epochs on ImageNet-1k. We considered such methods as BYOL (Grill et al., 2020), ReLICv2 (Tomasev et al., 2022) and SemPPL (Bošnjak et al., 2023).

A.1.1 Task-incremental experiments

As concrete benchmarks, we considered Split-CIFAR-100 with 10 tasks split (see Saha et al. (2021) for SOTA) as well as with 20 tasks split (see Mirzadeh et al. (2022) for SOTA). Moreover, we considered split TinyImageNet benchmarks with 10 tasks split (see Delange et al. (2021) for SOTA) and split-ImageNet1k benchmark with 60 tasks (see Mirzadeh et al. (2022) for SOTA).

For **linear evaluation** experiments, which are presented in Tables 1 and 2, we froze the pretrained parameters θ of the backbone $g(x;\theta)$ in equation equation 1 and only learned task-specific head parameters ϕ_i. We used a batch size of 1024. We used SGD optimizer with momentum of 0.1 and without any weight decay. We ran a sweep over learning rate parameter in a range of $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$ and ran each experiment for 3 seeds. For each of the task, we had train, validation and test set. For CIFAR-100, we used random crop augmentations producing images of size 24x24. For split-ImageNet1k, we used random crop augmentations producing images 200x200. For tiny-ImageNet1k, we used random crop augmentations producing images 48x48. We selected the best hyperparameters (learning rate) per experiment based on the performance on the validation set. As a selection metric, we used average accuracy. Then, we reported the performance of the method with corresponding hyperparameter on the test set.

For **full finetuning** experiments, which are presented in Table 3, we took the pretrained backbone parameters θ and do finetuning of θ as well as the task-specific head parameters ϕ_i. When finetuning these two sets of parameters, we used different objectives and learning rates. We used SGD optimizer with momentum of 0.1 and without any weight decay. For finetuning the head parameters, we used supervised learning objective with learning rate chosen from a range $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$. For CIFAR-100, we used random crop augmentations producing images of size 24x24. For split-ImageNet1k, we used random crop augmentations producing images 200x200. For tiny-ImageNet1k, we used random crop augmentations producing images 48x48. We used a batch size of 1024. For finetuning representation, we used either supervised, self-supervised (ReLICv2) or a sum of supervised and self-supervised objectives. For finetuning representation, we took the same learning rate as for the head parameters and multiplied it by $\beta \in (0, 1]$. We tried multiple values of β and saw that $\beta = 0.1$ provided the best performance. We report the results of using $\beta = 0.1$ as well as $\beta = 1$. The latter corresponds to the common way of finetuning pretrained models. Each experiment is run for 3 seeds. The best hyperparameters are chosen by looking at the average accuracy on the validation set. Then, we report performance on the test set for each of the method with found best hyperparameters.

A.1.2 Class-incremental experiments

In this setting, we assume a sequence of tasks $\mathcal{T}_1, \ldots, \mathcal{T}_n$ such that each subsequent task contains new unseen classes and there are no explicit task boundaries. The setting was studied in multiple works such as (Rebuffi et al., 2017; Wu et al., 2019; Prabhu et al., 2020). This becomes a challenging problem even for the linear classifier, since the softmax layer may push down the probabilities for the classes which are no longer observed in the batches. We consider class-incremental Split-CIFAR100 as in (Wu et al., 2019) with 10 and 20 tasks, where each task has disjoint classes.

We froze the pretrained parameters θ of the backbone $g(x;\theta)$ in equation equation 1 and sequentially (class-incrementally) learned the head parameters. The head parameters are learned in a supervised way by optimizing the cross entropy on the received data. Since the model uses softmax layer at the end, when it observes new classes, it starts to push down the probabilities for old classes. This results in forgetting of the old classes. To address the issue, we introduced replay strategy. For each gradient step on the batch of data for the current task, we do one gradient update on the batch of data sampled from replay. The replay batch size is 2 multiplied by a number of previously seen tasks, i.e. we replay 2 samples from each of the previous task, when we do a gradient update on the current task. We used a batch size of 1024 for learning on the current task. We used SGD optimizer with momentum of 0.1 and without any weight decay. We ran a sweep over learning rate parameter in a range of $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$ and ran each experiment for 3 seeds. For CIFAR-100, we used random crop augmentations producing images of size 24x24. For split-ImageNet1k, we used random crop augmentations producing images 200x200. For tiny-ImageNet1k, we used random crop augmentations producing images 48x48.

A.1.3 Out-of distribution finetuning experiment

This section covers results in Section 2.3. For this experiment, we take the model pretrained on ImageNet-1k using ReLICv2 loss. Then, for each of the considered VTAB task, we finetune the backbone parameters θ as well as the last layer parameters. Backbone parameters are finetuned either via supervised finetuning or via self-supervised finetuning using ReLICv2. The last layer parameters are always finetuned using supervised objective. On top of that, we report the results of linear evaluation (learning only last layer on top of the pretrained representation) as well as the learning from scratch, where the whole model is learned from scratch on a given VTAB task. Each of this experiment is done independently on each of the VTAB task. We used a batch size of 1024. We used SGD optimizer with momentum of 0.1 and without any weight decay. We sweep over learning rates in a range of $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$ as well as over β in a range $\{0.1, 1\}$. For VTAB, as data augmentations, we used random crops producing images of size 200x200. Parameter β is a scalar which multiplies the gradients of the backbone parameters. We run each experiment for 3 seeds. We consider a variant of VTAB benchmark where we have only 1000 training points in the training set and a large test set. We split 1000 points training set into training (800 points) and validation (200 points). We select the best hyperparameters based on the performance on the validation set. Then, we report the performance on the test set using the best hyperparameters.

A.2 Continual representation learning experiments

This section covers experiments from Section 3, i.e. **ImageNet: 100 to 900** as well as **ImageNet: Animals to Artifacts**. For both experiments, we split the original ImageNet-1k dataset in two subsets, $\mathcal{D}_A$ and $\mathcal{D}_B$. We pretrain models on dataset $\mathcal{D}_A$ using supervised and SSL (ReLICv2) objectives. When we use supervised objective, we pretrain for $N = 90$ epochs. When we use self-supervised objective, we pretrain for $N = 1000$ epochs. For each pretrained model on the datasets $\mathcal{D}_A$, we do finetuning on the dataset $\mathcal{D}_B$ for a different number of epochs. Moreover, during finetuning, we use the same objective as used for pretraining on $\mathcal{D}_A$.

For finetuning we use 3 strategies. **Continual finetuning** initalizes the model from the pretrained one on the dataset $\mathcal{D}_A$ and simply finetunes it on the dataset $\mathcal{D}_B$. **Warm starting** initializes the model from the pretrained one on the dataset $\mathcal{D}_A$ and finetunes it on the union of $\mathcal{D}_A \cup \mathcal{D}_B$. **From scratch** initializes the model randomly and trains on the union $\mathcal{D}_A \cup \mathcal{D}_{\mathcal{B}}$.

For both of the experiments, we use VTAB benchmark to measure performance. We consider a variant of VTAB containing only 1000 data points in the training set. We split training set into training (800 points) and validation (200 points) subsets. For each finetuning experiment on each of the VTAB task, we always sweep over learning rates in range of $\{0.1, 0.01, 0.001, 0.0001, 0.00001\}$ as well as over β in a range $\{0.1, 1\}$. We used SGD optimizer with momentum of 0.1 and without any weight decay. For VTAB, as data augmentations, we used random crops producing images of size 200x200. Parameter β is a scalar which multiplies the gradients of the backbone parameters. Each experiment is run for 3 seeds. We select the best hyperparameters based on the performance on the validation set. We report the performance on the test set based on the best hyperparameters. As a measure of performance we use a relative improvement in accuracy on given VTAB task of a model which is finetuned on this task from a pretrained one, against the accuracy of the model which is trained on the VTAB task from scratch.

On top of VTAB, we also report the performance on the ImageNet test set which is shown in Figure 2.

A.2.1 ImageNet: 100 to 900 experiment

In this experiment we split the ImageNet-1k dataset in two datasets $\mathcal{D}_A$ and $\mathcal{D}_B$. The dataset $\mathcal{D}_A$ contains the first 100 classes and the dataset $\mathcal{D}_B$ contains next 900 classes. We report performance in Figure 3 which is a relative difference in accuracy of the finetuned model against the model which was trained from random initialization on each of the VTAB task.

A.2.2 ImageNet: Animals to Artifacts experiment

In this experiment we split the ImageNet-1k dataset in two datasets $\mathcal{D}_A$ and $\mathcal{D}_B$. The dataset $\mathcal{D}_A$ contains the first 398 classes which correspond to a general category which we call Animals. The dataset $\mathcal{D}_B$ contains next 602 classes which correspond to a general category which we call Artifacts. We report performance on the whole VTAB in Figure 4 and on Oxford IIIT Pet and SUN397 tasks in Figure 5.