WHAT HAPPENS DURING FINETUNING OF VISION TRANSFORMERS: AN INVARIANCE BASED INVESTIGATION

Gabriele Merlin
MPI-SWS
Germany
gmerlin@mpi-sws.org

Vedant Nanda
MPI-SWS / University of Maryland
Germany / U.S.
vnanda@mpi-sws.org

Ruchit Rawal
MPI-SWS
Germany
rrawal@mpi-sws.org

Mariya Toneva
MPI-SWS
Germany
mtoneva@mpi-sws.org

ABSTRACT

The pretrain-finetune paradigm usually improves downstream performance over training a model from scratch on the same task, becoming commonplace across many areas of machine learning. While pretraining is empirically observed to be beneficial for a range of tasks, there is not a clear understanding yet of the reasons for this effect. In this work, we examine the relationship between pretrained vision transformers and the corresponding finetuned versions on several benchmark datasets and tasks. We present new metrics that specifically investigate the degree to which invariances learned by a pretrained model are retained or forgotten during finetuning. Using these metrics, we present a suite of empirical findings, including that pretraining induces transferable invariances in shallow layers and that invariances from deeper pretrained layers are compressed towards shallower layers during finetuning. Together, these findings contribute to understanding some of the reasons for the successes of pretrained models and the changes that a pretrained model undergoes when finetuned on a downstream task.

1 INTRODUCTION

In recent years much progress in deep learning has been driven by the reuse of models that were pretrained on large amounts of data. This is usually achieved by finetuning their parameters using a smaller amount of data from a target downstream task. This pretrain-finetune paradigm usually improves downstream performance over training a model from scratch on the same task, and has become commonplace across many areas of machine learning, including natural language processing (Howard & Ruder, 2018) and computer vision (Girshick et al., 2014).

While pretraining is empirically observed to be beneficial for a range of tasks, there is not a clear understanding yet of the reasons for this effect. Previous work has empirically examined various conditions for pretraining and found that for a given budget of pre-training images, training with fewer classes, but more images per class performs better (Huh et al., 2016). Pretraining has also been posited to elicit an accelerated convergence during finetuning (Kornblith et al., 2019b), suggesting that during pretraining, models learn transferable representations, particularly when the finetuning task domain is similar to the pretraining task.

In this work, we further examine the relationship between pretrained vision transformers and the corresponding finetuned versions on several benchmark tasks and datasets. A key to our study is that we leverage STIR (Nanda et al., 2022), a recent approach that estimates how much of the invariances to specific perturbations learned by one source model are shared with a second target model. We adopt this approach because of the observation that learning to be invariant to some perturbations has been shown to improve generalization ability in individual models (DeVries & Taylor, 2017; Zhang et al., 2018; Yun et al., 2019), and the transfer learning ability to other models (Salman et al., 2020). This suggests that learning invariant representations may enable generalization, and so it is possible that pretrained models are learning such invariant representations. STIR can help understand the extent to which invariance from the pretrained model is learned or forgotten by a finetuned model.

Using this approach, we define metrics that are useful for tracking the degree to which pretrained invariances are forgotten and new invariances are learned by finetuning a pretrained model. A well-known constraint that occurs

during training is the stability-plasticity dilemma (Mermillod et al., 2013), which refers to the trade-off between the ability of a neural network to retain information that has already been learned (stability) and the ability to learn new information (plasticity). Finding a balance between these two factors is thought to be crucial for the successful functioning of a neural network. Developing metrics that can capture the degree to which pretrained invariances are forgotten and new invariances are learned during finetuning allows us to characterize the trade-off between old and new invariances during the pretrain-finetune paradigm.

Using these metrics, we present a suite of empirical findings, including that pretraining induces transferable invariances, especially in the shallow layers of the network (*i.e.* closer to the inputs), and that invariances from deeper pretrained layers are *compressed* towards shallower layers during finetuning. Together, these findings contribute to understanding some of the reasons for the successes of pretrained models and the changes that a pretrained model undergoes when finetuned on a downstream task.

2 Related Works

Forgetting and learning. Forgetting and learning have been studied extensively in continual learning (Lesort et al., 2020; Kirkpatrick et al., 2017; Kemker et al., 2018). In this setting, a model is trained on a sequence of tasks and is required to maintain performance on previously learned tasks while learning new tasks. Toneva et al. (2018) study data instances that are learned and forgotten many times during the training process. These previous works primarily measure forgetting and learning from a behavioural perspective: they rely on task performance to quantify these measures and do not take into account invariances. In our work, we provide a fresh perspective on learning and forgetting through the lens of invariances. This unique lens allows us to propose two additional notions of compression and expansion, which provide a more complete picture of how representations change during finetuning and something that prior works have not considered. There are three notable works that go beyond task performance and are similar in spirit to ours: Ramasesh et al. (2021) use representation similarity and Davari et al. (2022) use linear probes, both to understand catastrophic forgetting in a continual learning setup. More recently, Ramasesh et al. (2022) investigated the role of pretraining scale in learning orthogonal class representations that lead to lower catastrophic forgetting during continual learning. We differ in two key aspects: we consider the setup of transfer learning which is more broadly applicable to a variety of machine learning tasks; and in addition to just using representations (via either representation similarity or linear probes), we also measure shared invariances that allow us to derive additional insights about changes that occur due to finetuning.

Using Similarity to Study Representations. In recent years, numerous works have adopted representational similarity measures (RSMs) to inspect representations of neural networks that differ in architecture family (Raghu et al., 2021), depth/width (Nguyen et al., 2021), and localization of information (Wu et al., 2020). Most similar to our work, Phang et al. (2021) utilize RSMs to investigate the changes in representations of a finetuned model with respect to the pretrained model. While conventional RSMs are useful in indicating 'representational-divergence' (*i.e.* how are the representations changing between two models for the same set of inputs), they are not equipped to quantify the degree to which two models share invariances. Thus, we build upon an approach proposed by Nanda et al. (2022), which reveals the extent to which the learned representations remain invariant to the same perturbation from one model to another. We analyze the pretraining-finetuning paradigm of ViT models that was not evaluated in Nanda et al. (2022). Moreover, we propose novel metrics that probe the extent and nature of *forgetting* and *learning* of shared-invariances due to finetuning.

3 Methods

We build upon a recently proposed general approach to estimate the shared invariances between two models, which we describe in Section 3.1. In Sections 3.2 and 3.3, we define metrics that are useful to characterize the forgetting of pretrained invariances and learning of new invariances by the finetuned model. In Section 3.4, we describe the experimental setup used to validate these metrics and to use them to reveal new insights about how models change during finetuning. Invariance is used in many contexts in the broad ML literature, however, here we adopt the terminology of Nanda et al. (2022) and use the word invariance to broadly mean *transformations of inputs that do not significantly change the representation of a model at a particular layer*.

3.1 Similarity Through Inverted Representations (STIR)

Similarity Through Inverted Representations (STIR) (Nanda et al., 2022) is a measure of shared invariances between two representations. To measure the degree to which the i^{th} layer of model m_2 shares invariances with the j^{th} layer

of model m_1, Nanda et al. (2022) define STIR as:

$$\texttt{STIR}(m_2^{[i]}|m_1^{[j]}, X, S_r) = \frac{1}{k}\sum_{X'} S_r(m_2^{[i]}(X), m_2^{[i]}(X')). \tag{1}$$

In equation 1, X is a set of samples and X' is a set of generated samples such that $m_1^{[j]}(X) \approx m_1^{[j]}(X')$ - *i.e.* a set of perturbed samples for which the j^{th} layer of m_1 is representationally invariant. S_r is a similarity metric, which is often taken to correspond to CKA (Kornblith et al., 2019a). Thus, using STIR one can compute the degree of shared-variances between the representations learned by any two layers (i & j) both across two models ($\texttt{STIR}(m_2^{[j]}|m_1^{[i]}, X, S_r)$) or within an individual model ($\texttt{STIR}(m_1^{[j]}|m_1^{[i]}, X, S_r)$). Note that STIR of a layer with itself within an individual model (*i.e.* $\texttt{STIR}(m_1^{[i]}|m_1^{[i]}, X, S_r)$) is 1. The sampling of X is repeated k times. Unlike the standard CKA that captures changes in the representation between two models, STIR is able to capture the robustness of a target model to perturbations on which a reference model is representationally invariant. Note that STIR is directional: $\texttt{STIR}(m_2^{[i]}|m_1^{[j]}, X, S_r) \neq \texttt{STIR}(m_1^{[j]}|m_2^{[i]}, X, S_r)$. For our purposes, STIR is useful for disentangling the `learning` and `forgetting` of invariances in a finetuned model, which cannot be easily disentangled using standard CKA. It is also useful to compare layers with different sizes, since it is based on CKA which by design has a normalization factor that ensures invariance to isotropic scaling (Kornblith et al., 2019a; Nanda et al., 2022)

STIR uses representations similarity (CKA by Kornblith et al. (2019a)) under the hood – which by design has a normalization factor that ensures invariance to isotropic scaling. This makes both representation similarity (CKA) and STIR suitable for comparison across layers of different sizes. Comparison of CKA and STIR across layers was also done by both Kornblith et al. (2019a) and Nanda et al. (2022). Thus we believe there's already proper normalization in both STIR and CKA which make them suitable for cross-layer comparisons.

3.2 Forgetting and Learning

To measure the extent to which the model learns and forgets invariances during finetuning, we propose two metrics: `learning` and `forgetting`. Unlike previous work described in Section 2, our aim is to characterize them from a representational robustness perspective using the STIR measure.

We define the $\texttt{forgetting}(ft^{[i]}|pt^{[j]})$, where $ft^{[i]}$ is a layer of the finetuned neural network as:

$$\texttt{forgetting}(ft^{[i]}|pt^{[j]}) = \texttt{STIR}(pt^{[i]}|pt^{[j]}) - \texttt{STIR}(ft^{[i]}|pt^{[j]}). \tag{2}$$

In the second term of Equation 2, we measure the shared invariances between the layer i of the finetuned model (ft) and the layer j of the pretrained model (pt). Thus, intuitively $\texttt{forgetting}(ft^{[i]}|pt^{[j]})$ measures the decrease in shared-invariances between i^{th} and j^{th} layers after finetuning (*i.e.* $\texttt{STIR}(ft^{[i]}|pt^{[j]})$) relative to after pretraining (*i.e.* $\texttt{STIR}(pt^{[i]}|pt^{[j]})$). We are interested in measuring the evolution of the same layer during finetuning. Therefore the `forgetting` of a finetuned layer ($i = j$) is measured by $\texttt{forgetting}(ft^{[i]}|pt^{[i]}) = \texttt{STIR}(pt^{[i]}|pt^{[i]}) - \texttt{STIR}(ft^{[i]}|pt^{[i]})$. If the first term, which is always 1, is greater than the second we can say that layer i is forgetting.

Similarly we define $\texttt{learning}(ft^{[i]}|pt^{[j]})$ as:

$$\texttt{learning}(ft^{[i]}|pt^{[j]}) = \texttt{STIR}(ft^{[i]}|ft^{[j]}) - \texttt{STIR}(pt^{[i]}|ft^{[j]}). \tag{3}$$

In the second term of Equation 3, we measure the shared invariances between the layer j of the finetuned model (ft) and layer i of the pretrained model (pt). Intuitively, if the invariances defined w.r.t a finetuned model are shared by a pretrained model (*i.e.* $\texttt{STIR}(pt^{[i]}|ft^{[j]})$), the degree of new invariances learned during finetuning is low. The `learning` of a particular finetuned layer ($i = j$) is measured by $\texttt{learning}(ft^{[i]}|pt^{[i]}) = \texttt{STIR}(ft^{[i]}|ft^{[i]}) - \texttt{STIR}(pt^{[i]}|ft^{[i]})$. If the first term, which is always 1, is greater than the second we can say that layer i is learning. These metrics can measure the relative levels of learning and forgetting between two models. As a result, they are useful for comparing the effect of different design choices involved in finetuning: pretrained models, finetuning tasks, or datasets.

Ramasesh et al. (2021) analyzed the phenomenon of catastrophic forgetting using representation similarity to identify the layers most responsible for forgetting. To compare our STIR-based metrics with CKA, we propose a metric similar to Ramasesh et al. (2021). We define the `cka divergence` as:

$$\texttt{cka divergence}(ft^{[i]}, pt^{[j]}) = \texttt{CKA}(pt^{[i]}, pt^{[j]}) - \texttt{CKA}(ft^{[i]}, pt^{[j]}). \tag{4}$$

Since we are interested in measuring the evolution of a specific layer ($i = j$), we can use $\texttt{cka divergence}(ft^{[i]}, pt^{[i]})$. Therefore, the first term of this equation is always 1. As a result, we are measuring how much the representations of layer i of the finetuned model (ft) vary from the pretrained model (pt). As result, when $\texttt{CKA}(ft^{[i]}, pt^{[i]})$ is low the $\texttt{cka divergence}$ is high. Since $\texttt{CKA}$ is not a directional metric, we cannot distinguish between learning and forgetting. Our metric is qualitatively the opposite of the one used by Ramasesh et al. (2021).

3.3 Compression and Expansion

During finetuning, a model may not only forget or learn invariances, but the invariances from a specific layer in the pretrained model may also migrate to a different layer in a finetuned model. To measure the migration of invariances during finetuning, we propose $\texttt{Invariance Flow}$. For two layers i and j we define the $\texttt{Invariance Flow}(ft^{[i]}, pt^{[j]})$ as:

$$\texttt{Invariance Flow}(ft^{[j]}, pt^{[i]}) = \texttt{STIR}(ft^{[j]}|pt^{[i]}) - \texttt{STIR}(ft^{[i]}|pt^{[i]}) \tag{5}$$

Equation 5 contrasts the ability of layer j of the finetuned model to share invariances with layer i of the pretrained model, with the same ability but of layer i. The result of this metric is the $\texttt{Invariance Flow Matrix}$, which describes the flow of invariances from a pretrained layer to another finetuned layer. As a result, if $\texttt{Invariance Flow Matrix}(i, j) > 0$ and $j < i$ we can say that the invariances of the pretrained model are *compressed* to earlier layers, otherwise if $j > i$ they are *expanded* to deeper layers. If $i = j$ the $\texttt{Invariance Flow}$ is 0 since the two terms of the equation would be equal.

Our proposed metrics can be generalized to other settings *i.e.* between any pair of reference and target models. We explicitly use the notation ft and pt for clarity.

3.4 Models and Tasks

Models. For our experiments, we use the Vision Transformer (ViT) model, proposed by Dosovitskiy et al. (2021). ViT is a variant of the Transformer architecture (Vaswani et al., 2017) for images. The architecture of Vision Transformer is similar to the standard Transformer, consisting of a stack of multi-head self-attention layers followed by fully connected layers. The key difference is the use of convolutional layers for image feature extraction rather than the embedding layer used in the original Transformer. We use a ViT model with 12 layers that use a patch size of 32x32 and 224x224 as image resolution. ViT models have proven to be effective end efficient in many tasks, achieving good performance in a shorter training time with respect to convolutional models. We use the implementation of ViT provided by Wightman (2019). For our experiment, ViT models are pretrained on ImageNet (Deng et al., 2009) classification or pretrained on CIFAR100 (Krizhevsky et al.) classification (see details in Appendix B).

Finetuning Tasks and Datasets. To understand how the difference between a pretraining and a finetuning task affects the shared invariances, we finetune the pretrained models on two different tasks: classification and reconstruction. Since all pretrained models that we use are trained using classification, the reconstruction task is helpful in analyzing changes in shared invariance that occur when the finetuning task is different from the pretraining task. For the reconstruction task, we reconstruct the original image from ViT representations using a convolutional decoder based on the Hugging Face implementation (Wolf et al., 2020). To train the model, we adopt the L1 loss as the objective function to ensure that the reconstructed image is as close as possible to the original image. For classification, we adopt the standard cross-entropy loss function. We performed experiments with 3 representative datasets from the Visual Task Adaptation Benchmark (Zhai et al., 2019), two naturalistic datasets - CIFAR100 (Krizhevsky et al.) and Oxford-IIT Pet (Parkhi et al., 2012) - and one specialized dataset - Eurosat (Helber et al., 2019), a satellite imagery dataset. We further experimented with CIFAR10 (Krizhevsky et al.) to compare the results of CIFAR100 with a dataset of similar domain and to use a well-known benchmark widely used in the literature. More details about training hyperparameters and methods are reported in Appendix B.

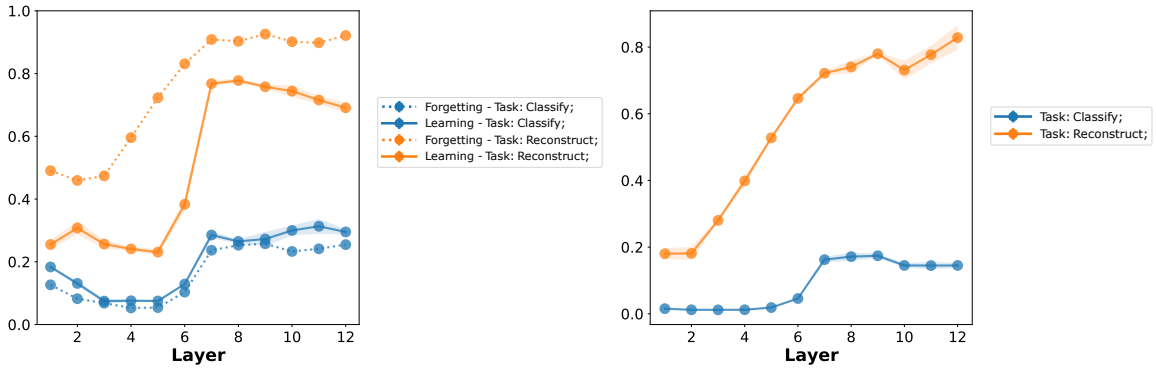
4 Results

Using the proposed metrics, we examine two main effects on the forgotten pretrained invariances and the newly learned invariances by the finetuned model: 1) the effect of the type of task a pretrained model is finetuned for (Section 4.1), and 2) the effect of the dataset on which the pretrained model is trained on (Section 4.2). We further investigate whether invariances that appear to have been forgotten in a certain finetuned layer have actually migrated to a different

layer in the model (either a shallower or a deeper layer) in Section 4.3. We finally study the training dynamics of forgotten and learned invariances in Section 4.4. All results are obtained by averaging over 3 random seeds of STIR computation. For each STIR computation, we set the number of sampled images (X in Equation 1) to 500, and we use 50 optimization iterations to find a representationally invariant input for each sampled image.

4.1 Effects of the finetuning task

To study the effect of different finetuning tasks on forgetting pretrained invariances and on learning new invariances during finetuning, we examine the `forgetting` and `learning` metrics across layers of a ViT model pretrained on ImageNet and finetuned on two tasks for the same CIFAR10 dataset: 1) classification and 2) reconstruction. We present the results in Figure 1a. The results for the remaining datasets (CIFAR100, Oxford-IIT Pet, EuroSAT) are qualitatively similar and can be viewed in Appendix C.



(a) Learning and forgetting CIFAR10

(b) CKA divergence CIFAR10

Figure 1: `learning`, `forgetting` and `cka divergence` values for ViT model pretrained on ImageNet and finetuned on CIFAR10 on reconstruction and classification task. `learning`, `forgetting` measure two different quantities that may differ from each other. The model finetuned on the reconstruction task shows different levels and dynamics of `learning` and `forgetting` with respect to the model finetuned on the classification task. Changes in `cka divergence` vary from task to task. The model finetuned on the reconstruction task shows higher `cka divergence` in each layer. Later layers for both task have a higher `cka divergence` with respect to earlier layers.

Learning and forgetting capture different phenomena. Intuitively, a model with a given capacity to store information may forget previously learned invariances in order to learn new ones. Therefore, the `forgetting` and `learning` metrics may be thought of as two consequences of the same phenomenon. However, the results in Figure 1a show that the two corresponding metrics can lead to different results. In particular, the metrics behave differently when the task for pretraining (*i.e.* classification) differs from the task for finetuning (*i.e.* reconstruction). The clearest example is for layers $3-5$, where the `learning` during the reconstruction finetuning is constant while the `forgetting` is steadily increasing, and for layers $9-12$, where the `forgetting` during the reconstruction finetuning is constant while the `learning` is steadily decreasing. Therefore, the proposed `learning` and `forgetting` metrics capture some unique information and may reveal different insights.

Representational similarity does not imply shared invariance. Analyzing Figure 1a, we can see that the model finetuned for the classification task has moderate `forgetting` and `learning` values for the initials layers. The two metrics decrease up to layer 5 and then start to increase, reaching values that are slightly above the ones evaluated at the earlier layers. This means that the initial layers and the later layers are the ones where more `forgetting` and `learning` of invariances occurs during finetuning. In contrast, if we were to use a representational similarity measure to quantify the differences in representations between the finetuned model and the pretrained model, we observe a very different trend in the early layers for the classification task (see Figure 1b). The representational similarity in the early layers is very high, corresponding to almost 0 values for the `cka divergence`. This difference between the `cka divergence` and the `forgetting` and `learning` metrics shows that representational similarity does not imply shared invariance. This result supports previous findings from Nanda et al. (2022), that similarly show that high values of CKA (or low values of `cka divergence`) can correspond to various degrees of shared invariance. Therefore, the proposed metrics based on relative invariance may provide additional insight into how a model changes during finetuning.

Earlier layers *do* change, even when the finetuning task matches the pretraining task. Previous work based on representational similarity has reported that during finetuning, later layers adapt more to the finetuning task than earlier layers (Ramasesh et al., 2021). Thus, we can expect not only that the representations change more in the later layers than in the earlier layers, but also that later layers learn new invariances and forget more pretrained invariances. Surprisingly, in Figure 1a, we observe that the initial layers (0,1) also learn and forget invariances similarly to the later layers, thus adapting to the new finetuning task and data set. As the model undergoes changes in the input distribution during finetuning (from ImageNet to CIFAR10), we hypothesise that earlier layers need to adapt to the new input distribution. Ramasesh et al. (2021) focused more on a continual learning setting and the result based on `forgetting` and `learning` may differ. However we show in Figure 1b and also in a pretraing-finetuning setting CKA is not able to capture changes in the model.

Transferring to a new task can require learning new invariances. Finetuning a model on a task that is different from the one on which it was trained may require learning new invariances, and to a larger degree than if the model was finetuned on the same pretraining task. We can examine this question by contrasting the `learning` values for the two tasks in Figure 1a: classification and reconstruction. We observe that new invariances are learned by both finetuned models. For the reconstruction task, the `learning` metric is higher in every layer compared to the classification task. This means that, as expected, finetuning a pretrained model on a new task may require learning new invariances.

Transferring to a lower-level task can require forgetting in earlier layers. The ability of deep neural networks to learn increasingly abstract concepts with increasing network depth has been widely studied in the literature (Zeiler & Fergus, 2014). Is this effect still observable when analysing learned and forgotten invariances during finetuning? As we can see in our experiments in Figure 1a, a low-level task, such as reconstruction, starts the increasing trend of `forgetting` from layer 3. Instead, a more abstract task, such as classification, starts the increasing trend of `forgetting` from layer 5 onwards. This means that `forgetting` shows an earlier growth trend for the model finetuned to the reconstruction task, compared to the model finetuned on a more abstract task such as classification. We hypothesize that because the reconstruction task is a lower-level task, the model forgets earlier invariances in the hierarchy of layers.

4.2 Effects of the pretraining dataset

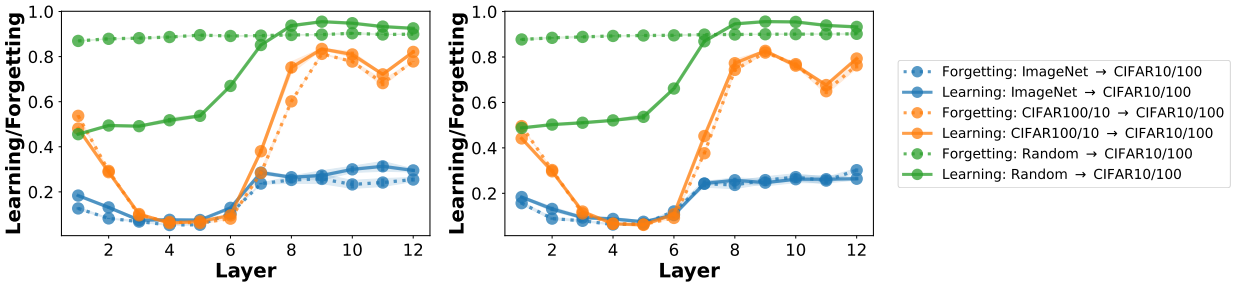
To analyze the effect of a pretrained dataset on the forgotten and learned invariances during finetuning, we analyze models that are initialized using different weights, and are then all finetuned for the same task (classification) and dataset (CIFAR10 in Fig 2a and CIFAR100 in Fig 2b). For each finetuning dataset, we consider 2 models with initial weights that are obtained by (pre) training on ImageNet, or on either CIFAR100 or CIFAR10 depending on the finetuning dataset. As a baseline, we additionally consider a third model that's trained from scratch for each dataset (*i.e.* starting from a random weight initialization). We present the values of the `forgetting` and `learning` metrics for all models in Fig 2 and `cka divergence` in Fig 3.

Pretraining instills reusable invariances in early layers. We observe that the finetuned models that start from pretrained models exhibit a substantially lower `learning` and `forgetting` in the early layers (1-6), than the model trained from scratch (green lines in Fig 2). This suggests that pretraining instills certain invariances during pretraining that can be reused in the finetuning task. This finding is consistent with other work showing that earlier layers of the pretrained model preserve more general knowledge that is still useful during finetuning (Ramasesh et al., 2021; Zeiler & Fergus, 2014).

ImageNet pretraining leads to more useful invariances even in later layers. In Fig 2 we observe that even for later layers (*i.e.* 7-12), the learning and forgetting values for the model pretrained on ImageNet are significantly lower than those for CIFAR10/100 pretraining or training from scratch. Our observation aligns with decades of empirical results that have found ImageNet pretraining to be an effective strategy for a variety of computer vision tasks Girshick et al. (2014); Kornblith et al. (2019b); Long et al. (2015).

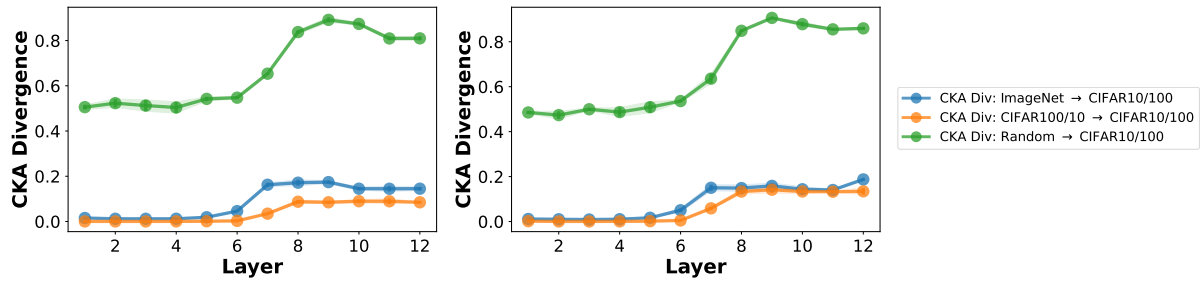
Training from scratch requires higher learning and forgetting across all layers For both finetuning on CIFAR10 and CIFAR100, we see that when learning from scratch (*i.e.*random weight initialization; shown in green in Fig 2) `learning` and `forgetting` both have higher values than pretrained models, across all layers. This aligns well with well-studied gains of pretraining in prior works Donahue et al. (2014); Erhan et al. (2010).

Representation similarity does not faithfully indicate the effects of pretraining. In Fig 3 we show `cka divergence` for the same setting evaluated in Fig 2. We observe that contrary to our finding about `learning`



(a) CIFAR10 (b) CIFAR100

Figure 2: **[Pretraining dataset affects `learning` and `forgetting`]** `learning` and `forgetting` across layers for a model finetuned for CIFAR10 (left) and CIFAR100 (right) classification, starting from different pretrained weights. The models are pretrained on CIFAR100/10 (orange), ImageNet (blue), or trained from scratch (random initialization, shown in green). For CIFAR10 (left), the orange line shows results for pretraining on CIFAR100, while for CIFAR100 (right), it shows results for pretraining on CIFAR10. Finetuning a model pretrained on *some* data leads to a lower `learning` and `forgetting` of invariances in early layers, whereas training from scratch leads to much higher learning (and forgetting), even in early layers. We also see that both pretraining datasets instill reusable invariances in early-to-mid layers. Further, ImageNet pretraining leads to useful invariances even in later layers, thus indicating why such pretraining is widely used as a recipe for a variety of computer vision tasks.



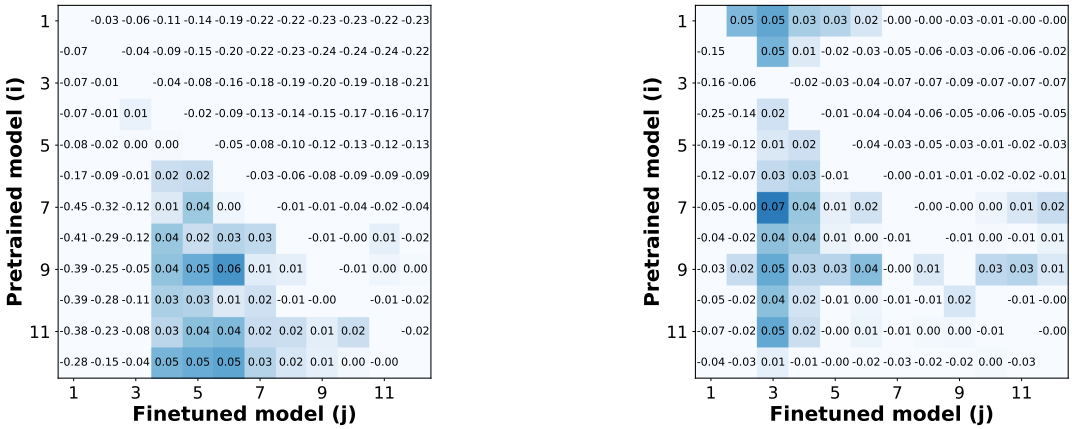
(a) CIFAR10 (b) CIFAR100

Figure 3: **[`cka divergence` does not capture changing nature of invariances]** `cka divergence` across layers for a model finetuned for CIFAR10 (left) and CIFAR100 (right) classification, starting from different pretrained weights. Finetuning a model pretrained on *some* data leads to almost no `cka divergence` in early layers, and only small values in later layers. However, training from scratch leads to a much higher `cka divergence`, even in early layers. Interestingly, contrary to the trend shown for `learning` and `forgetting` in Fig 2, we see that ImageNet pretraining leads to higher divergence than pertaining on the respective CIFAR dataset, as shown by the blue line being higher than orange in both plots.

and `forgetting`, `cka divergence` between pretrained and finetuned model shows higher values when using ImageNet pretrained weights than the corresponding CIFAR pretrained weights. Similar to observations of Nanda et al. (2022), CKA does not capture the nature of changing invariances and thus can give incomplete information about the effects of pretraining.

4.3 Compression and Expansion Analysis

Thus far we have focused on examining how the invariances learned by a specific layer in a pretrained model relate to the corresponding layer in the finetuned model. However, this may not capture all possible effects of finetuning as invariances in some pretrained layers may have migrated to other layers in a finetuned model due to task-dependent requirements. Here, we use the `Invariance Flow Matrix` to examine whether layer-wise invariances in the pretrained model closely correspond to the respective layer in the finetuned model, or whether there are better-matching layers.



(a) Classification Task

(b) Reconstruction Task

Figure 4: Invariance Flow Matrix for the finetuned model on classification and reconstruction task of CIFAR10, pretrained on ImageNet. For the classification task we observe a compression of the pretrained invariances. In particular, layers $6-12$ in the pretrained model correspond more closely to layers $4-8$ in the finetuned model. For the reconstruction task we observe both a compression and expansion of the pretrained invariances. In particular, we observe that layers $6-12$ in the pretrained model correspond more closely to layers $3-6$ in the finetuned model, which indicates compression. In contrast, layers $1-2$ in the pretrained model correspond more closely to layers $2-6$ in the finetuned model, which indicates expansion.

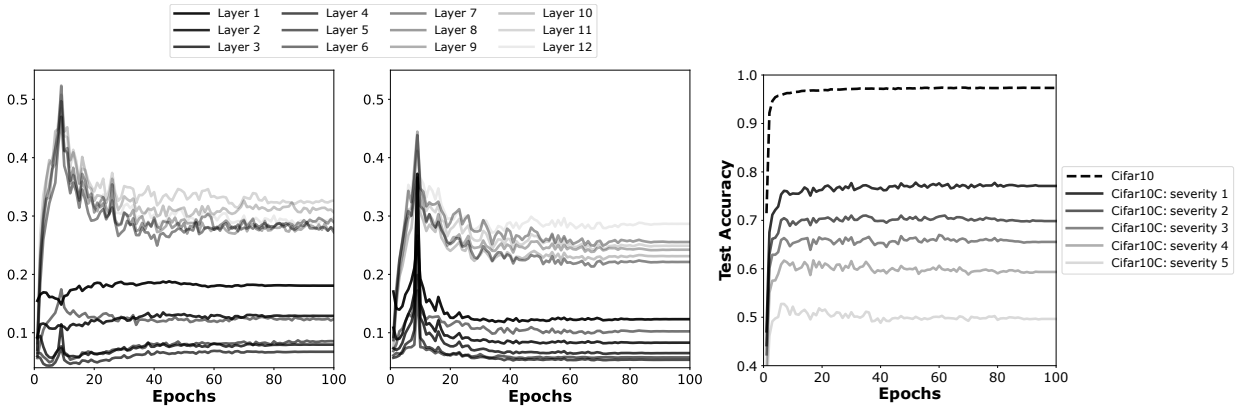
Finetuning compresses invariances. In Figures 4a and 4b, we present the `Invariance Flow Matrix` for the model pretrained on ImageNet and finetuned for classification and reconstruction of CIFAR10 respectively. For both tasks, we observe a compression of the pretrained invariances, visible in the lower left part of the matrix. In particular, layers $6-12$ in the pretrained model correspond more closely to layers $4-8$ in the model finetuned for classification and layers $4-11$ correspond more closely to layers $3-6$ in the model finetuned for reconstruction. This suggests that finetuning compresses some invariances from the pretrained model to earlier layers of the finetuned model. This is possibly a mechanism that allows for more capacity in later layers to support learning new invariances that are needed for the finetuning task and dataset. We observe similar results for the classification and reconstruction of the CIFAR100 dataset (see Appendix Figures 10a and 10b).

Transfering to lower-level tasks expands early-layer invariances. In the upper right triangles of Figures 4a and 4b, we can observe any possible expansion effects of finetuning as discussed in Section 3. Interestingly this effect is only observable when the model is finetuned on the reconstruction CIFAR10/CIFAR100 (see Appendix Figures 10a and 10b for similar CIFAR100 results). This observation is consistent with our expectations. Reconstruction is a lower-level task than classification: to reconstruct an image, more local information may be useful, and usually this kind of information is stored in earlier layers. The `Invariance Flow Matrix` shows that low-level invariances present in earlier layers of the pretrained model are now useful in deeper layers of finetuned model to solve the reconstruction task.

4.4 Forgetting and Learning Dynamics

So far we analyzed the `learning` and `forgetting` metrics across different layers to quantify the difference between the pretrained and the finetuned model. However, a neural network changes gradually during training, becoming more and more accurate on the finetuning task. In this section, we examine the evolution of the proposed metrics during finetuning.

Learning and Forgetting do not increase monotonically. Usually during training the accuracy on the test set increases gradually and the model becomes increasingly capable of performing the task. In Figure 5c we can clearly observe this trend (*i.e.* dotted line). However, analyzing the `learning` and `forgetting` during training and across different layers in Figure 5a and 5b we can observe that the two metrics do not increase monotonically. This is counter-intuitive from a behavioural perspective since one could expect an increasing `learning` and `forgetting` as the model continues to perform better during training. In particular later layers exhibit a peak in earlier epochs. This means that in earlier epochs the model has learned and forgotten invariances, as a result the model diverges more with



(a) `learning` during finetuning (b) `forgetting` during finetuning (c) Accuracies during finetuning

Figure 5: `(a-b) learning` and `forgetting` during finetuning on classification task of CIFAR10 of a model pretrained on ImageNet. We observe a peak of the two metrics in earlier epochs. Only the `learning` of earlier layer does not exhibit a peak. `(c)` Test accuracy during finetuning on classification task of CIFAR10 of a model pretrained on ImageNet. Different lines correspond to different corruption level of the original CIFAR10 test set. While the accuracy on the standard test set increase, the accuracy on corrupted dataset does not always increase, in particular for dataset with high level of corruption(*i.e.* severity 3, 4, 5). In these cases the accuracy has a peak on earlier epochs.

respect to the pretrained model. This observation is confirmed by the `cka divergence` in Appendix Figures 12a, 12b and it is true also for the CIFAR100 dataset (Appendix Figures 11a, 11b, 13,).

Variability in forgetting across layers could reveal more robust models. A question that naturally arises observing the different dynamics of `learning` and `forgetting` with respect to the test accuracy is whether these metrics could reveal additional properties of the model, such as robustness of the model to data corruption. In Figure 5c we report the accuracy values of the model on corrupted CIFAR10 test sets with different levels of corruption. For this purpose, we use the benchmark proposed by Croce et al. (2020). In correspondence with the peak of `learning` and `forgetting`, showed in Figures 5a, 5b, we observe higher robustness of the model in particular on the dataset with higher levels of corruption (*i.e.* severity 3, 4, 5). The hypothesis is that the interaction between learning and forgetting of different layers could be correlated with robustness accuracy. To test whether there is a notable relationship between `learning`/`forgetting` and the robustness accuracy, we compute the Pearson correlation between these metrics across training epochs (details in Appendix E.1). We observe a strong correlation (0.78 as correlation value, p-value 3.52e-18 that passes the Bonferroni correction used to take into account the multiple possible hypotheses) on average across the robustness accuracy on corrupted CIFAR10 test sets and the standard deviation of `forgetting` across layers $2-12$. The correlation is high even for the corrupted CIFAR100 test sets (0.877 as correlation value, p-value 2.51e-16 that passes the Bonferroni correction). This suggests that when `forgetting` varies a lot across layers, there is also more robustness. We repeated this test choosing the best aggregate metrics for both `cka divergence` and *subspace similarity* (Ramasesh et al., 2022), and we have found a weaker correlation between `cka divergence` and robustness accuracy across corrupted CIFAR10 test sets (standard deviation of cka divergence values, layers 1-8, 0.69) and CIFAR100 test sets (0.63) and weaker correlation between *subspace similarity* and robustness accuracy (0.44 CIFAR10c, 0.69 CIFAR100c).

We also note that both the standard deviation of `forgetting` across layers (Figure 5b) and average accuracies on corrupted datasets (Figure 5c) stabilize to constant values towards the end of training. Hence, a valid concern regarding the correlation analysis mentioned earlier is whether the high correlation value stems from the trend of stabilization observed towards the end of the training, rather than the trend of coinciding peaks in both the standard deviation of forgetting and average accuracy values during the initial phase of training. The latter is especially interesting for practitioners seeking to utilize higher values of the standard deviation of `forgetting` as an early stopping indicator for robustness. Therefore, we conduct further experiments to disentangle the contribution of the early epochs by computing the correlation between average accuracies up to an epoch n with the standard deviation of forgetting up to epoch n. Thus, if the correlation is still strong considering only the initial epochs, then this suggests that the higher overall correlation value is not solely attributable to the later epochs. Interestingly, in Figure 14 (Appendix), we observe precisely that, as the correlation up to early epochs is not only comparable but even higher than up to the epochs towards the end of training.

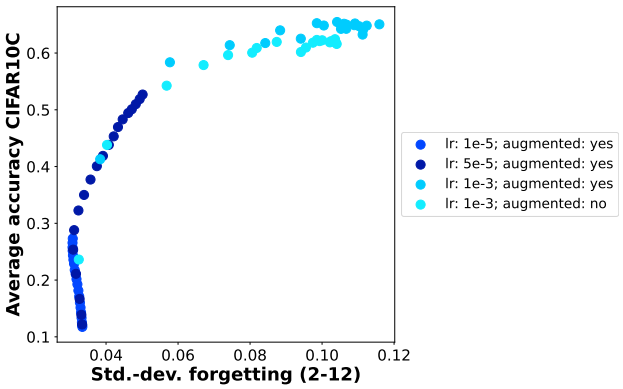


Figure 6: Average accuracy on CIFAR10C datasets and standard deviation of `forgetting` across layers 2-12 of 20 epochs of different models while training on classification of CIFAR10 with sgd optimizer. Higher forgetting variability across layers corresponds to higher accuracy on corrupted datasets.

The analysis of the correlation reported in the previous paragraph was conducted for a single model. To validate our findings beyond this model setting, we observed the corresponding relationship when varying several hyperparameters. We report in Figure 6 the average accuracy on the corrupted CIFAR10 datasets and the standard deviation of `forgetting` across layers 2-12 during the first 20 epochs of models trained on CIFAR10 classification with sgd as optimizer with different hyperparameter settings. The different models have similar trends, even with different learning rates and the usage of augmentations. Interestingly, when using $adam$ as optimizer we did not find a clear trend as with sgd and higher standard deviation of `forgetting` does not coincide with higher accuracy on corrupted datasets (Figure 15). We leave a deeper exploration into the effect of different optimizers on the relationship between model robustness and standard deviation of forgetting across layers for future work.

5 Conclusion

This work examined the relationship between pretrained vision transformers(ViT) and the corresponding finetuned versions on several benchmark datasets and tasks. We presented new metrics that specifically investigate the degree to which invariances learned by a pretrained model are learned or forgotten during finetuning (Section 3). Using these metrics, we presented empirical results on the effect of the finetuning task and the pretraining dataset on the invariances (Section 4.1). We further showed that invariances from deeper pretrained layers are compressed towards shallower layers during finetuning, which may be a mechanism that allows for more capacity in later layers to support learning new invariances that are needed for the finetuning task and dataset (Section 4.3). Analyzing the `learning` and `forgetting` dynamics during finetuning (Section 4.4), we show that they do not increase monotonically as was expected and we revealed a strong correlation between these metrics, and in particular the standard deviation of `forgetting` across layers, and the robustness of the model. This correlation becomes even stronger when fewer epochs are considered, making this measures particularly useful to analyze the robustness of the model during training. Together, these findings contribute to understanding some of the reasons for the successes of pretrained models and the changes that a pretrained model undergoes when finetuned on a downstream task.

The aim of our work was to provide a deeper understanding of what goes on in different layers during finetuning of ViTs. We offered a novel perspective on finetuning by analyzing model changes through the lens of shared invariances. There is already a rich body of ongoing studies that introduce better strategies for finetuning (Kumar et al., 2022; Lee et al., 2022; Evci et al., 2022). All these studies show that features from early layers can be leveraged for better transfer performance. Our work instead aims to shed light on *why* certain approaches work, by showing that early layers tend to learn transferable invariances. Our analysis can inspire future work to design even more effective architectures and finetuning strategies.

Acknowledgments

The authors would like to thank Camila Kolling and Till Speicher for helpful feedback on an earlier version of this manuscript. GM was supported by the CS@max planck graduate center. VN was supported in part by an ERC Advanced Grant "Foundations for Fair Social Computing" (no. 789373), NSF CAREER Award IIS-1846237, NSF D-ISN Award #2039862, NSF Award CCF-1852352, NIH R01 Award NLM013039-01, NIST MSE Award #20126334,

DARPA GARD #HR00112020007, DoD WHS Award #HQ003420F0035, ARPA-E Award #4334192. MT was supported in part by the German Research Foundation (DFG) - DFG Research Unit FOR 5368.

REFERENCES

Francesco Croce, Maksym Andriushchenko, Vikash Sehwag, Edoardo Debenedetti, Nicolas Flammarion, Mung Chiang, Prateek Mittal, and Matthias Hein. Robustbench: a standardized adversarial robustness benchmark. *arXiv preprint arXiv:2010.09670*, 2020.

MohammadReza Davari, Nader Asadi, Sudhir Mudur, Rahaf Aljundi, and Eugene Belilovsky. Probing representation forgetting in supervised and unsupervised continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 16712–16721, 2022.

Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255. Ieee, 2009.

Terrance DeVries and Graham W Taylor. Improved regularization of convolutional neural networks with cutout. *arXiv preprint arXiv:1708.04552*, 2017.

Jeff Donahue, Yangqing Jia, Oriol Vinyals, Judy Hoffman, Ning Zhang, Eric Tzeng, and Trevor Darrell. Decaf: A deep convolutional activation feature for generic visual recognition. In *International conference on machine learning*, pp. 647–655. PMLR, 2014.

Alexey Dosovitskiy, Lucas Beyer, Alexander Kolesnikov, Dirk Weissenborn, Xiaohua Zhai, Thomas Unterthiner, Mostafa Dehghani, Matthias Minderer, Georg Heigold, Sylvain Gelly, Jakob Uszkoreit, and Neil Houlsby. An image is worth 16x16 words: Transformers for image recognition at scale. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=YicbFdNTTy.

Dumitru Erhan, Aaron Courville, Yoshua Bengio, and Pascal Vincent. Why does unsupervised pre-training help deep learning? In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 201–208. JMLR Workshop and Conference Proceedings, 2010.

Utku Evci, Vincent Dumoulin, Hugo Larochelle, and Michael C Mozer. Head2toe: Utilizing intermediate representations for better transfer learning. In *International Conference on Machine Learning*, pp. 6009–6033. PMLR, 2022.

Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 580–587, 2014.

Patrick Helber, Benjamin Bischke, Andreas Dengel, and Damian Borth. Eurosat: A novel dataset and deep learning benchmark for land use and land cover classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, 12(7):2217–2226, 2019.

Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 328–339, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1031. URL https://aclanthology.org/P18-1031.

Minyoung Huh, Pulkit Agrawal, and Alexei A Efros. What makes imagenet good for transfer learning? *arXiv preprint arXiv:1608.08614*, 2016.

Ronald Kemker, Marc McClure, Angelina Abitino, Tyler Hayes, and Christopher Kanan. Measuring catastrophic forgetting in neural networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32, 2018.

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

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pp. 3519–3529. PMLR, 2019a.

Simon Kornblith, Jonathon Shlens, and Quoc V Le. Do better imagenet models transfer better? In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 2661–2671, 2019b.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL http://www.cs.toronto.edu/~kriz/cifar.html.

Ananya Kumar, Aditi Raghunathan, Robbie Jones, Tengyu Ma, and Percy Liang. Fine-tuning can distort pretrained features and underperform out-of-distribution. In *International Conference on Learning Representations*, 2022.

Yoonho Lee, Annie S Chen, Fahim Tajwar, Ananya Kumar, Huaxiu Yao, Percy Liang, and Chelsea Finn. Surgical fine-tuning improves adaptation to distribution shifts. *arXiv preprint arXiv:2210.11466*, 2022.

Timothée Lesort, Vincenzo Lomonaco, Andrei Stoian, Davide Maltoni, David Filliat, and Natalia Díaz-Rodríguez. Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges. *Information fusion*, 58:52–68, 2020.

Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3431–3440, 2015.

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

Vedant Nanda, Till Speicher, Camila Kolling, John P Dickerson, Krishna Gummadi, and Adrian Weller. Measuring representational robustness of neural networks through shared invariances. In *International Conference on Machine Learning*, pp. 16368–16382. PMLR, 2022.

Thao Nguyen, Maithra Raghu, and Simon Kornblith. Do wide and deep networks learn the same things? uncovering how neural network representations vary with width and depth. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=KJNcAkY8tY4.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, and C. V. Jawahar. Cats and dogs. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pp. 3498–3505, 2012. doi: 10.1109/CVPR.2012.6248092.

Jason Phang, Haokun Liu, and Samuel R. Bowman. Fine-tuned transformers show clusters of similar representations across layers. In *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, pp. 529–538, Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.blackboxnlp-1.42. URL https://aclanthology.org/2021.blackboxnlp-1.42.

Maithra Raghu, Thomas Unterthiner, Simon Kornblith, Chiyuan Zhang, and Alexey Dosovitskiy. Do vision transformers see like convolutional neural networks? *Advances in Neural Information Processing Systems*, 34:12116–12128, 2021.

Vinay Venkatesh Ramasesh, Ethan Dyer, and Maithra Raghu. Anatomy of catastrophic forgetting: Hidden representations and task semantics. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021. URL https://openreview.net/forum?id=LhY8QdUGSuw.

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

Hadi Salman, Andrew Ilyas, Logan Engstrom, Ashish Kapoor, and Aleksander Madry. Do adversarially robust imagenet models transfer better? *Advances in Neural Information Processing Systems*, 33:3533–3545, 2020.

Mariya Toneva, Alessandro Sordoni, Remi Tachet des Combes, Adam Trischler, Yoshua Bengio, and Geoffrey J Gordon. An empirical study of example forgetting during deep neural network learning. In *International Conference on Learning Representations*, 2018.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Ross Wightman. Pytorch image models. https://github.com/rwightman/pytorch-image-models, 2019.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pp. 38–45, 2020.

John Wu, Yonatan Belinkov, Hassan Sajjad, Nadir Durrani, Fahim Dalvi, and James Glass. Similarity analysis of contextual word representation models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4638–4655, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.422. URL https://aclanthology.org/2020.acl-main.422.

Sangdoo Yun, Dongyoon Han, Seong Joon Oh, Sanghyuk Chun, Junsuk Choe, and Youngjoon Yoo. Cutmix: Regularization strategy to train strong classifiers with localizable features. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 6023–6032, 2019.

Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pp. 818–833. Springer, 2014.

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.

Hongyi Zhang, Moustapha Cissé, Yann N. Dauphin, and David Lopez-Paz. mixup: Beyond empirical risk minimization. In *6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings*. OpenReview.net, 2018. URL https://openreview.net/forum?id=r1Ddp1-Rb.

A Appendix

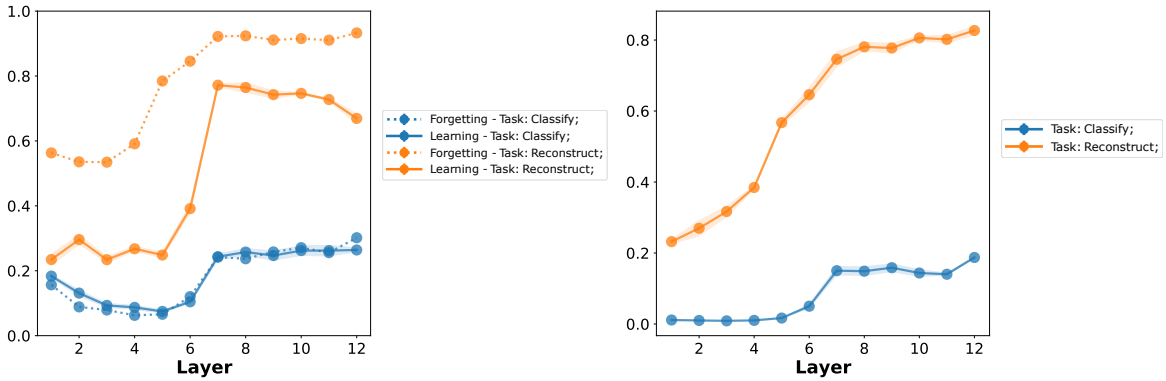
B Training details

For our experiment we use a ViT model with patch size of 32x32 and image resolution at 224x224. The pretrained model on ImageNet is provided by Wightman (2019). We train the other models for 100 epochs with $learning rate = 0.001$, $momentum = 0.9$ and $weight\ decay = 0.0001$. We use a cosine scheduler for the learning rate and the Stochastic Gradient Descent as optimizer. We use the `transformers` library from Hugging FaceWolf et al. (2020) to train the model and log training results. In table 1 we report the accuracy values on the test set for the model we use.

Table 1: Accuracy values ViT models

Model	Accuracy	Pretraining Accuracy
Pretrain ImageNet; finetune CIFAR10	0.97	0.51
Pretrain ImageNet; finetune CIFAR100	0.86	0.51
Pretrain CIFAR100; finetune CIFAR10	0.99	0.92
CIFAR10 from scratch	0.98	–
Pretrain CIFAR10; finetune CIFAR100	0.91	0.98
CIFAR100 from scratch	0.92	–
Pretrain ImageNet; finetune Eurosat	0.97	0.51
Pretrain ImageNet; finetune OxfordIIIT Pet	0.80	0.51

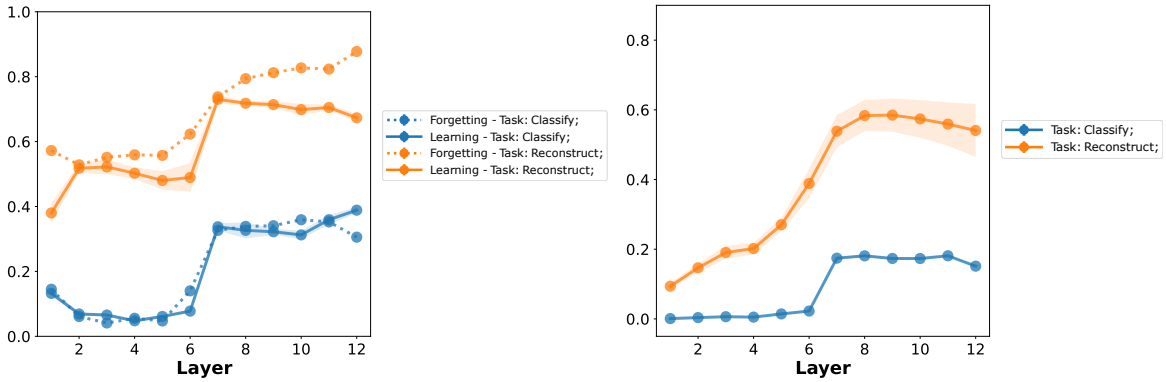
C EFFECT OF FINETUNING/PRETRAINING TASKS: ADDITIONAL RESULTS



(a) Learning and forgetting CIFAR100

(b) CKA divergence CIFAR100

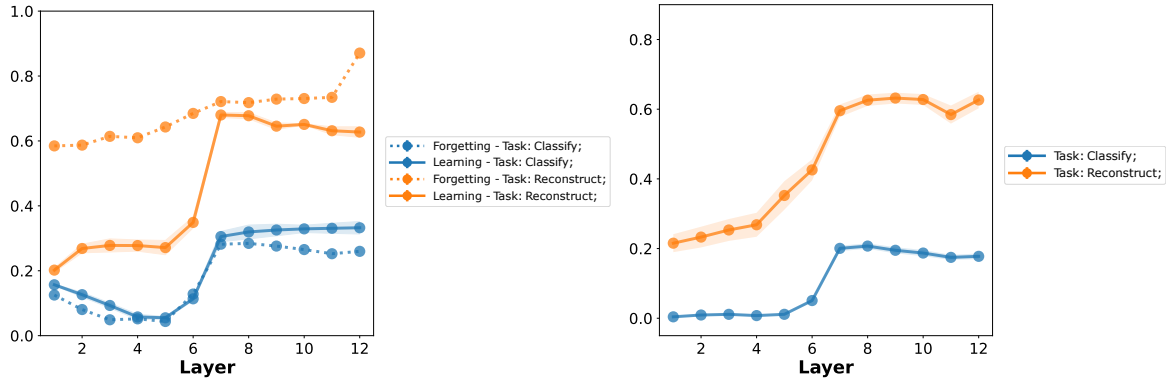
Figure 7: `learning`, `forgetting` and `cka divergence` values for ViT model pretrained on ImageNet and finetuned on CIFAR100 on reconstruction and classification task.



(a) Learning and forgetting EuroSAT

(b) CKA divergence EuroSAT

Figure 8: `learning`, `forgetting` and `cka divergence` values for ViT model pretrained on ImageNet and finetuned on EuroSAT on reconstruction and classification task.



(a) Learning and forgetting Oxford-IIIT Pet

(b) CKA divergence Oxford-IIIT Pet

Figure 9: `learning`, `forgetting` and `cka divergence` values for ViT model pretrained on ImageNet and finetuned on Oxford-IIT Pet dataset on reconstruction and classification task.

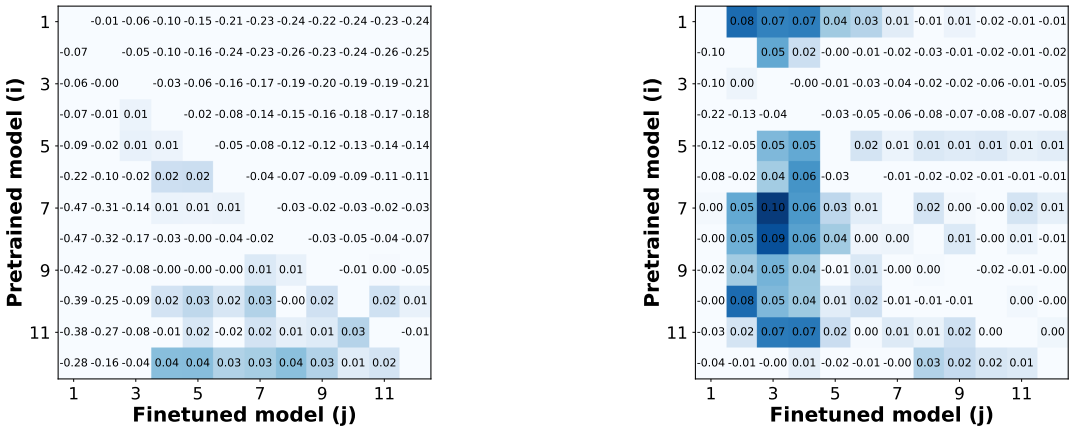
C.1 Correlation between learning and forgetting

In Figures 1a,7a,2a,2b we reported `learning`, `forgetting` metrics for different settings. Even if they may show different trends, they are significantly correlated.

- Imagenet → CIFAR100 classification: 0.97
- Imagenet → CIFAR10 classification: 0.97
- Imagenet → CIFAR100 reconstruction: 0.88
- Imagenet → CIFAR10 reconstruction: 0.86
- CIFAR100 → CIFAR10 classification: 0.98
- CIFAR10 → CIFAR100 classification: 0.99
- Random → CIFAR10 classification: 0.83
- Random → CIFAR100 classification: 0.84

The correlation however decreses for the recontruction tasks and for the models trained from scratch. The two measure therefore can be different even if they are correlated, and it is important to take into consideration both of them in future analysis.

D INVARIANCE FLOW MATRIX CIFAR100

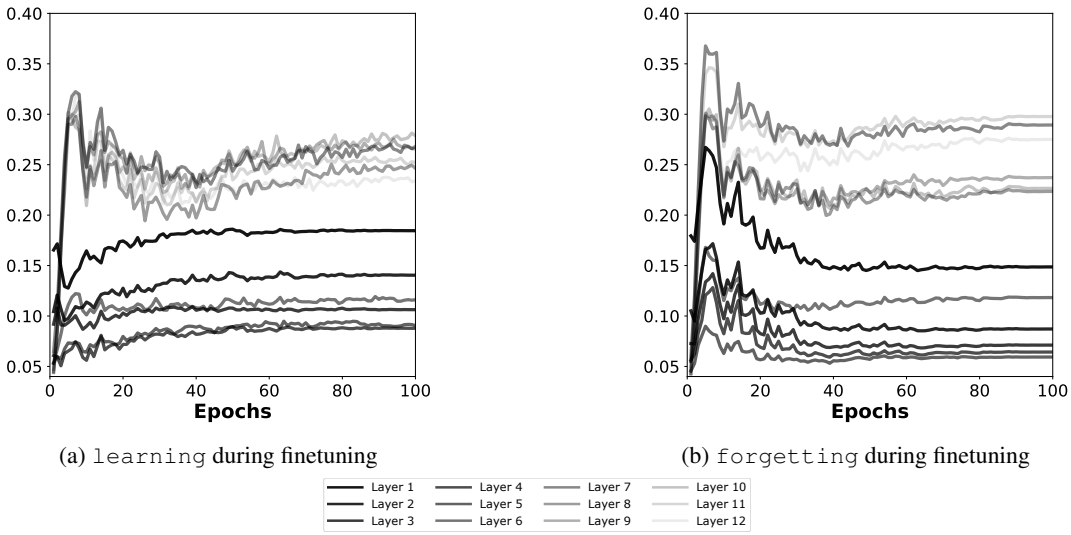


(a) Classification Task

(b) Reconstruction Task

Figure 10: Invariance Flow Matrix for finetuned model on classification/reconstruction task of CIFAR100, pretrained on ImageNet.

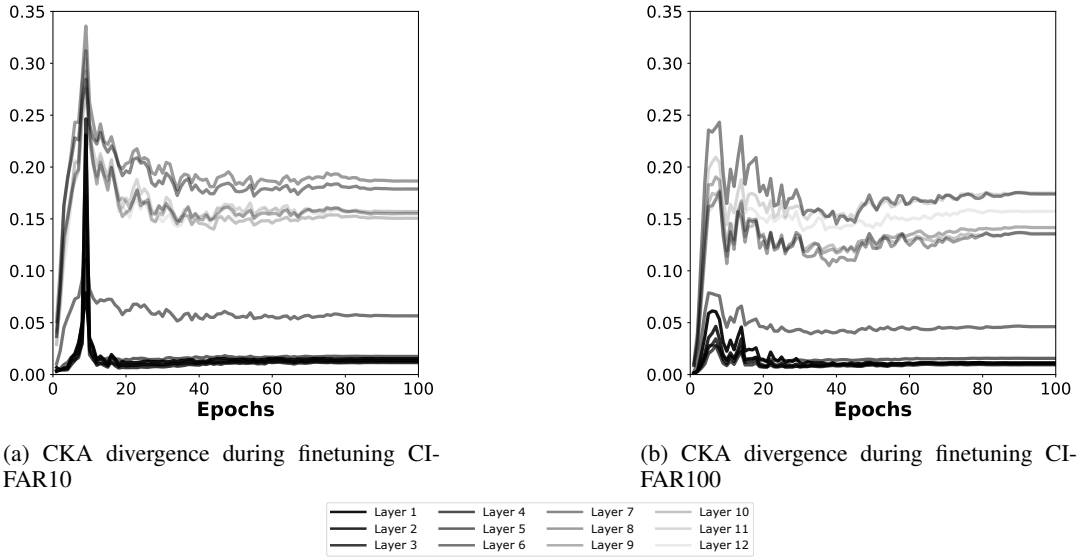
E LEARNING AND FORGETTING DYNAMICS ADDITIONAL RESULTS



(a) `learning` during finetuning

(b) `forgetting` during finetuning

Figure 11: `learning` and `forgetting` during finetuning on classification task of CIFAR100 of a model pretrained on ImageNet. We observe a peak of the two metrics in earlier epochs.Only the `learning` of earlier layer does not exhibit a peak.



(a) CKA divergence during finetuning CIFAR10

(b) CKA divergence during finetuning CIFAR100

Figure 12: `cka divergence` during finetuning on classification task of CIFAR10 and CIFAR100 of a model pretrained on ImageNet. We observe a peak in earlier epochs.

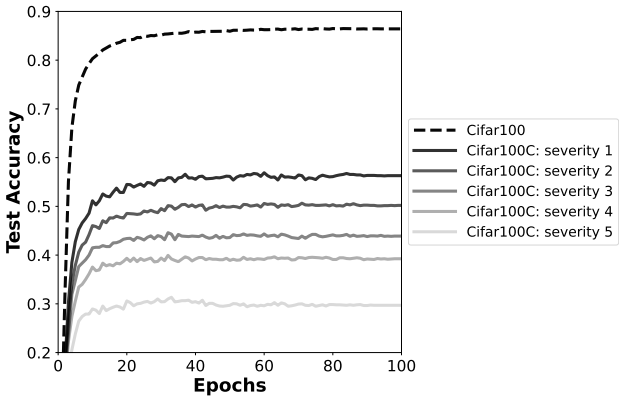


Figure 13: Test accuracy during finetuning on classification task of CIFAR100 of a model pretrained on ImageNet. Different lines correspond to different corruption level of the original CIFAR10 test set. While the accuracy on the standard test set increase, the accuracy on corrupted dataset does not always increase, in particular for dataset with high level of corruption(*i.e.* severity 3, 4, 5). In these cases the accuracy has a peak on earlier epochs.

E.1 Correlation with robustness

For this experiment we explored 352 hypotesis using aggregate metrics. We varied:

- Layer considered: only one layer, first n layers or last n layers.
- `learning` and `forgetting` operation: addition, subtraction, only `learning` or only `forgetting`
- Aggregate layers operation: mean, standard deviation, minimum or maximum.

Similarly to `learning` or only `forgetting` for the `cka divergence` we vearied the layers considered and the aggregate layers operations for a total number of 88 combinations To test the accuracy of the model we use the standard test set of CIFAR10 and CIFAR100, and we use 1000 randomly sampled inputs for each level of corruption.

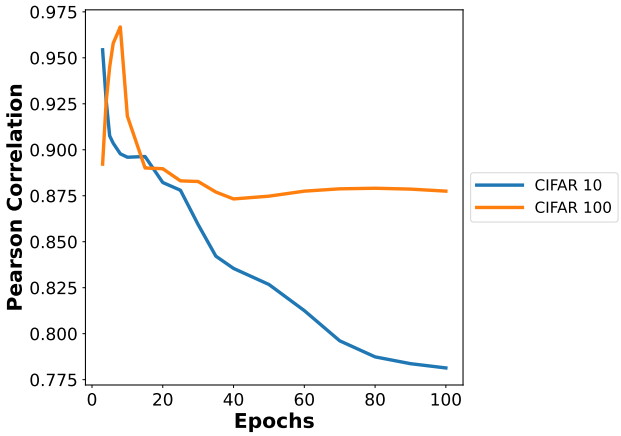


Figure 14: Correlation between average accuracy on CIFAR10C/CIFAR100C datasets and standard deviation of `forgetting` across layers 2-12 during training. Each point on the graph represents the correlation between the average of the accuracies on the CIFAR10C/CIFAR100C datasets and the standard deviation of `forgetting` across layers 2-12 considered up to the epoch indicated on the x-axis. For example at epoch n the correlation is computed between $[avg\_acc\_epoch\_0, avg\_acc\_epoch\_1, ..., avg\_acc\_epoch\_n]$ and $[std\_forgetting\_epoch\_0, std\_forgetting\_epoch\_1, ..., std\_forgetting\_epoch\_n]$.

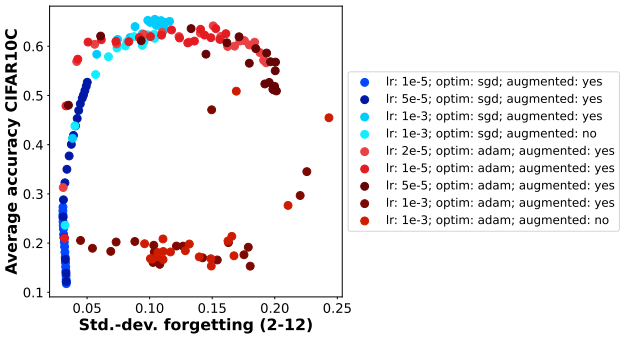


Figure 15: Average accuracy on Cifar10C datasets and standard deviation of `forgetting` across layers 2-12 of 20 epochs of different models while training on classification of CIFAR10 with sgd and $adam$ optimizers.