How to choose the right transfer learning protocol? A qualitative analysis in a controlled set-up

Federica Gerace *federica.gerace@unibo.it*
Scuola Internazionale Superiore di Studi Avanzati (SISSA)
Department of Mathematics, University of Bologna

Diego Doimo *diego.doimo@areasciencepark.it*
Area Science Park

Stefano Sarao Mannelli *s.saraomannelli@ucl.ac.uk*
Gatsby Computational Neuroscience Unit & Sainsbury Wellcome Centre
University College London

Luca Saglietti *luca.saglietti@unibocconi.it*
Department of Computing Sciences
Bocconi University

Alessandro Laio *laio@sissa.it*
Scuola Internazionale Superiore di Studi Avanzati (SISSA)
The Abdus Salam International Centre for Theoretical Physics (ICTP)

Reviewed on OpenReview: *https://openreview.net/forum?id=XWQgXLYwv2*

Abstract

Transfer learning is a powerful technique that enables model training with limited amounts of data, making it crucial in many data-scarce real-world applications. Typically, transfer learning protocols require first to transfer all the feature-extractor layers of a network pre-trained on a data-rich source task, and then to adapt only the task-specific readout layers to a data-poor target task. This workflow is based on two main assumptions: first, the feature maps of the pre-trained model are qualitatively similar to the ones that would have been learned with enough data on the target task; second, the source representations of the last hidden layers are always the most expressive. In this work, we demonstrate that this is not always the case and that the largest performance gain may be achieved when smaller portions of the pre-trained network are transferred. In particular, we perform a set of numerical experiments in a controlled setting, showing how the optimal transfer depth depends non-trivially on the amount of available training data and on the degree of source-target task similarity, and it is often convenient to transfer only the first layers. We then propose a strategy to detect the most promising source task among the available candidates. This approach compares the internal representations of a network trained entirely from scratch on the target task with those of the networks pre-trained on the potential source tasks.

1 Introduction

Machine learning models show a remarkable capacity to extrapolate rules and predict the behavior of complex systems. Still, this ability often comes at the cost of training with large amounts of data. In various domains – such as in medical applications – data collection is a slow and costly process. This goes at odds with the typical vast variability across data, which inevitably requires the usage of huge datasets to achieve satisfying generalization performance on a desired task Beam & Kohane (2018); Rajkomar et al. (2019). Data efficiency is thus a necessary condition.

Transfer learning emerged as an efficacious mitigation strategy to this problem Thrun & Pratt (2012); Shin et al. (2016); Raghu et al. (2019). Transferring the already meaningful representations learned on the source task to a network that has to solve a given target task allows it to work with dramatically smaller dataset sizes while keeping a comparable level of accuracy. With little data at disposal, one of the major risks is to incur overfitting Geirhos et al. (2020). However, transferring the layers directly from another network allows, in principle, to filter out irrelevant information present in the input of both the source and the target data sets and work with a representation of the data of reduced efficacious dimension. This allows a significant increase in data efficiency.

Unfortunately, deep neural networks tend to be sensitive even to tiny distribution shifts Gama et al. (2014). Therefore, it is essential to perform transfer learning starting from the most suitable pre-trained network or, in other words, from the most convenient parameter initialization. This becomes even more relevant when the transfer learning pipeline is followed by a further stage of fine-tuning, during which the transferred feature map is unfrozen, and the whole network is trained on the target set Bengio (2012). This step is performed under the assumption that the features are already appropriate, and one can use smaller learning rates to avoid overfitting.

Fig. 1 embodies the main motivation of this work: it shows that in a realistic data scenario transferring all the layers up to the last hidden representation does not always lead to the best performance. In particular, the two panels show the optimal test accuracy as a function of the transfer depth, namely the total number of frozen layers, when the source task is Imagenet and the target task is either a dataset of Breast Cancer images (panel (a)) or the Retinal Fundus Multi-Disease Image dataset (panel (b)). As we can see, while in the first case the optimal accuracy can be achieved by keeping almost all the layers frozen, in the second case it is instead necessary to retrain larger portions of the network. In the following, we will refer to these curves as *defrosting profiles*.

Since transfer learning can be so sensitive to parameter initialization, it is then natural to ask: *How could we detect which is the source pre-trained network whose features are the most promising to transfer?, how should one choose the layers that should be kept frozen? In other words, is it more convenient to transfer all the layers or just a portion of them?* Ideally, one should be able to identify, in the source network, a layer in which the representation is general enough to be meaningful also for the target task but not too general, as otherwise training it to a specialized task might be hindered by the scarcity of data in the target set.

In this work, we start addressing these questions with the following main contributions:

- We design a controlled experimental framework through which we show that the optimal transfer depth strongly depends on the source-target distribution shift and on the size of the target set;
- We propose a simple protocol for the identification of the optimal transfer depth (see sec. 3.1.1);
- We propose a strategy to detect the most promising source task by measuring the topological similarity between the internal representations of a network pre-trained on a candidate source task with a second one trained fully from scratch on the target set.

In particular, in sec. 3.1, we describe the experimental framework leading to the *defrosting profiles* as the ones displayed in Fig. 1. In sec. 3.1.1, we see how this framework can also be considered as a protocol for identifying the optimal transfer depth, which we call *incremental defrosting*. As we will see, one of the key advantages of this protocol is its simplicity. Indeed, no learning heuristics need to be employed to ensure the retention of useful information obtained in the pre-training phase. In sec. 3.2 and sec. 3.3, we also show how the shape of the resulting defrosting profiles is strongly affected by the amount of data in the target training set but also on the specific choice of the source task respectively. Given that the degree of source-target similarity can consistently impact transfer learning performances, up to the point of inducing negative transfer effects Gerace et al. (2022), in sec. 3.4, we also propose a strategy to select the most promising source task among all the possible candidate ones.

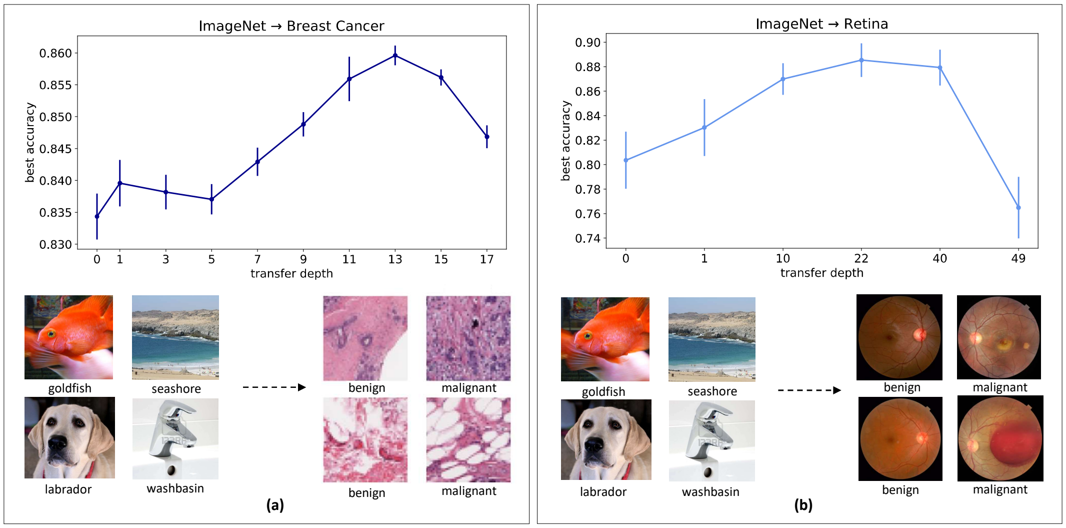

Figure 1: **Diverse effectiveness of freezing protocols** The figure shows two instances of transfer learning tasks: from Imagenet Deng et al. (2009) to a dataset composed of Breast Cancer images Janowczyk & Madabhushi (2016); Cruz-Roa et al. (2014) (panel (a)) and from ImageNet to the Retinal Fundus Multi-Disease Image Dataset Pachade et al. (2021) (panel (b)) (see SM for further details concerning the two datasets). In the first transfer experiment, we used a ResNet-18; in the second one, we used a ResNet-50. More details can be found in the Architecture and training protocol paragraph. We can see samples of each dataset in the lower part of the panels, while the curves above – the *defrosting profiles* – represent the test performance of the network. Each point of the curve is the accuracy reached when the first l layers have been transferred. The simulations are averaged over 20 (leftmost panel) and 5 (rigthmost panel) different realizations.

1.1 Related Works

Deep neural networks (DNNs) can build surprisingly general representations in various ways Deng et al. (2009); Solorio-Fernández et al. (2020); Jaiswal et al. (2020). A possible explanation for their effectiveness comes from their "implicit bias" towards regularized solutions Neyshabur et al. (2014). Numerous works provided evidence of a preference for *simple solutions* Rahaman et al. (2018); Pérez et al. (2019); Shah et al. (2020); Abbe et al. (2021), and it was shown that learning complex functions may require training the networks for longer times Goldt & Ingrosso (2022); Refinetti et al. (2023).

The layered nature of DNN architectures seems to play a crucial role, as it appears that increasingly complex features are progressively encoded in deeper layers of the network Kornblith et al. (2019); Abbe et al. (2021). Interestingly, the interaction between input data and labels has been shown to lead to non-trivial and often abrupt changes in the nature of the extracted features throughout the layers Tishby & Zaslavsky (2015); Doimo et al. (2020). Recent investigations highlighted the importance of intermediate representations Yosinski et al. (2014), but the question on how to select transferable DNN representations Raina et al. (2007); Long et al. (2015) remains largely open, given the complex dependence on the specific source-target distribution shifts Wenzel et al. (2022) and on the data scarcity Lee et al. (2022). The theoretical insight on transfer learning phenomenology in DNNs remains limited. Except for some results on training convergence rates Du et al. (2020), the simplifying assumption of linear activation function was necessary to obtain exact results on the generalization properties after a transfer, both in the asymptotic Dhifallah & Lu (2021); Dar & Baraniuk (2022) and in the dynamical Lampinen & Ganguli (2019) regimes. Currently, a systematic exploration of the generalization gain as a function of the source-target distribution shift in non-linear neural networks is only possible in the case of fully connected 2-layer networks Gerace et al. (2022).

The present work explores the less charted area between theory and real-world transfer learning applications through a carefully designed framework that allows some degree of control over the data distributions. The main goal is to investigate the connection between data and architecture, pinpointing the connection between distribution shift and sample size on one side and the quality of intermediate representations for a transfer task on the other. In particular, we show that, depending on these two factors, it is possible to identify an optimal number of layers to transfer. We provide evidence that this preliminary step can provide a better initial condition for subsequent fine-tuning stages, as shown in sec. 3.1. Our findings are complementary to what is already known about fine-tuning. For instance, it has been shown that fine-tuning the whole architecture seems to be beneficial only in the data-rich regime Wenzel et al. (2022). Similar observations can be made in the context of differentially private models Tobaben et al. (2023), where the authors notice how FiLM adapters are often competitive with learning just the final classifier layers or the entire network. Therefore, it is fundamental to understand whether the whole transferred information is always needed. For instance, from the perspective of knowledge distillation in a self-supervised learning context, it has been shown that distilled models are typically capable of surpassing the best available general purposed features of larger models in computer vision tasks Oquab et al. (2023).

In this work, we corroborate these results by investigating the case where either the whole network or just the head is adapted on the target set, and when a non-trivial number of layers is kept frozen to the source configurations. We believe this can provide a better starting point for later fine-tuning and knowledge distillation protocols. We conduct part of the experiments while having full control of the moments of the input data distribution shared between the source and target tasks. This allows us to directly relate the statistics of the input data distribution to the effectiveness of Transfer Learning protocols.

2 The Experimental setup

Architecture and training protocol We consider four different types of architectures. The first and mostly employed in our experiments is a Wide-ResNet-28-4 architecture Zagoruyko & Komodakis (2016), organized in 3 identical groups, each containing 4 blocks of alternating convolutional, batch-norm, and downsampling layers, plus a final batch norm and a fully connected one. This architecture is pretty simple but still reaches accuracies in image classifications close to the state-of-the-art Cubuk et al. (2020); Han et al. (2017). Therefore, it can be considered a good starting point for a generic image classification task. Moreover, wide versions of ResNet have been shown to foster a good degree of compatibility between model representations, whose corresponding similarity was demonstrated to increase with the width of the layers Kornblith et al. (2019). For the other tests, we also use a base version of the Vision Transformer Dosovitskiy et al. (2020) with an input patch size of 16, a ResNet-18 and a ResNet-50 from the PyTorch model zoo, all pre-trained on the ImageNet dataset He et al. (2016).

We train the Wide-ResNet-28-4 and the ResNet-18 for 200 epochs with stochastic gradient descent with momentum at 0.9, batch size of 128, weight decay at $5 \cdot 10^{-4}$, a cosine annealing scheduler for the learning rate, starting from a value of 0.1 Zagoruyko & Komodakis (2016); Loshchilov & Hutter (2016) and no data augmentation. Concerning the experiments involving the ResNet-50 architecture, we instead use the same learning protocol as in Dominik Müller & Kramer (2021). We train the ViT for 50 epochs, with a batch size of 64, learning rate 10^{-4}, weight decay of 10^{-4}, and Adam optimizer Kingma & Ba (2014). We provide the code to reproduce our experiments and analysis at `https://anonymous.4open.science/r/TL_gradient-DD27/`. For additional details, see supplementary material (SM) B.1.

Synthetic datasets To investigate the factors at play in determining the transfer learning behaviors in-depth, we consider three types of CIFAR-10 clones. These clones are meant to form a hierarchical family of datasets, approximating the true underlying distribution of CIFAR-10 with increasing fidelity.

Following Refinetti et al. (2023), the first level of the hierarchy (IsoGM) is obtained by fitting CIFAR-10 with a mixture of isotropic Gaussians and matching the first moments with the CIFAR-10 distribution. At the second level of the hierarchy (GM), also the covariances are learned from CIFAR-10. To go beyond the second moment, we propose to obtain finer approximations of CIFAR-10 images by using a deep autoencoder architecture consisting of a convolutional encoder network and a deconvolutional decoder, both connected to a bottleneck layer of variable size (we refer to SM for additional details). Note that, by construction, this setup is devoid of the confounding effect of misalignments between labeling rules Gerace et al. (2022);

Lee et al. (2022) in the source and target tasks since the synthetic datasets share the same label structure, independently of the alterations induced in the input distributions.

Real-world datasets We analyze the defrosting profiles for transfer and fine-tuning setups for three real-world datasets for image classification: the Food101 dataset Bossard et al. (2014), a dataset of Retinal Images Pachade et al. (2021), and a Breast Cancer dataset Janowczyk & Madabhushi (2016); Cruz-Roa et al. (2014). Food101 comprises 101 classes of different types of foods; the Retinal Dataset has 46 different types of retinal pathologies, but we treat the task as a binary classification of the presence and absence of a disease. Similarly also, in the Breast Cancer dataset, the task is a binary classification task. In all cases, we report the accuracies of the defrosting profiles on the test sets. More information about the dataset construction can be found in Sec. B.1 of the Appendix.

Information imbalance One of the main factors affecting whether a layer can be effectively transferred to a target network is the similarity between the source and target feature maps. To measure this quantity, we use the *Information Imbalance* (II), a topological measure recently proposed in Glielmo et al. (2022), and estimate if a given feature map can be used to predict another feature map. To compute the II, we first take both a pre-trained network on a given source task and a network trained entirely from scratch on the target task. We then make the two networks process the target test set by extracting the corresponding internal representations at each layer of both networks and for each test example. Since feature maps are typically very high-dimensional, estimating the conditional entropy between the two representations would be impractical. The II can be seen as a computationally viable proxy for entropy and other information-theoretical quantities.

The II is estimated as follows. Given a layer l and a test data point μ, we first find the closest data point μ' according to their Euclidean distance in the source representation. Then, in the target representation of the same data set, we count the number of data points that are closer to μ than μ'. The II is proportional to the average of this number. Formally

$$\Delta_l\left(S \to T\right) = \frac{2}{M} \langle r^T | r^S = 1 \rangle$$

with M being the total number of samples in the test set, r^T and r^S being the rank matrices associated with the representations in target and source networks respectively (therefore $r^S_{\mu,\mu'}$ is equal to 1 if μ and μ' are first neighbors). The average $\langle \cdot \rangle$ is taken over all the data points in the target test set. The II is, therefore, a measure of the local similarity between the internal representations space of the source and target task. In particular, if two local neighborhoods are approximately equivalent, one finds $\Delta_l \sim 0$. In general, the more the first representation is predictive with respect to the second, the smaller the II is. In the limit where the source local neighborhood does not provide any information on the target one, one will find $\Delta_l = 1$ Glielmo et al. (2022). Note that, crucially, the II circumvents all the known ambiguities caused by re-scaling, permutations, and symmetries in neural network functions since this measure relies solely on the distance ranks, which by definition are invariant to the transformations implemented by neural networks.

3 Results

3.1 Experimental framework to identify the optimal transfer depth detection

To identify the optimal transfer depth for a given source-target pair, we perform a series of transfer learning cycles, as exemplified in the cartoon of Fig. 2. In particular, we start from a completely frozen setting, except the output layer. We then gradually increase the number of layers to be defrosted and re-trained, starting from the very last layer up to the early layers in the network. At each iteration of this procedure, to avoid vanishing gradient scenarios, we re-initialize the defrost layers to random values (following typical heuristics He et al. (2015)), and we then train them from scratch on the target training data, with the hyper-parameter setting described in the architecture and training protocol paragraph. The *defrosting profiles* in Fig. 2, show the validation accuracy recorded at each iteration of this procedure, starting from the transfer

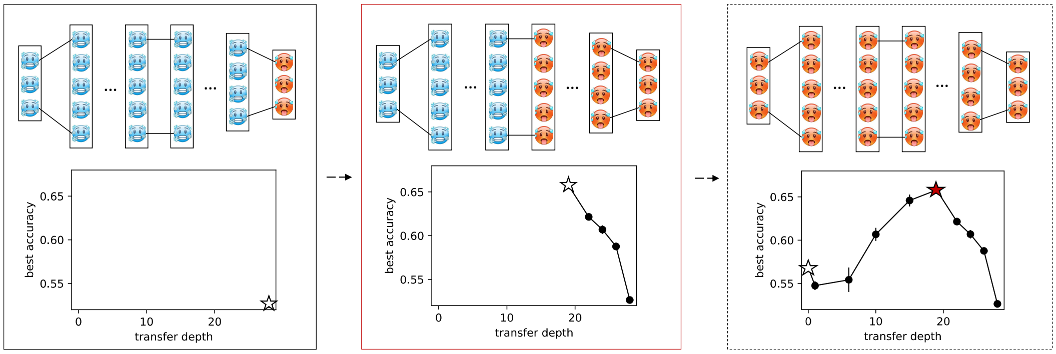

Figure 2: **Incremental defrosting procedure.** The cartoon illustrates the transfer procedure employed in the experiments, where the first layers are transferred from the architecture trained on the source, and only the last layers are fine-tuned to the target set. On the right, with a star, we show the accuracy reached on the target set by a network trained up to the kth layer.

scenario where all layers are kept frozen except the final read-out (leftmost panel), up to the transfer setting where the whole network is let to adapt to the target task (rightmost panel).

While it is common practice to transfer all feature-extractor layers except for the fully-connected segment (or even just the readout layer), in many scenarios transferring a smaller number of layers may largely improve the downstream accuracy. This is, for instance, the case of the example shown in Fig. 2, where we highlight the optimal transfer depth with a red star. As we can see, in this transfer setting where the source task is GM while the target task is CIFAR-10, it is neither convenient to transfer all the feature-extractor layers nor to train the entire network entirely from scratch.

As shown in the left panel of Fig. 3, this behavior is observed even with a Transformer. The plot shows the defrosting profile of a Vision Transformer pre-trained on Imagenet and then adapted to the Retinal Fundus Multi-Disease Image Dataset (Pachade et al. (2021), top) and the Food101 dataset (Bossard et al. (2014), bottom). As for convolutional neural networks, even in this case, training from scratch or transferring all the blocks is not the optimal transfer learning strategy. The optimal performance is obtained by keeping the first 7-9 layers frozen.

There are at least two reasons for the performance deterioration when a sub-optimal number of layers is frozen. On the one hand, one can see a defrosting profile that decreases when too many layers are kept frozen. This is natural when the input/output mappings associated with the two tasks are too dissimilar: a trained network will not retain all the information contained in the data points as it learns to select the meaningful features for classification. Some information that is relevant to the target task might be dropped completely by the pre-trained model, given the presence of non-linearities and down-sampling operations throughout the layers. On the other hand, when the target dataset is small, one can see a performance deterioration if too many layers are defrosted and re-trained on the new data. If the ratio of training samples and tunable parameters becomes too small, one can trace over-fitting behaviors, as the model can not choose a good generalizing solution among the vast number of zero training error models.

The effect of fine-tuning. Identifying the optimal number of layers to transfer can be beneficial for later fine-tuning stages. This is shown in Fig. 3. In particular, we display the test accuracy achieved by a Vision Transformer on the Retina (top) and on the Food101 (bottom) datasets when an increasing number of layers is kept frozen to the ImageNet feature map, while the remaining ones are re-trained on the target set starting from random initialization (transfer only on the left) or fine-tuning the source feature-map (fine-tuning on the right). In both cases, fine-tuning drastically improves the generalization performances at early frozen layers if we compare the resulting defrosting profile with the one obtained at random initialization. However, we can still identify for both datasets an optimal transfer depth in the same region of the optimal depth detected at random initialization (layer 6 for Retina and layer 7 for Food101). At the optimal transfer depth,

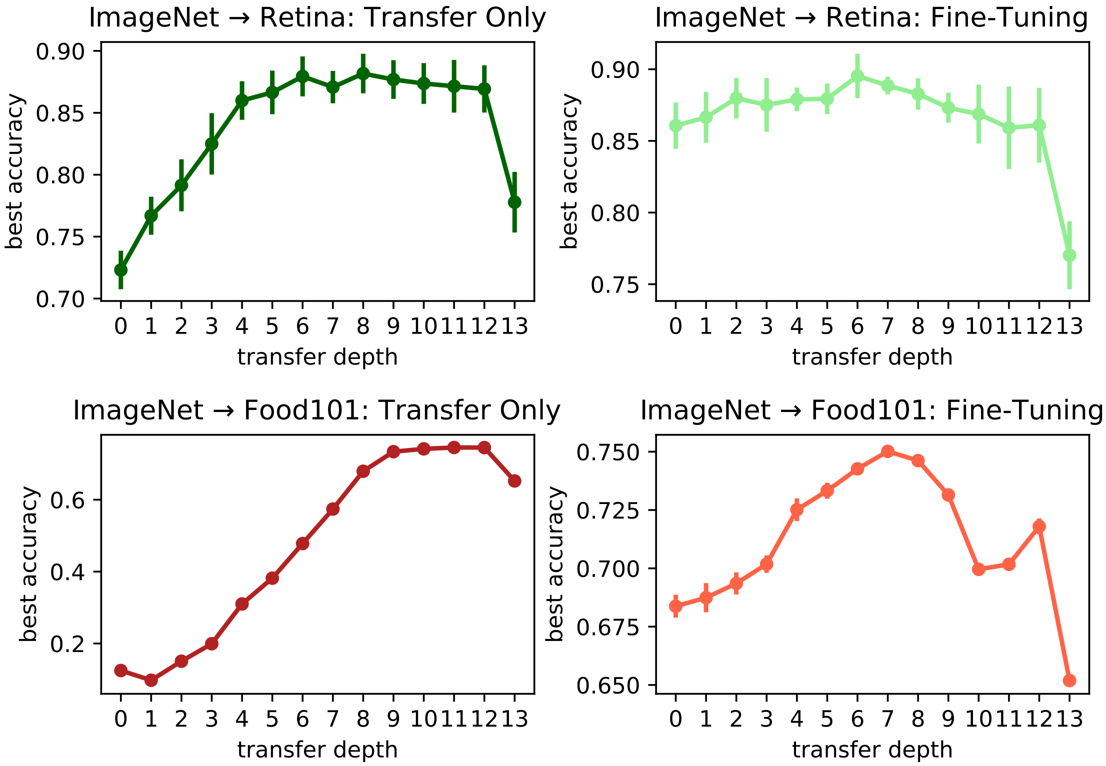

Figure 3: **Optimal transfer depth and fine-tuning.** The figure shows a Vision Transformer pre-trained on ImageNet and then adapted to the Retinal Fundus Multi-Disease Image Dataset (top) and Food101 dataset (bottom). Each point of the curve is the validation accuracy reached when the first l layers have been frozen, and the remaining ones are re-trained from a random initial condition (leftmost panel), fine-tuning from the ImageNet configurations (middle panel) In these experiments, we use a *vit_b_16* architecture from Dosovitskiy et al. (2020). More details can be found in the Architecture and training protocol paragraph. The curves – the *defrosting profiles* – represent the validation performance of the network. The simulations are averaged over 3 different realizations.

the accuracy achieved is higher than the one obtained starting from random initialization, meaning that the detection of the optimal number of layer to transfer, combined with a subsequent fine-tuning stage, allows to achieve better test scores. For instance, we can see that fine-tuning at the optimal transfer depth improves the generalization performances that one would achieve by directly fine-tuning all the layers (first point on each fine-tuning defrosting profiles), as standard practice in deep learning applications.

3.1.1 Incremental Defrosting Algorithm.

The vanilla transfer learning protocol, where only the read-out layers are retrained on the target data, may represent a cost-effective and easily implementable strategy. However, as we can see from the example in Fig. 1 for different architectures and source-target pairs, transferring all the feature-extractor layers does not always represent the optimal strategy. Indeed, there exist transfer learning scenarios where considering different depths in the defrosting process may allow large performance gains in the downstream task.

The multiple defrosting and re-training process exemplified in the previous paragraph can be equivalently thought of as a preliminary and simple protocol for identifying the optimal transfer depth. We call this algorithm incremental defrosting, whose pseudocode is provided in Algorithm 1. Since transfer learning is mostly relevant when the target dataset is effectively small, this procedure should not require high computational costs compared with typical large-scale deep network training. Moreover, the computational cost of training is further reduced by pre-processing the dataset a single time, up until the last frozen layer, and then just training the remaining part of the architecture as a separate network receiving the frozen representations as inputs. For instance, training a wide ResNet-28 on all the 50000 examples of CIFAR10 from scratch takes approximately 2 hours, while training just the last layer only a couple of minutes.

Algorithm 1 Incremental Defrosting

```
Input: Target task, Source pre-trained network.
Load the network pre-trained on the source task
For $l$ = last layer up to $l$ = first layer do:
    Train on the target task freezing all layers up to layer $l$ and intializing the remaining ones at random;
    Evaluate the test metrics.
Return: Defrosting profile
```

The salient trait of the defrosting profiles is that they show a shape that is generally quite regular and predictable from a few points in the profile. Indeed, the main drops in the representation similarities happen after the lossy down-sampling layers (see sec. 3.4). At the same time, due to the smoothness of the curve, once the maximum has been identified, it is not necessary to compute the points in the branch between the maximum and the input layer. Therefore, in cases of limited computational budgets, the position of the maximum could also be inferred by sampling just a few transfer depths, ranging from next to last to after the first down-sampling operation.

The identification of the optimal transfer depth by means of the incremental defrosting algorithm will then provide the best initialization for all the possible subsequent fine-tuning strategies, whose first step is the choice of a source pre-trained network.

3.2 Dependency of the optimal transfer depth on the training set size

At a qualitative level, the interplay between data scarcity and transfer learning seems clear: With enough samples in the target dataset, all the relevant data features can be extracted directly from the task at hand. Instead, with a reduced sample size, learning good representations from scratch becomes impossible and transfer learning becomes thus a convenient option.

We conduct a quantitative investigation of the transition between these two scenarios by considering two distinct datasets as test cases. The first dataset is the GM CIFAR-10 clone (described in sec. 2). The second one is a balanced sub-sample of a publicly available medical dataset for the classification of breast cancer Janowczyk & Madabhushi (2016); Cruz-Roa et al. (2014) (more on the Breast Cancer dataset in SM sec. B.1). Given these two datasets, we then consider two different transfer learning experiments. In the first one, we train a Wide-ResNet-28-4 by selecting GM as the source task and CIFAR-10 as the target task. In the second experiment, we take instead a ResNet-18 pre-trained on ImageNet from the PyTorch model zoo, and we select as the target task the Breast Cancer dataset (this is a common choice in medical applications Khan et al. (2019); Chouhan et al. (2020)). For both transfer learning experiments, we sub-sample the target data, thus spanning over a wide range of data-scarce scenarios.

Fig. 4 shows the outcome of the experiments involving CIFAR-10 (left) and Breast Cancer (right), respectively. The defrosting profiles in the two cases show some common features. As expected, the generalization accuracy on the target task generally increases with larger amounts of training data. However, a nontrivial observation is that the optimum of the defrosting profile shifts backward in the layers as the size of the target training set is increased (darker colors). Instead of just transitioning from a regime of convenient transfers (with some fixed optimal transfer depth) to one of sub-optimal transfer, we find that the number of transferred layers needs to be adjusted according to the sample size, as early layers maintain a higher degree of compatibility when more data is made available. Note also that optimally defrosting the pre-trained network can be more effective than employing the vanilla freezing protocol after acquiring more data. For example, in the right panel of Fig. 4, the optimal transfer with 5×10^3 samples achieves a comparable performance with vanilla freezing with 10 times more data. The fact that this accuracy boost can be obtained almost

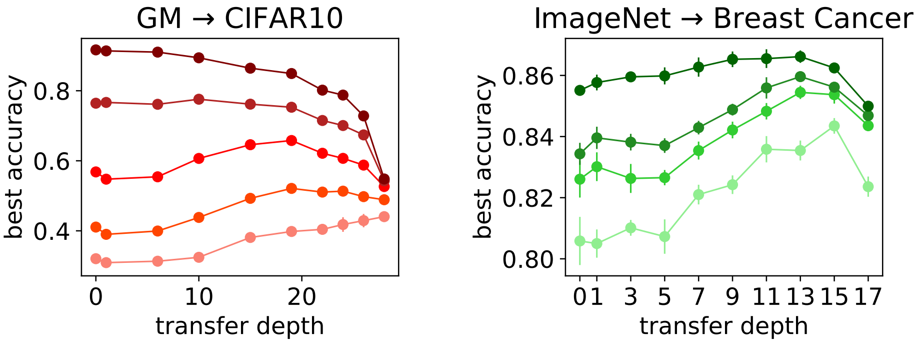

Figure 4: **Effect of dataset size.** The figures show the defrosting curves obtained in two transfer learning tasks for different target set sizes: from GM to CIFAR-10 and from ImageNet to Breast Cancer. In the left panel, we use a Wide-ResNet-28-4, while in the right panel, a ResNet-18. More details can be found in the paragraph Architecture and training protocol. The darkness of the curve represents the set size, the darker the larger. In particular, in the GM to CIFAR10 transfer, the different shades of red correspond to the training set sizes $\{2^4, 2^6, 2^8, 2^{10}, 5 \times 10^3\}$ per class, while, in the ImageNet to Breast Cancer transfer, the different shades of green correspond to the training sizes $\{5 \times 10^2, 5 \times 10^3, 10^4, 5 \times 10^4\}$ per class. The simulations are averaged over 20 and 5 different realizations in the top and bottom panels, respectively.

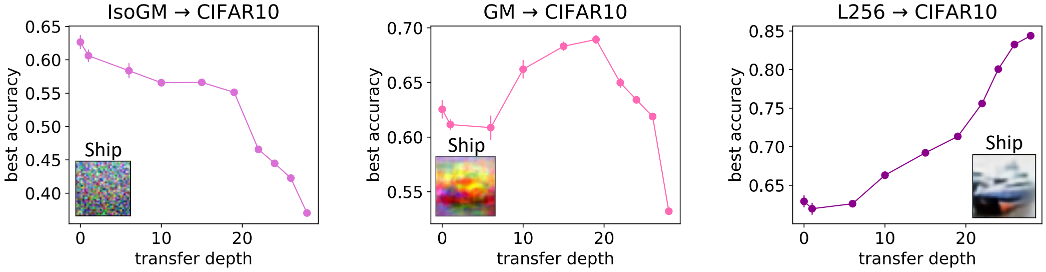

Figure 5: **Defrosting profile for CIFAR-10 clones.** The four panels show three different CIFAR-10 clones to CIFAR-10 transfer directions in increasing levels of CIFAR-10 approximation: IsoGM, GM, and L256. The figure shows the corresponding defrosting profiles at 384 training samples per class. It provides evidence of how better approximating clones convey richer features, and therefore, it is best to transfer a larger portion of the architecture. Each of the three insets displays a prototypical example of how the images look in the corresponding clone. The network architecture is a Wide-ResNet-28-4. More details can be found in the paragraph Architecture and training protocols. The simulations are averaged over 5 different realizations.

for free, just by estimating a defrosting profile for a given transfer task, is one of the main messages of this work.

3.3 Dependency of the optimal transfer depth on source-target similarity

We now investigate the role played by the statistical similarity between the input distributions of the source and target tasks in determining the optimal transfer depth. Comparing experiments with no control over the input statistics and with different labeling rules may be insufficient to answer this question, as these factors cannot be easily disentangled. With the hierarchy of generative models described in Section 2, instead, one can isolate this crucial aspect of transfer learning. In this controlled synthetic framework, we can gradually move from a scenario where only the first moment of the source and target task distribution are matched (IsoGM), to the case in which the first two moments are matched (GM), up to the scenario where the source and target distributions almost perfectly match (L256).

Our results are summarised by the incremental defrosting profiles in Fig. 5 at the same training set size. As expected, if the similarity of the two data sources is insufficient (left panel), fully retraining the network and giving up the transfer learning approach is preferable, even in the data-scarce regime. At pre-training, the early layers of the network have adapted to significantly different input distributions and are thus unsuitable to extract the information needed to solve the target task. As the similarity between the two tasks increases (middle and right panel), the optimal cut for the defrosting protocol shifts towards the later layers of the network. In the SM, Fig. 7 complements this picture by showing how the defrosting profiles at different sizes of the training set can significantly change depending on the relatedness between the source and the target task. Indeed, when the source and the target task are poorly statistically correlated, we can observe negative transfer effects even in an extreme data scarcity regime. On the other hand, when the two tasks are extremely correlated, it is almost always convenient to transfer up to the very last layer, except when one can count on a huge number of training examples.

From this analysis, two relevant and related aspects emerge. First of all, depending on the complexity of the source task and its relatedness to the target one, we can observe different scenarios, ranging from always negative to almost always positive transfer. It is then natural to ask whether it is possible to design a strategy to identify the most promising source task among several other candidates. The second aspect is that the experiments in Fig. 5 strongly hint at the fact that there is a precise order in which the layers learn the different moments of the input data distribution. Indeed, if we consider the way CIFAR10-clones have been designed, it seems that the first layers are just focusing on capturing the first and second-order statistics of the data distribution, thus playing a more generic image pre-processing role Yosinski et al. (2014), compatible with multiple downstream classification tasks on similar input distributions. On the contrary, deeper layers are more sensitive to high-order moments. We discuss these two points in the next section.

3.4 Source Task Detection

The experiments in Fig. 5 highlight how crucial is the choice of the source task in transfer learning protocols. Indeed, the generalization performances of the target network can consistently improve or downgrade, simply depending on the initial conditions or, in other words, on how closely related to the target the source pre-trained network is. It is then natural to ask: How can we choose the most promising source task among several candidates? In principle, we would expect that similar tasks induce similar internal representations in the neural networks trying to solve them. To select the most promising source task, we propose quantifying the similarity level between the internal representations of the network pre-trained on the source task and the one trained from scratch on the few examples in the target set with the *Information Imbalance* (II) described in Sec. 2.

Fig. 6 shows how the II can be used as a good proxy to identify the most promising source task when the target task is CIFAR-10 and the available source candidates are the three clones IsoGM, GM, and L256. In particular, the left panel of Fig. 6 shows the II computed at each layer l in the context where the target task is CIFAR10 and the source task is one of its clones, e.g. IsoGM (magenta curve), GM (blue curve) and L256 (green curve). As can be seen, the CIFAR-10 clone closer to CIFAR-10 in data distribution (e.g. L256 green curve) is also the one closer in terms of internal network representation and, at the same time, the one leading to the best defrosting profile, as shown in the right panel of Fig. 6. By simply looking at the II profile, we can then predict which source candidate leads to the more informative feature map for solving the target task.

A further thing to notice is that II generally increases as a function of the depth, but its rate of increase depends on the similarity between the datasets on which the networks to be compared have been trained on. The main II jumps happen in correspondence with the down-sampling layers in the network, where some information that might be relevant for one of the tasks is filtered out in the other task, inevitably inducing different local neighborhoods in the representations. As already pointed out in sec. 3.1.1, this

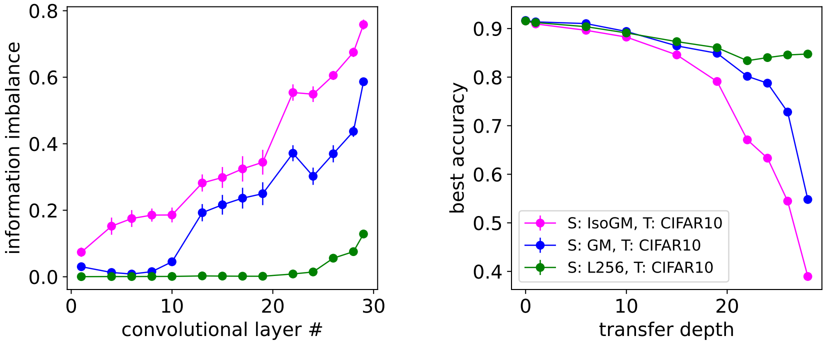

Figure 6: **Role of the topological similarity of representations** In the (top) panel, we display the Information Imbalance (II) between CIFAR10 and three CIFAR10 clones to be employed as source dataset: IsoGM, GM, and the largest autoencoder with bottleneck 256. On the (bottom), we display the incremental defrosting profile obtained in these three transfer learning scenarios, assuming the source and target datasets are full (50K images each). The simulations are averaged over 5 different realizations. In the bottom panel, the error bars are smaller than the point size.

allows us to infer the optimal transfer depth by sampling just a few transfer depths rather than identifying the whole defrosting profile. Finally, the II seems to suggest the existence of a hierarchical order according to which deep neural network layers learn the increasingly higher-order moments of the input data distribution. Indeed, the topological similarity in the case of L256 is almost perfect until the very last layers, indicating the presence of small differences in the high-order features of the two data distributions. Instead, in the case of GM, we see that the representations are almost equivalent up to the 10-th layer and become increasingly different thereafter. Finally, the IsoGM case shows that the difference in the statistical distribution of training data affects the learning process dramatically, inducing visibly dissimilar representations that are incompatible with a good generalization performance. This finding complements the picture presented in Goldt & Ingrosso (2022), where the learning dynamics is shown to learn the different moments of the input distribution in a precise temporal order.

There are many alternative similarity indices that can be measured in place of the II, each one clearly unveiling the connection between the topological similarity of the representations and the transfer learning behavior observed in the incremental defrosting profile. In SM C, we also show the results for the CKA Kornblith et al. (2019), the neighborhood overlap Doimo et al. (2020) and the Spearman correlation of local neighborhoods, which provide qualitatively similar information.

4 Discussion

In the present work, we propose an experimental framework able to identify the key factors determining the number of layers to keep frozen in transfer learning settings. This framework can also be considered a simple algorithm to improve the accuracy of networks trained via transfer learning. We call this algorithm incremental defrosting. For data-poor target tasks, incremental defrosting can be run at a negligible computational cost. However, we further propose a strategy that can be employed to reduce its cost also for larger target data sets. Thanks to the controlled experimental framework, we have shown that the optimal transfer depth depends non-trivially on the amount of training data and on the similarity between source and target tasks and that its identification can be more impactful on downstream performance than acquiring 10 times more training data. Finally, we have proposed a strategy to identify the most promising source task based on topological similarity measures of network representations.

We show that the incremental defrosting algorithm and the source task detection strategy can make a difference in real-world transfer learning applications, as it allows squeezing the best accuracy for a given architecture in an elementary and intuitive manner. Moreover, when used in conjunction with subsequent fine-tuning stages, both protocols can overall improve fine-tuning performance, especially if we consider that the small learning rates employed in fine-tuning stages prevent the learning dynamics from deviating too much from the initial condition dictated by the source network. Therefore starting from the most convenient initialization is definitely crucial for transfer learning effectiveness.

Recent developments in the theory of neural networks Jacot et al. (2018); Canatar et al. (2021) achieved exact descriptions of learning neural networks in different regimes. However, there is still a lack of mathematical tools for analyzing the role played by representations in the feature-learning regime relevant to transfer learning. Our work provides calibrated experiments that showed the topological similarity between representations – obtained via independent training on source and target tasks – is a qualitative indication of the suitability of a representation transfer between tasks. This highlights the strong interplay among feature learning, dataset similarity and transfer learning effectiveness. On the theoretical side, a first step in the direction of understanding this interplay has been already done, for two-layer neural networks, in Gerace et al. (2022), where, using tools from statistical physics of learning, the authors show that negative transfer effects can occur if the source and target task are poorly related, up to the point that it is more convenient to randomly chose the network parameters rather than transferring a feature map from the source task. Extending these results to a generic number of layers in the feature-learning phase of infinite-width networks Yang et al. (2020) or in the proportional limit of deep neural networks Pacelli et al. (2023); Aiudi et al. (2023); Cui et al. (2023); Seroussi et al. (2023), is a promising future direction which could help to theoretically rationalize the efficacy of transfer learning in deep learning settings.

In perspective, we believe that understanding – both on the theoretical and the practical sides – how to disentangle input similarity from the labeling rule might be an interesting direction for future investigations in this research field.

Acknowledgments

The authors acknowledge Sebastian Goldt for useful discussions. SSM acknowledges support from the Sainsbury Wellcome Centre Core Grant from Wellcome (219627/Z/19/Z) and the Gatsby Charitable Foundation (GAT3755). AL acknowledges the Italian National Centre for HPC, Big Data, and Quantum Computing (grant number CN00000013).

References

Emmanuel Abbe, Enric Boix-Adsera, Matthew S Brennan, Guy Bresler, and Dheeraj Nagaraj. The staircase property: How hierarchical structure can guide deep learning. *Advances in Neural Information Processing Systems*, 34:26989–27002, 2021.

R. Aiudi, R. Pacelli, A. Vezzani, R. Burioni, and P. Rotondo. Local Kernel Renormalization as a mechanism for feature learning in overparametrized Convolutional Neural Networks. *arXiv e-prints*, art. arXiv:2307.11807, July 2023. doi: 10.48550/arXiv.2307.11807.

Andrew L Beam and Isaac S Kohane. Big data and machine learning in health care. *Jama*, 319(13): 1317–1318, 2018.

Yoshua Bengio. Deep learning of representations for unsupervised and transfer learning. In *Proceedings of ICML workshop on unsupervised and transfer learning*, pp. 17–36. JMLR Workshop and Conference Proceedings, 2012.

Lukas Bossard, Matthieu Guillaumin, and Luc Van Gool. Food-101 – mining discriminative components with random forests. In *European Conference on Computer Vision*, 2014.

Abdulkadir Canatar, Blake Bordelon, and Cengiz Pehlevan. Spectral bias and task-model alignment explain generalization in kernel regression and infinitely wide neural networks. *Nature communications*, 12(1): 1–12, 2021.

Vikash Chouhan, Sanjay Kumar Singh, Aditya Khamparia, Deepak Gupta, Prayag Tiwari, Catarina Moreira, Robertas Damaševičius, and Victor Hugo C De Albuquerque. A novel transfer learning based approach for pneumonia detection in chest x-ray images. *Applied Sciences*, 10(2):559, 2020.

Angel Cruz-Roa, Ajay Basavanhally, Fabio González, Hannah Gilmore, Michael Feldman, Shridar Ganesan, Natalie Shih, John Tomaszewski, and Anant Madabhushi. Automatic detection of invasive ductal carcinoma in whole slide images with convolutional neural networks. In *Medical Imaging 2014: Digital Pathology*, volume 9041, pp. 904103. SPIE, 2014.

Ekin D Cubuk, Barret Zoph, Jonathon Shlens, and Quoc V Le. Randaugment: Practical automated data augmentation with a reduced search space. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition workshops*, pp. 702–703, 2020.

Hugo Cui, Florent Krzakala, and Lenka Zdeborová. Optimal learning of deep random networks of extensive-width. *arXiv preprint arXiv:2302.00375*, 2023.

Yehuda Dar and Richard G Baraniuk. Double double descent: on generalization errors in transfer learning between linear regression tasks. *SIAM Journal on Mathematics of Data Science*, 4(4):1447–1472, 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.

Oussama Dhifallah and Yue M. Lu. Phase transitions in transfer learning for high-dimensional perceptrons. *Entropy*, 23(4):400, 2021. doi: 10.3390/e23040400. URL `https://doi.org/10.3390/e23040400`.

Diego Doimo, Aldo Glielmo, Alessio Ansuini, and Alessandro Laio. Hierarchical nucleation in deep neural networks. *Advances in Neural Information Processing Systems*, 33:7526–7536, 2020.

Iñaki Soto-Rey Dominik Müller and Frank Kramer. Multi-disease detection in retinal imaging based on ensembling heterogeneous deep learning models. *Studies in Health Technology and Informatics*, 283, 2021. doi: 10.3233/shti210537. URL `https://doi.org/10.3233/shti210537`.

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

Simon S Du, Wei Hu, Sham M Kakade, Jason D Lee, and Qi Lei. Few-shot learning via learning the representation, provably. *arXiv preprint arXiv:2002.09434*, 2020.

João Gama, Indrė Žliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia. A survey on concept drift adaptation. *ACM computing surveys (CSUR)*, 46(4):1–37, 2014.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard Zemel, Wieland Brendel, Matthias Bethge, and Felix A Wichmann. Shortcut learning in deep neural networks. *Nature Machine Intelligence*, 2(11):665–673, 2020.

Federica Gerace, Luca Saglietti, Stefano Sarao Mannelli, Andrew Saxe, and Lenka Zdeborová. Probing transfer learning with a model of synthetic correlated datasets. *Machine Learning: Science and Technology*, 3(1):015030, 2022.

Aldo Glielmo, Claudio Zeni, Bingqing Cheng, Gábor Csányi, and Alessandro Laio. Ranking the information content of distance measures. *PNAS Nexus*, 1(2), 04 2022. ISSN 2752-6542. doi: 10.1093/pnasnexus/pgac039. URL `https://doi.org/10.1093/pnasnexus/pgac039`. pgac039.

Sebastian Goldt and Alessandro Ingrosso. Data-driven emergence of convolutional structure in neural networks. *Proceedings of the National Academy of Sciences*, 119(40):2201854119, 2022.

Dongyoon Han, Jiwhan Kim, and Junmo Kim. Deep pyramidal residual networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 5927–5935, 2017.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

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.

Arthur Jacot, Franck Gabriel, and Clément Hongler. Neural tangent kernel: Convergence and generalization in neural networks. *Advances in neural information processing systems*, 31, 2018.

Ashish Jaiswal, Ashwin Ramesh Babu, Mohammad Zaki Zadeh, Debapriya Banerjee, and Fillia Makedon. A survey on contrastive self-supervised learning. *Technologies*, 9(1):2, 2020.

Andrew Janowczyk and Anant Madabhushi. Deep learning for digital pathology image analysis: A comprehensive tutorial with selected use cases. *Journal of pathology informatics*, 7(1):29, 2016.

SanaUllah Khan, Naveed Islam, Zahoor Jan, Ikram Ud Din, and Joel JP C Rodrigues. A novel deep learning based framework for the detection and classification of breast cancer using transfer learning. *Pattern Recognition Letters*, 125:1–6, 2019.

Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*, 2014.

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, 2019.

Andrew K. Lampinen and Surya Ganguli. An analytic theory of generalization dynamics and transfer learning in deep linear networks. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL https://openreview.net/forum?id=ryfMLoCqtQ.

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.

Phillip Lippe. UvA Deep Learning Tutorials. https://uvadlc-notebooks.readthedocs.io/en/latest/, 2022.

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.

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

Behnam Neyshabur, Ryota Tomioka, and Nathan Srebro. In search of the real inductive bias: On the role of implicit regularization in deep learning. *arXiv preprint arXiv:1412.6614*, 2014.

Maxime Oquab, Timothée Darcet, Théo Moutakanni, Huy Vo, Marc Szafraniec, Vasil Khalidov, Pierre Fernandez, Daniel Haziza, Francisco Massa, Alaaeldin El-Nouby, et al. Dinov2: Learning robust visual features without supervision. *arXiv preprint arXiv:2304.07193*, 2023.

R Pacelli, S Ariosto, M Pastore, F Ginelli, M Gherardi, and P Rotondo. A statistical mechanics framework for bayesian deep neural networks beyond the infinite-width limit. *Nature Machine Intelligence*, 5(12): 1497–1507, 2023.

Samiksha Pachade, Prasanna Porwal, Dhanshree Thulkar, Manesh Kokare, Girish Deshmukh, Vivek Sahasrabuddhe, Luca Giancardo, Gwenolé Quellec, and Fabrice Mériaudeau. Retinal fundus multi-disease image dataset (rfmid): A dataset for multi-disease detection research. *Data*, 6(2):14, 2021. URL https://doi.org/10.3390/data6020014.

Guillermo Valle Pérez, Chico Q. Camargo, and Ard A. Louis. Deep learning generalizes because the parameter-function map is biased towards simple functions. In *7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019*. OpenReview.net, 2019. URL `https://openreview.net/forum?id=rye4g3AqFm`.

Maithra Raghu, Chiyuan Zhang, Jon Kleinberg, and Samy Bengio. Transfusion: Understanding transfer learning for medical imaging. *Advances in neural information processing systems*, 32, 2019.

Nasim Rahaman, Devansh Arpit, Aristide Baratin, Felix Draxler, Min Lin, Fred A Hamprecht, Yoshua Bengio, and Aaron C Courville. On the spectral bias of deep neural networks. 2018.

Rajat Raina, Alexis Battle, Honglak Lee, Benjamin Packer, and Andrew Y Ng. Self-taught learning: transfer learning from unlabeled data. In *Proceedings of the 24th international conference on Machine learning*, pp. 759–766, 2007.

Alvin Rajkomar, Jeffrey Dean, and Isaac Kohane. Machine learning in medicine. *New England Journal of Medicine*, 380(14):1347–1358, 2019.

Maria Refinetti, Alessandro Ingrosso, and Sebastian Goldt. Neural networks trained with sgd learn distributions of increasing complexity. In *International Conference on Machine Learning*, pp. 28843–28863. PMLR, 2023.

Inbar Seroussi, Gadi Naveh, and Zohar Ringel. Separation of scales and a thermodynamic description of feature learning in some cnns. *Nature Communications*, 14(1):908, 02 2023. ISSN 2041-1723. doi: 10.1038/s41467-023-36361-y. URL `https://doi.org/10.1038/s41467-023-36361-y`.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585, 2020.

Hoo-Chang Shin, Holger R Roth, Mingchen Gao, Le Lu, Ziyue Xu, Isabella Nogues, Jianhua Yao, Daniel Mollura, and Ronald M Summers. Deep convolutional neural networks for computer-aided detection: Cnn architectures, dataset characteristics and transfer learning. *IEEE transactions on medical imaging*, 35(5): 1285–1298, 2016.

Saúl Solorio-Fernández, J Ariel Carrasco-Ochoa, and José Fco Martínez-Trinidad. A review of unsupervised feature selection methods. *Artificial Intelligence Review*, 53(2):907–948, 2020.

Sebastian Thrun and Lorien Pratt. *Learning to learn*. Springer Science & Business Media, 2012.

Naftali Tishby and Noga Zaslavsky. Deep learning and the information bottleneck principle. In *2015 ieee information theory workshop (itw)*, pp. 1–5. IEEE, 2015.

Marlon Tobaben, Aliaksandra Shysheya, John Bronskill, Andrew Paverd, Shruti Tople, Santiago Zanella-Beguelin, Richard E Turner, and Antti Honkela. On the efficacy of differentially private few-shot image classification. *arXiv preprint arXiv:2302.01190*, 2023.

Florian Wenzel, Andrea Dittadi, Peter Gehler, Carl-Johann Simon-Gabriel, Max Horn, Dominik Zietlow, David Kernert, Chris Russell, Thomas Brox, Bernt Schiele, et al. Assaying out-of-distribution generalization in transfer learning. *Advances in Neural Information Processing Systems*, 35:7181–7198, 2022.

Qiang Yang, Yu Zhang, Wenyuan Dai, and Sinno Jialin Pan. *Transfer learning*. Cambridge University Press, 2020.

Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *Advances in neural information processing systems*, 27, 2014.

S. Zagoruyko and N. Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

A Appendix

B Supplementary Material

B.1 Numerical Details

In this section, we provide details concerning the autoencoder clones and the Breast Cancer medical dataset.

Autoencoder Clones We follow Lippe (2022) to build an encoder with 5 3×3-convolutional layers and GeLU activations, alternating stride 2 and stride 1 filters to reduce the dimensionality and ending with a single dense layer, whose size determines the bottleneck. The decoder has the same structure with the stride-2 convolutional layers replaced by deconvolutional ones. Training is performed over the CIFAR-10 training set in batches of size 256 over 500 epochs using Adam Kingma & Ba (2014). We used a warmup schedule, peaking the learning rate up to 10^{-3} after 100 epochs and then annealing to 10^{-5}. At the end of the training, we collect the produced images for each bottleneck size in a new dataset. In the experiments, the so-produced CIFAR-10 autoencoder clones are standardized in such a way to have zero mean and standard deviation equal to one for each channel.

Medical Dataset Concerning the transfer experiments in the leftmost panel of Fig. 1, we consider the Retinal Fundus Multi-Disease Image Dataset Pachade et al. (2021) which is composed of 3200 fundus images, each one associated to a given retinal condition, including various examples of rare and challenging to detect diseases for a total of 46 different classes. This dataset has been associated with the Retinal Image Analysis for Multi-Disease Classification (RIADD) challenge from the ISBI 2021, where the main goal was to classify the 46 different types of retina conditions through sized retinal microscope images. In our experiments, to make the task more similar to the one of breast cancer detection (rightmost panel of Fig. 1), we simply train the ResNet50 to recognize whether the retina is affected by some disease or not, thus switching from a multi-class to a binary classification task. Concerning the transfer experiments in the bottom panel of Fig. 4, we instead consider as a test case a publicly available medical dataset for the classification of breast cancer, consisting of 277.524 50x50 patches extracted from 162 scans Janowczyk & Madabhushi (2016); Cruz-Roa et al. (2014). The images are labeled as positive or negative according to the presence or absence of invasive ductal carcinoma (IDC) tissue in the patch. Among all the examples, we use the 80% of them for the training set and the 20% for the test set. With the purpose of speeding up the simulations and describing more realistic transfer learning scenarios, we further randomly sub-sampled the test set up to 10^4 images at each simulation run, preserving class balance. In the transfer experiment, the images have been rescaled to 224×224 pixels and then standardized as required in the PyTorch documentation for pre-trained ResNet-18.

Food101 dataset In the bottom panel of Fig. 3, we show the defrosting profiles for transfer and fine-tuning setups for the Food101 dataset Bossard et al. (2014). This dataset has 101 classes of different kinds of foods, with 1000 images each. The images are divided into training and validation sets containing 750 and 250 examples. In our experiments, we select 30 images per class to create our training set and report the accuracy of the defrosting profiled on the full validation set.

B.2 More on CIFAR10 Clones: Defrosting Profile and Information Imbalance

As pointed out in the main text, the optimal transfer depth strongly depends on the size of the target training set and the statistical similarity between the source and the target task.

Fig. 7 complements the picture described in sec.3.2 and sec.3.3, by showing the defrosting profile for different CIFAR-10 training set sizes (lighter colors identify smaller training set size), when the source task is IsoGM or one of the autoencoder clones at increasingly higher bottleneck size, e.g. L4, L32, L256. As we can see, in the transfer direction IsoGM to CIFAR-10, transferring up to the very layer is never beneficial, no matter the size of the target task. IsoGM and CIFAR-10 share only the first moment of the underlying CIFAR-10 distribution; therefore, negative transfer effects can occur due to the scarce statistical similarity between the two datasets. By including higher-order moments of the CIFAR-10 distribution and refining their approximations through the bottleneck size of the CIFAR-10 autoencoder clones, transfer learning becomes increasingly more effective. Indeed, in the transfer direction L256 to CIFAR-10, we can see it is always convenient to transfer up to the very last layer, except at very huge training set sizes. Almost the same picture occurs when transferring the pre-trained feature map on L32 to CIFAR10. However, similarly to the picture that emerged in the transfer experiment concerning the Breast Cancer dataset, in this case, a peak starts appearing in correspondence with the second-to-last defrosted layer. The optimal transfer depth then shifts to the left in the L4 to CIFAR-10 transfer experiment with 64 images per class. This phenomenon is probably due to the smaller degree of similarity between the two datasets, even if it is less exacerbated than what is observable in the GM to CIFAR-10 transfer direction.

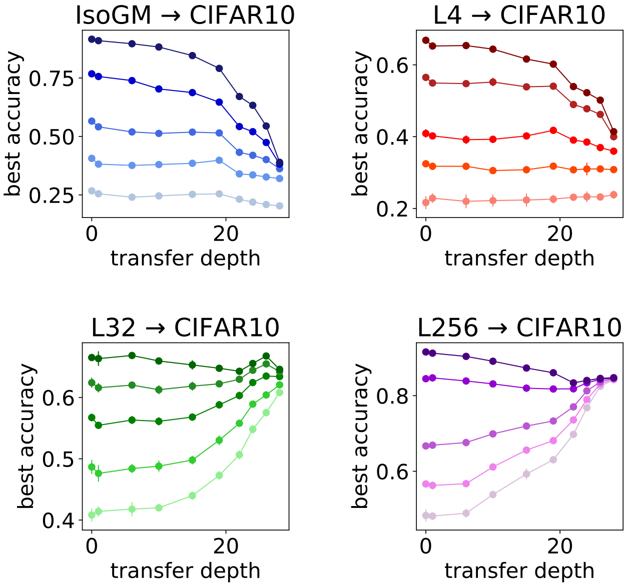

Figure 7: **Effect of dataset size and source-target relatedness.** The plot shows the defrosting profiles at varying training set sizes of CIFAR-10 for four different transfer couples, similar to what was shown in Fig. 4 (top panel) of the main text. The transfers reported are IsoGM to CIFAR-10 (top left panel), L4 to CIFAR-10 (top right panel), L32 to CIFAR-10 (bottom left panel), and L256 to CIFAR-10 (bottom right panel). In particular, in the Iso-GM to CIFAR-10 transfer direction, the curves are relative from the lighter to the darker colors to the following CIFAR-10 training set sizes $\{8, 64, 256, 1024, \text{full dataset size}\}$; in the L4 to CIFAR-10 transfer direction, the curves are relative from the lighter to the darker colors to the following CIFAR-10 training set sizes $\{4, 16, 64, 256, 512\}$; in the L32 to CIFAR-10 transfer direction, the curves are relative from the lighter to the darker colors to the following CIFAR-10 training set sizes $\{64, 128, 256, 384, 512\}$; in the L256 to CIFAR-10 transfer direction, the curves are relative from the lighter to the darker colors to the following CIFAR-10 training set sizes $\{128, 256, 512, 2048, \text{full dataset size}\}$. The simulations are averaged over 5 different realizations and the error bars are smaller than the point size.

Fig. 8 shows the II across layers, quantifying the representation similarity of two networks trained from scratch on CIFAR-10 and on one of the clones among IsoGM, L4, L32 and L256. The representations are extracted on the CIFAR-10 test set. As we can see, the profile of the II reflects the hierarchical nature of the CIFAR-10 clone family. Indeed, the larger the size of the bottleneck, the more and the deeper similar the representations are across the layers.

C Similarity measures

In the main manuscript, we have analyzed the similarity among representations while relying on the II metrics. However, as already pointed out in Sec.3.4, there are other similarity measures that can be taken

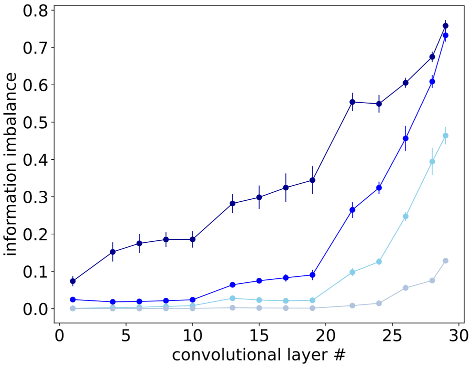

Figure 8: **Representation Similarity.** The plot displays the information imbalance across the layers to increase the fidelity of the CIFAR-10 clones. In particular, lighter colors refer to more similar clones (in sequence IsoGM, L4, L32, L256). The source and target network are trained at full dataset size (50K images). The simulations are averaged over 5 different realizations.

into account and that are equivalently blind to some of the re-scaling, permutation, and symmetries of the parametric function implemented by a neural network model.

Fig. 9 shows, from left to right, the CKA Kornblith et al. (2019), the Spearman correlation of the first 100 neighborhoods and the neighborhood overlap among the first 30 Doimo et al. (2020) neighbors as a function of the network depth. The representations are extracted once again on the CIFAR-10 test set and correspond to the feature map of two different networks, one trained from scratch on the CIFAR-10 dataset and the other one on one of the CIFAR-10 clones, e.g., IsoGM (magenta curve), GM (blue curve) and L256 (green curve). As can be seen, the behavior of the three metrics is qualitatively similar to the one already observed for the II in Fig. 6 (top panel). Indeed, even in this case, the representation similarity is stronger at early layers, providing a further hint at the generality of the feature maps extracted at the bottom of the networks. Moreover, the hierarchical structure of the CIFAR10-clones is reflected in the profiles of the similarity measured, signaling which task is the most promising one as the source task for the CIFAR-10 classification target task.

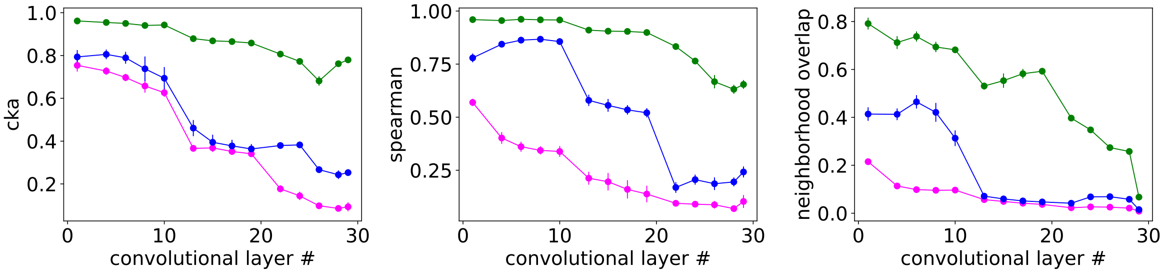

Figure 9: **Other Similarity Measures.** From left to right, CKA, Spearman Correlation, and Neighborhood Overlap across layers. The different colors refer to different source tasks, e.g., IsoGM (magenta curve), GM (blue curve), and L256 (green curve). The setting is the same as reported in the main text Fig. 6, i.e. source and target networks are trained on the full training set size (50K images). The simulations are averaged over 5 different realizations.