From Alexnet to Transformers: Measuring the Non-linearity of Deep Neural Networks with Affine Optimal Transport

Quentin Bouniot[⋆,1,2,3,4] Ievgen Redko[⋆,5] Anton Mallasto[6] Charlotte Laclau[1]
Oliver Struckmeier[7] Karol Arndt[7,8] Markus Heinonen[9] Ville Kyrki[7] Samuel Kaski[9,10]

[1]LTCI, Télécom Paris, Institut Polytechnique de Paris, France, [2]Technical University of Munich, [3]Helmholtz Munich, [4]Munich Center for Machine Learning (MCML), [5]Noah's Ark Lab, Paris, [6]Readpeak, [7]Aalto University, Finland, Dept. of Electrical Engineering and Automation, [8]Nomagic, [9]Aalto University, Finland, Department of Computer Science, [10]University of Manchester, UK, Department of Computer Science

Abstract

In the last decade, we have witnessed the introduction of several novel deep neural network (DNN) architectures exhibiting ever-increasing performance across diverse tasks. Explaining the upward trend of their performance, however, remains difficult as different DNN architectures of comparable depth and width – common factors associated with their expressive power – may exhibit a drastically different performance even when trained on the same dataset. In this paper, we introduce the concept of the non-linearity signature of DNN, the first theoretically sound solution for approximately measuring the non-linearity of deep neural networks. Built upon a score derived from closed-form optimal transport mappings, this signature provides a better understanding of the inner workings of a wide range of DNN architectures and learning paradigms, with a particular emphasis on the computer vision task. We provide extensive experimental results that highlight the practical usefulness of the proposed non-linearity signature and its potential for long-reaching implications. The code for our work is available at https://github.com/qbouniot/AffScoreDeep.

1. Introduction

Deep neural networks (DNNs) are undoubtedly the most powerful AI models currently available [17, 29, 33, 35, 54]. Their performance on many tasks, including natural language processing (NLP) [23] and computer vision [20], is already on par or exceeds that of a human being. One of the reasons explaining such progress is of course the increasing computational resources [43, 57]. Another one is the endeavour for finding ever more efficient neural architectures pursued by researchers over the last decade. As of today, the transformer architecture [65] has firmly imposed itself as a number one choice for most, if not all, of the recent breakthroughs [3, 44, 63] in machine learning.

Limitations But why transformers are more capable than other architectures? Answering this question requires finding a meaningful measure to compare the different famous models over time gauging the trend of their intrinsic capacity. For such a comparison to be informative, it is particularly appropriate to consider the computer vision field that produced many of the landmark neural architectures improving upon each other over the years. Indeed, the decade-long revival of deep learning started with Alexnet's [31] architecture, the winner of the ImageNet Large Scale Visual Recognition Challenge [50] in 2012. By achieving a significant improvement over the traditional approaches, Alexnet was the first truly deep neural network to be trained on a dataset of such scale, suggesting that deeper models were likely to bring even more gains. In the following years, researchers proposed novel ways to train deeper models with hundreds of layers [21, 28, 55, 59] pushing the performance frontier even further. The AI research landscape then reached a turning point with the proposal of transformers [65], starting their unprecedented dominance first in NLP and then in computer vision [9]. Surprisingly, transformers are not particularly deep, and the size of their landmark vision architecture is comparable to that of Alexnet, and this despite a significant performance gap between the two. Ultimately, this gap should be explained by the differences in the expressive power [18] of the two models: a term used to denote the ability of a DNN to approximate functions of a certain

⋆Equal contribution.

complexity. Unfortunately, the existing theoretical results related to this either associate higher expressive power with depth [2, 11, 51] or width [38, 41, 48, 64] falling short in comparing different families of architectures. This, in turn, limits our ability to understand what underpins the achieved progress and what challenges and limitations still exist in the field, guiding future research efforts.

Contributions We argue that quantifying the non-linearity of a DNN may be what we were missing so far to understand the evolution of the deep learning models at a more fine-grained level. To verify this hypothesis in practice, we propose a first theoretically sound measure, called the *affinity score*, that estimates the non-linearity of a given (activation) function using optimal transport (OT) theory. We use the proposed affinity score to introduce the concept of the *non-linearity signature* of DNNs defined as a sequence of affinity scores of all its activation functions.

Summary of findings

1. The non-linearity of activation functions depends on the range of input values and its overall shape (Section 3, Figure 2). This impacts the overall complexity of the model even for randomly initialized MLPs (Figure 4).
2. Until the arrival of vision transformers in 2020, the trend in state-of-the-art vision DNNs was to decrease the average non-linearity of activation functions, rather than to increase it. Vision transformers changed this pattern drastically and are the most non-linear architectures among all landmark ones (Section 4, Figure 5 (A)).
3. Non-linearity signatures of DNNs capture and allow to identify in a meaningful way the similarities in the architectural choices of the different models (Section 4, Figure 5 (B), (C)). Non-linearity signature also reflects the exact role of residual connections in increasing the linearity of activation functions, and thus, of the network (Section 4, Figure 7).
4. Simple statistics extracted from the non-linearity signatures of DNNs correlate with their performance on Imagenet dataset (Section 4, Figure 6).
5. Non-linearity signature is unique, as there is no other metric correlating strongly with it across all architectures (Section 4, Tab. 1).

2. Background

Optimal Transport Let (X, d) be a metric space equipped with a lower semi-continuous *cost function* $c : X \times X \to \mathbb{R}_{\geq 0}$, e.g the Euclidean distance $c(x, y) = \|x - y\|$. Then, the Kantorovich formulation of the OT problem between two probability measures $\mu, \nu \in \mathcal{P}(X)$ is given by

$$\mathrm{OT}_c(\mu, \nu) = \min_{\gamma \in \mathrm{ADM}(\mu,\nu)} \mathbb{E}_\gamma[c], \tag{1}$$

where $\mathrm{ADM}(\mu, \nu)$ is the set of joint probabilities with marginals μ and ν, and $\mathbb{E}_\nu[f]$ denotes the expected value of f under ν. The optimal γ minimizing equation 1 is called the *OT plan*. Denote by $\mathcal{L}(X)$ the law of a random variable X. Then, the OT problem extends to random variables X, Y and we write $\mathrm{OT}_c(X, Y)$ meaning $\mathrm{OT}_c(\mathcal{L}(X), \mathcal{L}(Y))$.

Assuming that either of the considered measures is *absolutely continuous*, then the Kantorovich problem is equivalent to the *Monge problem*

$$\mathrm{OT}_c(\mu, \nu) = \min_{T : T_{\#}\mu = \nu} \mathbb{E}_{X \sim \mu}[c(X, T(X))], \tag{2}$$

where the unique minimizing T is called the *OT map*, and $T_{\#}\mu$ denotes the *push-forward measure*, which is equivalent to the *law* of $T(X)$, where $X \sim \mu$.

Wasserstein distance Let X be a random variable over $\mathbb{R}^d$ satisfying $\mathbb{E}[\|X - x_0\|^2] < \infty$ for some $x_0 \in \mathbb{R}^d$, and thus for any $x \in \mathbb{R}^d$. We denote this class of random variables by $\mathcal{P}_2(\mathbb{R}^d)$. Then, the 2-Wasserstein distance W_2 between $X, Y \in \mathcal{P}_2(\mathbb{R}^d)$ is defined as

$$W_2(X, Y) = \mathrm{OT}_{||x-y||^2}(X, Y)^{\frac{1}{2}}. \tag{3}$$

We now proceed to the presentation of our main contribution.

3. Non-linearity signature of DNNs

Among all non-linear operations introduced into DNNs in the last several decades, activation functions remain the only structural piece that they all inevitably share. For instance, without non-linear activation functions, popular multi-layer perceptrons, no matter how deep, reduce to a linear function unable to learn complex patterns. Activation functions were also early identified [1, 6, 25, 32] as a key to making even a shallow network capable of approximating any function, however complex it may be, to arbitrary precision.

We thus build our study on the following intuition: if activation functions play an important role in making DNNs non-linear, then measuring their degree of non-linearity can provide us with an approximation of the DNN's non-linearity itself. To implement this intuition in practice, however, we first need to find a way to measure the non-linearity of an activation function. Surprisingly, there is no widely accepted measure for this, neither in the field of mathematics nor in the field of computer science. To fill this gap, we will use the OT theory to develop a so-called *affinity score* below.

3.1. Affinity score

Identifiability We consider the pre-activation signal X of an activation function within a neural network, and the post-activation signal $\sigma(X)$ denoted by Y as input and output random variables. Our first step to build the affinity score then is to ensure that we can identify when σ is linear with respect to (wrt) X (for instance, when an otherwise non-linear activation is *locally linear* at the support of X). To

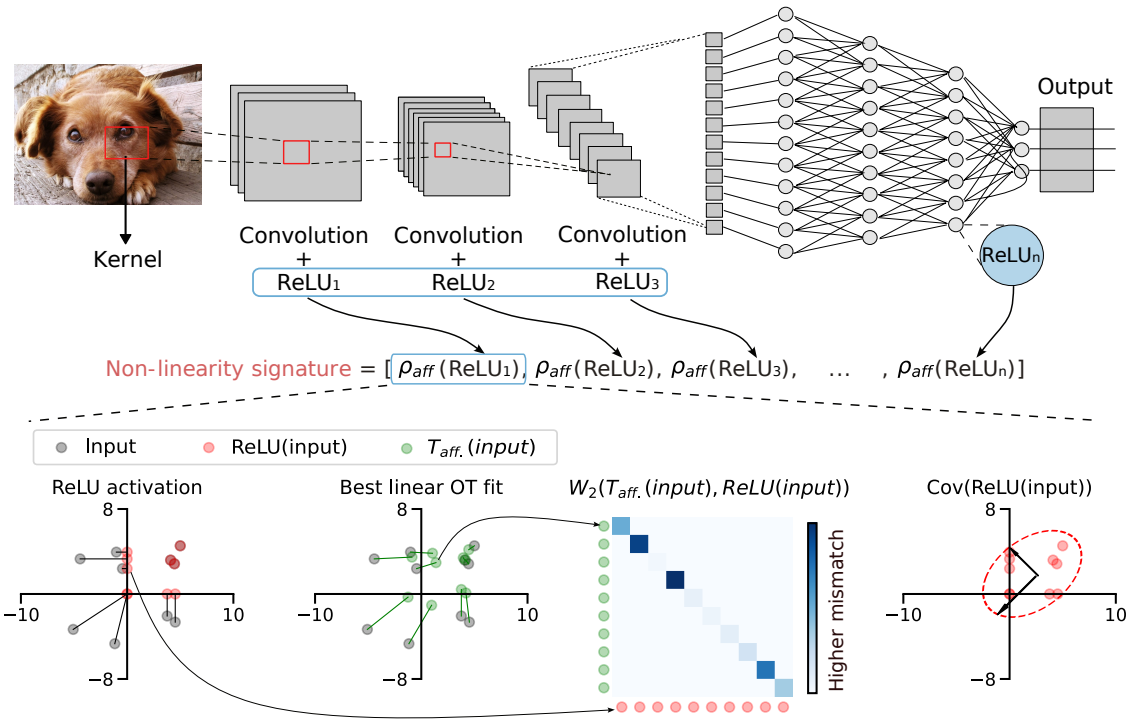

Figure 1. Illustration of how the non-linearity of a given neural network is measured. (**Top**) The non-linearity signature of a DNN is a collection of affinity scores calculated for each activation function spread across its hidden layers. (**Bottom**) The affinity score is calculated based on 3 main steps. First, given an input (grey) and an output (red) of an activation function (*left*), we estimate the best affine OT fit $T_{\text{aff}}(X)$ (green) transporting the input to the output (*middle-left*). Second, we measure the mismatch between the two by summing the transportation costs (*middle-right*) to obtain the Wasserstein distance $W_2(T_{\text{aff}}X, Y)$. Finally, this distance is normalized with the magnitudes of variance (arrows in the rightmost plot) of the output data based on its covariance matrix.

show that such an identifiability condition can be satisfied with OT, we first recall the following classic result from the literature characterizing the OT maps.

Theorem 3.1 ([56]). *Let $X \in \mathcal{P}_2(\mathbb{R}^d)$, $T(x) = \nabla\phi(x)$ for a convex function ϕ with $T(X) \in \mathcal{P}_2(\mathbb{R}^d)$. Then, T is the unique optimal OT map between μ and $T_{\#}\mu$.*

Using this theorem about the uniqueness of OT maps expressed as gradients of convex functions, we can prove the following result (all proofs are in Appendix A.1):

Corollary 3.2. *Without loss of generality, let $X, Y \in \mathcal{P}_2(\mathbb{R}^d)$ be centered, and let $Y = \sigma(X) = TX$, where T is a positive definite linear transformation. Then, T is the OT map from X to Y.*

Whenever the activation function σ is linear, the solution to the OT problem T exactly reproduces it.

Characterization We now seek to understand whether T can be characterized more explicitly. For this, we prove the following theorem stating that T can be computed in closed-form using the normal approximations of X and Y.

Theorem 3.3. *Let $X, Y \in \mathcal{P}_2(\mathbb{R}^d)$ be centered and $Y = TX$ for a positive definite matrix T. Let $N_X \sim \mathcal{N}(\mu(X), \Sigma(X))$ and $N_Y \sim \mathcal{N}(\mu(Y), \Sigma(Y))$ be their normal approximations where μ and Σ denote mean and covariance, respectively. Then, $W_2(N_X, N_Y) = W_2(X, Y)$ and $T = T_{\text{aff}}$, where T_{aff} is the OT map between N_X and N_Y and can be calculated in closed-form*

$$\begin{aligned} T_{\text{aff}}(x) &= Ax + b, \\ A &= \Sigma(Y)^{\frac{1}{2}} \left(\Sigma(Y)^{\frac{1}{2}}\Sigma(X)\Sigma(Y)^{\frac{1}{2}}\right)^{-\frac{1}{2}} \Sigma(Y)^{\frac{1}{2}} \\ b &= \mu(Y) - A\mu(X). \end{aligned} \tag{4}$$

Theorem 3.3 is reminiscent of the classical statement about the 2-Wasserstein distance W_2 between two normal distributions [45] but is significantly more general. The statement is for arbitrary random variables X and Y that satisfy $Y = T(X)$ for a symmetric positive-definite linear transformation T. The classical result about normal distributions is then a direct corollary.

Upper bound When the activation σ is non-linear wrt X, the affine OT mapping $T_{\text{aff}}(X)$ will deviate from the true activation outputs Y. One important step toward quantify-

ing this deviation is given by the famous Gelbrich bound, formalized by means of the following theorem:

Theorem 3.4 (Gelbrich bound [14]). *Let $X, Y \in \mathcal{P}_2(\mathbb{R}^d)$ and let N_X, N_Y be their normal approximations. Then, $W_2(N_X, N_Y) \leq W_2(X, Y)$.*

This upper bound provides a first intuition of why OT can be a great tool for measuring non-linearity: the cost of the affine map solving the OT problem on the left-hand side increases when the map becomes non-linear. We now upper bound the difference between $W_2(N_X, N_Y)$ and $W_2(X, Y)$, two quantities that coincide *only* when σ is linear.

Proposition 3.5. *Let $X, Y \in \mathcal{P}_2(\mathbb{R}^d)$ and N_X, N_Y be their normal approximations. Then,*

1. $|W_2(N_X, N_Y) - W_2(X, Y)| \leq \frac{2\,\mathrm{Tr}\left[(\Sigma(X)\Sigma(Y))^{\frac{1}{2}}\right]}{\sqrt{\mathrm{Tr}[\Sigma(X)]+\mathrm{Tr}[\Sigma(Y)]}}$.
2. *For T_{aff} as in (4),* $W_2(T_{\text{aff}}X, Y) \leq \sqrt{2\,\mathrm{Tr}\,[\Sigma(Y)]}$.

To have a more informative non-linearity measure, we now need to normalize the non-negative Wasserstein distance $W_2(T_{\text{aff}}X, Y)$ to an interpretable interval of $[0, 1]$. The bound given in Proposition 3.5 lets us define the following *affinity score*

$$\rho_{\text{aff}}(X, \sigma(X)) = 1 - \frac{W_2(T_{\text{aff}}X, \sigma(X))}{\sqrt{2\,\mathrm{Tr}[\Sigma(\sigma(X))]}}. \tag{5}$$

The proposed affinity score quantifies how far a given activation σ is from an affine transformation. It is equal to 1 for any input for which the activation function is linear, and 0 when it is maximally non-linear, i.e., when $T_{\text{aff}}X$ and $\sigma(X)$ are independent random variables. We illustrate the computation of the affinity score in Figure 1.

Remark 3.6. *One may wonder whether a simpler alternative to the affinity score can be to use, instead of T_{aff}, a mapping $T_W(x) = Wx$ defined as a solution of a linear regression problem $\min_W ||Y - WX||_F^2$. Then, one can use the coefficient of determination (R^2 score) to measure how well T_W fits the observed data. This approach, however, has two drawbacks. First, following the famous Gauss-Markov theorem, T_W is an optimal linear (linear in Y) estimator. On the contrary, T_{aff} is a globally optimal non-linear mapping aligning X and Y. Second, R^2 compares the fit of T_W with that of a mapping outputting $\mu(Y)$ for any value of X. This is contrary to ρ_{aff} that compares how well T_{aff} fits the data wrt to the worst possible cost incurred by T_{aff} as quantified in Proposition 3.5. This gives us a bounded score, i.e. $\rho_{\text{aff}} \in [0, 1]$, whereas R^2 is not lower bounded, i.e. $R^2 \in [-\infty, 1]$. We confirm experimentally in Section 4 that the two coefficients do not correlate consistently across the studied DNNs suggesting that R^2 is a poor proxy to ρ_{aff}.*

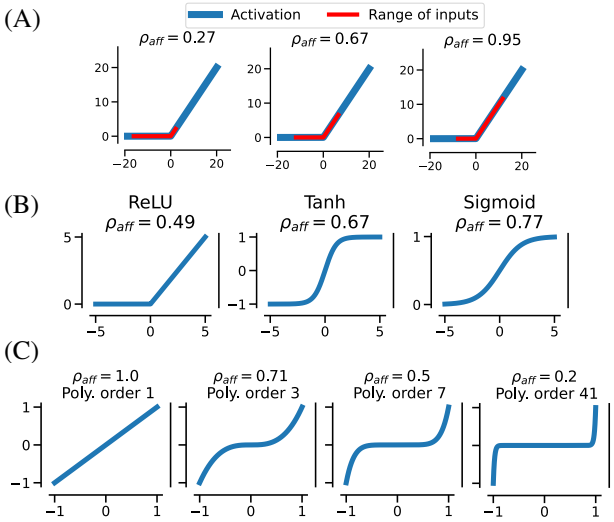

Figure 2. **(A)** Non-linearity of activation functions depends on the range of input values (*red*), illustrated here on ReLU; **(B)** ReLU, Tanh, and Sigmoid exhibit different degrees of non-linearity for the same input; **(C)** Affinity score captures the increasing non-linearity of polynomials of different degrees.

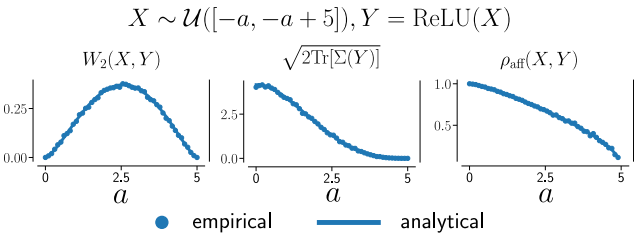

Figure 3. The behavior of **(Left)** $W_2(X, Y)$ term, **(Middle)** $\sqrt{2\,\mathrm{Tr}[\Sigma(Y)]}$ and **(Right)** $\rho_{\text{aff}}(X, Y)$.

3.2. Non-linearity and activation functions

What makes an activation function non-linear? We now want to understand the mechanism behind achieving a lower or higher non-linearity with a given (activation) function. This will explain what the different values of the affinity scores stand for when studying trained DNNs. In Figure 2 (A), we show how the ReLU function [42], defined element-wise as $\mathrm{ReLU}(x) = \max(0, x)$, achieves its varying degree of non-linearity. Interestingly, this degree depends only on the range of the input values. Second, in Figure 2 (B) we also show how the shape of activation functions impacts their non-linearity for a fixed input: surprisingly, piece-wise linear ReLU function is more non-linear than $\mathrm{Sigmoid}(x) = 1/(e^{-x} + 1)$ [49] or $\mathrm{Tanh}(x) = (e^{-x} - e^{x})/(e^{-x} + e^{x})$. Similar observations also apply to compare polynomials of varying degrees (Figure 2 (C)). We include more visualizations of the affinity score of popular activation functions in Appendix B.

A perfect non-linearity and asymmetry In Figure 2 (A), we showed that the affinity score is not symmetric around the origin. This is mainly due to the difference in the behavior of the two terms used to calculate ρ_{aff}. To explain this asymmetry, we derive an analytical expression for $W_2(X, Y)$ and $\sqrt{\text{Tr}[\Sigma(Y)]}$ terms for a special case when $X \sim \mathcal{U}([b, a]), Y = \text{ReLU}(X)$ for $b < 0$ and $a > 0$ (see Appendix A.2) and illustrate it for $b = -a$ and $a = -a + 5$ for $a \in [0, 5]$ in Figure 3. We note that the numerator – illustrated in the left part of the plot – is roughly symmetric and the main source of asymmetry comes from the denominator term (middle part) that captures the amount of variance in the output of the activation function. The affinity score (right part) in 1D then approaches 0 only when both terms tend to 0: a situation where the activation function receives only negative values. Surprisingly, this shows that in our formalism, the "perfect" non-linearity doesn't exist. It would be interesting to study whether this holds in higher dimensions.

Neural redshift revisited As a follow-up to the previous experiment, we now revisit a recent work by [62] that studied the complexity biases carried by the different activation functions in randomly initialized DNNs. The conclusion reached by the authors is that ReLU and its variations have a strong bias toward low complexity which is unaffected by the change in the magnitude of weights when compared, for instance, to tanh. We follow the protocol of [62] and consider an MLP with 3 hidden layers and scalar output initialized using Glorot initialization $\mathcal{U}(-s, s)$ [15] for weight matrices and $\mathcal{U}(-1, 1)$ for biases. The inputs to the network are 64^2 evenly spaced 2D coordinates on the grid in $[-1, 1]^2$ so that the network output 64^2 scalars that can be visualized as a grayscale image. In Figure 4 (A), we present the results from [62] reproducing their claim regarding the comparison between ReLU and tanh when using the different weight magnitudes in the Glorot initialization. This indeed shows that ReLU visually appears to lead to a much simpler function that is independent of the weights' magnitudes. The appearance, however, seems misleading as the average affinity scores of ReLU activations in the considered MLP decrease slightly in the case of higher magnitude weights, yet remain higher than that of tanh. However, our experiments in Figure 2 suggest that this behaviour should not be universal and that the domain of the activation function should have a strong influence on its complexity. To verify this, we slightly change the s parameter in Glorot initialization by setting it to $s' = s - 0.05$. We redo the experiment as before and plot the obtained results in Figure 4 (B). It is now apparent that both visually, and quantitatively, the ReLU activations became much more complex within the considered random MLP with the visualization of the function approximating the considered networks becoming almost indistinguishable between ReLU and tanh. Our proposed affinity score captures this change of complexity and provides a more fine-grained quantitative measure for it.

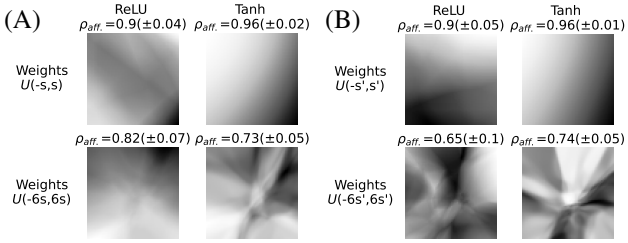

Figure 4. Revisiting the neural redshift phenomenon. **(A)** MLP with 3 hidden layers [62] equipped with either ReLU or Tanh activation functions; **(B)** Same MLP initialized by setting $s' = s - 0.05$ in Glorot initialization. By changing the domain of ReLU, we see that its simplicity bias with changing weight magnitudes vanishes.

Non-linearity signature We now turn our attention to the definition of a non-linearity signature of deep neural networks. We define a neural network N as a composition of layers F_i where each layer F_i is a function taking as input a tensor $\text{X}_i \in \mathbb{R}^{h_i \times w_i \times c_i}$ (for instance, an image of size $224 \times 224 \times 3$ for $i = 1$) and outputting a tensor $\text{Y}_i \in \mathbb{R}^{h_{i+1} \times w_{i+1} \times c_{i+1}}$ used as an input of the following layer F_{i+1}. This defines $\text{N} = F_L \odot ... \odot F_i ... \odot F_1 = \bigodot_{k=1,...,L} F_k$ where $\odot$ stands for a composition. Non-linearity signature, illustrated in Figure 1 (top), associates to each network N a vector of affinity scores calculated over the inputs and outputs of all activation functions across layers.

3.3. Related work

Layer-wise similarity analysis of DNNs A line of work that can be distantly related to our main proposal is that of quantifying the similarity of the hidden layers of the DNNs as proposed [47] and [30] (see [7] for a complete survey of the subsequent works). [47] extracts activation patterns of the hidden layers in the DNNs and use CCA on the singular vectors extracted from them to measure how similar the two layers are. Their analysis brings many interesting insights regarding the learning dynamics of the different convnets, although they do not discuss the non-linearity propagation in the convnets, nor do they propose a way to measure it. [30] proposed to use a normalized Frobenius inner product between kernel matrices calculated on the extracted activations of the hidden layers and argued that such a similarity measure is more meaningful than that proposed by [47].

Impact of activation functions [10] provides the most comprehensive survey on the activation functions used in DNNs. Their work briefly discusses the non-linearity of the different activation functions suggesting that piecewise linear activation functions with more linear components are more non-linear (e.g., ReLU vs. ReLU6). The choice of activation function can lead to different implicit biases carried by the DNNs [62]. [19] show theoretically that smooth

versions of ReLU allow for more efficient information propagation in DNNs with a positive impact on their performance. Our work provides a first extensive comparison of all popular activation functions; we also show that smooth version of ReLU exhibit wider regions of high non-linearity (see Appendix B). Furthermore, we illustrate that the choice of activation function by itself doesn't have a determining effect on the inductive bias of a given randomly initialized DNN. A simple rescaling of the weights obtained either as a result of training or through a different initialization strategy, changes the non-linearity of the activation function within a DNN and, subsequently, its inductive bias (see Figure 4).

Non-linearity measure [46] proposed the non-linearity coefficient in order to predict the train and test error of DNNs. Their coefficient is defined as a square root of the Jacobian of the neural network calculated wrt its input, multiplied by the covariance matrix of the Jacobian, and normalized by the covariance matrix of the input. The presence of the Jacobian in it calls for the differentiability assumption making its application to most of the neural networks with ReLU non-linearity impossible as is. The authors didn't provide any implementation of their coefficient and we were not able to find any prior study reproducing their results.

4. Experimental evaluations

We consider computer vision models trained and evaluated on the same Imagenet dataset with 1,000 output categories (Imagenet-1K) publicly available at [40]. The non-linearity signatures of different studied models presented in the paper is calculated by passing batches of size 512 through the pre-trained models for the entirety of the Imagenet-1K validation set (see Appendix F for more datasets) with a total of 50,000 images. We include the following landmark architectures in our study: Alexnet [31], four VGG models [55], Googlenet [58], Inception v3 [59], five Resnet models [21], four Densenet models [28], four MNASNet models [61], four EfficientNet models [60], five ViT models, three Swin transformer [36] and four Convnext models [37]. We include MNASNet and EfficientNet models as prominent representatives of the neural architecture search approach [13]. Such models are expected to explicitly maximize the accuracy for a given computational budget. Swin transformer and Convnext models are introduced as ViTs with traditional computer vision priors. Their presence will be useful to better grasp how such priors impact ViTs. We refer the reader to Appendix C for more practical details.

History of deep vision models at a glance We give a general outlook of the developments in computer vision over the last decade when seen through the lens of their non-linearity. In Figure 5 (A) we present the minimum, median, and maximum values of the affinity scores calculated for the considered neural networks (see Appendix D for raw non-linearity signatures). We immediately see that until the arrival of transformers, the trend of the landmark models was to increase their linearity, rather than to decrease it. On a more fine-grained level, we note that pure convolution architectures such as Alexnet (2012) and VGGs (2014) exhibit a very low spread of the affinity score values. This trend changes with the arrival of the inception module first used in Googlenet (2014): the latter includes activation functions that extend the range of the non-linearity on both ends of the spectrum. Importantly, we can see that the trend toward increasing the maximum and average non-linearity of the neural networks has continued for almost the whole decade. Even more surprisingly, EfficientNet models (2019), trained through neural architecture search, have strong negative skewness toward higher linearity, although they were state-of-the-art in their time. The second surprising finding comes with the arrival of ViTs (2020): they break the trend and leverage the non-linearity of their hidden activation functions becoming more or more non-linear with the varying size of the patches (see Appendix D for a more detailed comparison with raw signatures). This trend remains valid also for Swin transformers (2021), although introducing the computer vision priors into them makes their non-linearity signature look more similar to pure convolutional networks from the early 2010s, such as Alexnet and VGGs. Finally, we observe that the non-linearity signature of a modern Convnext architecture (2022), designed as a convnet using best practices of Swin transformers, further confirms this.

Similarities between architectures In Figure 5 (B) and (C) we highlight the differences between the DNNs discussed earlier. For this, we first calculate the pairwise dynamic time warping (DTW) distances [52] between the non-linearity signatures of the representative models of each architecture. We plot the obtained result in Figure 5 (B). From the first column of it, we can see that models proposed over the years were becoming more and more dissimilar to Alexnet, until making a "U-turn" with the introduction of Swin transformer and Convnext. Likewise, we note that ViT is highly dissimilar to all other models highlighting its uniqueness. Finally, we qualitatively confirm the findings from Figure 5 (A) and (B) by plotting in Figure 5 (C) the estimated kernel density estimates of the representatives of each architecture. We see that Alexnet has very similar non-linearity signature to Swin Base model, with the latter having heavier tails. This also holds for VGG16 and Convnext. One can note the distinct bi-modal structure of Resnet that we discuss more later.

Closer look at accuracy/non-linearity trade-off Different families of vision models leverage different characteristics of their internal non-linearity to achieve better performance. To better understand this phenomenon, we now turn our attention to a more detailed analysis of the accuracy/non-linearity trade-off by looking for a statistic extracted from their non-linearity signatures that is the most predictive of their accuracy as measured by the R^2 score. Additionally, we

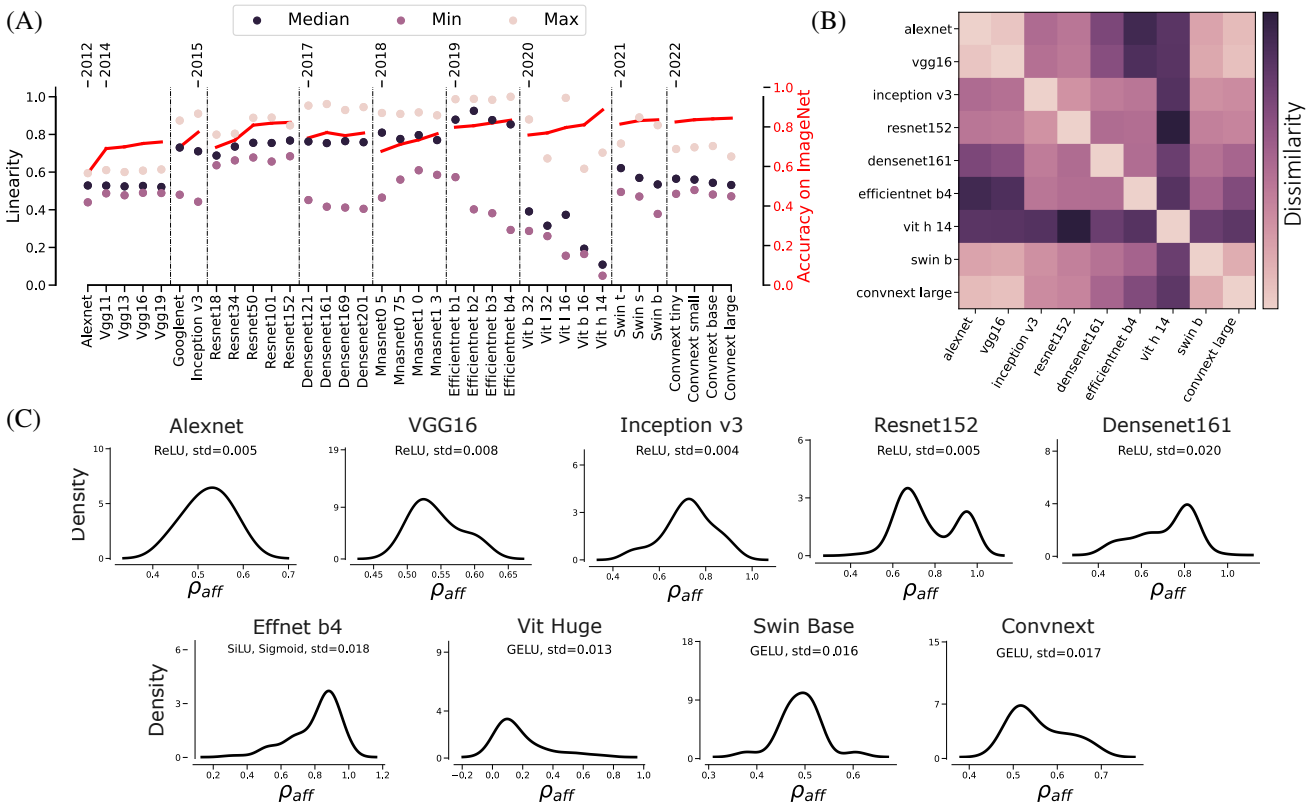

Figure 5. **(A)** Median, minimum, and maximum values of non-linearity signatures of the different architectures spanning a decade (2012-2022) of computer vision research. **(B)-(C)** Pairwise similarities and densities of affinity scores' distributions of representative architectures.

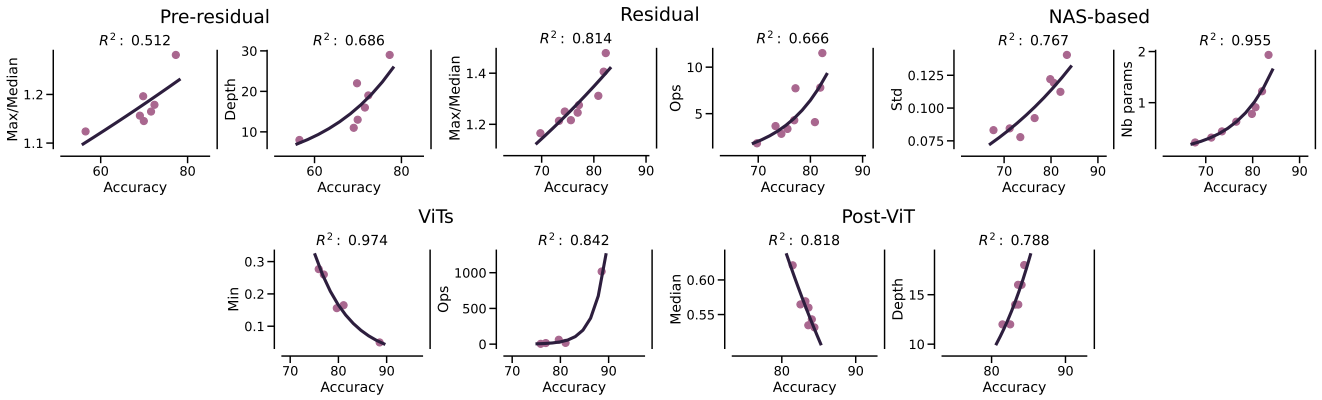

Figure 6. Best found dependency between the different statistics extracted from the non-linearity signatures of the DNN families and their respective Imagenet-1K accuracy. The results are compared in terms of the R^2 score against the most precise of the other common DNN characteristics such as depth, size, and the GFLOPS.

also want to understand whether the non-linearity of DNNs can explain their performance better than the traditional characteristics such as the number of parameters, the number of giga floating point operations per second (GFLOPS), and the depth. From the results presented in Figure 6, we observe the following. First, the information extracted from the non-linearity signatures often correlates more with the final accuracy, than the usual DNN characteristics. This is the case for Residual networks (ResNets and DenseNets), ViTs, and vision models influenced by transformers (Post-ViT). Unsurprisingly, for models based on neural architecture search (NAS-based, i.e. EfficientNets and MNASNets) the number of parameters is the most informative metric as they are specifically designed to reach the highest accuracy with

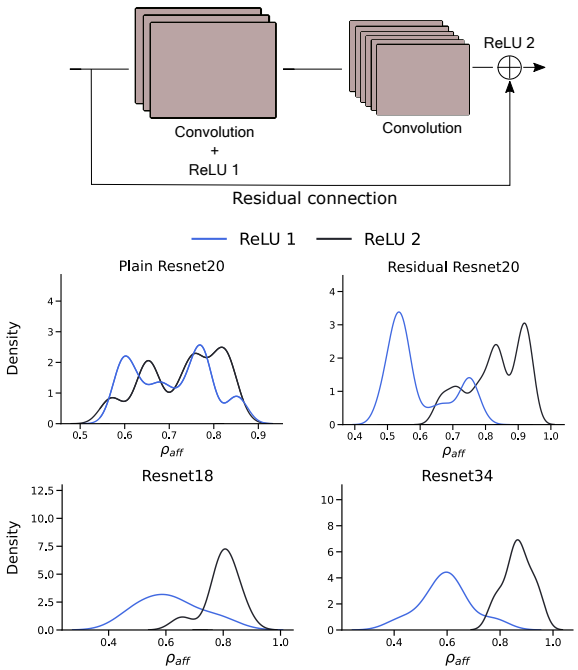

Figure 7. Comparing the same convnet with 20 layers when trained with (Residual Resnet20) and without (Plain Resnet20) residual connections (top row). Residual connections introduce a clear trend toward a bimodal distribution of affinity scores; the same effect is observed for Resnet18 and Resnet34 (bottom row).

the increasing model size and compute. For Pre-residual pure convolutional models (Alexnet, VGGs, Googlenet, and Inception), the spread of the non-linearity explains the accuracy increase similarly to depth. Second, we observe that all models preceding ViTs were implicitly optimizing the spread of their affinity score values to achieve better performance. After the arrival of the transformers, the observed trend is to increase either the median or the minimum values of the non-linearity. This suggests a fundamental shift in the implicit bias that the transformers carry.

Role of residual connections From Figure 5 (C), we note that Resnet is the first architecture that has a distinct bi-modal distribution of affinity scores. As this landmark architecture introduced for the first time the residual connections, we decide to explore what exact effect they have on the non-linearity signature of the same model trained with and without residual connection. Figure 7 reveals vividly that residual connections make the distribution of the affinity scores bimodal with one such mode centered around highly linear activation functions. This confirms in a principled way that residual connections indeed tend to enable the learning of the identity function just as suggested in the seminal work that proposed them [21]. Finally, we note that only ResNets exhibit this particular bi-modal distribution, thanks to the activation functions being exactly after the residual

Models	CKA	NORM	SPARSITY	ENTROPY	R^2
VGGs	0.0 (±0.05)	−0.67 (±0.06)	−0.18 (±0.03)	**-0.90** (±0.04)	−0.21 (±0.06)
ResNets	0.53 (±0.04)	−0.41 (±0.19)	**-0.68** (±0.02)	−0.38 (±0.12)	−0.48 (±0.24)
DenseNets	0.88 (±0.02)	−0.76 (±0.02)	**-0.89** (±0.02)	−0.66 (±0.03)	0.85 (±0.04)
MNASNets	**0.67** (±0.11)	−0.54 (±0.14)	−0.63 (±0.07)	−0.55 (±0.16)	0.45 (±0.17)
EffNets	**0.42** (±0.10)	−0.16 (±0.22)	−0.17 (±0.23)	−0.16 (±0.14)	0.21 (±0.12)
ViTs	−0.22 (±0.40)	**-0.67** (±0.20)	−0.09 (±0.56)	0.17 (±0.25)	−0.10 (±0.34)
Swins	−0.15 (±0.13)	**-0.53** (±0.10)	−0.26 (±0.17)	0.06 (±0.35)	−0.13 (±0.13)
Convnexts	0.69 (±0.08)	0.21 (±0.15)	0.23 (±0.16)	0.02 (±0.09)	**0.79** (±0.05)
Average	0.33 (±0.45)	**-0.44** (±0.34)	−0.32 (±0.42)	−0.31 (±0.39)	0.14 (±0.49)

Table 1. Pearson correlations between the non-linearity signature and other metrics, for all the architectures evaluated in this study. The highest absolute value in each group is reported in **bold**.

connections (illustrated in Figure 7 (top)). However, we still observe rather high maximum affinity score in all subsequent architectures that use residual connections.

Uniqueness of the affinity score No other metric extracted from the activation functions of the considered networks exhibits a strong consistent correlation with the non-linearity signature. To validate this claim, we compare in Tab. 1 the Pearson correlation between the non-linearity signature and several other metrics comparing the inputs and the outputs of the activation functions. We can see that for different models the non-linearity correlates with different metrics suggesting that it captures information that other metrics fail to capture consistently across all architectures. This becomes even more apparent when analyzing the individual correlation values (in Appendix E). Overall, the proposed affinity score and the non-linearity signatures derived from it offer a unique perspective on the developments in ML.

5. Discussions

We proposed the affinity score, the first sound approach to measure non-linearity of activation functions in neural networks, and defined their non-linearity signature based on it. We then showed what impacts the non-linearity of activation functions, both in isolation and as part of a randomly initialized DNN. We further used non-linearity signatures to provide a meaningful overview of the evolution of neural architectures proposed over the last decade with clear interpretable patterns. We showed the distinct features of the different landmark architectures and highlighted that the affinity score is unique in identifying them as it does not correlate with any other reasonable candidate metric.

Future work and applications Our work can be applied to study non-linearity of large-scale foundation models to better understand the effect of different architectural and training choices. As the computation of the affinity score is fully differentiable, one future application can be to control the bias induced by each activation function through a regularization of the affinity score. Deviations between signatures of the same architecture can be used to monitor or identify the peculiarities of different training strategies since they lead to different non-linearity signatures (see Appendix G).

Acknowledgements

This work was supported by the EU Horizon project ELIAS (No. 101120237) and by the French National Research Agency (ANR) under the PEPR IA FOUNDRY (ANR-23-PEIA-0003), the LIMPID (ANR-20-CE23-0028) and the FAR-SEE (ANR-24-CE23-0921) projects. We also thank Florence d'Alché-Buc, Rémi Flamary and Devis Tuia for providing feedback on early versions of the manuscript.

References

[1] Andrew R. Barron. Approximation and estimation bounds for artificial neural networks. *Mach. Learn.*, 14(1):115–133, 1994. 2

[2] Peter L. Bartlett, Nick Harvey, Christopher Liaw, and Abbas Mehrabian. Nearly-tight vc-dimension and pseudodimension bounds for piecewise linear neural networks. *Journal of Machine Learning Research*, 20(63):1–17, 2019. 2

[3] Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*, 2020. 1

[4] Mathilde Caron, Ishan Misra, Julien Mairal, Priya Goyal, Piotr Bojanowski, and Armand Joulin. Unsupervised learning of visual features by contrasting cluster assignments. *Advances in neural information processing systems*, 33:9912–9924, 2020. 9

[5] Mathilde Caron, Hugo Touvron, Ishan Misra, Hervé Jégou, Julien Mairal, Piotr Bojanowski, and Armand Joulin. Emerging properties in self-supervised vision transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 9650–9660, 2021. 9

[6] G. Cybenko. Approximation by superpositions of a sigmoidal function. *Mathematics of Control, Signals, and Systems (MCSS)*, 2(4):303–314, 1989. 2

[7] MohammadReza Davari, Stefan Horoi, Amine Natik, Guillaume Lajoie, Guy Wolf, and Eugene Belilovsky. Reliability of CKA as a similarity measure in deep learning. In *ICLR*, 2023. 5

[8] Julien Denize, Jaonary Rabarisoa, Astrid Orcesi, Romain Hérault, and Stéphane Canu. Similarity contrastive estimation for self-supervised soft contrastive learning. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 2706–2716, 2023. 9

[9] 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 *ICLR*, 2021. 1

[10] Shiv Ram Dubey, Satish Kumar Singh, and Bidyut Baran Chaudhuri. Activation functions in deep learning: A comprehensive survey and benchmark. *Neurocomput.*, 503(C): 92–108, 2022. 5, 3

[11] Ronen Eldan and Ohad Shamir. The power of depth for feedforward neural networks. In *29th Annual Conference on Learning Theory*, pages 907–940, 2016. 2

[12] Stefan Elfwing, Eiji Uchibe, and Kenji Doya. Sigmoid-weighted linear units for neural network function approximation in reinforcement learning. *Neural networks*, 107: 3–11, 2018. 3

[13] Thomas Elsken, Jan Hendrik Metzen, and Frank Hutter. Neural architecture search: A survey. *Journal of Machine Learning Research*, 20(55):1–21, 2019. 6

[14] Matthias Gelbrich. On a formula for the l2 wasserstein metric between measures on euclidean and hilbert spaces. *Mathematische Nachrichten*, 147(1):185–203, 1990. 4

[15] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, pages 249–256, 2010. 5

[16] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, pages 315–323, Fort Lauderdale, FL, USA, 2011. PMLR. 3

[17] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. MIT Press, 2016. 1

[18] Ingo Gühring, Mones Raslan, and Gitta Kutyniok. Expressivity of deep neural networks. *arXiv:2007.04759*, 2020. 1

[19] Soufiane Hayou, Arnaud Doucet, and Judith Rousseau. On the impact of the activation function on deep neural networks training. In *Proceedings of the 36th International Conference on Machine Learning*, pages 2672–2680, 2019. 5

[20] 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*, page 1026–1034, 2015. 1

[21] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2016. 1, 6, 8, 9

[22] Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 9729–9738, 2020. 9

[23] Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Wei Chen. Deberta: Decoding-enhanced bert with disentangled attention. In *Proceedings of the International Conference on Learning Representations*, 2021. 1

[24] Dan Hendrycks and Kevin Gimpel. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*, 2016. 3

[25] Kurt Hornik. Multilayer feedforward networks are universal approximators. *Neural Networks*, 2(5):359–366, 1989. 2

[26] Andrew Howard, Mark Sandler, Grace Chu, Liang-Chieh Chen, Bo Chen, Mingxing Tan, Weijun Wang, Yukun Zhu, Ruoming Pang, Vijay Vasudevan, et al. Searching for mobilenetv3. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 1314–1324, 2019. 3

[27] Andrew G. Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Wei Wang, Wenhan Weng, Marco Andreetto, and Hartwig Adam. Mobilenets: Efficient convolutional neural networks for mobile vision applications. In *Proceedings of the 2017 IEEE Conference on Computer Vision and Pattern Recognition*, pages 4200–4210. IEEE, 2017. 3

[28] Gao Huang, Zhuang Liu, Laurens Van Der Maaten, and Kilian Q Weinberger. Densely connected convolutional networks. *arXiv preprint arXiv:1608.06993*, 2017. 1, 6

[29] Michael I Jordan and Tom M Mitchell. Machine learning: Trends, perspectives, and prospects. *Science*, 349(6245):255–260, 2015. 1

[30] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *ICML*, pages 3519–3529. PMLR, 2019. 5, 7

[31] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25:1097–1105, 2012. 1, 6

[32] Kurt and Hornik. Approximation capabilities of multilayer feedforward networks. *Neural Networks*, 4(2):251–257, 1991. 2

[33] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. Deep learning. *Nature*, 521(7553):436–444, 2015. 1

[34] Olivier Ledoit and Michael Wolf. Honey, i shrunk the sample covariance matrix. *Journal of Portfolio Management*, 30(4): 110–119, 2004. 4

[35] Geert Litjens, Thijs Kooi, Babak Ehteshami Bejnordi, Arnaud A.A. Setio, Francesco Ciompi, Mohsen Ghafoorian, Jeroen A.W.M. van der Laak, Bram van Ginneken, and Clara I. Sánchez. A survey on deep learning in medical image analysis. *Medical image analysis*, 42:60–88, 2017. 1

[36] Ze Liu, Yutong Lin, Yue Cao, Han Hu, Yixuan Wei, Zheng Zhang, Stephen Lin, and Baining Guo. Swin transformer: Hierarchical vision transformer using shifted windows. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021. 6

[37] Zhuang Liu, Hanzi Mao, Chao-Yuan Wu, Christoph Feichtenhofer, Trevor Darrell, and Saining Xie. A convnet for the 2020s. *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022. 6

[38] Zhou Lu, Hongming Pu, Feicheng Wang, Zhiqiang Hu, and Liwei Wang. The expressive power of neural networks: a view from the width. In *Advances in Neural Information Processing Systems*, page 6232–6240, 2017. 2

[39] Andrew L Maas, Awni Y Hannun, and Andrew Y Ng. Rectifier nonlinearities improve neural network acoustic models. In *Proceedings of the ICML Workshop on Deep Learning for Audio, Speech and Language Processing*, 2013. 3

[40] TorchVision maintainers and contributors. Torchvision: Pytorch's computer vision library. *GitHub repository*, 2016. 6

[41] Guido Montúfar, Razvan Pascanu, KyungHyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks. In *NeurIPS*, pages 2924–2932, 2014. 2

[42] Vinod Nair and Geoffrey E. Hinton. Rectified linear units improve restricted boltzmann machines. In *Proceedings of the International Conference on Machine Learning*, pages 807–814, 2010. 4

[43] OpenAI. Ai and compute. 2018. Accessed: March 13, 2024. 1

[44] OpenAI. Gpt-4 technical report. *ArXiv*, abs/2303.08774, 2023. 1

[45] Gabriel Peyré and Marco Cuturi. Computational optimal transport. *Foundations and Trends® in Machine Learning*, 11:355–607, 2019. 3

[46] George Philipp. The nonlinearity coefficient - A practical guide to neural architecture design. *CoRR*, abs/2105.12210, 2021. 6

[47] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep learning dynamics and interpretability. In *NIPS'17*, page 6078–6087, 2017. 5

[48] Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks. In *Proceedings of the International Conference on Machine Learning*, pages 2847–2854, 2017. 2

[49] D. E. Rumelhart, G. E. Hinton, and R. J. Williams. Learning representations by back-propagating errors. *Nature*, 323 (6088):533–536, 1986. 4, 3

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

[51] Itay Safran and Ohad Shamir. Depth-width tradeoffs in approximating natural functions with neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, pages 2979–2987, 2017. 2

[52] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26 (1):43–49, 1978. 6, 7

[53] Mert Bülent Sarıyıldız, Yannis Kalantidis, Karteek Alahari, and Diane Larlus. No reason for no supervision: Improved generalization in supervised models. In *The Eleventh International Conference on Learning Representations*, 2023. 9

[54] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015. 1

[55] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 1, 6

[56] Cyril S Smith and Martin Knott. Note on the optimal transportation of distributions. *Journal of Optimization Theory and Applications*, 52(2):323–329, 1987. 3

[57] Emma Strubell, Ananya Ganesh, and Andrew McCallum. Energy and policy considerations for deep learning in nlp. *arXiv preprint arXiv:1906.02243*, 2019. 1

[58] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. *arXiv preprint arXiv:1409.4842*, 2014. 6

[59] Christian Szegedy, Sergey Ioffe, Vincent Vanhoucke, and Arman Alemi. Rethinking the inception architecture for computer vision. *arXiv preprint arXiv:1512.00567*, 2016. 1, 6

[60] Mingxing Tan and Quoc Le. EfficientNet: Rethinking model scaling for convolutional neural networks. In *Proceedings of the International Conference on Machine Learning*, pages 6105–6114, 2019. 6

[61] Mingxing Tan, Bo Chen, Ruoming Pang, Vijay Vasudevan, Mark Sandler, Andrew Howard, and Quoc V. Le. Mnasnet: Platform-aware neural architecture search for mobile. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2019. 6

[62] Damien Teney, Armand Mihai Nicolicioiu, Valentin Hartmann, and Ehsan Abbasnejad. Neural redshift: Random networks are not random functions. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4786–4796, 2024. 5

[63] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. 1

[64] Gal Vardi, Gilad Yehudai, and Ohad Shamir. On the optimal memorization power of relu neural networks. In *The Tenth International Conference on Learning Representations, ICLR*, 2022. 2

[65] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017. 1

[66] Guangrun Wang, Keze Wang, Guangcong Wang, Philip HS Torr, and Liang Lin. Solving inefficiency of self-supervised representation learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9505–9515, 2021. 9

[67] Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. In *Procedings of the British Machine Vision Conference 2016*. British Machine Vision Association, 2016. 9

[68] Mingkai Zheng, Shan You, Fei Wang, Chen Qian, Changshui Zhang, Xiaogang Wang, and Chang Xu. Ressl: Relational self-supervised learning with weak augmentation. *Advances in Neural Information Processing Systems*, 34:2543–2555, 2021. 9

From Alexnet to Transformers: Measuring the Non-linearity of Deep Neural Networks with Affine Optimal Transport

Supplementary Material

A. Main theoretical results

A.1. Proofs

In this section, we provide proofs of the main theoretical results from the paper.

Corollary 3.2. Without loss of generality, let $X, Y \in \mathcal{P}_2(\mathbb{R}^d)$ be centered, and such that $Y = TX$, where T is a positive semi-definite linear transformation. Then, T is the OT map from X to Y.

Proof. We first prove that we can consider centered distributions without loss of generality. To this end, we note that

$$W_2^2(X,Y) = W_2^2(X - \mathbb{E}[X], Y - \mathbb{E}[Y]) + \|\mathbb{E}[X] - \mathbb{E}[Y]\|^2, \tag{6}$$

implying that splitting the 2-Wasserstein distance into two independent terms concerning the L^2 distance between the means and the 2-Wasserstein distance between the centered measures.

Furthermore, if we have an OT map T' between $X - \mathbb{E}[X]$ and $Y - \mathbb{E}[Y]$, then

$$T(x) = T'(x - \mathbb{E}[X]) + \mathbb{E}[Y], \tag{7}$$

is the OT map between X and Y.

To prove the statement of the Corollary, we now need to apply Theorem 3.1 to the convex $\phi(x) = x^T T x$, where T is positive semi-definite. □

Theorem 3.3. Let $X, Y \in \mathcal{P}_2(\mathbb{R}^d)$ be centered and $Y = TX$ for a positive definite matrix T. Let $N_X \sim \mathcal{N}(\mu(X), \Sigma(X))$ and $N_Y \sim \mathcal{N}(\mu(Y), \Sigma(Y))$ be their normal approximations where μ and Σ denote mean and covariance, respectively. Then, $W_2(N_X, N_Y) = W_2(X, Y)$ and $T = T_{\text{aff}}$, where T_{aff} is the OT map between N_X and N_Y and can be calculated in closed-form

$$\begin{aligned} T_{\text{aff}}(x) &= Ax + b, \\ A &= \Sigma(Y)^{\frac{1}{2}} \left(\Sigma(Y)^{\frac{1}{2}} \Sigma(X) \Sigma(Y)^{\frac{1}{2}}\right)^{-\frac{1}{2}} \Sigma(Y)^{\frac{1}{2}}, \\ b &= \mu(Y) - A\mu(X). \end{aligned} \tag{8}$$

Proof. Corollary 3.2 states that T is an OT map, and

$$\Sigma(TN_X) = T\Sigma(X)T = \Sigma(Y).$$

Therefore, $TN_X = N_Y$, and by Theorem 3.1, T is the OT map between N_X and N_Y. Finally, we compute

$$\begin{aligned} W_2^2(N_X, N_Y) &= \operatorname{Tr}[\Sigma(X)] + \operatorname{Tr}[T\Sigma(X)T] \\ &\quad - 2\operatorname{Tr}[T^{\frac{1}{2}}\Sigma(X)T^{\frac{1}{2}}] \\ &= \underset{T:T(X)=Y}{\arg\min}\ \mathbb{E}_X[\|X - T(X)\|^2] \\ &= W_2^2(X, Y). \end{aligned}$$

□

Proposition 3.5. Let $X, Y \in \mathcal{P}_2(\mathbb{R}^d)$ and N_X, N_Y be their normal approximations. Then,

1. $|W_2(N_X, N_Y) - W_2(X, Y)| \leq \frac{2\operatorname{Tr}\left[(\Sigma(X)\Sigma(Y))^{\frac{1}{2}}\right]}{\sqrt{\operatorname{Tr}[\Sigma(X)] + \operatorname{Tr}[\Sigma(Y)]}}$.
2. For T_{aff} as in (4), $W_2(T_{\text{aff}}X, Y) \leq \sqrt{2}\operatorname{Tr}\left[\Sigma(Y)\right]^{\frac{1}{2}}$.

Proof. By Theorem 3.4, we have $W_2(N_X, N_Y) \leq W_2(X, Y)$. On the other hand,

$$\begin{aligned} W_2^2(X, Y) &= \min_{\gamma \in \text{ADM}(X,Y)} \int_{\mathbb{R}^d \times \mathbb{R}^d} \|x - y\|^2 d\gamma(x, y) \\ &\leq \int_{\mathbb{R}^d \times \mathbb{R}^d} \left(\|x\|^2 + \|y\|^2\right) d\gamma(x, y) \\ &= \operatorname{Tr}[\Sigma(X)] + \operatorname{Tr}[\Sigma(Y)]. \end{aligned}$$

Combining the above inequalities, we get

$$\begin{aligned} &|W_2(N_X, N_Y) - W_2(X, Y)| \\ &\leq \left|\sqrt{\operatorname{Tr}[\Sigma(X)] + \operatorname{Tr}[\Sigma(Y)]} - W_2(N_X, N_Y)\right|. \end{aligned}$$

Let $a = \operatorname{Tr}[\Sigma(X)] + \operatorname{Tr}[\Sigma(Y)]$, and so $W_2^2(N_X, N_Y) = a - b$, where $b = 2\operatorname{Tr}\left[(\Sigma(X)\Sigma(Y))^{\frac{1}{2}}\right]$. Then the RHS of can be written as

$$\left|\sqrt{a} - \sqrt{a - b}\right| = \frac{|a - (a - b)|}{\sqrt{a} + \sqrt{a - b}} \leq \frac{b}{\sqrt{a}},$$

where the inequality follows from positivity of $W_2(N_X, N_Y) = \sqrt{a - b}$. Letting $X = T_{\text{aff}}X$ in the obtained bound gives 2). □

A.2. Analytical expression in 1D for ReLU

Let $X \sim \mathcal{U}[b, a]$, $b < 0$ and $a > 0$. Furthermore, let $f : x \mapsto \text{ReLU}(x) = x\chi(x \geq 0)$ and $Y = f(X)$.

We are interested in whether the affinity score

$$\rho_{\text{aff}}(X,Y) = 1 - \frac{W_2(T_{\text{aff}}(X),Y)}{\sqrt{2\,\text{Tr}\left[\Sigma(Y)\right]}} \tag{9}$$

is symmetric wrt. a and b. Here W_2 is the 2-Wasserstein distance between the laws of two random variables and T_{aff} is the affine transport map between X and Y given by

$$\begin{aligned} T_{\text{aff}}(x) &= A_{\text{aff}}x + b_{\text{aff}}, \\ A_{\text{aff}} &= \Sigma(Y)^{\frac{1}{2}}\left(\Sigma(Y)^{\frac{1}{2}}\Sigma(X)\Sigma(Y)^{\frac{1}{2}}\right)^{-\frac{1}{2}}\Sigma(Y)^{\frac{1}{2}}, \\ b_{\text{aff}} &= \mu(Y) - A_{\text{aff}}\mu(X). \end{aligned} \tag{10}$$

Source Mean and Variance. Recall the formulae for mean and variance for a uniform distribution

$$\mu(X) = \frac{a+b}{2}, \quad \Sigma(X) = \frac{(a-b)^2}{12}$$

Target Mean and Variance. This time we are forced to compute a bit. Let's tart with the mean

$$\begin{aligned} \mu(Y) &= \frac{1}{a-b}\int_b^a f(x)dx \\ &= \frac{1}{a-b}\left(\int_b^0 0dx + \int_0^a xdx\right) \\ &= \frac{a^2}{2(a-b)}. \end{aligned} \tag{11}$$

Moving on to the variance

$$\begin{aligned} \Sigma(Y) &= \frac{1}{a-b}\int_b^a \left(f(x)-\mu(Y)\right)^2 dx \\ &= \frac{1}{a-b}\left(\int_b^0 \mu(Y)^2 dx + \int_0^a \left(x-\mu(Y)\right)^2 dx\right) \\ &= \frac{1}{a-b}\left((a-b)\mu(Y)^2 - a^2\mu(Y) + \frac{1}{3}a^3\right) \\ &= \left(\frac{2a}{3} - \mu(Y)\right)\mu(Y) \\ &= \frac{a^3(a-4b)}{12(a-b)^2} \end{aligned} \tag{12}$$

Affine transport map. Substituting the computed statistics into equation 9 and abusing their scalar nature, we get

$$\begin{aligned} A_{\text{aff}} &= \frac{\Sigma(Y)^{\frac{1}{2}}}{\Sigma(X)^{\frac{1}{2}}} \\ &= \frac{\sqrt{a^3(a-4b)}}{(a-b)^2}, \\ b_{\text{aff}} &= \mu(Y) - A_{\text{aff}}\mu(X), \\ &= \frac{a}{2(a-b)}\left(a - \left(\frac{a+b}{a-b}\right)\left(\sqrt{a(a-4b)}\right)\right) \end{aligned} \tag{13}$$

2-Wasserstein Distance. Recall that the 2-Wasserstein distance between scalars is simply a sorting problem: sort the source and target and match the elements with similar indices. Luckily in our case, both T_{aff} and $f = ReLU$ preserve order as increasing functions, and hence

$$\begin{aligned} W_2^2\left(T_{\text{aff}}(X),Y\right) &= W_2^2\left(T_{\text{aff}}(X), ReLU(X)\right) \\ &= \frac{1}{a-b}\int_b^a \left(T_{\text{aff}}(x) - ReLU(x)\right)^2 dx \end{aligned} \tag{14}$$

Before continuing the computation, remember that due to affine transport, $\mu(T(X)) = \mu(Y)$ and $\Sigma(T(X)) = \Sigma(Y)$. Therefore

$$\begin{aligned} \Sigma(T_{\text{aff}}(X)) &= \Sigma(Y) \\ \Rightarrow \mathbb{E}\left[T_{\text{aff}}(X)^2\right] - \mathbb{E}\left[T_{\text{aff}}(X)\right]^2 &= \mathbb{E}\left[Y^2\right] - \mathbb{E}\left[Y\right]^2 \\ \Rightarrow \mathbb{E}\left[T_{\text{aff}}(X)^2\right] &= \mathbb{E}\left[Y^2\right]. \end{aligned} \tag{15}$$

Using this, we can continue the computation in equation 14

$$\begin{aligned} &W_2^2(T_{\text{aff}}(X),Y) \\ =&\frac{2}{a-b}\left(\int_b^a ReLU^2(x)dx - \int_b^a ReLU(x)T_{\text{aff}}(x)dx\right) \\ =&\frac{2}{a-b}\left(\int_0^a x^2dx - \int_0^a (A_{\text{aff}}x^2 + b_{\text{aff}}x)dx\right) \\ =&\frac{a^2}{3(a-b)}\left(2a\left(1-A_{\text{aff}}\right) - 3b_{\text{aff}}\right) \\ =&\frac{2}{3}\mu(Y)(2a(1-A_{\text{aff}}) - 3b_{\text{aff}}) \\ =&\frac{a^3}{6(a-b)^2}\left((a-4b) + \sqrt{a(a-4b)}\left(\frac{-a+3b}{a-b}\right)\right). \end{aligned}$$

B. Affinity scores of other popular activation functions

Many works aimed to improve the way how the non-linearity – represented by activation functions – can be defined in DNNs. As an example, a recent survey on the commonly used activation functions in deep neural networks [10] identifies over 40 activation functions with first references to sigmoid dating back to the seminal paper [49] published in late 80s. The fashion for activation functions used in deep neural networks evolved over the years in a substantial way, just as the neural architectures themselves. Saturating activations, such as sigmoid and hyperbolic tan, inspired by computational neuroscience were a number one choice up until the arrival of rectifier linear unit (ReLU) in 2010. After being the workhorse of many famous models over the years, the arrival of transformers popularized Gaussian Error Linear Unit (GELU) which is now commonly used in many large language models including GPTs.

We illustrate in Figure 8 the affinity scores obtained after a single pass of the data through the following activation functions: Sigmoid, ReLU [16], GELU [24], ReLU6 [27], LeakyReLU [39] with a default value of the slope, Tanh, HardTanh, SiLU [12], and HardSwish [26]. As the non-linearity of activation functions depends on the domain of their input, we fix 20 points in their domain equally spread in $[-20, 20]$ interval. We use these points as means $\{m_i\}_{i=1}^{20}$ of Gaussian distributions from which we sample 1000 points in $\mathbb{R}^{300}$ with standard deviation (std) σ taking values in $[2, 1, 0.5, 0.25, 0.1, 0.01]$. Each sample denoted by $X_{m_i}^{\sigma_j}$ is then passed through the activation function act $\in \{\text{sigmoid}, \text{ReLU}, \text{GELU}\}$ to obtain $\rho_{\text{aff}}^{m_i,\sigma_j} := \rho_{\text{aff}}(X_{m_i}^{\sigma_j}, \text{act}(X_{m_i}^{\sigma_j}))$. Larger std values make it more likely to draw samples that are closer to the region where the studied activation functions become non-linear. We present the obtained results in Figure S2 where each of 20 boxplots showcases median$(\rho_{\text{aff}}^{m_i,\sigma_\cdot})$ values with 50% confidence intervals and whiskers covering the whole range of obtained values across all σ_j.

This plot allows us to derive several important conclusions. We observe that each activation function can be characterized by 1) the lowest values of its non-linearity obtained for some subdomain of the considered interval and 2) the width of the interval in which it maintains its non-linearity. We note that in terms of 1) both GELU and ReLU may attain affinity scores that are close to 0, which is not the case for Sigmoid. For 2), we observe that the non-linearity of Sigmoid and GELU is maintained in a wide range, while for ReLU it is rather narrow. We can also see a distinct pattern of more modern activation functions, such as SiLU and HardSwish having a stronger non-linearity pattern in large subdomains. We also note that despite having a shape similar to Sigmoid, Tanh may allow for much lower affinity scores. Finally, the variations of ReLU seem to have a very similar shape with LeakyReLU being on average more linear than ReLU and ReLU6.

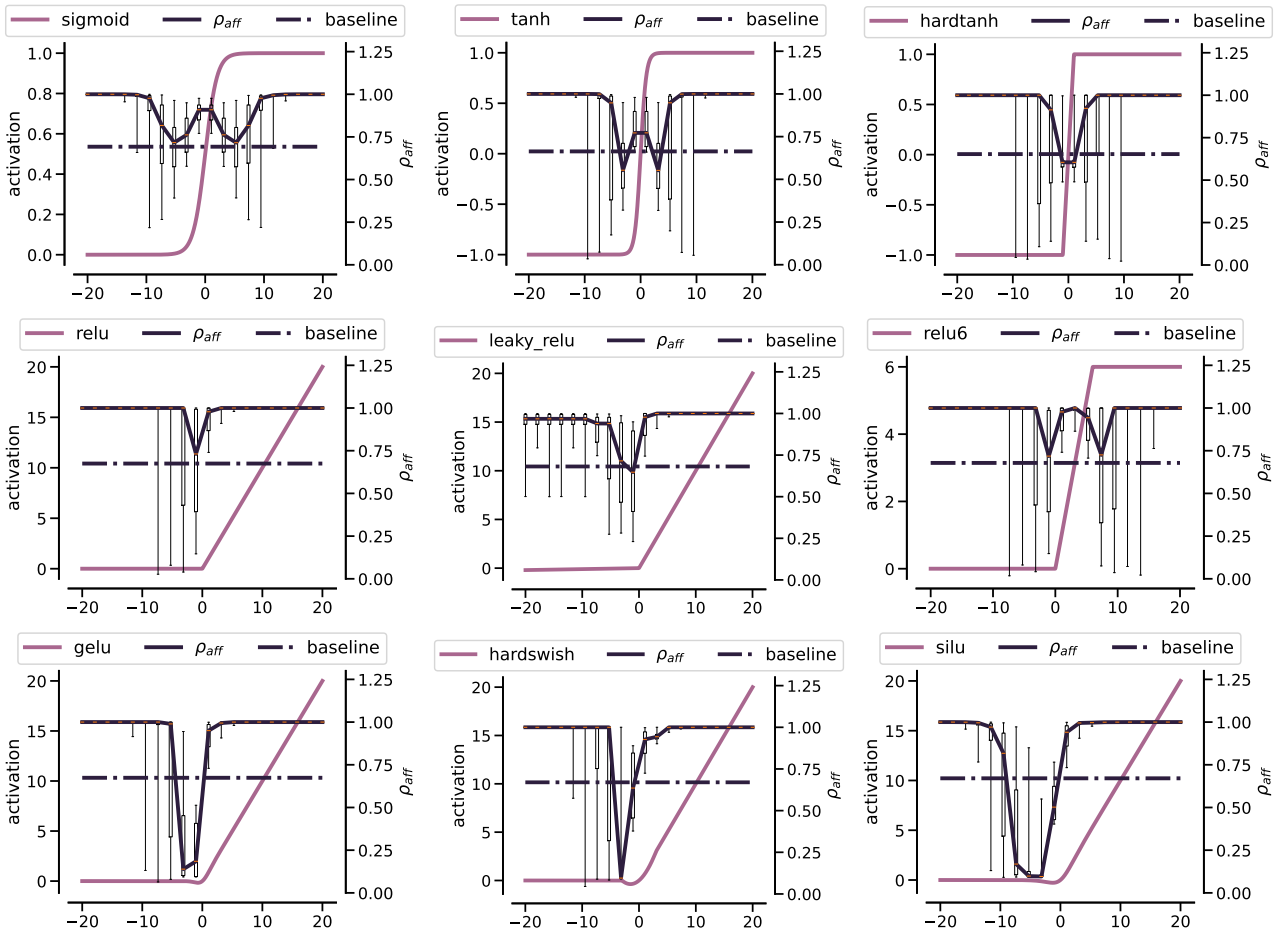

Figure 8. Median affinity scores of Sigmoid, ReLU, GELU, ReLU6, LeakyReLU with a default value of slope, Tanh, HardTanh, SiLU, and HardSwish obtained across random draws from Gaussian distribution with a sliding mean and varying stds used as their input. Whiskers of boxplots show the whole range of values obtained for each mean across all stds. The baseline value is the affinity score obtained for a sample covering the whole interval. The ranges and extreme values of each activation function over its subdomain are indicative of its non-linearity limits.

C. Implementation details

Dimensionality reduction Manipulating 4-order tensors is computationally prohibitive and thus we need to find an appropriate lossless function r to facilitate this task. One possible choice for r may be a vectorization operator that flattens each tensor into a vector. In practice, however, such flattening still leads to very high-dimensional data representations. In our work, we propose to use averaging over the spatial dimensions to get a suitable representation of the manipulated tensors. In Figure 9 (top), we show that the affinity score is robust wrt such an averaging scheme and maintains the same values as its flattened counterpart.

Computational considerations The non-linearity signature requires calculating the affinity score over "wide" matrices. Indeed, after the reduction step is applied to a batch of n tensors of size $h \times w \times c$, we end up with matrices of size $n \times c$ where n may be much smaller than c. This is also the case when input tensors are 2D when the batch size is smaller than the dimensionality of the embedding space. To obtain a well-defined estimate of the covariance matrix in this case, we use a known tool from the statistics literature called Ledoit-Wolfe shrinkage [34]. In Figure 9 (bottom), we show that shrinkage allows us to obtain a stable estimate of the affinity scores that remain constant in all regimes.

Robustness to batch size and different seeds In this section, we highlight the robustness of the non-linearity signature with respect to the batch size and the random seed used for training. To this end, we concentrate on VGG16 architecture and CIFAR10 dataset to avoid costly Imagenet retraining. In Figure 10, we present the obtained result

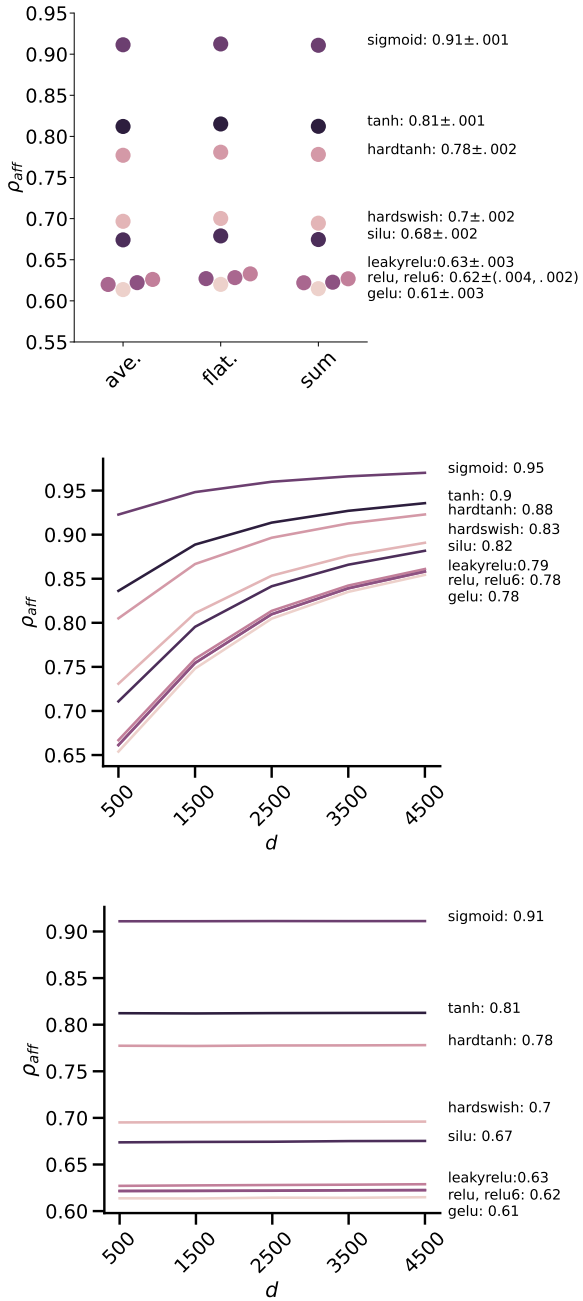

Figure 9. **(Top)** Affinity score is robust to the dimensionality reduction both when using averaging and summation over the spatial dimensions; **(Middle)** When $d > n$, sample covariance matrix estimation leads to a lack of robustness in the estimation of the affinity score; **(Bottom)** Shrinkage of the covariance matrix leads to constant values of the affinity scores with increasing d.

where the batch size was varied between 128 and 1024 with an increment of 128 (left plot) and when VGG16 model was retrained with seeds varying from 1 to 9 (right plot).

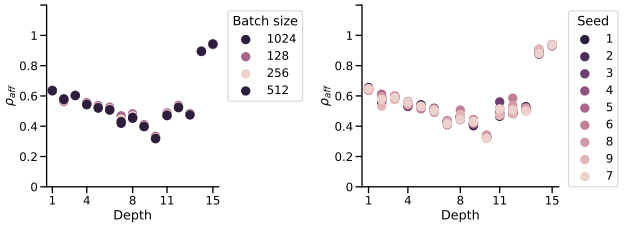

Figure 10. Non-linearity signature of VGG16 on CIFAR10 with a varying batch size (left) and when retrained from 9 different random seeds (right).

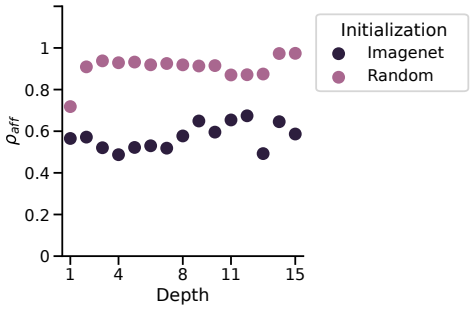

Figure 11. Non-linearity signatures of VGG16 on CIFAR10 in the beginning and end of training on Imagenet.

The obtained results show that the affinity score is robust to these parameters suggesting that the obtained results are not subject to a strong stochasticity.

Impact of training Finally, we also show how a non-linearity signature of a VGG16 model looks like at the beginning and in the end of training on Imagenet. We extract its non-linearity signature at initialization when making a feedforward pass over the whole CIFAR10 dataset and compare it to the non-linearity signature obtained in the end. In Figure 11, we can see that at initialization the network's non-linearity signature is increasing, reaching almost a perfectly linear pattern in the last layers. Training the network enhances the non-linearity in a non-monotone way. Importantly, it also highlights that the non-linearity signature is capturing information from the training process.

D. Raw signatures

In Figure 12, we portray the raw non-linearity signatures of several representative networks studied in the main paper. We use different color codes for distinct activation functions appearing repeatedly in the considered architecture (for instance, every first ReLU in a residual block of a Resnet). We also indicate the mean standard deviation of the affinity scores over batches in the title.

We see that the non-linearities across ReLU activations in all of Alexnet's 8 layers remain stable. Its successor, VGG network, reveals tiny, yet observable, variations in the non-linearity propagation with increasing depth and, slightly lower overall non-linearity values. We attribute this to the decreased size of the convolutional filters (3x3 vs. 7x7). The Googlenet architecture was the first model to consider learning features at different scales in parallel within the so-called inception modules. This add more variability as affinity scores of activation in Googlenet vary between 0.6 and 0.9. Despite being almost 20 times smaller than VGG16, the accuracy of Googlenet on Imagenet remains comparable, suggesting that increasing and varying the linearity is a way to have high accuracy with a limited computational complexity compared to predecessors. This finding is further confirmed with Inception v3 that pushed the spread of the affinity score toward being more linear in some hidden layers. When comparing this behavior with Alexnet, we note just how far we are from it. Resnets achieve the same spread of values of the non-linearity but in a different, and arguably, simpler way. Indeed, the activation after the skip connection exhibits affinity scores close to 1, while the activations in the hidden layers remain much lower. Densenet, that connect each layer to all previous layers and not just to the one that precedes it, is slightly more non-linear than Resnet152, although the two bear a striking similarity: they both have an activation function that maintains the non-linearity low with increasing depth. Additionally, transition layers in Densenet act as linearizers and allow it to reset the non-linearity propagation in the network by reducing the feature map size. ViTs (Large with 16x16 and 32x32 patch sizes, and Huge with 14x14 patches) are all highly non-linear models to the degree yet unseen. Interestingly, as seen in Figure 13 the patch size affects the non-linearity propagation in a non-trivial way: for 16x16 size a model is more non-linear in the early layers, while gradually becoming more and more linear later, while 32x32 patch size leads to a plateau in the hidden layers of MLP blocks, with a steep change toward linearity only in the final layer. We hypothesize that attention modules in ViT act as a focusing lens and output the embeddings in the domain where the activation function is the most non-linear.

Finally, we explore the role of increasing depth for VGG and Resnet architectures. We consider VGG11, VGG13, VGG16 and VGG19 models in the first case, and Resnet18, Resnet34, Resnet50, Resnet101 and Resnet152. The results are presented in Figure 14 and Figure 15 for VGGs and Resnets, respectively. Interestingly, VGGs do not change their non-linearity signature with increasing depth. In the case of Resnets, we can see that the separation between more linear post-residual activations becomes more distinct and approaches 1 for deeper networks.

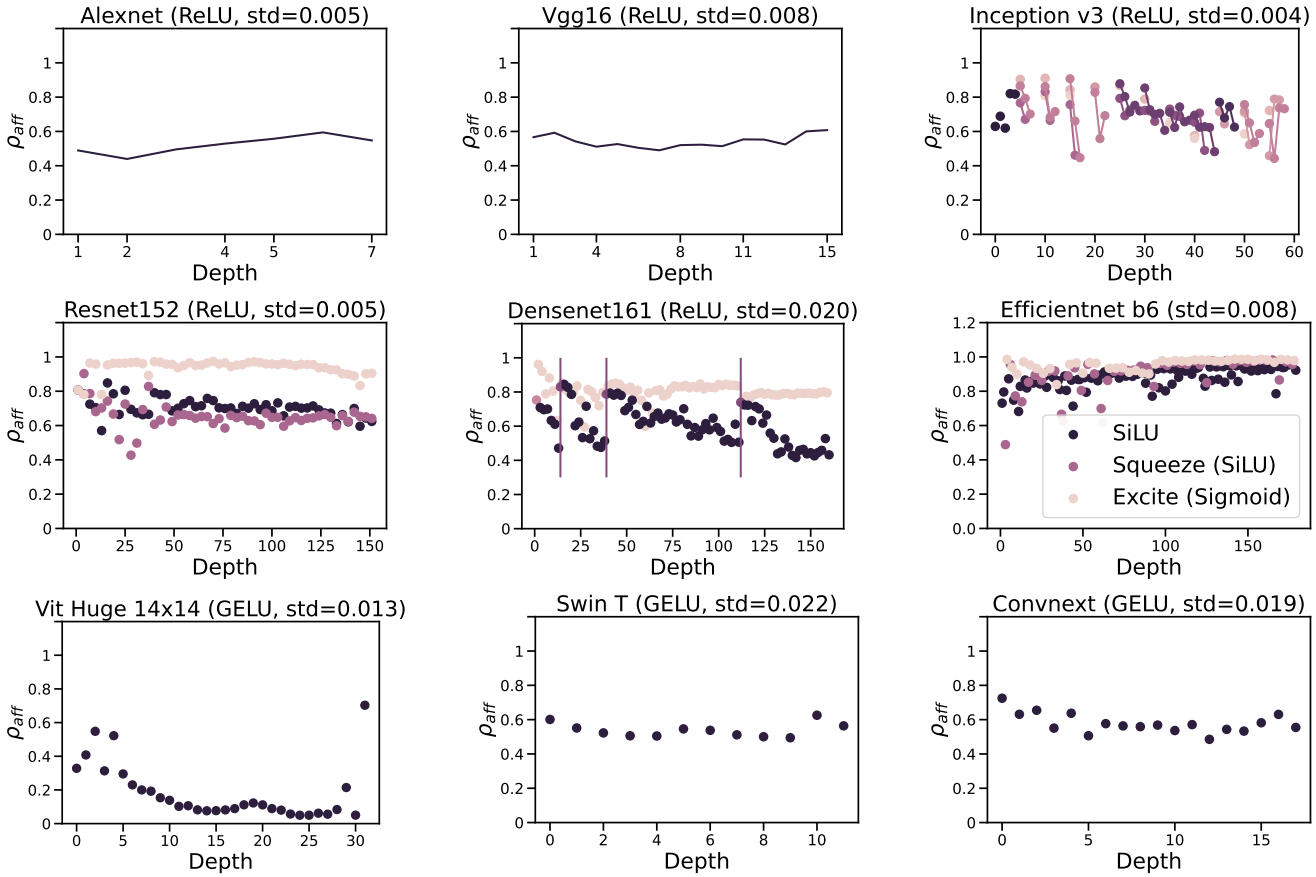

Figure 12. Raw non-linearity signatures of popular DNN architectures, plotted as affinity scores over the depth throughout the network.

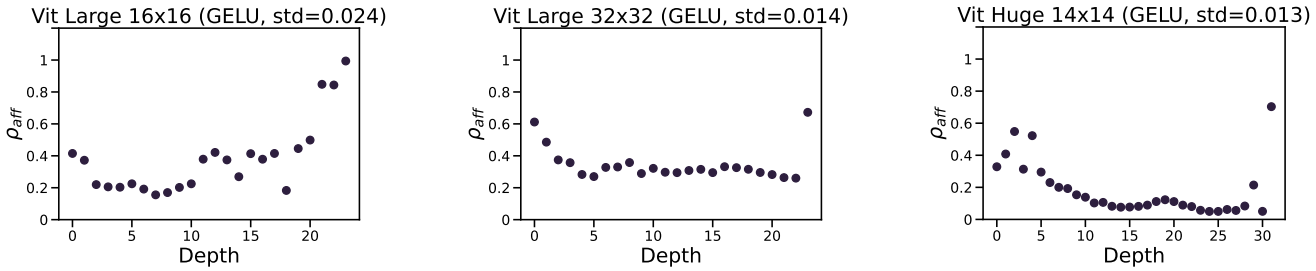

Figure 13. ViTs: Large ViT with 16x16 and 32x32 patch sizes and Huge ViT.

E. Detailed comparisons between architectures

We consider the following metrics as 1) the linear CKA [30] commonly used to assess the similarity of neural representations, the average change in 2) SPARSITY and 3) ENTROPY before and after the application of the activation function as well as the 4) Frobenius NORM between the input and output of the activation functions, and the 5) R^2 score between the linear model fitted on the input and the output of the activation function. We present in Tab. 2, the detailed values of Pearson correlations obtained for each architecture and all the metrics considered in this study. In Figure 16, we show the full matrix of pairwise DTW distances [52] obtained between architectures, then used to obtain the clustering presented alongside. For the latter, we applied multi-dimensional scaling algorithms to the linkage matrix of the 36 considered architectures.

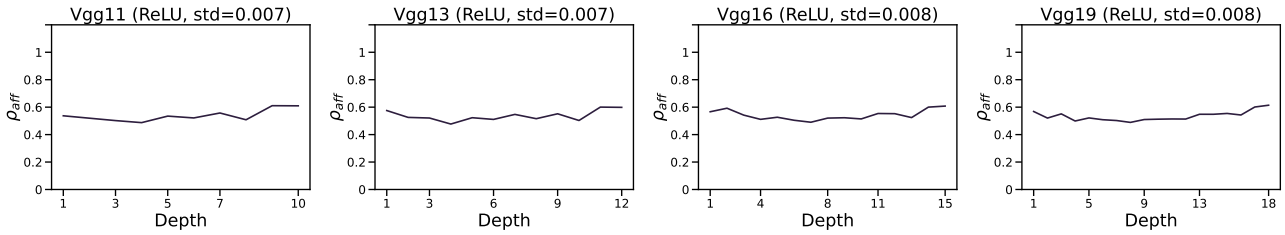

Figure 14. Impact of depth on the non-linearity signature of VGGs.

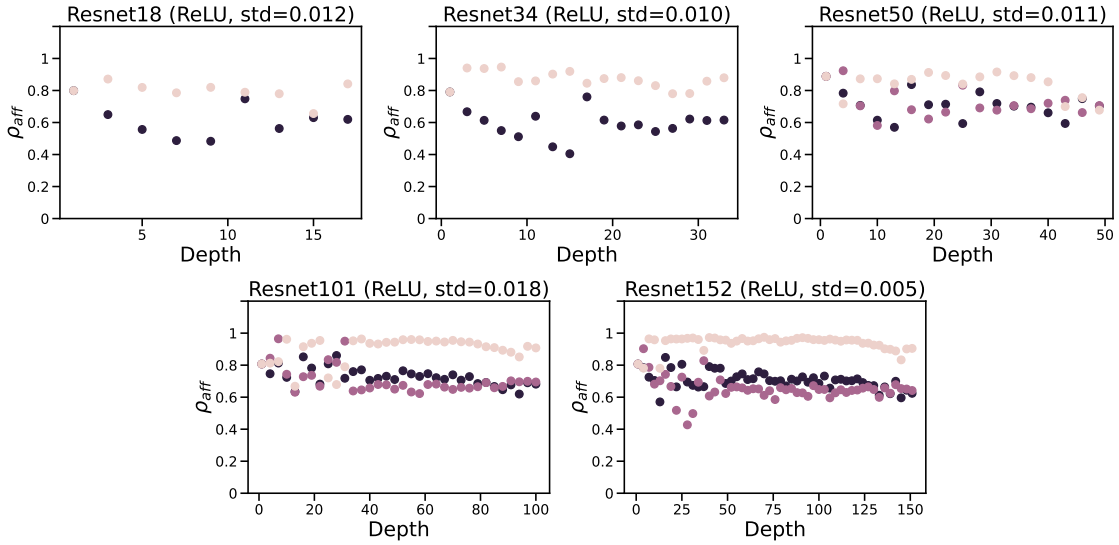

Figure 15. Impact of depth on the non-linearity signature of Resnets.

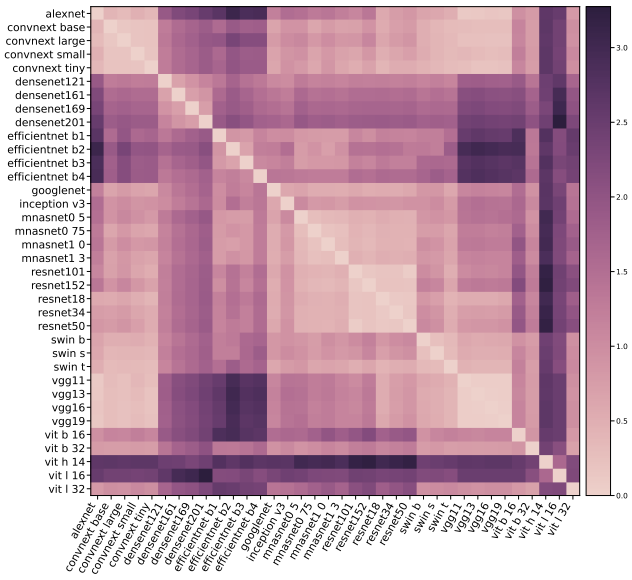

Figure 16. Full matrix of DTW distances between non-linearity signatures.

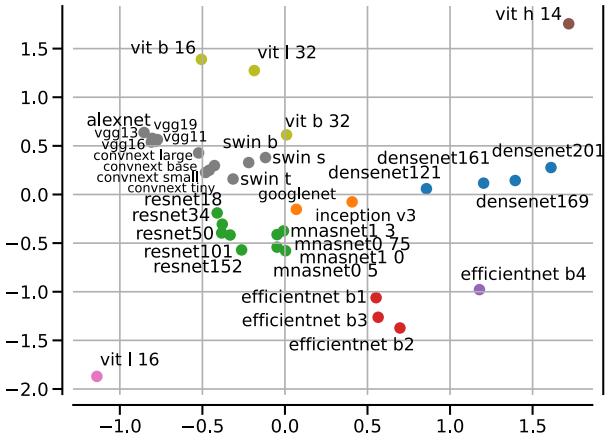

Figure 17. Multi-dimensional scaling of the linkage matrix obtained from the pairwise DTW distances between non-linearity signatures.

F. Results on more datasets

Below, we compare the results obtained on CIFAR10, CIFAR100 datasets as well as when the random data tensors

Model	CKA	Norm	Sparsity	Entropy	R^2
alexnet	-0.75	**-0.86**	0.14	-0.80	-0.41
vgg11	-0.07	-0.76	-0.15	**-0.95**	-0.27
vgg13	0.08	-0.66	-0.23	**-0.93**	-0.26
vgg16	0.01	-0.63	-0.19	**-0.88**	-0.17
vgg19	-0.01	-0.62	-0.15	**-0.86**	-0.14
googlenet	0.74	-0.60	**-0.83**	-0.49	0.73
inception v3	0.69	-0.66	**-0.75**	-0.45	0.35
resnet18	0.59	-0.17	**-0.67**	-0.30	-0.44
resnet34	0.48	-0.18	**-0.65**	-0.19	-0.08
resnet50	0.56	-0.60	-0.71	-0.50	**-0.78**
resnet101	0.51	-0.57	**-0.70**	-0.51	-0.64
resnet152	0.52	-0.51	**-0.68**	-0.42	-0.48
densenet121	0.84	-0.75	**-0.87**	-0.62	0.82
densenet161	**0.87**	-0.74	**-0.87**	-0.67	0.81
densenet169	**0.87**	-0.74	**-0.87**	-0.67	0.81
densenet201	0.89	-0.75	**-0.91**	-0.67	0.90
efficientnet b1	0.35	**-0.41**	-0.39	0.01	0.03
efficientnet b2	**0.49**	-0.02	-0.44	-0.06	0.34
efficientnet b3	**0.32**	-0.12	-0.18	-0.13	0.18
efficientnet b4	0.30	**-0.51**	-0.29	-0.44	0.11
vit b 32	0.47	-0.31	-0.29	0.39	**0.51**
vit l 32	-0.14	**-0.61**	-0.47	-0.02	-0.06
vit b 16	-0.27	**-0.71**	0.04	0.39	-0.22
vit l 16	-0.39	**-0.89**	-0.66	-0.23	-0.24
vit h 14	-0.77	-0.83	**0.92**	0.31	-0.49
swin t	-0.12	-0.39	-0.02	**-0.42**	-0.06
swin s	-0.003	**-0.61**	-0.31	0.18	-0.03
swin b	-0.32	**-0.59**	-0.43	0.42	-0.32
convnext tiny	0.77	-0.01	-0.04	0.09	**0.80**
convnext small	0.57	0.22	0.25	0.13	**0.72**
convnext base	0.67	0.41	0.35	-0.03	**0.82**
convnext large	0.75	0.23	0.35	-0.10	**0.84**
Average	0.31 ± 0.45	$\mathbf{-0.44 \pm 0.35}$	-0.31 ± 0.43	-0.29 ± 0.39	0.13 ± 0.50

Table 2. Pearson correlations between the affinity score and other metrics, for all the architectures evaluated in this study. We see that no other metric can reliably provide the same information as the proposed non-linearity signature across different neural architectures.

are passed through the network. As the number of plots for all chosen 33 models on these datasets will not allow for a meaningful visual analysis, we rather plot the differences – in terms of the DTW distance – between the non-linearity signature of the model on Imagenet dataset with respect to three other datasets. We present the obtained results in Figure 18.

We can see that the overall deviation for CIFAR10 and CIFAR100 remains lower than for Random dataset suggesting that these datasets are semantically closer to Imagenet.

G. Results for self-supervised methods

In this section, we show that the non-linearity signature of a network remains almost unchanged when considering other pertaining methodologies such as for instance, self-supervised ones. To this end, we use 17 Resnet50 architecture pre-trained on Imagenet within the next 3 families of learning approaches:

1. SwAV [4], DINO [5], and MoCo [22] that belong to the family of contrastive learning methods with prototypes;
2. Resnet50 [21], Wide Resnet50 [67], TRex, and TRex* [53] that are supervised learning approaches;
3. SCE [8], Truncated Triplet [66], and ReSSL [68] that perform contrastive learning using relational information.

From the dendrogram presented in Figure 19, we can observe

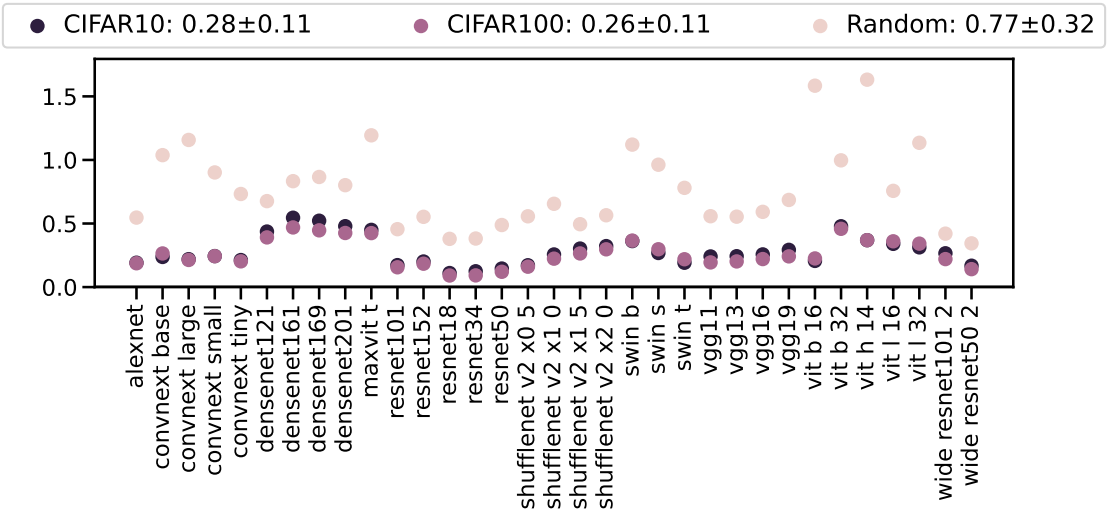

Figure 18. Deviation in terms of the Euclidean distance of the non-linearity signature obtained on CIFAR10, CIFAR100, and Random datasets from the non-linearity signature of the Imagenet dataset.

Criterion	Mean ± std
ρ_{aff}	0.76±**0.04**
Linear CKA	0.90±0.07
Norm	448.56±404.61
Sparsity	0.56±0.16
Entropy	0.39±0.46

Table 3. Robustness of the different criteria when considering the same architectures pre-trained for different tasks. Affinity score achieves the lowest standard deviation suggesting that it is capable of correctly identifying the architecture even when it was trained differently.

that the DTW distances between the non-linearity signatures of all the learning methodologies described above allow us to correctly cluster them into meaningful groups. This is rather striking as the DTW distances between the different instances of the Resnet50 model are rather small in magnitude suggesting that the affinity scores still retain the fact that it is the same model being trained in many different ways.

While providing a fine-grained clustering of different pre-trained models for a given fixed architecture, the average affinity scores over batches remain surprisingly concentrated as shown in Tab. 3. This hints at the fact that the non-linearity signature is characteristic of architecture but can also be subtly multi-faceted when it comes to its different variations.

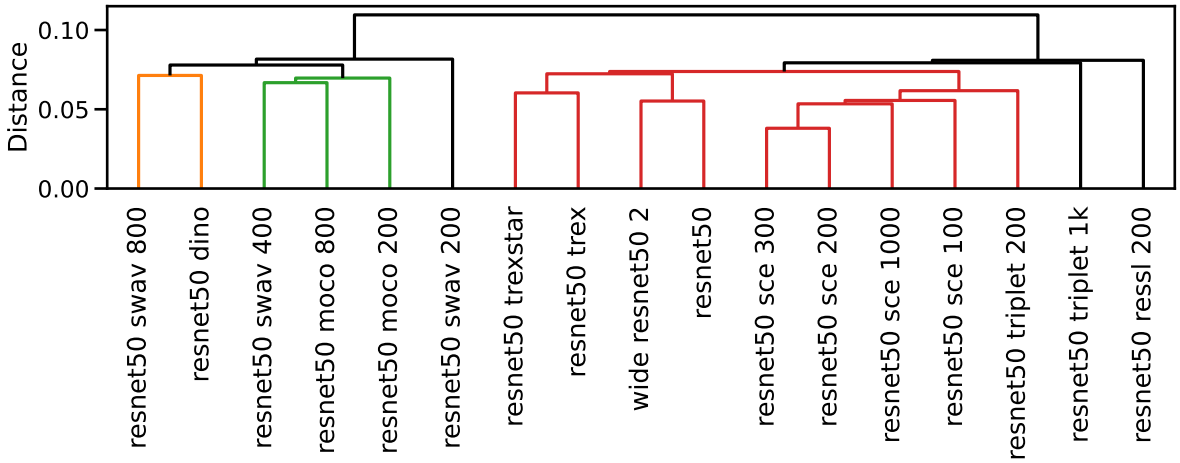

Figure 19. Hierarchical clustering of supervised and self-supervised pre-trained Resnet50 using the DTW distances between their non-linearity signatures.

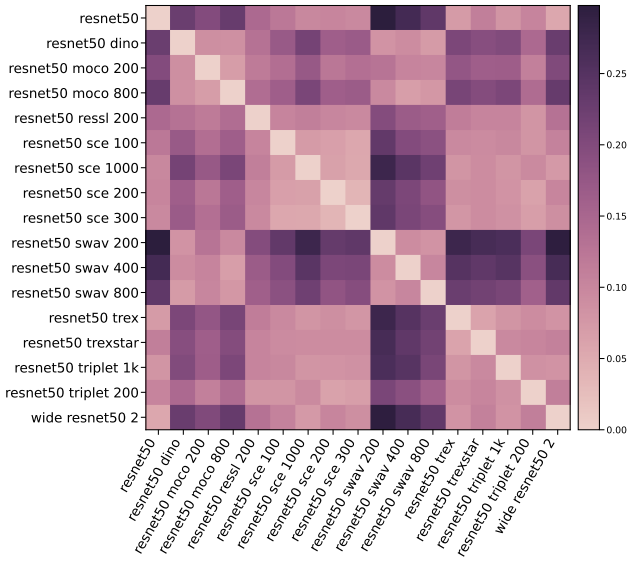

Figure 20. DTW distances associated with the clustering presented in Figure 19. We can see distinct clusters as revealed by the dendrogram.