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

Anonymous Author(s)
Affiliation
Address
email

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.

1 Introduction

Deep neural networks (DNNs) are undoubtedly the most powerful AI models currently available [1, 2, 3, 4, 5]. Their performance on many tasks, including natural language processing (NLP) [6] and computer vision [7], 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 [8, 9]. 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 [10] has firmly imposed itself as a number one choice for most, if not all, of the recent breakthroughs [11, 12, 13] in the machine learning and artificial intelligence fields.

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 [14] architecture, the winner of the ImageNet Large Scale Visual Recognition Challenge [15] 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 [16, 17, 18, 19] pushing the performance frontier even further. The AI research landscape then reached a turning

Submitted to 38th Conference on Neural Information Processing Systems (NeurIPS 2024). Do not distribute.

point with the proposal of transformers [10], starting their unprecedented dominance first in NLP and then in computer vision [20]. 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 [21] of the two models: a term used to denote the ability of a DNN to approximate functions of a certain complexity. Unfortunately, the existing theoretical results related to this either associate higher expressive power with depth [22, 23, 24] or width [25, 26, 27, 28] 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 put forward the following contributions:

1. 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 set of affinity scores of all its activation functions.
2. We compare non-linearity signatures of a wide range of popular DNNs used in computer vision: from Alexnet to vision transformers (ViT) and their more recent variations. Through this, we clearly illustrate the disruptive patterns in the evolution of the deep learning field.
3. We demonstrate that non-linearity signature can be predictive of DNNs performance and used to meaningfully identify the family of approaches to which a given DNN belongs. We further show that the non-linearity signature is unique as it doesn't correlate strongly with other potential candidates used for this task.

The rest of the paper is organized as follows. We start by presenting the relevant background knowledge on OT in Section 2. Then, we introduce the affinity score together with its different theoretical properties in Section 3. Section 4 presents experimental evaluations on a wide range of popular convolutional neural networks. Finally, we conclude in Section 5.

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 deep neural networks

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. Without non-linear activation functions, most of DNNs, no matter how deep, reduce to a linear function unable to learn complex patterns. Activation functions were also early identified [29, 30, 31, 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 in 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 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 ([33]). *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 can be found in the Appendix C):

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*

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

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

Theorem 3.4 (Gelbrich bound [34]). *Let $X, Y \in \mathcal{P}_2(\mathbb{R}^d)$ and let N_X, N_Y be their normal approximations. Then, $W_2(N_X, N_Y) \le W_2(X, Y)$.*

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. $|W_2(N_X, N_Y) - W_2(X, Y)| \le \frac{2\,\mathrm{Tr}\left[(\Sigma(X)\Sigma(Y))^{\frac{1}{2}}\right]}{\sqrt{\mathrm{Tr}[\Sigma(X)] + \mathrm{Tr}[\Sigma(Y)]}}$.

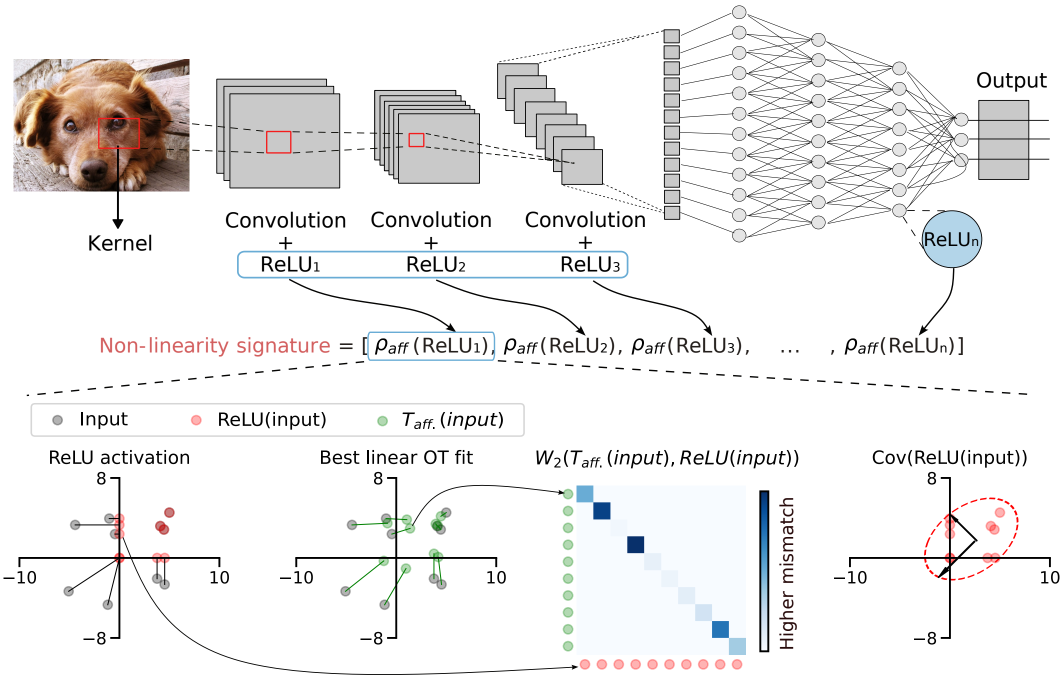

Figure 1: Illustration of how the non-linearity of a given neural network is measured. (**Top**) The non-linearity signature of a DNN is a collection of affinity scores calculated for each activation function spread across its hidden layers. (**Bottom**) The affinity score is calculated based on 3 main 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.

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.

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}.*

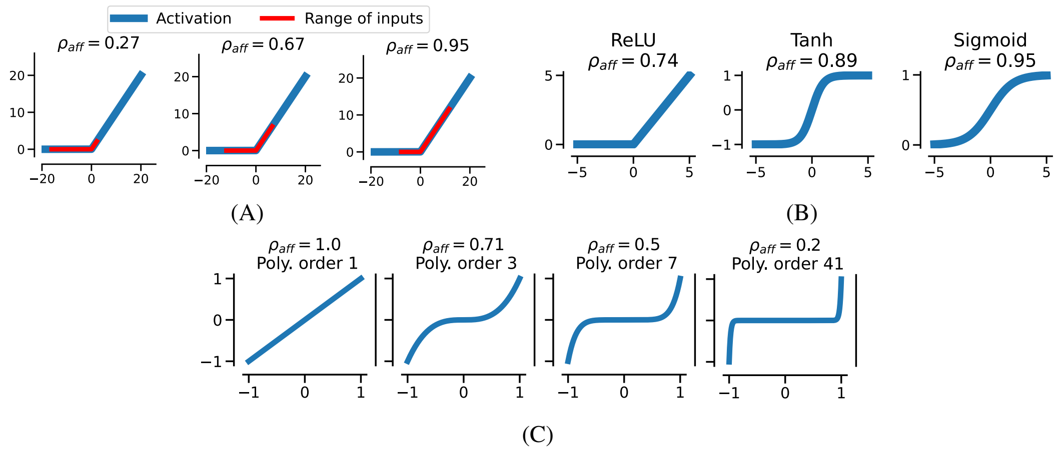

Figure 2: **(A)** Non-linearity of ReLU depends on the range of input values (*red*); **(B)** ReLU, Tanh, and Sigmoid exhibit different degrees of non-linearity for the same input; **(C)** Affinity score captures the increasing non-linearity of polynomials of different degrees.

3.2 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 $\mathrm{X}_i \in \mathbb{R}^{h_i \times w_i \times c_i}$ (for instance, an image of size $224 \times 224 \times 3$ for $i = 1$) and outputting a tensor $\mathrm{Y}_i \in \mathbb{R}^{h_{i+1} \times w_{i+1} \times c_{i+1}}$ used as an input of the following layer F_{i+1}. This defines $\mathrm{N} = F_L \odot ... \odot F_i ... \odot F_1 = \bigodot_{k=1,\dots,L} F_k$ where $\odot$ stands for a composition.

We now present the definition of a non-linearity signature of a network N. Below, we abuse the compositional structure of F_i and see it as an ordered sequence of functions.

Definition 3.1. *Let $N = \bigodot_{k=1,\dots,L} F_k$ be a neural network. Define by $\mathcal{A}$ a finite set of common activation functions such that $\mathcal{A} := \{\sigma | \sigma : \mathbb{R}^{h \times w \times c} \to \mathbb{R}^{h \times w \times c}\}$. Let r be a pooling operation such that $r : \mathbb{R}^{h \times w \times c} \to \mathbb{R}^c$. Then, the non-linearity signature of N given an input X is defined as follows:*

$$\rho_{\mathrm{aff}}(N; \mathrm{X}) = \{\rho_{\mathrm{aff}}(r(\mathrm{X}_i), \sigma(r(\mathrm{X}_i))), \quad \forall \sigma \in F_i \cap \mathcal{A}, \quad i = \{1, \dots, L\}\}.$$

Non-linearity signature, illustrated in Figure 1, associates to each network N a vector of affinity scores calculated over the inputs and outputs of all activation functions encountered across its layers.

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 defining the non-linearity signature of a DNN. In Figure 2(A), we show how the ReLU function [35], 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 $\mathrm{Sigmoid}(x) = 1/(e^{-x} + 1)$ [36] 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 refer the reader to Appendix D for more visualizations of the affinity score of popular activation functions.

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 [37] and [38] (see [39] for a complete survey of the subsequent works). [37] 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. [38] 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 [37].

Impact of activation functions [40] 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). [41] 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 D).

Non-linearity measure The only work similar to ours in spirit is the paper by [42] proposing 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 other study reporting the reproduced results from this work.

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 [43]. 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 H for more datasets) with a total of 50,000 images. We include the following landmark architectures in our study: Alexnet [14], four VGG models [16], Googlenet [44], Inception v3 [17], five Resnet models [18], four Densenet models [19], four MNASNet models [45], four EfficientNet models [46], five ViT models, three Swin transformer [47] and four Convnext models [48]. We include MNASNet and EfficientNet models as prominent representatives of the neural architecture search approach [49]. 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 E 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 3 we present the minimum, median, and maximum values of the affinity scores calculated for the considered neural networks (see Appendix F for raw non-linearity signatures). We immediately see that until the arrival of transformers, the trend of the landmark models was to decrease their non-linearity, rather than to increase 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 F 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 for 2020s using the best practices of Swin transformers, further confirms this observation.

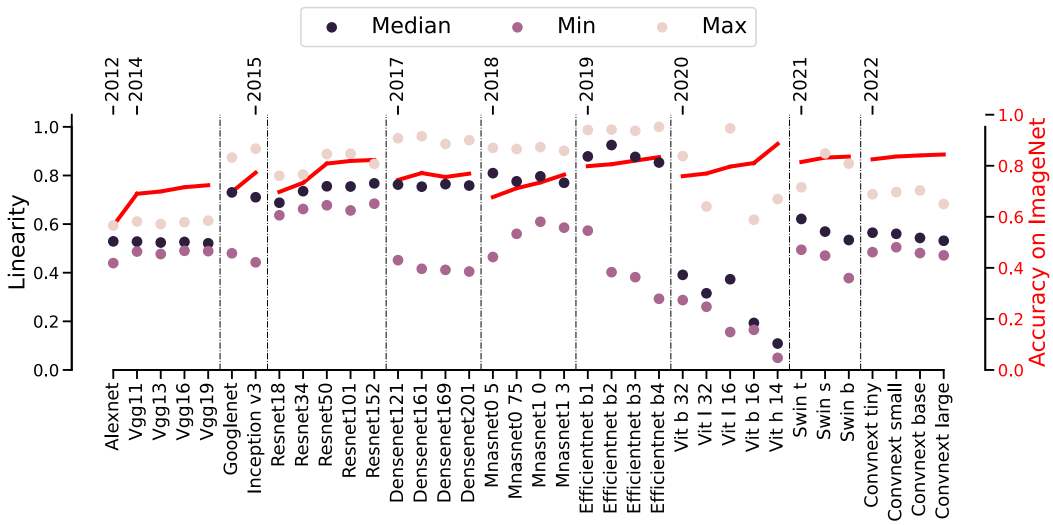

Figure 3: Median, minimum, and maximum values of non-linearity signatures of the different architectures spanning a decade (2012-2022) of computer vision research. We observe a clear trend toward the increase of the spread and the maximum values of the linearity in neural networks lasting until the arrival of transformers in 2020. ViTs have a distinct pattern of maximizing the non-linearity of their activation functions. Swin transformers and Convnext models retain this property from them while remaining close to the pure convolutional networks.

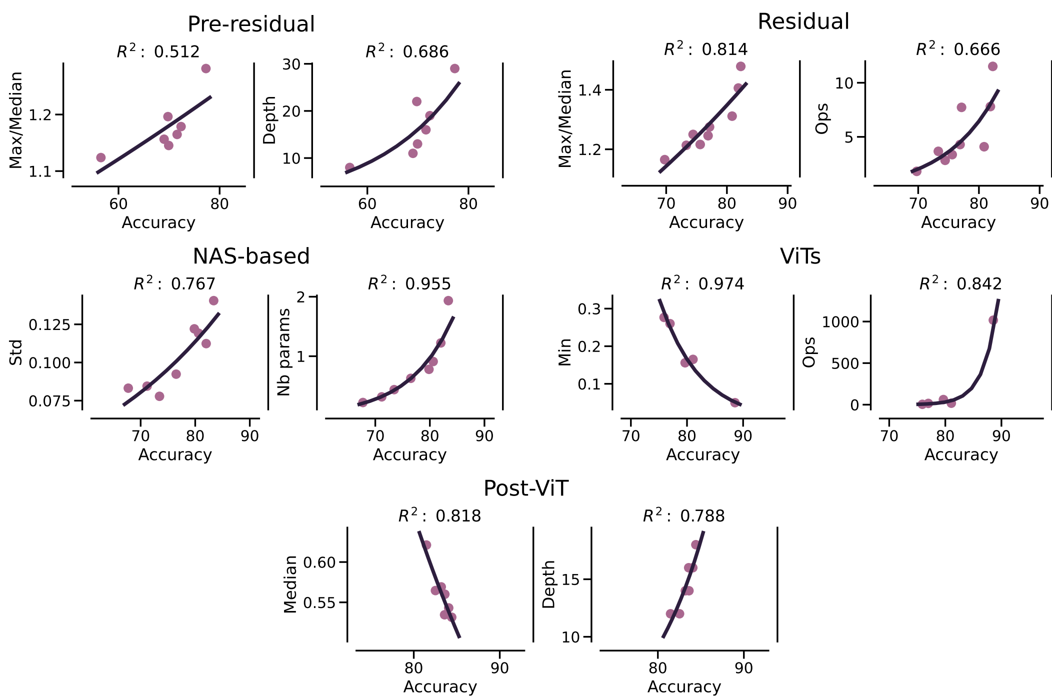

Figure 4: Best found dependency between the different statistics extracted from the non-linearity signatures of the DNN families and their respective Imagenet-1K accuracy. The results are compared in terms of the R^2 score against the most precise of the other common DNN characteristics such as depth, size, and the GFLOPS.

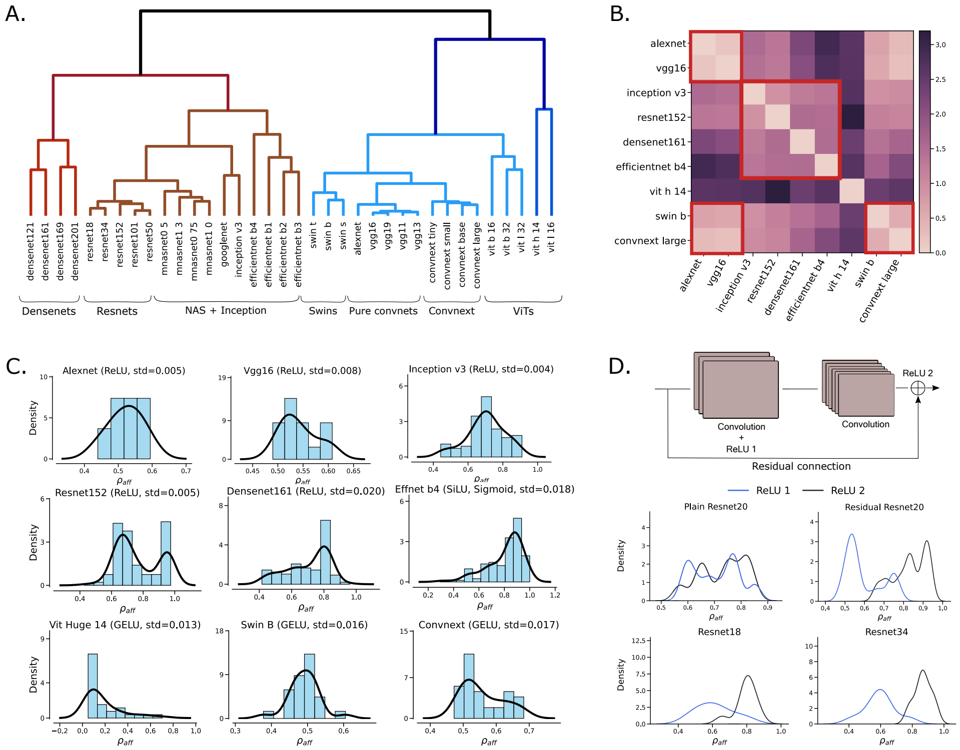

Figure 5: Comparing the different families of the neural architectures based on their non-linearity signatures. **(A)** Hierarchical clustering of all DNNs considered in our study revealing meaningful clusters with close architectural characteristics; **(B)** 9 representative architectures from all studied families and the similarities between them. Note how the similarities between early convnets and other models is decreasing with time until computer vision priors are introduced into Swin transformers in 2021; **(C)** Distributions of affinity scores in each network. Most models expand the non-linearity ranges of their activation functions compared to early convnets. ViTs are dominated by highly non-linear activation functions, Resnets have a bimodal distribution, Densenets, and EfficientNets have a diametrically skewed distribution compared to ViTs. **(D)** 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).

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 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 4, 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 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.

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**.

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

Distinct signature for every architecture Non-linearity signature correctly identifies the different families of neural architectures. To show this, we perform hierarchical clustering using pairwise dynamic time warping (DTW) distances [50] between the non-linearity signatures of the models from Figure 3. The results in Figure 5 (A), as well as the pairwise distance matrix between a representative of each studied family in Figure 5 (B) (see Appendix G for the full matrix), show that we correctly cluster all similar models together, both within their respective families (such as the different variations of the same architecture) and across them (such as the cluster of Swin and pure convolution models). Additionally, we highlight the individual affinity scores' distributions of representative models in Figure 5 (C). Finally, we highlight the exact effect of residual connections proposed in 2016 and used ever since by every benchmark model in Figure 5 (D). It 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 [18]. Non-linearity signatures can also be applied to meaningfully identify training methods, such as popular nowadays self-supervised approaches, for a fixed architecture (see Appendix I).

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 Table 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 the information that other metrics fail to capture consistently across all architectures. This becomes even more apparent when analyzing the individual correlation values (in Appendix G). Overall, the proposed affinity score and the non-linearity signatures derived from it offer a unique perspective on the developments in the ML field.

5 Discussions

We proposed the first sound approach to measure non-linearity of activation functions in neural networks and defined their non-linearity signature based on it. 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 that until the arrival of transformers, the trend in DNNs was to decrease their non-linearity, rather than to increase it. Vision transformers changed this pattern drastically. We also showcased that our measure is unique, as no other metric correlates strongly with it across all architectures.

In the future, our work can be applied to study the non-linearity of the LLM models to better understand the effect of different architectural choices in them. On a higher level, our approach can also be used to identify new disruptive neural architectures by identifying those of them that leverage different internal non-linearity characteristics to obtain better performance. This capacity of identifying novel technologies is even more crucial in the age of very large models where experimenting with the building blocks of the optimized backbone comes at a very high cost.

References

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

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

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

[4] I. Goodfellow, Y. Bengio, and A. Courville. *Deep Learning*. Adaptive computation and machine learning. MIT Press, 2016.

[5] 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.

[6] 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.

[7] 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.

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

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

[10] 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.

[11] 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.

[12] 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.

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

[14] 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.

[15] 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.

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

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

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

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

[20] 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.

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

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

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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.

[28] 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.

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

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

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

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

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

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

[35] 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.

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

[37] 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.

[38] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *ICML*, volume 97, pages 3519–3529. PMLR, 09–15 Jun 2019.

[39] 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.

[40] 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.

[41] 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.

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

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

[44] 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.

[45] 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.

[46] 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.

[47] 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.

[48] 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.

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

[50] 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.

[51] Xavier Glorot, Antoine Bordes, and Yoshua Bengio. Deep sparse rectifier neural networks. In Geoffrey Gordon, David Dunson, and Miroslav Dudík, editors, *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA, 11–13 Apr 2011. PMLR.

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

[53] 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.

[54] 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.

[55] 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.

[56] 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.

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

[58] 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.

[59] 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.

[60] 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.

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

[62] 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.

[63] 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.

[64] 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.

[65] 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.

A Broader Impacts

This paper presents work whose goal is to advance the field of Machine Learning and better understand the underlying behavior of Deep Neural Networks architectures. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

B Limitations

An important assumption of Theorem 3.3, is that the activation function that we want to analyze through ρ_{aff} needs to be a positive definite transformation of the inputs. Fortunately, this is the case for activation functions, that we consider in this paper. Finally, we note that despite the strong correlation between the statistics extracted from the non-linearity signatures for certain DNNs' architectures, we are yet to show that explicitly optimizing affinity scores through backpropagation can have an actionable impact on DNNs performance or its other properties, such as robustness or transferability.

C Proofs of main theoretical results

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 proof 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

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

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

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

□

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

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

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

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

Combining the above inequalities, we get

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

Let $a = \mathrm{Tr}[\Sigma(X)] + \mathrm{Tr}[\Sigma(Y)]$, and so $W_2^2(N_X, N_Y) = a - b$, where $b = 2\,\mathrm{Tr}\left[(\Sigma(X)\Sigma(Y))^{\frac{1}{2}}\right]$. 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). $\square$

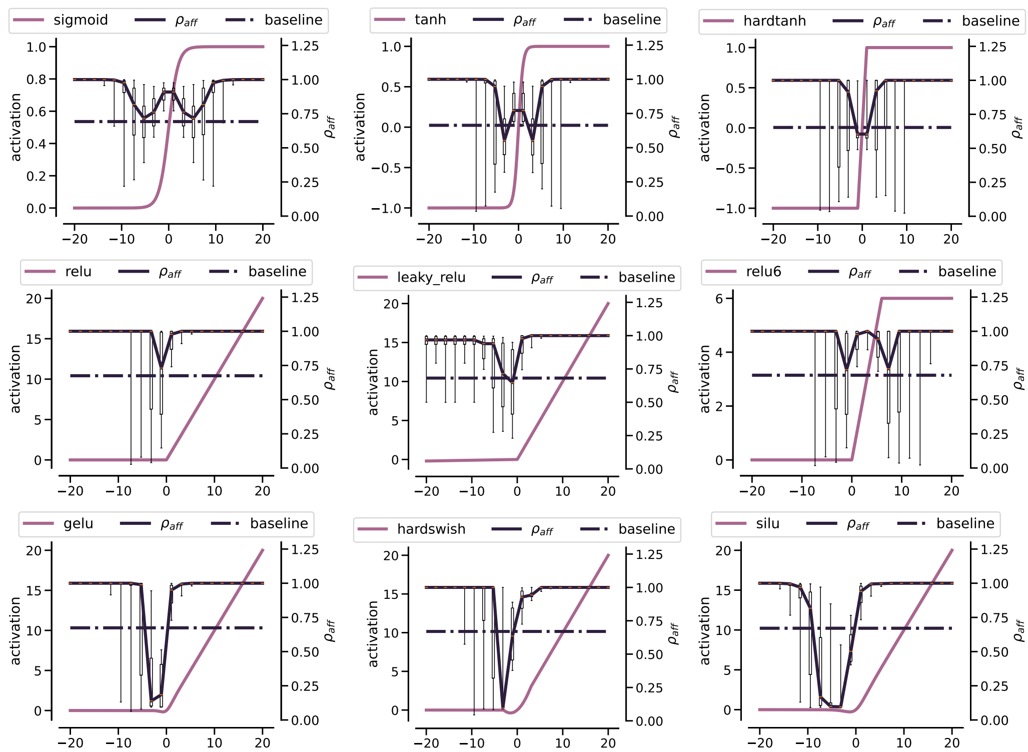

Figure 6: Median affinity scores of Sigmoid, ReLU, GELU, ReLU6, LeakyReLU with a default value of slope, Tanh, HardTanh, SiLU, and HardSwish obtained across random draws from Gaussian distribution with a sliding mean and varying stds used as their input. Whiskers of boxplots show the whole range of values 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.

D 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 [40] identifies over 40 activation functions with first references to sigmoid dating back to the seminal paper [36] 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 6 the affinity scores obtained after a single pass of the data through the following activation functions: Sigmoid, ReLU [51], GELU [52], ReLU6 [53], LeakyReLU [54] with a default value of the slope, Tanh, HardTanh, SiLU [55], and HardSwish [56]. 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.

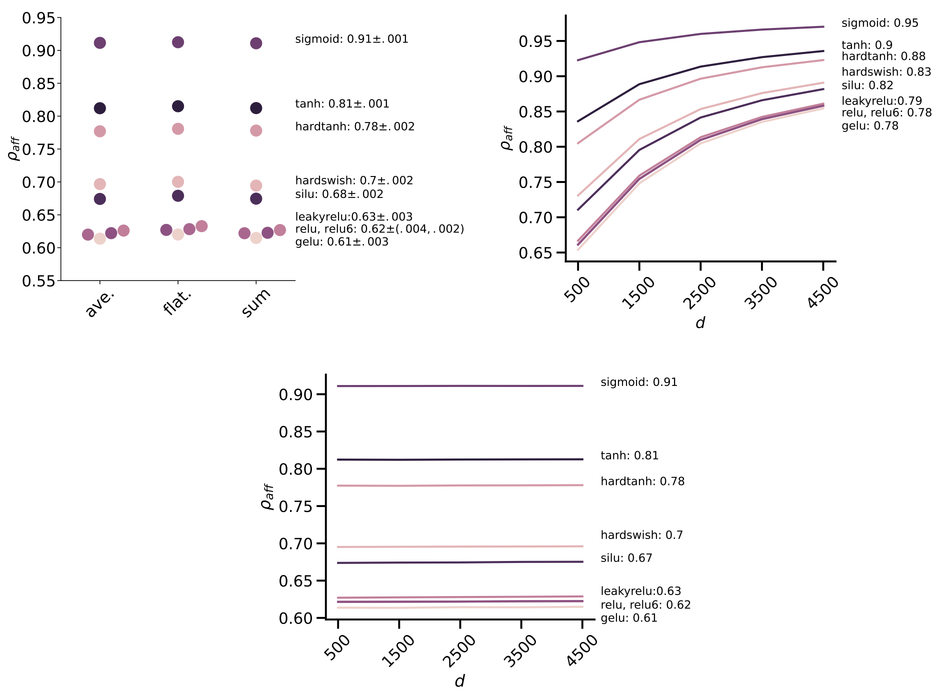

Figure 7: **(Top left)** Affinity score is robust to the dimensionality reduction both when using averaging and summation over the spatial dimensions; **(Top right)** When $d > n$, sample covariance matrix estimation leads to a lack of robustness in the estimation of the affinity score; **(Bottom)** Shrinkage of the covariance matrix leads to constant values of the affinity scores with increasing d.

E 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 7 (left), 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 [57]. In Figure 7 (right), 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 8, we present the obtained result 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). 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.

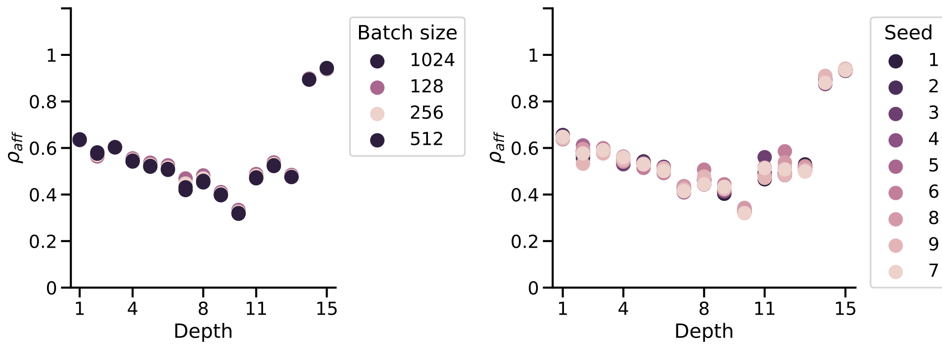

Figure 8: Non-linearity signature of VGG16 on CIFAR10 with a varying batch size (left) and when retrained from 9 different random seeds (right).

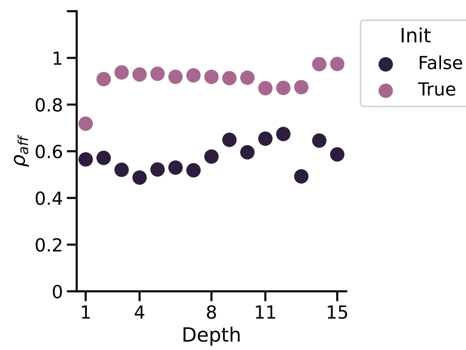

Figure 9: Non-linearity signatures of VGG16 on CIFAR10 in the beginning and end of training on Imagenet.

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 9, 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.

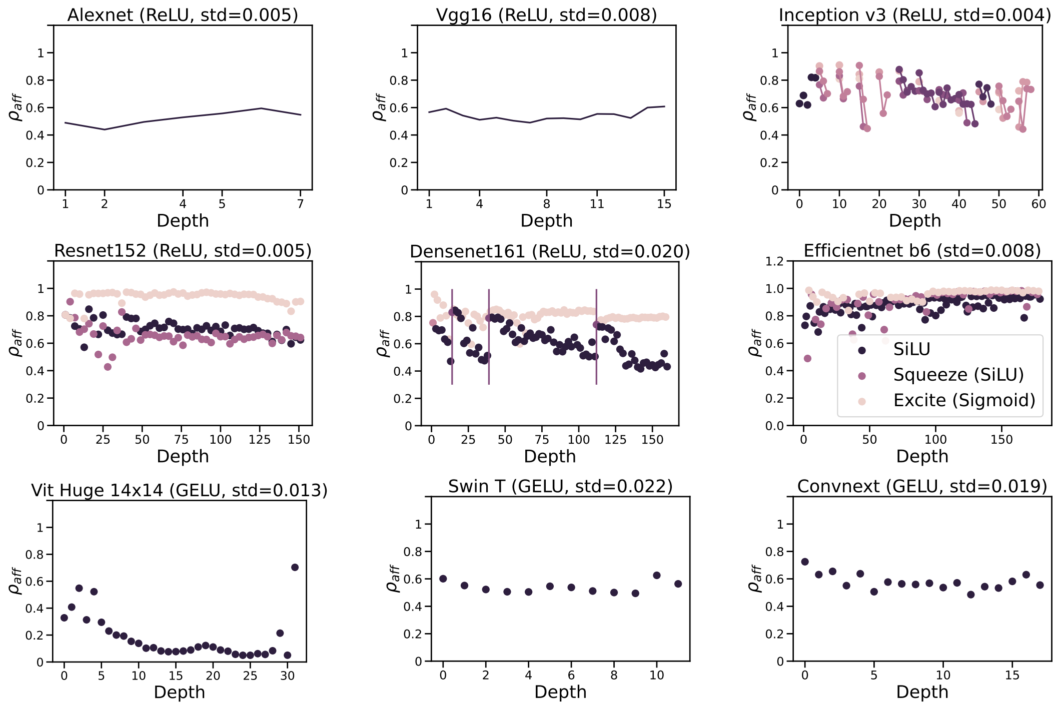

Figure 10: Raw non-linearity signatures of popular DNN architectures, plotted as affinity scores over the depth throughout the network.

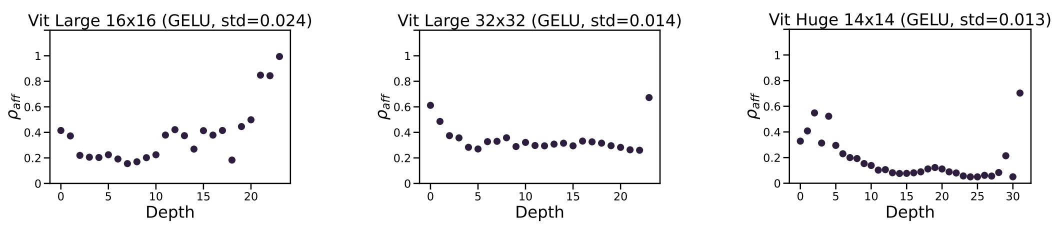

Figure 11: ViTs: Large ViT with 16x16 and 32x32 patch sizes and Huge ViT.

F Raw signatures

In Figure 10, 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

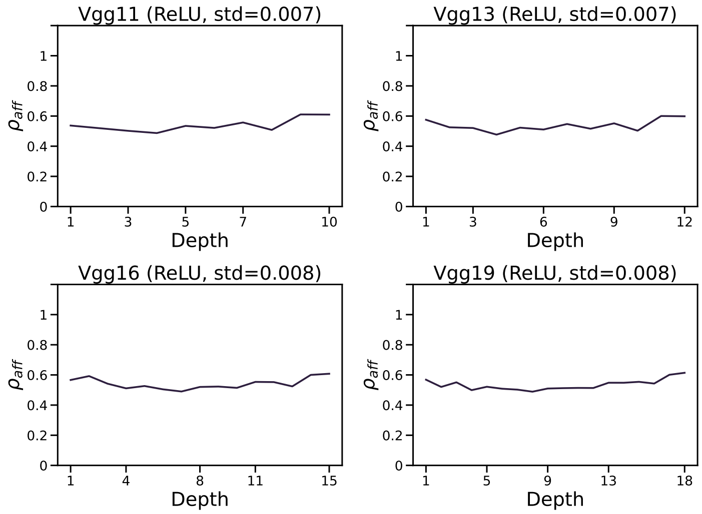

Figure 12: Impact of depth on the non-linearity signature of VGGs.

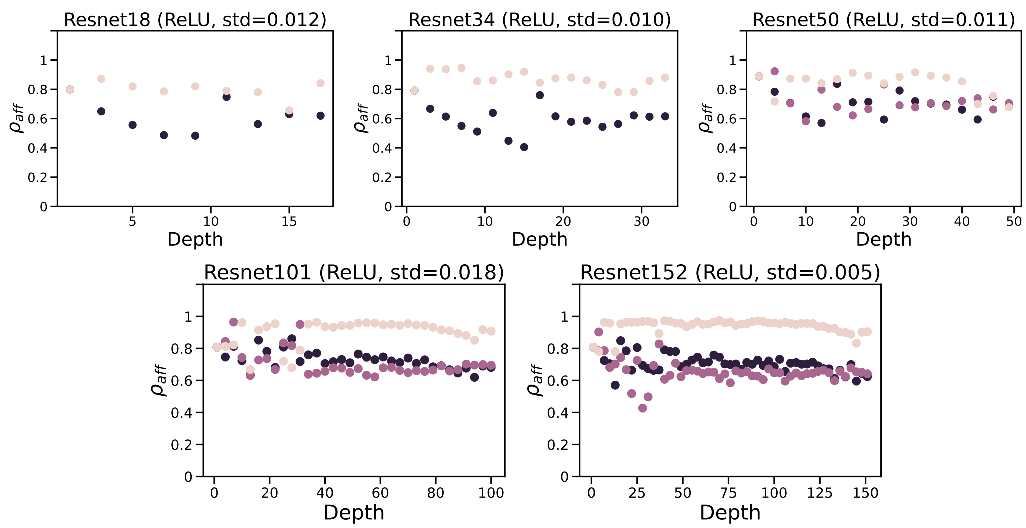

Figure 13: Impact of depth on the non-linearity signature of Resnets.

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 11 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 12 and Figure 13 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.

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.

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	**-0.44 ± 0.35**	-0.31 ± 0.43	-0.29 ± 0.39	0.13 ± 0.50

G Detailed comparisons between architectures

We consider the following metrics as 1) the linear CKA [38] 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 Table 2, the detailed values of Pearson correlations obtained for each architecture and all the metrics considered in this study. In Figure 14, we show the full matrix of pairwise DTW distances [50] obtained between architectures, then used to obtain the clustering presented in the main text.

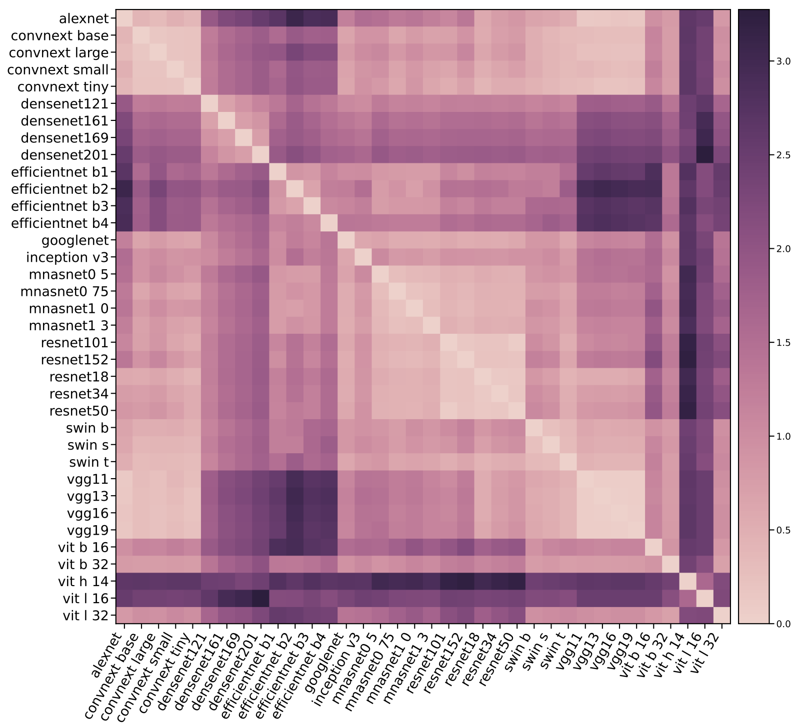

Figure 14: Full matrix of DTW distances between non-linearity signatures.

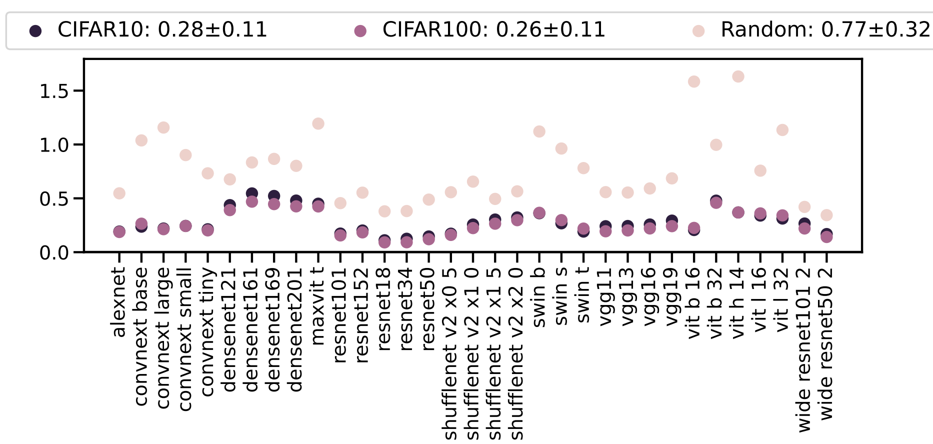

Figure 15: Deviation in terms of the Euclidean distance of the non-linearity signature obtained on CIFAR10, CIFAR100, and Random datasets from the non-linearity signature of the Imagenet dataset.

H Results on more datasets

Below, we compare the results obtained on CIFAR10, CIFAR100 datasets as well as when the random data tensors 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 15.

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.

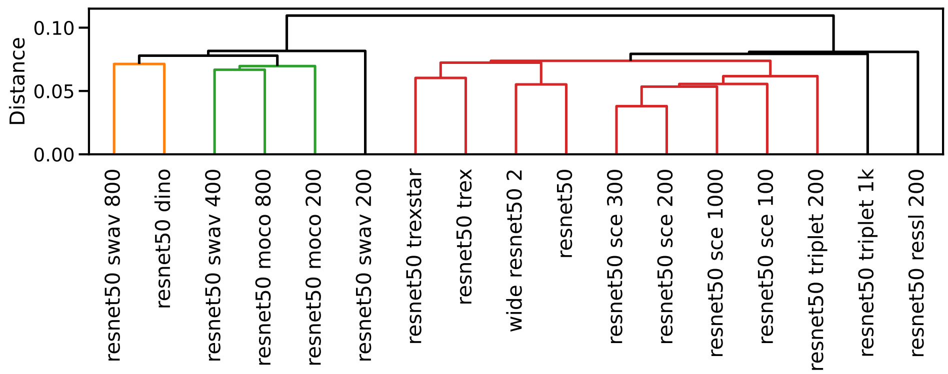

Figure 16: Hierarchical clustering of supervised and self-supervised pre-trained Resnet50 using the DTW distances between their non-linearity signatures.

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.

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

I 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 [58], DINO [59], and MoCo [60] that belong to the family of contrastive learning methods with prototypes;
2. Resnet50 [18], Wide Resnet50 [61], TRex, and TRex* [62] that are supervised learning approaches;
3. SCE [63], Truncated Triplet [64], and ReSSL [65] that perform contrastive learning using relational information.

From the dendrogram presented in Figure 16, we can observe 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 Table 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.

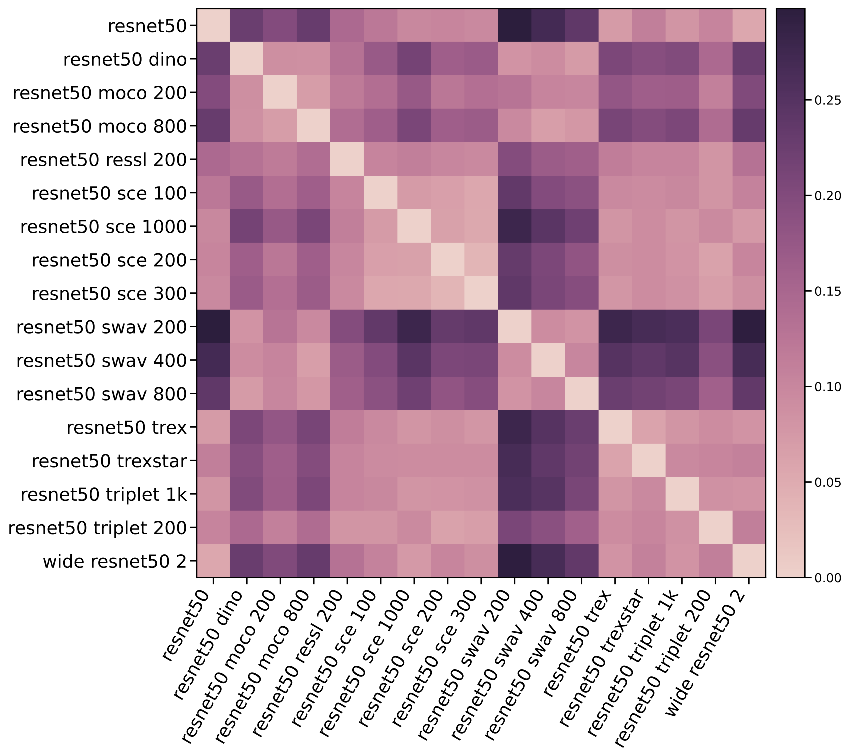

Figure 17: DTW distances associated with the clustering presented in Figure 16. We can see distinct clusters as revealed by the dendrogram.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: Proposition of affinity score and non-linearity signature in Section 3. Experiments showing non-linearity signatures of DNNs, prediction of performance, clustering and uniqueness in Section 4.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in Appendix B.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: Full proofs in Appendix C.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: All models are pretrained checkpoints from torchvision. Experiments are conducted on Imagenet, publicly available.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Anonymized code to reproduce experiments is available as a zip file, with a README file to explain how to run it.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: Experimental details are described in Section 4 and Appendix E.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: Standard deviations across multiple batch of data are reported.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: All experiments are carried out on a single A100 GPU.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: Standard and public datasets used, no experiments on human subjects.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We discuss broader impacts in Appendix A.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: No such risks, no checkpoints released.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Torchvision contributors credited for checkpoints, and datasets as well, in Section 4.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.

- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Anonymized code to reproduce experiments is available as a zip file, with a README file to explain how to run it.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: No experiments on human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: No experiments on or with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.