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

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

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.

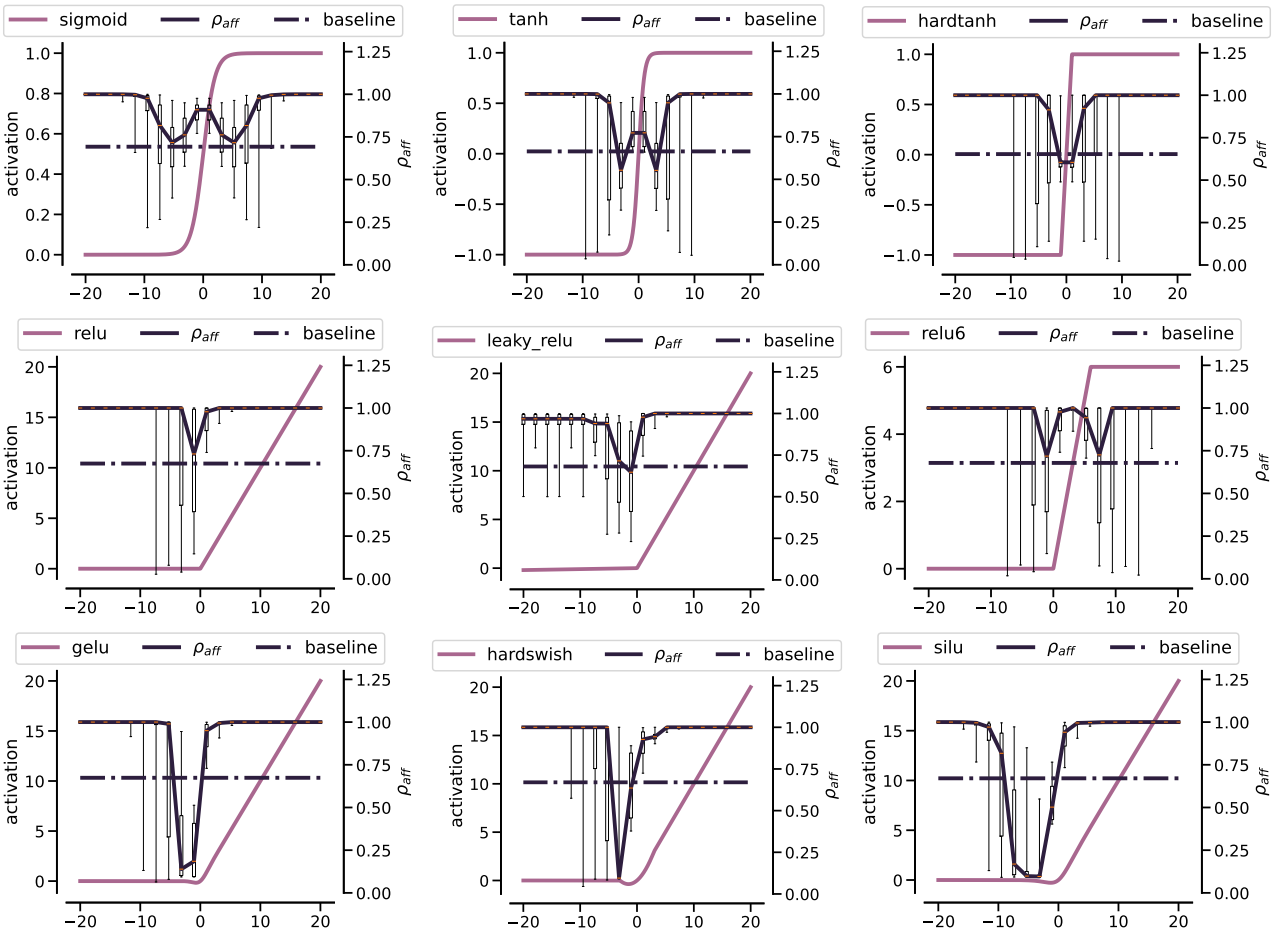

Figure 8. Median affinity scores of Sigmoid, ReLU, GELU, ReLU6, LeakyReLU with a default value of slope, Tanh, HardTanh, SiLU, and HardSwish obtained across random draws from Gaussian distribution with a sliding mean and varying stds used as their input. Whiskers of boxplots show the whole range of values 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

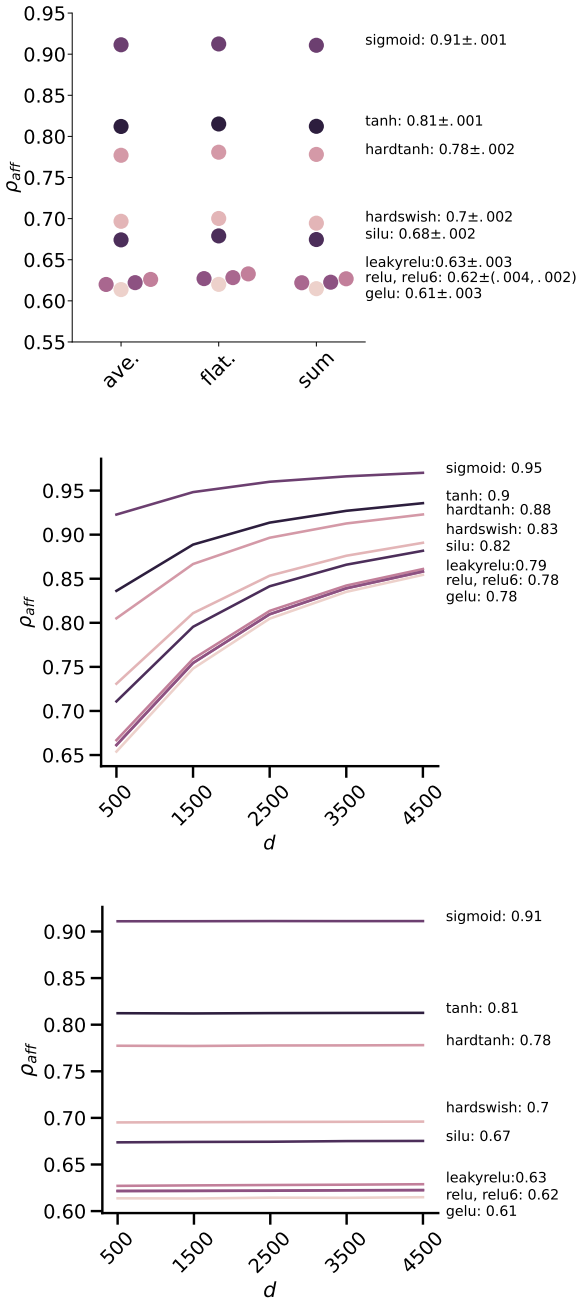

Figure 9. **(Top)** Affinity score is robust to the dimensionality reduction both when using averaging and summation over the spatial dimensions; **(Middle)** When $d > n$, sample covariance matrix estimation leads to a lack of robustness in the estimation of the affinity score; **(Bottom)** Shrinkage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

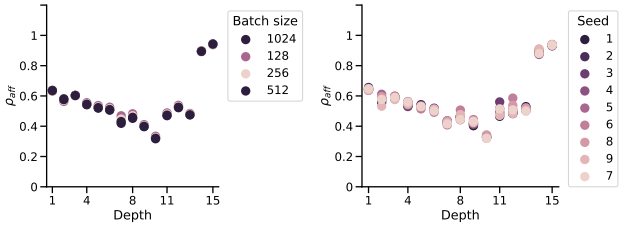

Figure 10. Non-linearity signature of VGG16 on CIFAR10 with a varying batch size (left) and when retrained from 9 different random seeds (right).

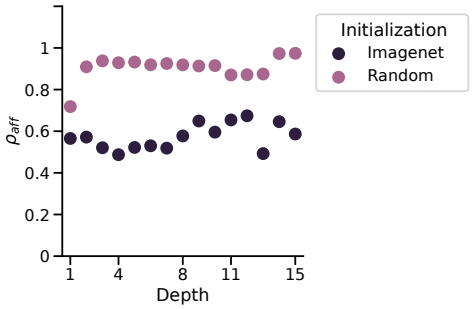

Figure 11. Non-linearity signatures of VGG16 on CIFAR10 in the beginning and end of training on Imagenet.

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.

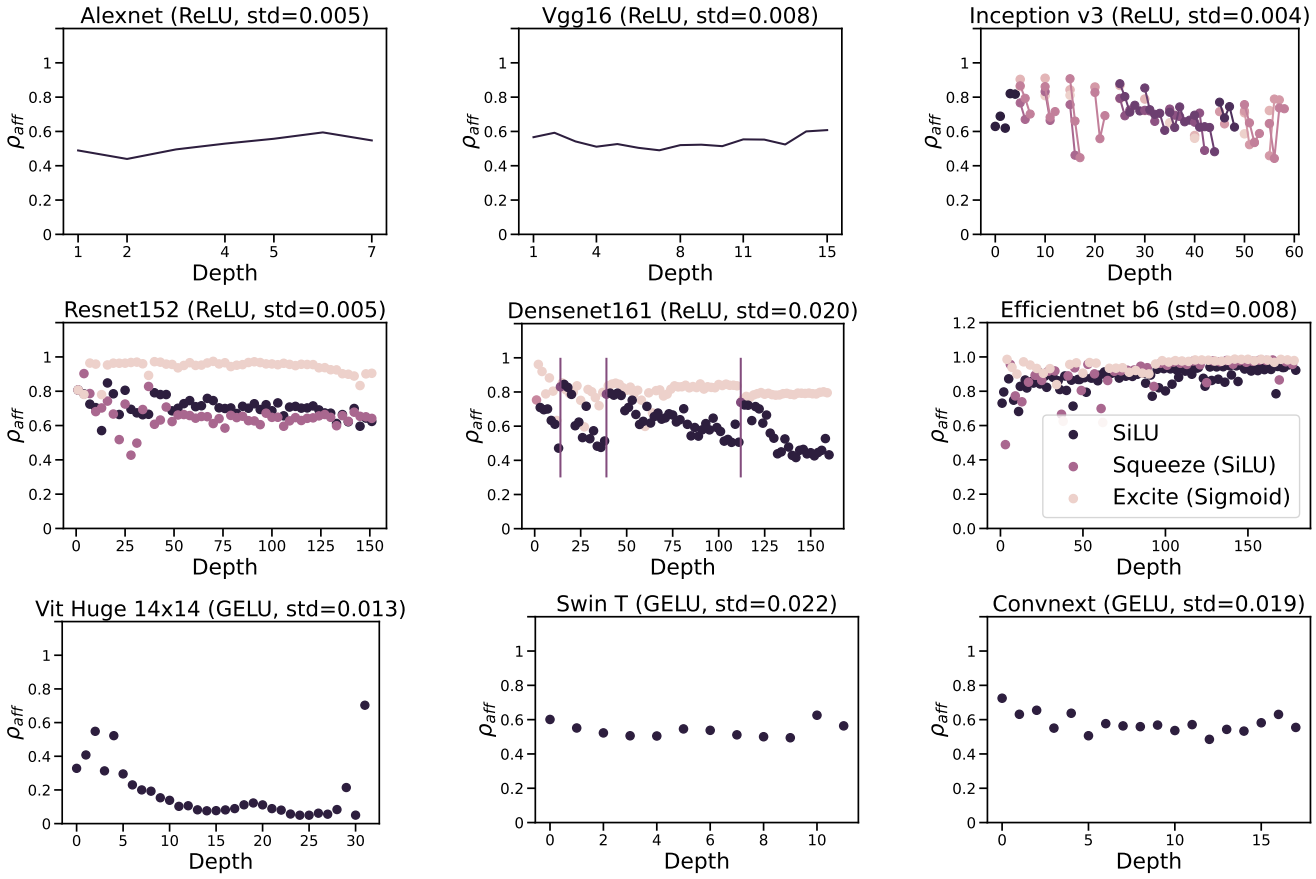

Figure 12. Raw non-linearity signatures of popular DNN architectures, plotted as affinity scores over the depth throughout the network.

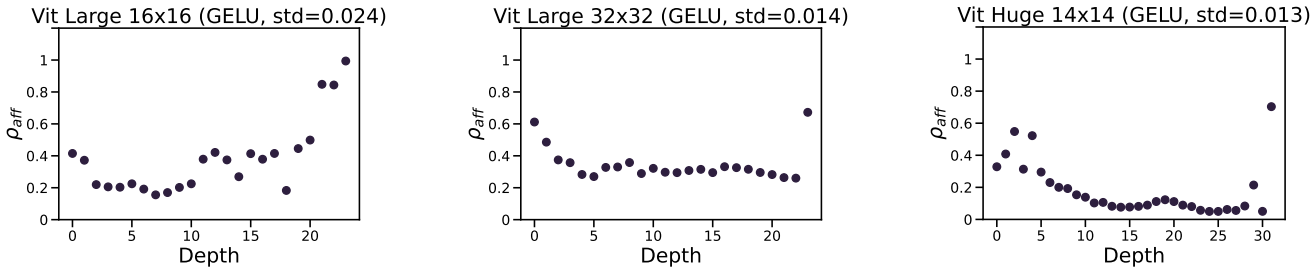

Figure 13. ViTs: Large ViT with 16x16 and 32x32 patch sizes and Huge ViT.

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.

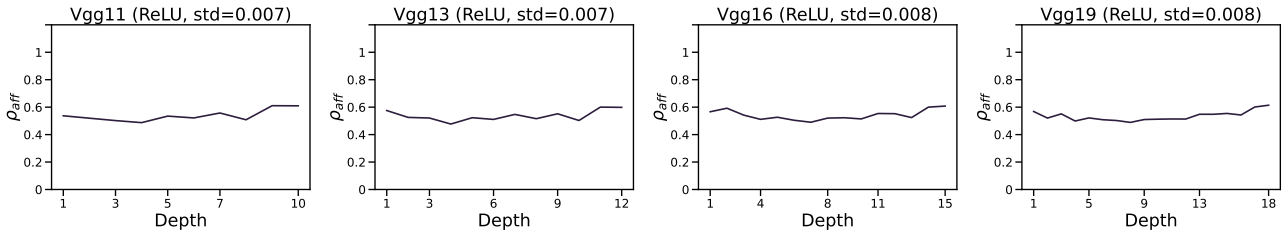

Figure 14. Impact of depth on the non-linearity signature of VGGs.

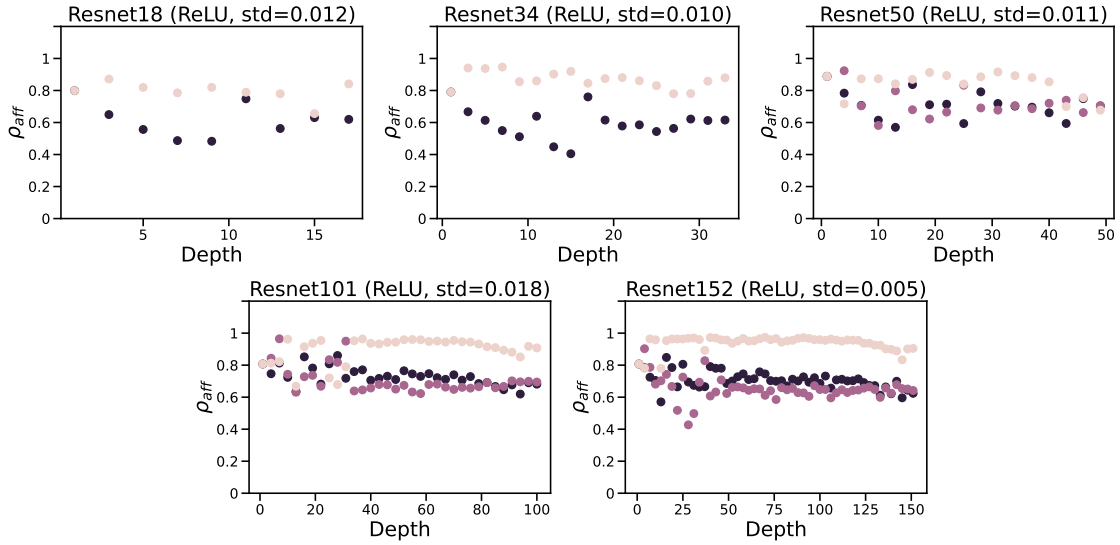

Figure 15. Impact of depth on the non-linearity signature of Resnets.

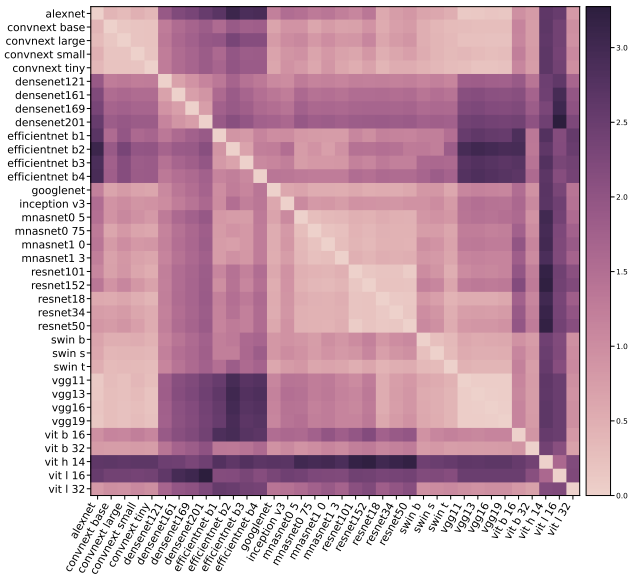

Figure 16. Full matrix of DTW distances between non-linearity signatures.

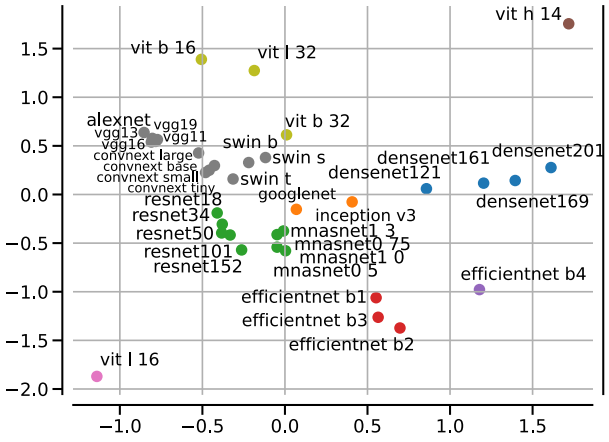

Figure 17. Multi-dimensional scaling of the linkage matrix obtained from the pairwise DTW distances between non-linearity signatures.

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

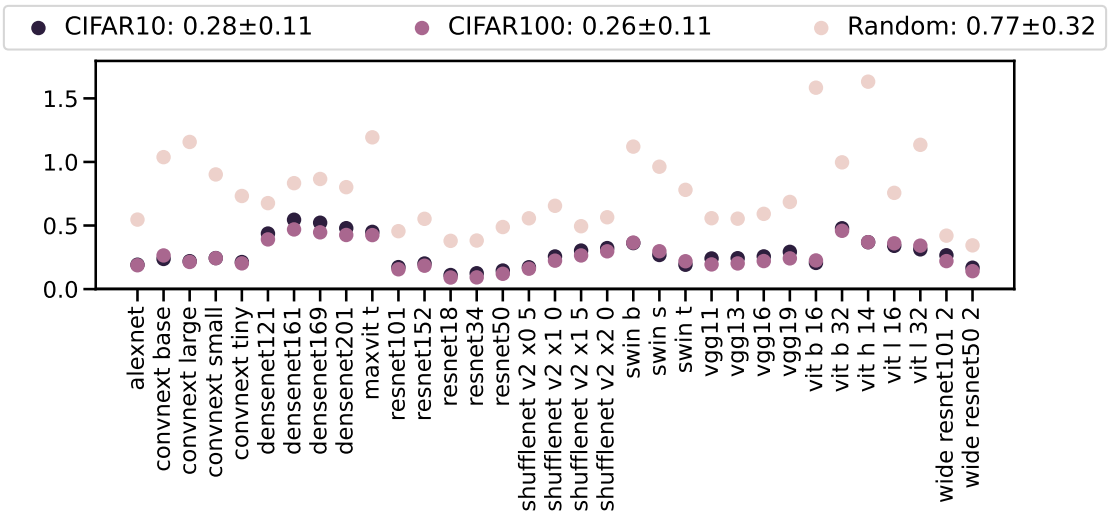

Figure 18. Deviation in terms of the Euclidean distance of the non-linearity signature obtained on CIFAR10, CIFAR100, and Random datasets from the non-linearity signature of the Imagenet dataset.

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.

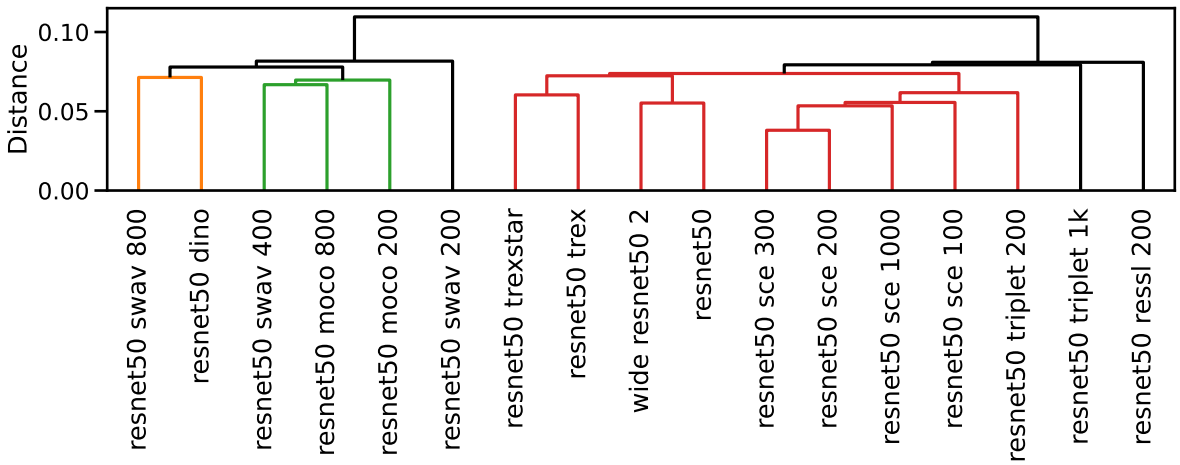

Figure 19. Hierarchical clustering of supervised and self-supervised pre-trained Resnet50 using the DTW distances between their non-linearity signatures.

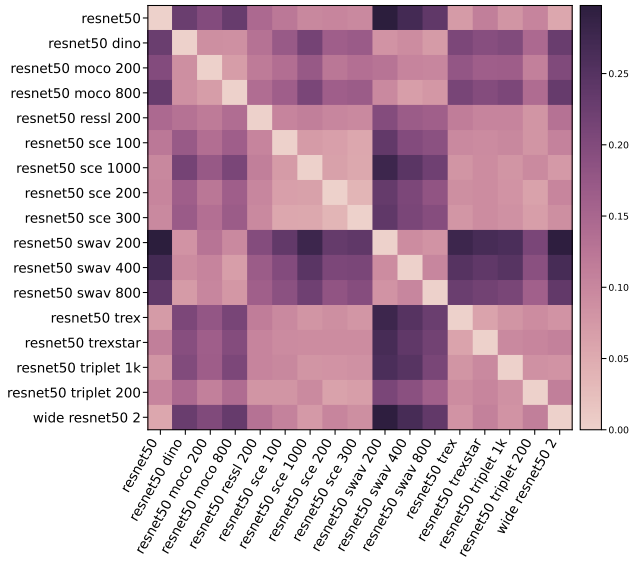

Figure 20. DTW distances associated with the clustering presented in Figure 19. We can see distinct clusters as revealed by the dendrogram.