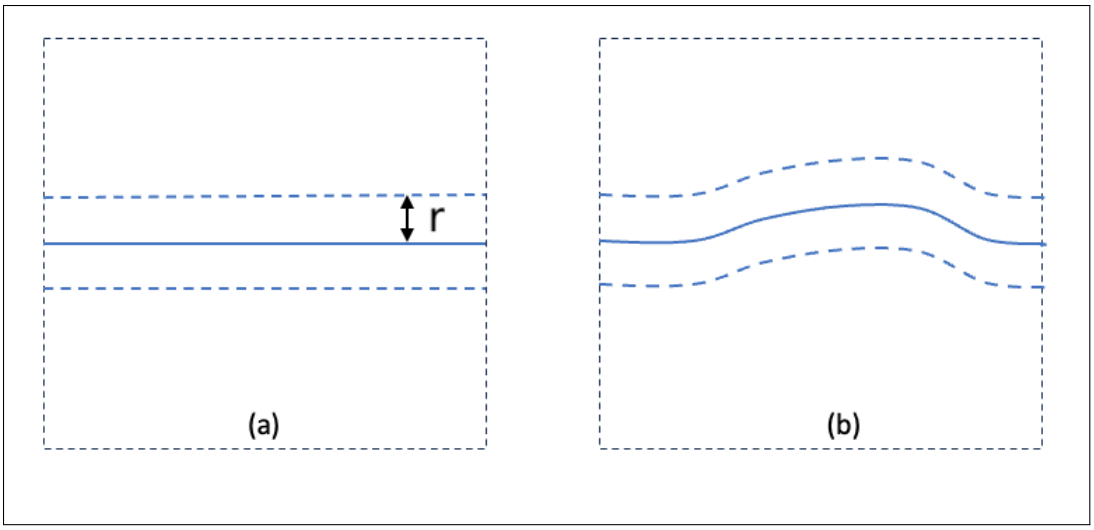


Figure 1: Two areas of decision space, one with a short boundary (a) and a longer boundary (b). For a given radius r, anharmoniticity is non-zero between the dotted lines a distance r from the boundaries.

1 Harmonic Functions as Minimal Interpolators

We state here a very relevant and interesting property of Harmonic Functions, namely that, given fixed values on a closed n-dimensional boundary, there is a unique function f that interpolates inside this boundary satisfying the Laplace Condition ($\nabla^2 f = 0$), and that this function has the minimum average curvature over all possible f satisfying the same boundary conditions. This is known as the Dirichlet Problem and always has a solution for harmonic functions (https://en.wikipedia.org/wiki/Dirichlet_problem).

In some cases we can explicity see this. For example, in 1-dimension it is trivial to see that between two points x_1 and x_2, with values a and b, respectively, there is exactly one function $f(x)$ that interpolates with minimal curvature, namely the straight line connecting x_1 and x_2 with $f(x_1) = a$ and $f(x_2) = b$, i.e. $f(x) = a + (b-a)\frac{x-x_1}{x_2-x_1}$.

In two dimensions we also have the explicit case of a function defined on the circular boundary ∂D enclosing the unit disk D in $\mathbf{R}^2$, with the Poisson Integral Formula

$$f(z) = \frac{1}{2\pi}\int_0^{2\pi} f(e^{i\psi})\frac{1-|z|^2}{|1-ze^{-i\psi}|^2}d\psi \quad if \;\; z \in D \tag{1}$$

$$f(z) \quad if \;\; z \in \partial D \tag{2}$$

$$\tag{3}$$

Practically speaking, for any boundary with fixed values you can also find the interior values of the function by iteratively applying the Mean Value Condition on a grid until the solution relaxes to equilibrium.

2 Gamma correlates with Decision Boundary Length

In the main text we stated that γ indicated the 'wiggliness' of the decision boundary. Let us argue this more mathematically here, for the case of a smooth decision boundary in feature space, that the average value of γ is proportional to the "length" of the decision boundary. So for the two areas of decision space shown in Fig 1, we are setting out to show the region on the left (a) should have a higher average anharmoniticity than the one on the right (b), for any given choice of the radius parameter r. Note anharmoniticity is nonzero only for points within r of the boundary, as the ball around any point will only have disparate values when part of it crosses the boundary.

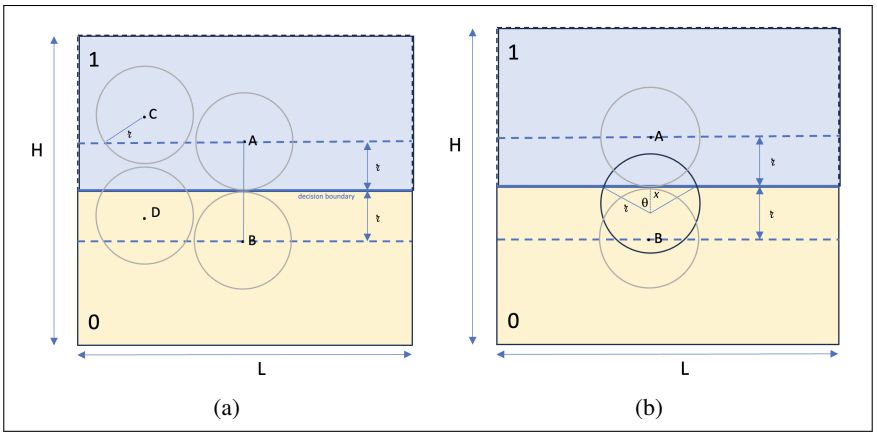


Figure 2: Region of 2d feature space of length L and Height H, with sharp decision boundary in the middle. For a given choice of radius r, anharmoniticity is nonzero only for points within r of the boundary.

Now for a mathematical 'proof' of this. At any given point x in feature space, the anharmoniticity is defined as

$$\gamma(x) \equiv |f(x) - \frac{1}{S_{r,n}} \int_{B(x,r)} f d\Omega_{r,n}| \tag{4}$$

for some choice of ball radius r. In any bounded region R of feature space, then, the average anharmoniticity in this region is defined as

$$\overline{\gamma} \equiv \frac{1}{V_R} \int_R \gamma(x)\, dx \tag{5}$$

Note there is no need to make any assumptions about how the function behaves in feature space, e.g., whether it is smooth or discontinuous, full of pockets of minima and maxima and the like. We simply apply Eqn 5 as-is, as a statement of how close the function is to being harmonic, on average.

But for definiteness, consider the case of a binary classifier, with a region where a sharp decision boundary divides feature space into the '1' class and the '0' class (see Fig 2).

At some points in this region, such as point C in Fig 2(a), the anharmoniticity is zero because the ball around point C is completely on one side of the decision boundary. At other points, e.g., point D, the ball crosses the boundary and thus γ will be nonzero. Indeed for this local patch of feature space, the only points with non-zero γ are those in the locus of points distance r or less from the decision boundary. Integrating γ along the line segment $\overline{AB}$, for example, multiplying by the boundary length L, then dividing by the volume of the whole region ($H \cdot L$), gives us the average γ in this region.

This integral along $\overline{AB}$ is straightforward: following Fig 2(b), we first integrate a sliding circle from distance x below the decision boundary (0 to r), and then again above the decision boundary (0 to r); in each case the integrand being the difference between the function at the center of the circle and the average value on the circle:

$$\overline{\gamma} = \frac{1}{HL}\left(|0 - \int_0^r \frac{2\ arccos\frac{x}{r}}{2\pi} dx| + |1 - \int_0^r \frac{2\pi - 2\ arccos\frac{x}{r}}{2\pi} dx|\right) \tag{6}$$

$$= \frac{2}{\pi}\left(x\ arccos\frac{x}{r} - r\sqrt{1 - (\frac{x}{r})^2}\right)_0^r \tag{7}$$

$$= \frac{2r}{\pi} \tag{8}$$

Thus the average γ of the straight line region Fig. 1(a) is $L \cdot \frac{2r}{\pi}/(LH) = \frac{2r}{\pi H}$. For the curved line region in Fig. 1(b), the computation along a corresponding path $\overline{AB}$ follows exactly the same except there is *more* decision boundary L' to integrate along ($L' > L$), so overall that region will have greater average $\gamma = L' \cdot \frac{2r}{\pi}/(LH) > \frac{2r}{\pi H}$. QED.

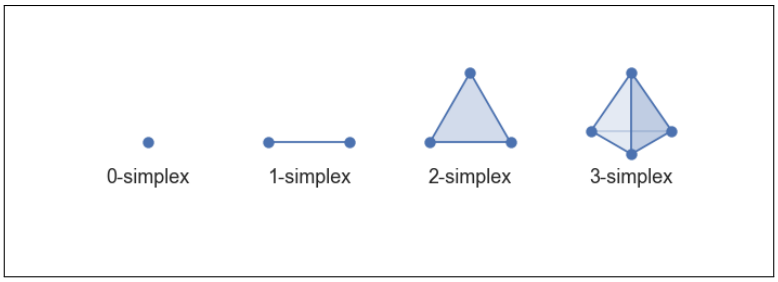


Figure 3: The first few n-simplices ...

3 Simplex Computation

To compute the n-dimensional simplex about any n-dimensional point $\overrightarrow{p}$, we first compute it about the n-dimensional origin. One way to do this is to add an auxiliary (n+1)-st dimension and construct the vectors corresponding to the vertices of the (n+1)-dimensional hypercube:

$$\begin{aligned} v_1 &= (1,0,0,..0)_{n+1} && (9)\\ v_2 &= (0,1,0,..0)_{n+1} && (10)\\ ... &= ... && (11)\\ v_{n+1} &= (0,0,0,..1)_{n+1} && (12)\\ & && (13) \end{aligned}$$

Centering these vectors around the origin by translating by the group average, $\overline{v} = \frac{1}{n+1}(1,1,1,...)_{n+1}$,

$$\begin{aligned} v'_1 &= (1-\frac{1}{n+1},\frac{-1}{n+1},..\frac{-1}{n+1})_{n+1} && (14)\\ v'_2 &= (\frac{-1}{n+1},1-\frac{1}{n+1},..\frac{-1}{n+1})_{n+1} && (15)\\ & ... && (16)\\ v'_{n+1} &= (\frac{-1}{n+1},...,\frac{-1}{n+1},1-\frac{1}{n+1})_{n+1} && (17)\\ & && (18) \end{aligned}$$

we then 'rotate away' the auxiliary (n+1)-st dimension by the angle θ formed by the (n+1)-st unit vector $\hat{n}_1 = (0,0,...,0,1)_{n+1}$ and the normal to the hyperplane formed by the n+1 vectors, $\hat{n}_2 = \frac{1}{\sqrt{n+1}}(1,1,...,1)_{n+1}$ using a generalization of Rodrigues' formula for the rotation by θ in a hyperplane formed by any two orthonormal vectors $\mathbf{n}_1$ and $\mathbf{n}_2$:

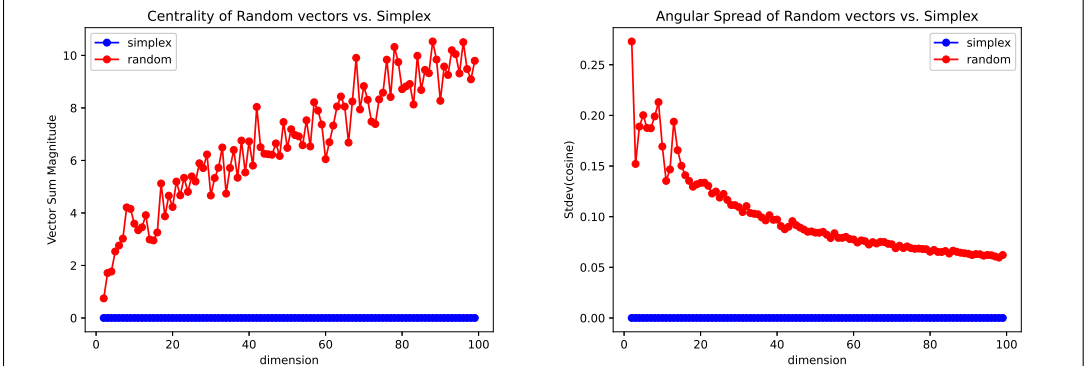
$$\mathcal{R} = I + (\mathbf{n}_2\mathbf{n}_1^T - \mathbf{n}_1\mathbf{n}_2^T)sin\theta + (\mathbf{n}_1\mathbf{n}_1^T + \mathbf{n}_2\mathbf{n}_2^T)(cos\theta - 1) \quad (19)$$

with $\mathbf{n}_1 = \hat{n}_1$, $\mathbf{n}_2 = |\hat{n}_2 - (\hat{n}_2 \cdot \hat{n}_1)\hat{n}_1| = \frac{1}{\sqrt{n}}(1,1,...,1,0)_{n+1}$, and $cos\theta = 1/\sqrt{n+1}$. Thus

$$\begin{aligned} v''_1 &= \mathcal{R}v'_1 && (20)\\ v''_2 &= \mathcal{R}v'_2 && (21)\\ ... &= ... && (22)\\ v''_{n+1} &= \mathcal{R}v'_{n+1} = (0,0,...,0)_{n+1} && (23)\\ & && (24) \end{aligned}$$

where the last vector is 'rotated away' by construction and can be dropped. The other n vectors form the vertices of the origin-centered, symmetric n-simplex in n-dimensions. Now to form the simplex ball about $\overrightarrow{p}$ as in Algorithm 1, one simply adds them vectorially, scaled to magnitude r, i.e., $\overrightarrow{p} + r\mathbf{v}''_1$, $\overrightarrow{p} + r\mathbf{v}''_2$, etc. The simplex + anti-simplex ball adds the negative displacements in as well, i.e., $\overrightarrow{p} - r\mathbf{v}''_1$, $\overrightarrow{p} - r\mathbf{v}''_2$, etc.

Now taking the limit of $n \to \infty$, we see that $\mathcal{R} \to I$, $v' \to v$, and thus $v'' = \mathcal{R}v' \to v$, i.e., the n-simplex vertices for high dimensions converge to the vertices of the n-dimensional hypercube.



(a) (b)

Figure 4: Comparison of space coverage between n random vectors and the n vertex vectors of a (n-1) simplex. Random vectors trace out a random walk and actually never achieve symmetry. The simplex points are all consistent with zero, meaning they are exactly centered and spread evenly.

4 Application to Simple Functions

Let us compare how our harmoniticity metric performs on simple known functions, using two different approximations to the n-ball: one with n-dimensional random vectors, and one using the vertices of the n-simplex.

We first observe how well these two different balls cover space in terms of centrality (how close their vector sum comes to zero) and isotropy (how small is the standard deviation of their angles with respect to a fixed unit vector). For each dimension n, we randomly generate n vectors and compare to the deterministic vectors given by the n-simplex. Adding n random vectors of course replicates the random-walk phenomenon, thus centrality is expected to diverge from the origin as $\sqrt{n}$, as we indeed see in Fig. 4. Isotropy, meanwhile, slowly converges towards zero at high n. For the n-simplices, on the other hand, we have by construction perfect centrality and isotropy.

Now since γ is supposed to measure closeness to "harmoniticity", let us demonstrate this actually works on known harmonic and anharmonic functions. For our first test, consider this pair:

$$f_1(x_0, x_1, ..., x_n) = x_0^2 - x_1^2 + x_2^2 - ... - x_n^2 \quad (25)$$

$$f_2(x_0, x_1, ..., x_n) = x_0^2 + x_1^2 + x_2^2 + ... + x_n^2 \quad (26)$$

$$(27)$$

The first function f_1 is easily seen as harmonic (for even n) since the 2nd derivative of each term is ± 2 which sums to 0, while the second function f_2 is similarly not harmonic as all 2nd derivatives are positive. Choosing 1000 points randomly within the n-dimensional unit-hypercube, constructing an approximate ball around each (either randomly or with simplices as described), we compute the average anharmoniticity (i.e., γ) as per Algorithm 1.

As seen in Fig. 5, the simplex method actually gives the ideal anharmonic value of 0 for the harmonic function, and 1 for the anharmonic function, for all n tested. The random method does not distinguish the functions as well, although this might improve for higher n (or more random vectors).

Testing on another harmonic-anharmonic function pair:

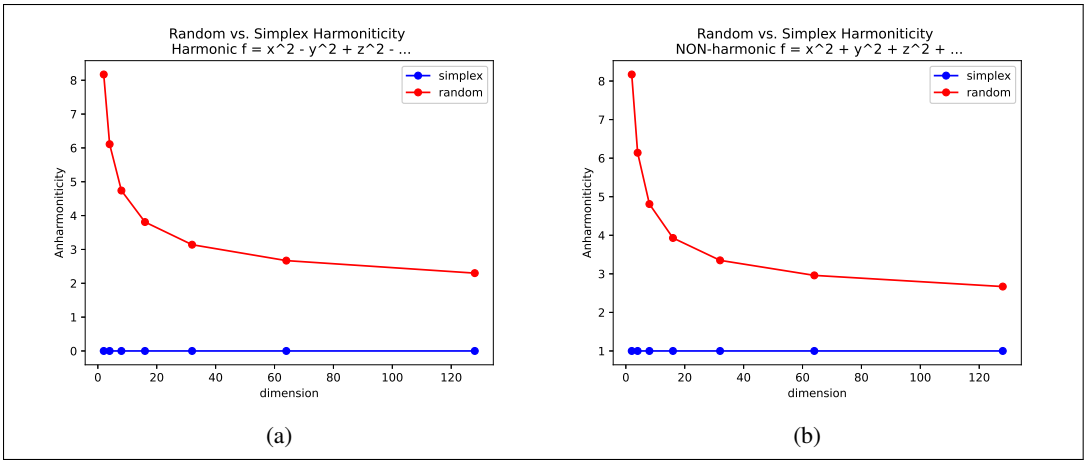
$$f_3(x_0, x_1, ..., x_n) = sin(x_0)e^{x_1} sin(x_2)e^{x_3}...e^{x_n} \quad (28)$$

$$f_4(x_0, x_1, ..., x_n) = e^{x_0+x_1+x_2+...+x_n} \quad (29)$$

$$(30)$$

we repeat the comparison and again the simplex method is superior in Fig. 6. While not getting exactly $\gamma = 0$ for f_3, it is still significantly lower than that of f_4.

So for these simple cases the technique seems to work on pure functions in any dimension, and especially well with the simplex method. Actual ML functions that do something interesting (e.g.,



(a) (b)

Figure 5: Measuring harmoniticity of a known harmonic and anharmonic function, using n random vectors or the n vertex vectors of a (n-1) simplex. The simplex method happens to work ideally here in all dimensions, while random vectors only very slowly improve at higher dimensions.

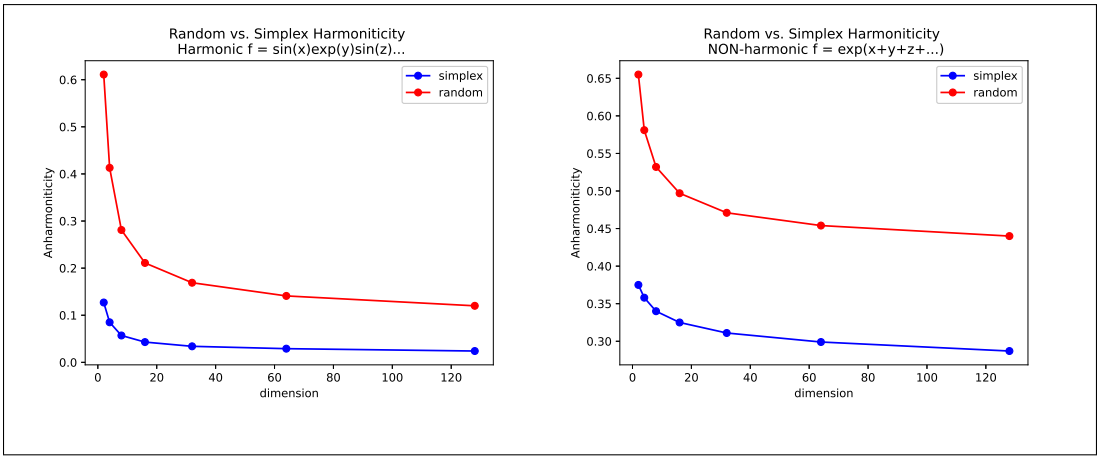


Figure 6: Measuring harmoniticity of another known harmonic and anharmonic function. The simplex method is still better than random vectors, and shows a significantly lower anharmoniticity for the harmonic function.

classifiy MNIST digits, predict credit card fraud, etc.) will of course be much more complicated functions that will have different convergence rates. But based on the above basic observations, we go with the simplex method to most efficiently measure harmoniticity.

5 Dependence on Radius

Here we demonstrate the mild dependence of γ on the choice of r in our main algorithm **??**. For the GBDT and MLP-type models discussed in Section **??**, for example, we have the behavior of γ over a range of radii shown in Fig. 7.

Thus, while 0 is certainly not a good choice of r, anything 'reasonable' will bring out the expected hierarchy in γ between an overfit and regularized model. The exact choice is not important, but one should be consistent across comparisons.

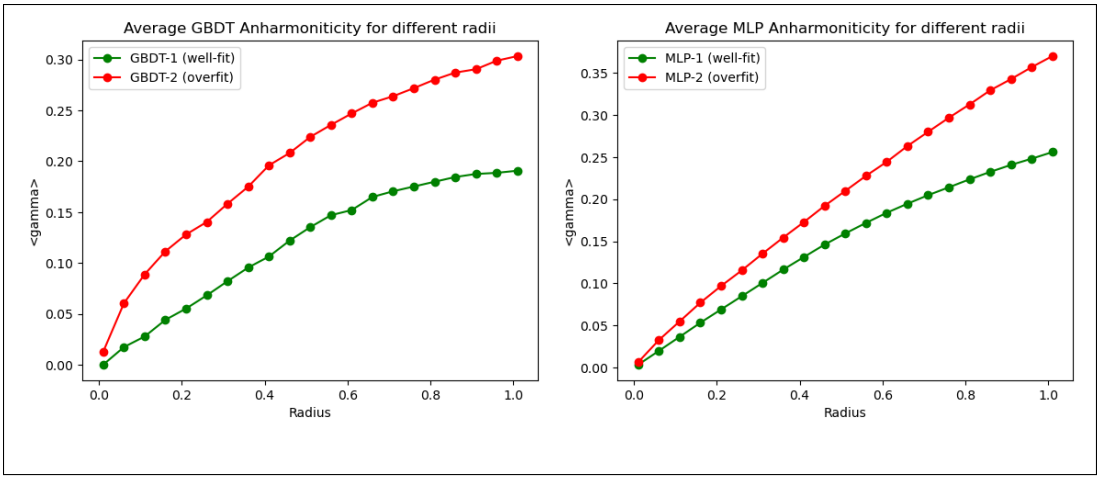


Figure 7: Average anharmoniticity for the GBDT well-fit and overfit models (left) and MLP models (right) at various choices of the radius parameter r. It is significantly greater for the overfit function at any choice of radius r (statistical error is too small to show).

6 ResNet vs. ViT performance

In Table 1 we show more details on the ResNet-50 vs. ViT performance on the animals test set. As explained in the main text, γ by itself is not sufficient to indicate image instability, for one also needs the 'scale' of relevance, i.e. the average size of the logits $\overline{L}$ in the output as well as the value of the average predicted class logit $\overline{\mathcal{P}}_C$ itself. In the combination $\overline{\mathcal{P}}_C e^{-N\overline{\gamma}}$, this does correlate with measured image stability (averaged over 100 images per animal class).

Table 1: Average Accuracy and Stability metrics of 10 animal classes (100 images each) for ResNet-50 and Vision Transformer (ViT). $\overline{\mathcal{P}}_C$ is average softmax probability for the class, and Stability measures percentage of samples with stable classification after N=25 adversarial steps. ViT is typically more accurate and robust but not always (e.g. Cow and Squirrel classes). The 'stability metric' $\overline{\mathcal{P}}_C e^{-N\overline{\gamma}}$ correlates well with observed image Stability in aggregate. Exceptions do arise because the stochastic nature of the gradient ascent procedure does not guarantee that the class logit changes by $N\gamma$ after N steps. The Cat class, for example, has average logit changes of 0.67 and 0.81 for ResNet and ViT, respectively, while $N\overline{\gamma}$ is overestimating at 1.34 and 2.05, respectively. On average across classes it is close: the average logit drift after $N = 25$ iterations is $0.88(2)$ while N times the average γ is 0.94.

Class	Model	$\overline{L}_C$	$\overline{L}$	$\overline{\mathcal{P}}_C$	$\overline{\gamma}$	$\overline{\mathcal{P}}_C e^{-N\overline{\gamma}}$	Accuracy %	Stability %
Chicken	ResNet	-0.12	-9.35	0.911	0.042	0.32	34	57
	ViT	8.91	-6.9 $\cdot 10^{-5}$	0.881	0.027	0.45	**70**	**68**
Butterfly	ResNet	0.038	-9.09	0.929	0.038	0.36	36	52
	ViT	8.84	-3.0 $\cdot 10^{-5}$	0.873	0.034	0.37	**60**	**67**
Sheep	ResNet	0.05	-9.86	0.953	0.037	0.38	40	58
	ViT	8.07	4.5 $\cdot 10^{-5}$	0.762	0.022	0.44	**71**	**81**
Cat	ResNet	0.96	-9.55	0.973	0.054	0.25	80	73
	ViT	10.18	2.8 $\cdot 10^{-4}$	0.963	0.082	0.12	**83**	**76**
Dog	ResNet	0.99	-10.03	0.984	0.040	0.36	87	72
	ViT	11.07	1.2 $\cdot 10^{-4}$	0.985	0.039	0.37	**94**	**80**
Elephant	ResNet	0.96	-10.08	0.984	0.041	0.35	89	75
	ViT	12.98	1.8 $\cdot 10^{-6}$	0.995	0.027	0.51	**90**	**81**
Horse	ResNet	1.71	-9.56	0.987	0.038	0.38	67	72
	ViT	8.95	3.9 $\cdot 10^{-5}$	0.885	0.020	0.54	**89**	**88**
Spider	ResNet	1.94	-9.74	0.992	0.035	0.41	66	78
	ViT	10.99	2.6 $\cdot 10^{-5}$	0.983	0.029	0.48	**73**	**82**
Cow	ResNet	1.84	-10.00	0.993	0.033	0.44	79	**92**
	ViT	9.72	-2.2 $\cdot 10^{-5}$	0.944	0.022	0.54	**90**	90
Squirrel	ResNet	4.08	-9.72	0.999	0.044	0.33	**79**	**88**
	ViT	11.72	1.0 $\cdot 10^{-4}$	0.992	0.044	0.33	77	84