

What does a deep neural network confidently perceive?

The effective dimension of high certainty class manifolds and their low confidence boundaries

Stanislav Fort^{*1}, Ekin Dogus Cubuk², Surya Ganguli¹, and Samuel S. Schoenholz²

¹Stanford University, Stanford, CA, USA

²Google Research, Mountain View, CA, USA

October 12, 2022

Abstract

Deep neural network classifiers partition input space into high confidence regions for each class. The geometry of these class manifolds (CMs) is widely studied and intimately related to model performance; for example, the margin depends on CM boundaries. We exploit the notions of Gaussian width and Gordon’s escape theorem to tractably estimate the effective dimension of CMs and their boundaries through tomographic intersections with random affine subspaces of varying dimension. We show several connections between the dimension of CMs, generalization, and robustness. In particular we investigate how CM dimension depends on 1) the dataset, 2) architecture (including ResNet, WideResNet & Vision Transformer), 3) initialization, 4) stage of training, 5) class, 6) network width, 7) ensemble size, 8) label randomization, 9) training set size, and 10) robustness to data corruption. Together a picture emerges that higher performing and more robust models have higher dimensional CMs. Moreover, we offer a new perspective on ensembling via intersections of CMs. Our code is on [Github](#).

1 Introduction

Training neural networks to classify data is a ubiquitous and classic problem in deep learning. In K -way classification, trained networks naturally partition the space of inputs into K types of regions, $S_k \subset R^D$, containing points that the network confidently predicts have class k . We call these regions *class manifolds* (CMs) of the neural network. In this paper, we analyze the high-dimensional geometry of these CMs, focusing primarily on their *effective dimension* that we define using the Gordon’s escape through a mesh theorem ([Gordon, 1988](#)) and the concept of Gaussian width from high-dimensional geometry ([Vershynin, 2018](#)).

To estimate the dimension of these class manifolds, we perform constrained optimization on random d -dimensional sections (affine subspaces, which are d -dimensional generalizations of lines, planes etc) of input space to actively seek out regions that the neural network assigns to a target class with high confidence. Using optimization in this way allows us to beat the curse of dimensionality ([Bellman, 1957](#)) and find points in the input space that are unlikely to be discovered with other diagnostic techniques such as random sampling. Through a theoretical analysis of high-dimensional geometry, we link the success of such constrained optimization to the effective dimension of the target class manifold using the Gordon’s escape through a mesh theorem ([Gordon, 1988](#)) and the concept of Gaussian width of a set ([Vershynin, 2018](#)). Using extensive experiments, we leverage this method to show deep connections between the geometry of CMs, generalization, and robustness. In particular we investigate how CM dimension depends on the dataset,

^{*}Now at Anthropic. Work done while at Stanford University.

architecture (including ResNet He et al. (2015), WideResNet (Zagoruyko and Komodakis, 2017), and the Vision Transformer (Dosovitskiy et al., 2020)), random initialization, stage of training, class, network width, ensemble size, label randomization, training set size, and model robustness to data corruption. Together a picture emerges that well-performing, robust, models have class manifolds that have higher dimension than inferior models. As a corollary, we offer a unique geometric perspective on ensembling via intersections of CMs.

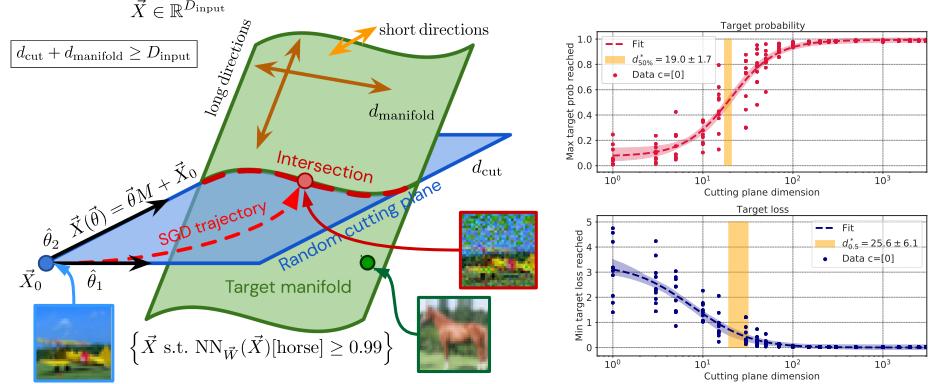


Figure 1: An illustration of finding a point in the intersection between a random cutting plane of dimension d_{cut} (affine subspace) and a high-confidence class manifold (CM) of effective dimension d_{manifold} . If the $d_{\text{cut}} \gtrsim D_{\text{input}} - d_{\text{manifold}}$, there likely exists an intersection between the two. We use optimization from a random point (image) \vec{X}_0 on the d_{cut} affine subspace to find a point in the intersection using gradient descent. The panels below show an example of the dependence of the maximum class probability and minimum loss reached vs. cut dimension d_{cut} . The higher dimensional the cut, the less constrained the available images \vec{X} are, and the more likely we are to find one of high class confidence.

Related work. There has been significant research into understanding linear regions of neural networks, both trained and untrained. Montúfar et al. (2014) studied the number of linear regions in deep neural networks, Raghu et al. (2016) looked at their expressive power with depth, while Serra et al. (2017); Novak et al. (2018) tried to bound and count them. Hanin and Rolnick (2019a) showed that deep ReLU networks have surprisingly few activation patterns, and Hanin and Rolnick (2019b) did the same for the linear regions in the input space. The spectral properties of neural nets were studied in Rahaman et al. (2018), and the stiffness of the functional approximations defined through gradient alignment was coined in Fort et al. (2019). Balestriero and Baraniuk (2018) and Balestriero et al. (2019) use splines to understand class boundaries. While revealing interesting aspects of neural network input space and activations space behavior, the methods used so far have not been able beat the curse of dimensionality – they have stayed local, and analyzed either one- or two-dimensional sections of input space. While our method makes a global estimate of the dimension, local methods based on Maximum Likelihood Estimate pioneered in Levina and Bickel (2005) sparked a fruitful research direction, for example continued by Ma et al. (2018) and their application to adversarial examples.

The exploration of constrained optimization on random, d -dimensional planes in the weight space was employed successfully in Li et al. (2018) to estimate the intrinsic dimension of loss landscapes. Fort and Scherlis (2019) extended this analysis geometrically, and Fort and Jastrzebski (2019) used this and other observations to build a geometric model of the low-loss basins weight-space basins.

Another closely related area concerns adversarial examples and robustness. Szegedy et al. (2013) first noted that there exist points in input space very close to test examples that are mispredicted by neural networks, suggesting CMs of different classes can come very close to each other. Gilmer et al. (2018) showed that the existence of adversarial examples is related to the dimensionality of input space and the accuracy of the classifier. Ford et al. (2019) further link this interplay between dimension, generalization, and adversarial

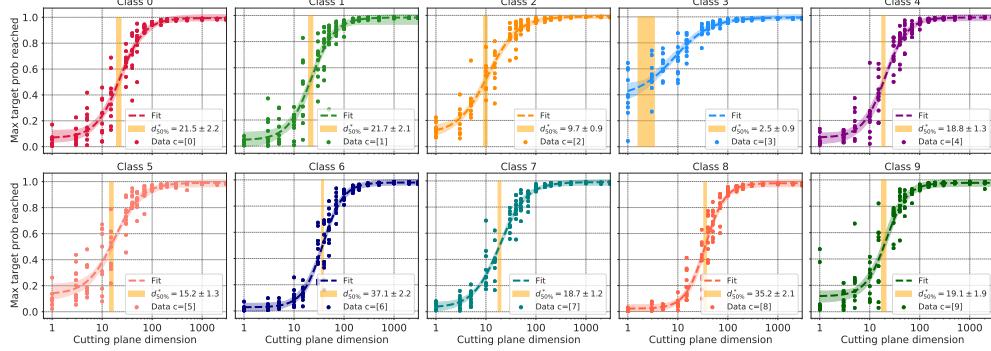


Figure 2: Maximum probability of single classes of CIFAR-10 reached on random cutting planes of dimension d through optimization. The figure shows dependence of the probability of a single class of CIFAR-10 (y-axes) reached via optimization on random cutting affine subspaces of different dimensions (x-axes). The results shown are for a well-trained ($> 90\%$ test accuracy) ResNet20v1 on CIFAR-10. The experiment for each dimension d is repeated $10\times$ with random spans and offsets chosen from the training set. $d_{50\%}^*$ is extracted using a fit. The $d_{50\%}^* \ll 3072$, which implies that the class manifolds are surprisingly high dimensional ($3072 - d_{50\%}^*$) (*all in excess of 3000*).

robustness to more general corruption robustness. In a similar spirit [Salman et al. \(2019\)](#) produce more robust models by convolving neural networks with Gaussian noise in input space. [Ovadia et al. \(2019\)](#) explore model uncertainty in general. Whereas these studies are local, the techniques discussed in this paper are primarily concerned with global properties of CMs.

2 Methods

We seek to determine the effective dimension of class manifolds (CMs). To that end, consider a neural network whose last layer is a softmax yielding normalized probabilities for a given input, $\vec{p}(\vec{X})$. We define a class manifold for class k as the pre-image, $S_k = \vec{p}^{-1}(\{p_k > p_{\text{threshold}}\})$ for some confidence threshold $p_{\text{threshold}}$. We seek to identify the effective dimension of S_k by introducing the *Subspace Tomography Method* (see also Fig. 1).

The Subspace Tomography Method: For a neural network (NN) mapping input $\vec{X} \in \mathbb{R}^D$ into probabilities $\vec{p}(\vec{X}) \in \mathbb{R}^C$, take a random d -dimensional affine subspace defined by orthonormal basis vectors given by rows of $M \in \mathbb{R}^{d \times D}$ and a point $\vec{X}_0 \in \mathbb{R}^D$ (from the training set in our case). Inputs in this subspace are parameterized by $\vec{\theta} \in \mathbb{R}^d$ as $\vec{X}(\vec{\theta}) = \vec{\theta}M + \vec{X}_0$. Given a target probability vector \vec{p}_{target} , we seek to optimize the cross entropy loss, $\mathcal{L}(\vec{p}(\vec{X}(\vec{\theta})), \vec{p}_{\text{target}})$ with respect to $\vec{\theta}$. This will identify points constrained to the affine subspace (M, \vec{X}_0) that have probabilities as close as possible to p_{target} . We study the dependence of \mathcal{L}_{\min} and \vec{p}_{\max} (the loss and probability after optimization) over many repetitions of the procedure on the cut dimension d . We show this analysis estimates the effective codimension of the pre-image of p_{target} : $\{\vec{X} \text{ s.t. } \vec{p}(\vec{X}) = \vec{p}_{\text{target}}\}$ by observing for which d_{cut} the expected \vec{p}_{\max} reaches a threshold.

As discussed in the summary box, we use the cross entropy loss $\mathcal{L}(p(\vec{X}), \hat{p}) = -\hat{p} \cdot \log[\vec{p}(\vec{X})]$ between the target probability vector \vec{p}_{target} and the output of the network to reach the intersection. We use [Adam](#) ([Kingma and Ba, 2017](#)) to minimize \mathcal{L} with respect to $\vec{\theta}$, starting from $\vec{\theta}_0 = \vec{0}$, which corresponds to an initial *random* input $\vec{X}(\vec{\theta}_0) = \vec{X}_0$. We choose this \vec{X}_0 such that it is not of any of the target classes whose dimension we are trying to measure, as discussed in Section 2.1. We found no effect of choosing \vec{X}_0 from different distributions, and decided to use the training set. Through optimization, we take $\vec{\theta}_0 \rightarrow \vec{\theta}_{\min}$. The $\vec{\theta}_{\min}$ defines an optimized input $\vec{X}_{\min} = \vec{\theta}_{\min}M + \vec{X}_0$ and corresponding output $\vec{p}_{\max} = \vec{p}(\vec{X}_{\min})$ that is as

close as possible to \vec{p}_{target} while confining \vec{X} to the random affine subspace (cut) defined by (M, \vec{X}_0) . As a technical detail, we discuss the weak effect of sparsity of M in Fig. 13.

The optimization thus starts with a tuple $(\text{NN}, d, M, \vec{X}_0)$ and maps it to the final probability vector \vec{p}_{\max} and the associated \mathcal{L}_{\min} . By analyzing the dependence of \vec{p}_{\max} and \mathcal{L}_{\min} on the dimension d of the cut we can estimate the effective dimension of the pre-image in input space of a region around \vec{p}_{target} in output space (Fig. 1).

Larsen et al. (2021) use the Subspace Tomography Method to explore the manifold of solutions in the weight space by looking for low-loss parameter configurations on affine subspaces of various types.

2.1 Class manifolds (CMs) and multi-way class boundary manifolds (CBMs)

There are several interesting choices of \vec{p}_{target} . Consider $\vec{p}_{\text{target}} = (0, 0, \dots, 1, \dots, 0)$, a 1-hot vector on a single class k . The pre-image of \vec{p}_{target} is the CM S_k , the set of points in the input space that map to high-confidence class k predictions. The cutting plane method allows us to estimate the effective co-dimension of S_k by computing the dimension d^* at which we reliably obtain a \vec{p}_{\max} whose k 'th component is close to 1 within some tolerance (Fig. 2). More precisely, by choosing a threshold p^* , we are detecting the super-level set of inputs $\{\vec{X} \in \mathbb{R}^D \text{ s.t. } \vec{p}(\vec{X})[k] \geq p^*\}$ (see e.g. Fig. 2).

The cross entropy loss formulation allows us to also study regions that lie in between classes. For example, by setting $\vec{p}_{\text{target}} = (\frac{1}{2}, \frac{1}{2}, 0, \dots, 0)$, our optimization finds regions of input space that lie on a class boundary manifold (CBM) between classes 0 and 1. We can even find multi-way CBMs. For example, a three-way CBM between classes 0, 1, and 2 corresponds to $\vec{p}_{\text{target}} = (\frac{1}{3}, \frac{1}{3}, \frac{1}{3}, 0, \dots, 0)$. At the extreme, we can study the region where all classes have equal probability by setting $\vec{p}_{\text{target}} = (\frac{1}{C}, \frac{1}{C}, \dots, \frac{1}{C})$, where C is the number of classes. The subspace tomography method, therefore, allows us to study the intertwined geometry of multiple CMs and their boundaries. The nature of this geometry is deeply linked with generalization, via the margin, and adversarial examples. See Fig. 7 for results on multi-way CBMs.

2.2 Extracting the critical cutting plane dimension and CM co-dimension $d_{50\%}^*$

Given a particular class target vector \vec{p}_{target} (e.g. $\vec{p}_{\text{target}} = (1, 0, \dots, 0)$ corresponding to the CM S_k with $k = 1$), we perform the subspace tomography experiment multiple times for random \vec{X}_0 and M (both randomly chosen for every experiment) for a sweep of different values of d . For each random draw of M and \vec{X}_0 , we obtain a final probability vectors \vec{p}_{\max} as a function of cutting plane dimension d , as shown e.g. in Fig. 1 and 2. When targeting a single class manifold S_k we plot the k 'th component $\vec{p}_{\max}[k]$. For small values of d , the affine cutting plane will not intersect the target manifold, S_k , and \vec{p}_{\max} will be far from \vec{p}_{target} . For large dimensions, e.g. $d = D$, the subspace is now the full space of inputs, and we can always find a point on the plane such that $\vec{p}_{\max} \approx \vec{p}_{\text{target}}$. For intermediate values of d , the k 'th component of \vec{p}_{\max} will gradually increase with d in expectation. To extract a single cutting plane dimension from this data we 1) fit an empirical curve to the data (Equation 8; typically a good fit), 2) use the mean and covariance of the fitting parameters to obtain a distribution of valid fitting functions, and 3) extract the range of values of d^* where these functions cross a threshold probability, often $p = 50\%$. We call this value $d_{50\%}^*$; in some cases we use thresholds of 25% and 75%, and in principle we can choose whichever we like, understanding that it measures the appropriate superlevel set and we note that in the figures. This cutting plane dimension d^* is the *effective co-dimension* of the CM S_k . Thus the *effective dimension* of the CM S_k is $D - d^*$ (as derived in Section 3).

3 A theory for estimating class manifold dimension through the Tomographic Subspace Method (TSM)

We begin with a simple theoretical description of our method for the case of affine subspaces, after which we will consider the case of realistic CMs.

High-level description. Two affine subspaces of dimensions d_A and d_B generically intersect provided that their dimensions add to at least the dimension of the ambient space D they are embedded in, $d_A + d_B \geq D$.

In algebraic geometry, this statement is known as *dimension counting*, and is equivalent to the statement that the co-dimensions of subspaces are at most additive under intersection (Bourbaki, 1998) (recall that the co-dimension of a subspace of dimension d in a space of ambient dimension D is $D - d$). An illustration of what such intersections can look like for $D = 2$ and $D = 3$ are shown in Fig. 3.

If we know that there reliably exists an intersection, we can use this fact to bound $d_B \geq D - d_A$. The same intuition carries over to a situation where an affine subspace A of dimension d_A intersects a generic manifold B of effective dimension d_B . Our Tomographic Subspace Method uses constrained optimization on randomly chosen affine subspaces, A , to measure the lowest dimension, d^* , at which A reliably intersects B , a class manifold in the input space. This may therefore be used to bound the dimension of B as $d_B \geq D - d^*$. By replacing the linear algebra dimension of the subspace B with the effective dimension of the class manifold, the condition for intersection remains unchanged.

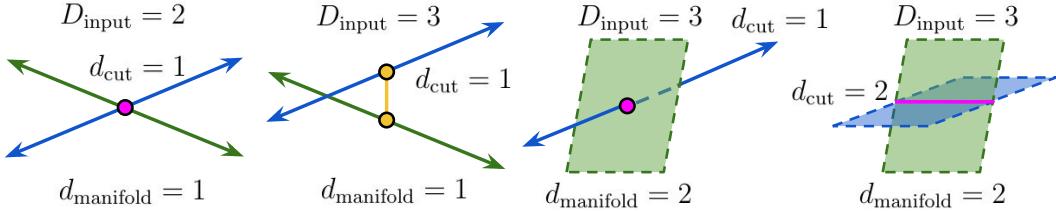


Figure 3: An illustration of the way two affine subspaces of dimensions d_A and d_B can intersect. If the dimensions add up to at least the ambient dimension D , $d_A + d_B \geq D$, the subspaces likely intersect, otherwise they typically do not. If we know D and d_A , we can use the reliable existence of an intersection to bound $d_B \geq D - d_A$. An equivalent result holds beyond affine subspaces.

3.1 Gaussian width and the diameter of a set

Our goal is to study the class manifolds (CMs) and class boundary manifolds (CBMs) in the space of inputs of deep neural networks. For a NN : $\vec{X} \in \mathbb{R}^D \rightarrow \vec{p} \in \mathbb{R}^C$ mapping inputs of dimension D to class probabilities of dimension C , the manifolds in question are the pre-images of a particular target output p_{target} : $\{\vec{X} \text{ s.t. } \vec{p}(\vec{X}) = p_{\text{target}}\}$.

Our method uses the empirically estimated probability of an intersection of an affine subspace with a manifold to measure the dimension of the manifold. The probability of an intersection therefore depends on the extent of the manifold in different directions. For an affine subspace of dimension n , we have n dimensions of length ∞ , and $D - n$ dimensions of length 0. For a generic set, however, the situation is more complicated.

To estimate the effective dimension (which is equal to the statistical dimension (Amelunxen et al., 2014)) of a subset $T \subseteq \mathbb{R}^D$, we need to measure its Gaussian width as defined in Vershynin (2018). We denote the Gaussian width of the set T by $w(T)$. In words, $w(T)$ is defined to be half of the expected diameter of T as measured over all directions and rescaled by the length of a random vector $\vec{g} \in \mathcal{N}(0, 1)^D$. The expected length of this vector is bounded by $D/\sqrt{D+1} < \mathbb{E}(|\vec{g}|_2) < \sqrt{D}$ (Mixon, 2014), and as $D \rightarrow \infty$, $|\vec{g}| \rightarrow \sqrt{D}$. Along a direction \hat{g} , the width of the set T is $\max_{\vec{x}, \vec{y} \in T} (\hat{g} \cdot (\vec{x} - \vec{y}))$. Therefore, mathematically,

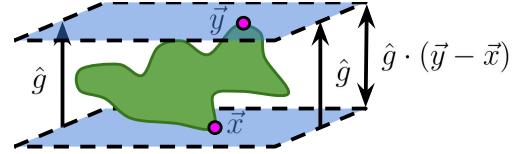


Figure 4: An illustration of measuring the Gaussian width of a set T (in green) in a direction \hat{g} by identifying $\vec{x}, \vec{y} \in T$ in $\max_{\vec{x}, \vec{y} \in T} \hat{g} \cdot (\vec{y} - \vec{x})$. The expectation width using random vectors $\vec{g} \sim \mathcal{N}(0, 1)^D$ instead of \hat{g} is a half of the Gaussian width $w(T)$. Intuitively, it is the characteristic extent of the set T over all directions rescaled by a factor between $D/\sqrt{D+1}$ and \sqrt{D} .

the Gaussian width is defined as

$$\begin{aligned} w(T) &= \frac{1}{2} \mathbb{E}_{\vec{g} \sim \mathcal{N}(0,1)^D} \max_{\vec{x}, \vec{y} \in T} (\vec{g} \cdot (\vec{x} - \vec{y})) \\ &= \mathbb{E}_{\vec{g} \sim \mathcal{N}(0,1)^D} \max_{\vec{x} \in T} (\vec{g} \cdot \vec{x}). \end{aligned} \quad (1)$$

By contrast, the diameter of the set is its maximum extent over all directions

$$\text{diam}(T) = \max_{\vec{g} \sim \mathcal{N}(0,1)^D} \max_{\vec{x}, \vec{y} \in T} (\vec{g} \cdot (\vec{x} - \vec{y}) / \|\vec{g}\|_2). \quad (2)$$

3.2 Effective and statistical dimension

The linear algebra concept of a dimension of a subset T is the smallest dimension of an affine subspace that contains T . This definition is very brittle – an infinitesimal perturbation to a single point in T can change the resulting dimension (Vershynin, 2018). In high-dimensional geometry, *effective dimension* Vershynin (2018) and *statistical dimension* Amelunxen et al. (2014) are both robust and can be estimated using our Subspace Tomography Method.

Gordon's escape through a mesh theorem. As described above, when the target manifold is affine, we know the exact condition for there to exist an intersection with a random affine subspace: their dimensions must add up to at least the dimension of the ambient space, $d_{\text{cut}} + d_{\text{target}} \geq D$. For generic target subsets, T , the condition turns out to be very similar. To show that, we will use the Gordon's escape through a mesh theorem (Gordon, 1988; Mixon, 2014; Amelunxen et al., 2014).

A complication, however, is that the theorem is defined for subsets of the unit sphere $S \subseteq \mathbb{S}^{D-1}$ centered on the point \vec{X}_0 contained in the cutting plane, rather than a generic subset $T \subseteq \mathbb{R}^D$. We resolve this by noticing that were we to project T to the surface of the unit sphere as $\text{proj}_{\vec{X}_0}(T) = \{\vec{X}_0 + (\vec{x} - \vec{X}_0) / \|(\vec{x} - \vec{X}_0)\|_2 \text{ for } \forall \vec{x} \in T\}$, for any cutting plane A passing through \vec{X}_0 the probabilities of intersection are exactly the same,

$$\Pr(A_{\vec{X}_0} \cap T \neq \emptyset) = \Pr(A_{\vec{X}_0} \cap \text{proj}_{\vec{X}_0}(T) \neq \emptyset). \quad (3)$$

Since $\text{proj}_{\vec{X}_0}(T) \subseteq \mathbb{S}_{\vec{X}_0}^{D-1}$, we will refer to $S = \text{proj}_{\vec{X}_0}(T)$ and derive the result below, noting that the same holds for T . The effective dimension measured in this way will therefore be \vec{X}_0 dependent. In practice, we marginalize over different values of \vec{X}_0 to produce a consistent estimate of effective dimension.

The Gordon's escape through mesh theorem allows us to bound the probability that a linear subspace A of dimension d , and co-dimension $k = D - d$, will not intersect the subset S in terms of its Gaussian width $w(S)$.

$$\Pr(A \cap S = \emptyset) \geq 1 - \frac{7}{2} e^{-\frac{1}{18}(a_k - w(S))^2}, \quad (4)$$

where $k/\sqrt{k+1} < a_k < \sqrt{k}$ and the bound is valid only for $w(S) < a_k$. Since we typically have $k \gg 1$ we can assume $a_k = \sqrt{k}$.

The probability of a miss goes down up to the point where $w(S) = a_k$, which we will use to define the effective dimension. Since $a_k \approx \sqrt{D-d}$, this corresponds to $w^2(S) = D-d$. Comparing this to the affine subspace case, we see that $w^2(S)$ now acts as the effective dimension of the target set T whose projection $S = \text{proj}(T)$ we're studying.

$d_{\text{effective}}(T \in \mathbb{R}^D, \vec{X}_0) = w^2 \left(\text{proj}_{\vec{X}_0}(T) \in \mathbb{S}_{\vec{X}_0}^{D-1} \right)$

(5)

3.3 Affine subspaces as a corollary and numerical experiments

In a way, the Gaussian width allows us to count the number of *long* directions of the set T as compared to the distance of T from the origin of the cutting plane. To help build some intuition, we will now apply

Gordon’s escape through the mesh theorem to the case of an affine target space considered above. Imagine an n -dimension affine subspace T ; as described above, such a space is characterized by n dimensions that are infinite in extent and $D - n$ dimensions that have no extent at all. The projection $\text{proj}(T)$ will wrap around an angle π of the unit sphere along the n axes of infinite extent, and will have 0 extent along the others. Therefore $w(S) = \sqrt{n}$ (assuming $D, n \gg 1$). Using Eq. 5, $d_{\text{effective}} = n$, recovering the dimension of the affine subspace we chose to use.

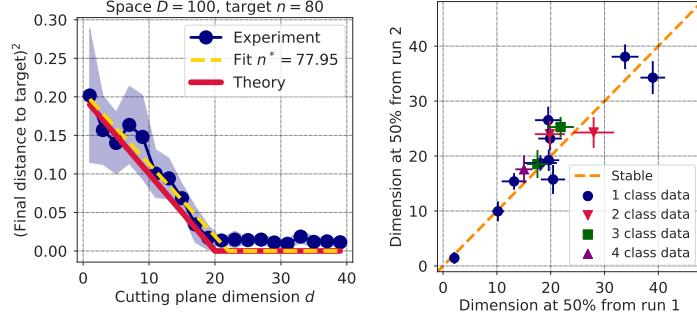


Figure 5: **Left panel:** Distance between random affine subspaces – numerical experiments vs theory from Eq. 6. The figure compares the distance between two subspaces of dimensions d & n in a D -dimensional space. The numerical experiment in JAX is in blue, theory in red, and numerical fit in yellow. **Right panel:** Comparison of the cutting plane dimension needed to get 50% of the target class ($d_{50\%}^*$) for two independently initialized and trained ResNets on CIFAR-10, showing the stability of our method to reinitialization and retraining.

In our subspace tomography method, we control the dimension d_A of a randomly chosen cutting affine subspace and use optimization constrained to it to find an intersection with a class manifold in order to estimate its effective dimension d_B . The dimension $d_A = d_{50\%}^*$ where we can first reliably find a 50% probability image of the target class will be an estimate of the *codimension* of such a CM. An estimate of the dimension of the CM will therefore be $d_B = D - d_{50\%}^*$.

In Sec. A.2 we analytically derive the expected closest distance between two such affine subspaces. The result for $d_A + d_B \geq D$ is exactly 0 (they intersect), while for $d_A + d_B < D$ the $\mathbb{E}[l(A, B)] \propto (\sqrt{D} - d_A - d_B)/\sqrt{D}$. To compare this analytic result to reality, we ran a numerical experiment using automatic differentiation in JAX (Bradbury et al., 2018) where we generated random affine subspaces of different dimensions and measured their closest approach using optimization to locate the point of closest approach (or intersection). An example is shown in Fig. 5.

4 Experiments

We now present our experiments using the tomographic subspace method to measure the dimension of CMs and CBMs, and to make connections to generalization and robustness. The details of the architectures, datasets and precise training procedures are in the Appendix Sec. A.1. The majority of our experiments are done with a standard ResNet20v1 He et al. (2016) and WideResNet on CIFAR-10 and CIFAR-100. To see how architecture-dependent our conclusions were, we also include results from the Vision Transformer model Dosovitskiy et al. (2020), pretrained on ImageNet (Deng et al., 2009), of a radically different design. For the cuts, we choose the random starting point \vec{X}_0 from the train set, making sure it is of a different class than contained in the target vector \vec{p}_{target} .

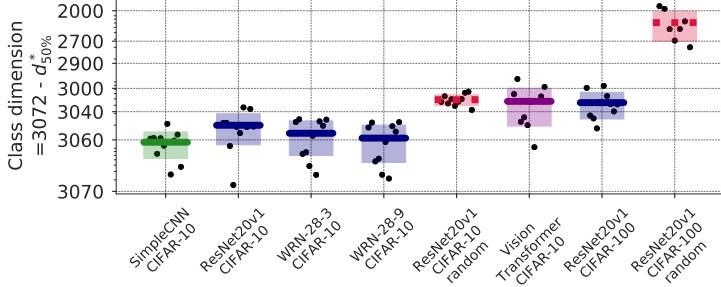


Figure 6: Comparison of the class manifold dimensions ($3072 - d_{50\%}^*$) for SimpleCNN, ResNet, WideResNet and Vision Transformer on CIFAR-10/100 with real and *randomized* labels. Label randomization decreases the class manifold dimension. The dimensions of the learned class manifolds are very high as compared to dimensions of the data alone (Sec. A.10).

4.1 Re-initialization and re-training stability.

If the class manifold dimension is to be seen as a robust property, the results should be stable under reinitialization and retraining of a model. We verified that that is the case, as shown in Fig. 5, comparing the $d_{50\%}^*$ dimensions extracted from single class regions of CIFAR-10, as well several regions between 2, 3 and 4 classes. The results are consistent between the 2 runs.

4.2 Single class manifolds.

The main object of interest for us are the high-confidence single class manifolds. To be precise, we study the supersets of class probability of a class k above a threshold, most often 50% to guarantee that $\text{argmax}(\vec{p}) = k$. Given the continuity of the $\vec{p}(\vec{X})$, the supersets corresponding to higher thresholds will be subsets of the lower thresholds. We present our results for a well-trained ResNet20v1 on CIFAR-10 in Fig. 2, and for CIFAR-100 in Fig. 20, for a SimpleCNN on CIFAR-10 in Fig. 19, and Vision Transformer in Fig. 17. The results show that the $d_{50\%}^*$ (dimension of the cutting plane) is \ll the dimension of the input, therefore the class manifold dimension is very high (summary in Fig. 6), close to the full 3072 dimensions for CIFAR (compared to small estimates of the dimension of the data itself, Sec. A.10).

4.3 Class boundary manifolds between multiple classes.

As described in Sec. 2.1, our method allows us to study the dimension of boundary manifolds between multiple classes. We show results for a well trained ResNet20v1 on CIFAR-10 ($> 91\%$ test accuracy) for several selected sets of classes in Fig. 7. In particular, we look at the region in between all 10 classes, where the network is equally uncertain about all. There, we primarily focus on the loss (Sec. 3) in the bottom row of Fig. 7, since the probability always sums up to 1.

4.4 Training on random labels.

Due to the structure of the training data and the neural network prior, we expect the learned class manifolds to inherit a lot of structure from both. To disentangle the role of the class label, we trained a ResNet20v1 on CIFAR-10 with randomly reshuffled labels. As shown in Zhang et al. (2017), we can reach 100% training accuracy on random labels with a network of high enough capacity. However, as shown in Fig. 5 and 14 the class manifolds learned a significantly higher $d_{50\%}^*$ and therefore smaller dimension than the ones corresponding to the semantically meaningful labels. Since these models completely fail to generalize, this

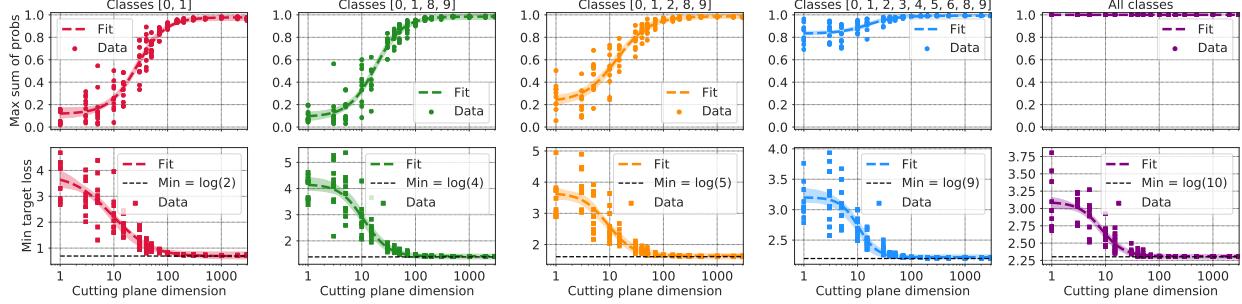


Figure 7: Maximum probability and minimum loss (y-axes) of in-between-classes regions of CIFAR-10 reached on cutting planes of dimension d (x-axes). The results shown are for a well-trained ($> 90\%$ test accuracy) ResNet20v1 on CIFAR-10. Each dimension d is repeated $10\times$ with random planes and offsets (training examples of different than target classes). The last column shows the results for the 10-class region where the network assigns equal probability to each class. The small spread of results for a given d shows that the local difference in dimension are small and that we can estimate it well globally with our cutting plane method.

result is consistent with the hypothesis that generalization and class manifold dimension are intimately related.

4.5 The effect of training set size.

During the course of training, a neural network has to learn to partition the D -dimensional space of inputs into generalizable regions of high class confidence that contain both the training points (directly enforced by loss minimization) and the test points (generalization). To see the role of training set size, we repeated our cutting plane experiments for networks trained to 100% training set accuracy on subsets of CIFAR-10 of size 250, 500, 1.5k, 5k, 15k, and 50k images (=full training set) and added a final point using data augmentation on top, effectively mimicking a larger dataset. The bigger the training set, the smaller the $d_{50\%}^*$, and therefore the larger the dimension of the CMs, as shown in Fig. 8. This trend held across all classes, and continued with augmentation. Better generalization is associated with higher CM dimensionality here. We hypothesize that the larger number of training points might allow the learned partitioning of the input space to connect previously disconnected and lower dimensional CMs through interpolation, thereby effectively increasing CM dimensions with training set size.

4.6 The effect of robustness to data corruptions.

We measure the effect of cutting plane dimension on out-of-domain robustness of neural networks, which has recently been gaining in theoretical and practical importance (see e.g. Ovadia et al. (2019)). Robustness to Gaussian noise was found to be a useful predictor for general robustness as well as adversarial robustness (Ford et al., 2019; Yin et al., 2019). For this reason, we first measure the robustness of WideResNet models to Gaussian noise applied at test time, where noise is sampled from a Gaussian with a standard deviation of 0.05, for each pixel independently. The left panel of Fig. 9 shows the correlation between $d_{50\%}^*$ and error due to Gaussian noise, calculated as the accuracy on corrupted data minus the accuracy on clean data. We see that the models with smaller $d_{50\%}^*$, therefore higher class manifold dimension, are more robust to this type of noise.

Next, we calculate the correlation between the $d_{50\%}^*$ and the accuracy on CIFAR-10-C (Hendrycks and Dietterich, 2018), which includes 15 different corruption types applied at test time (right panel of Fig. 9). These results together show that the effective dimension of neural networks class manifolds is correlated with their robustness to a variety of test-time distortions. The higher the class CM dimension, the better the robustness. Note that the results in Fig. 9 are obtained across a large number of models and hyperparameter

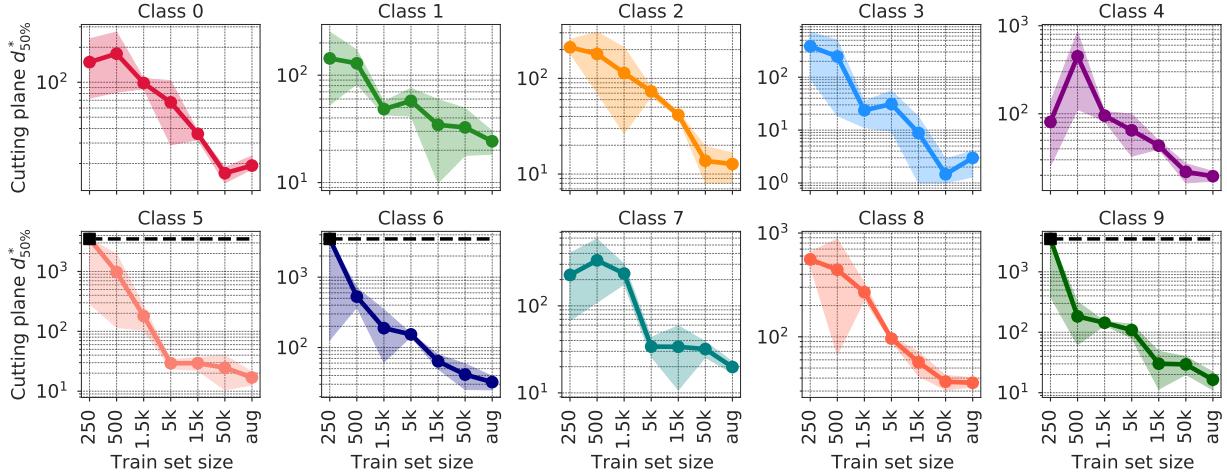


Figure 8: Comparison of the cutting plane dimension needed to get 50% of the target class for ResNets trained to 100% training accuracy on subsets of the training set of CIFAR-10 (mean and standard deviation of 2 networks shown). The bigger the training set, the smaller the $d_{50\%}^*$, therefore the higher the class manifold dimension. The trend continues with the addition of data augmentation (aug), and takes the manifolds from low-D ($\ll D$) for small sets, to high-D ($\approx D$) for large set + augmentation. All classes shown in Fig. 22.

combinations to show the generality of the effect.

4.7 Evolution of dimension with training.

We study the effect of training on the high confidence manifolds in Fig. 10. The early epochs are heavily influenced by the initialization. After a small amount of training, there seems to be an intermediate stage when it is very hard to find high confidence class manifolds ($d_{25\%}^*$ is high, and therefore the manifold dimension low). Towards the end of training, $d_{25\%}^*$ goes down for all classes (details in Figs. 18, 20 and 19). The non-monotonic behavior of the dimension points towards something unusual happening in the intermediate stages of training, and it could be related to the host of phenomena pointing towards the high impact of early stages of training. The causal reason for this remains for future work.

4.8 The effect of network width.

We found that the wider the neural network, the higher the dimension of the CMs (the smaller the dimension $d_{50\%}^*$). Our results for WideResNet-28-K (Zagoruyko and Komodakis, 2017), where K is specifying the width of the layers, are shown in Fig. 11 for the average of all CIFAR-10 classes (individual classes are in Fig. 21).

4.9 Model ensembling.

We found that model ensembling (taking N independently trained models, giving them the same input, and averaging their predicted probabilities) reliably leads to class manifolds of lower dimension, as well as between-class regions of lower dimension. The bigger the ensemble, the lower the dimension, as shown in a summary plot in Fig. 12 (average over all 10 classes). This is atypical, as all other methods of improving performance (e.g. larger training set, more training (towards the end), width) correlated with higher dimensional CMs. This suggests that ensembles might be doing something geometrically distinct from the other methods. This could be related to the observation that, unlike other techniques, deep ensembles combine models from distinct loss landscape basins Fort et al. (2020), which can be partially reached by architectures such as MIMO Havasi et al. (2020). In Fig. 12 we show the effect of ensemble size on the class manifold codimension individually

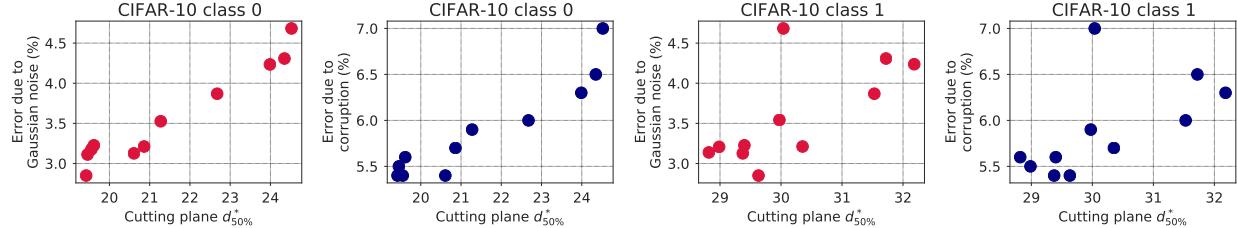


Figure 9: The correlation between class manifold dimension and model robustness to test-time distortions. The left panels show the error due to Gaussian noise applied at test time vs. $d_{50\%}^*$ for classes 0 and 1 (restricted to due the high cost of the experiment). The right panels show the effect of $d_{50\%}^*$ on error due to corruptions in CIFAR-10-C. Models with higher CM dimension are more robust to both Gaussian noise and to distortions in the Common Corruptions Benchmark. The plots show averages over a large number of models and random hyperparameter choices.

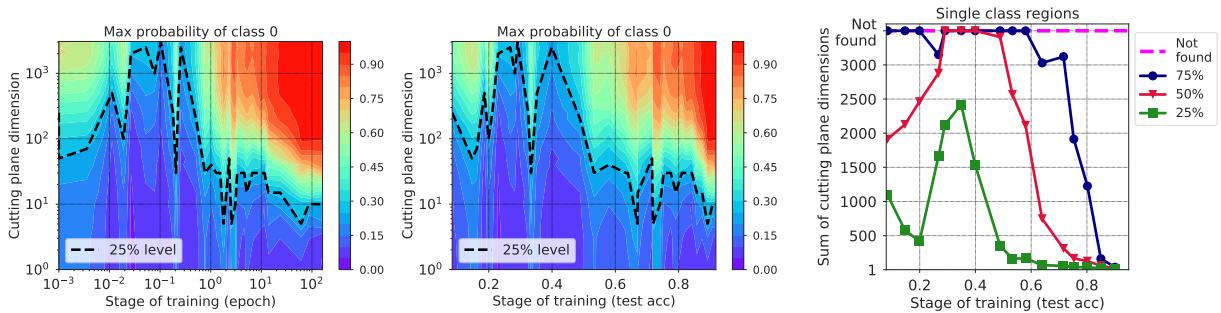


Figure 10: The effect of training on the dimension of cutting plane necessary to reach a particular probability. The two left panels show the maximum probability of class 0 reached for cutting planes of different dimensions over training for ResNet20v1 on CIFAR-10. The probability 25% level (superset) is highlighted. The right panel shows $d_{25\%}^*$, $d_{50\%}^*$ and $d_{75\%}^*$ for the average of all single class regions. High confidence regions become hard to find at intermediate stages. Towards the end of training, the dimension of the manifolds grows (codimension d^* goes down). The breakdown by classes is shown in Fig. 18.

for all 10 classes of CIFAR-10. A very simple model, predicting that the codimension of class manifold for an ensemble of N models scales linearly with N is born out well for small N there. This is supported by the right panels in Fig. 12 which show a 2D section of the input space with class regions of different classes highlighted for 3 different models and ensembles of different sizes. The class regions seem relatively randomly oriented, leading to the addition of manifold codimensions, explaining the relation $\text{codim} \propto N$ observed in Fig. 12.

5 Conclusion

We propose a new tool that we call the Subspace Tomography method for estimating the dimension of class manifolds and multi-way class boundary manifolds in the space of inputs for deep neural networks. To circumvent the curse of dimensionality, we use optimization constrained to randomly chosen affine subspaces (cutting planes) of varying dimension. This allows us to extract the effective dimension of the class manifolds as well regions between classes. Our mathematical analysis uses the concept of Gaussian width and the Gordon's escape through mesh theorem from high-dimensional geometry to define a robust, effective dimension. We study the manifold dimension as a function the network, architecture, stage of training, accuracy and

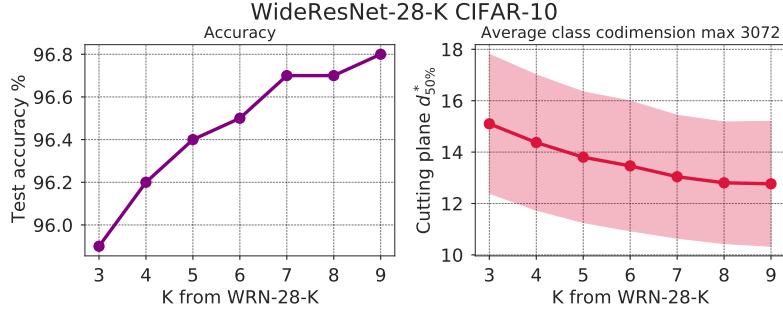


Figure 11: The effect of network width on the dimension. The left panel shows the final test accuracy of a WRN-28-K on CIFAR-10 for different values of the width K . The right panel shows $d_{50\%}^*$, the dimension of a cutting plane need to reach 50% averaged over 10 classes (individual results shown in Fig. 21). The wider the network, the higher the dimension of the class manifolds.

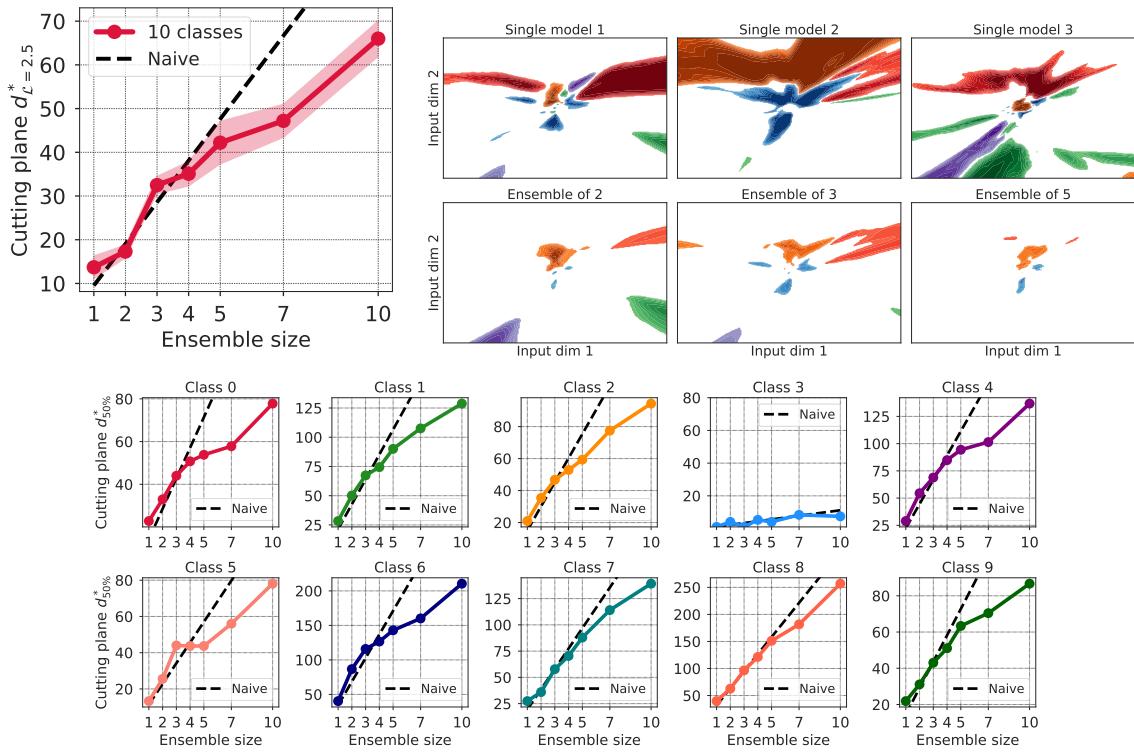


Figure 12: The effect of model ensembling on the dimension $d_{50\%}^*$ needed to reach 50% accuracy averaged over all CIFAR-10 classes (top left) (individual class in the bottom row) for ResNet20 trained for 50 epochs. Across all classes, the larger the ensemble, the higher the $d_{50\%}^*$ and therefore the lower the class manifold dimension. A naive model of addition of codimensions between models is overlayed, showing a surprisingly good fit for small ensembles. The right panels show a fixed random section of the input space for 3 single models (top row) and 3 ensemble sizes (bottom). The colors indicate 4 different classes > 50%. The elongated high-probability structures disappear with ensembled, as they get averaged.

robustness and find a ubiquitous correlation between higher class manifold dimension and better performance and robustness along the many axes tested points towards an intimate link between the geometry of the input

space class partitioning and generalization. Ensembling is the only technique amongst the ones we explored that both increases performance and decreases the manifold dimension at the same time, suggesting that its beneficial effects might be geometrically distinct from other ways of improving performance.

Acknowledgments

We would like to thank Ilya Tolstikhin from Google Research Zurich who was instrumental in the early phases of development of this project, and Dustin Mixon from Ohio State University for discussions on Gordon's escape through a mesh theorem.

References

- Yehoram Gordon. On milman’s inequality and random subspaces which escape through a mesh in \mathbb{R}^n . In *Geometric aspects of functional analysis*, pages 84–106. Springer, 1988.
- Roman Vershynin. *High-Dimensional Probability: An Introduction with Applications in Data Science*. Cambridge Series in Statistical and Probabilistic Mathematics. Cambridge University Press, 2018. doi: 10.1017/9781108231596.
- Richard Bellman. *Dynamic Programming*. Dover Publications, 1957. ISBN 9780486428093.
- Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition, 2015.
- Sergey Zagoruyko and Nikos Komodakis. Wide residual networks, 2017.
- 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, 2020.
- Guido Montúfar, Razvan Pascanu, Kyunghyun Cho, and Yoshua Bengio. On the number of linear regions of deep neural networks, 2014.
- Maithra Raghu, Ben Poole, Jon Kleinberg, Surya Ganguli, and Jascha Sohl-Dickstein. On the expressive power of deep neural networks, 2016.
- Thiago Serra, Christian Tjandraatmadja, and Sri Kumar Ramalingam. Bounding and counting linear regions of deep neural networks, 2017.
- Roman Novak, Yasaman Bahri, Daniel A Abolafia, Jeffrey Pennington, and Jascha Sohl-Dickstein. Sensitivity and generalization in neural networks: an empirical study. *arXiv preprint arXiv:1802.08760*, 2018.
- Boris Hanin and David Rolnick. Deep relu networks have surprisingly few activation patterns, 2019a.
- Boris Hanin and David Rolnick. Complexity of linear regions in deep networks, 2019b.
- Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred A. Hamprecht, Yoshua Bengio, and Aaron Courville. On the spectral bias of neural networks, 2018.
- Stanislav Fort, Paweł Krzysztof Nowak, Stanisław Jastrzebski, and Srinivas Narayanan. Stiffness: A new perspective on generalization in neural networks, 2019.
- Randall Balestrieri and Richard Baraniuk. Mad max: Affine spline insights into deep learning, 2018.
- Randall Balestrieri, Romain Cosentino, Behnaam Aazhang, and Richard Baraniuk. The geometry of deep networks: Power diagram subdivision, 2019.
- Elizaveta Levina and Peter J Bickel. Maximum likelihood estimation of intrinsic dimension. In *Advances in neural information processing systems*, pages 777–784, 2005.
- Xingjun Ma, Bo Li, Yisen Wang, Sarah M. Erfani, Sudanthi Wijewickrema, Grant Schoenebeck, Dawn Song, Michael E. Houle, and James Bailey. Characterizing adversarial subspaces using local intrinsic dimensionality, 2018.
- Chunyuan Li, Heerad Farkhoor, Rosanne Liu, and Jason Yosinski. Measuring the intrinsic dimension of objective landscapes, 2018.

Stanislav Fort and Adam Scherlis. The goldilocks zone: Towards better understanding of neural network loss landscapes. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33:3574–3581, Jul 2019. ISSN 2159-5399. doi: 10.1609/aaai.v33i01.33013574. URL <http://dx.doi.org/10.1609/aaai.v33i01.33013574>.

Stanislav Fort and Stanislaw Jastrzebski. Large scale structure of neural network loss landscapes, 2019.

Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2013.

Justin Gilmer, Luke Metz, Fartash Faghri, Samuel S. Schoenholz, Maithra Raghu, Martin Wattenberg, and Ian Goodfellow. Adversarial spheres, 2018.

Nic Ford, Justin Gilmer, Nicolas Carlini, and Dogus Cubuk. Adversarial examples are a natural consequence of test error in noise, 2019.

Hadi Salman, Jerry Li, Ilya Razenshteyn, Pengchuan Zhang, Huan Zhang, Sebastien Bubeck, and Greg Yang. Provably robust deep learning via adversarially trained smoothed classifiers. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d’Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 11292–11303. Curran Associates, Inc., 2019. URL <http://papers.nips.cc/paper/9307-provably-robust-deep-learning-via-adversarially-trained-smoothed-classifiers.pdf>.

Yaniv Ovadia, Emily Fertig, Jie Ren, Zachary Nado, D Sculley, Sebastian Nowozin, Joshua V. Dillon, Balaji Lakshminarayanan, and Jasper Snoek. Can you trust your model’s uncertainty? evaluating predictive uncertainty under dataset shift, 2019.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2017.

Brett W. Larsen, Stanislav Fort, Nic Becker, and Surya Ganguli. How many degrees of freedom do we need to train deep networks: a loss landscape perspective, 2021. URL <https://arxiv.org/abs/2107.05802>.

N. Bourbaki. *Algebra I: Chapters 1-3*. Actualités scientifiques et industrielles. Springer, 1998. ISBN 9783540642435. URL <https://books.google.cz/books?id=STS9aZ6F204C>.

D. Amelunxen, M. Lotz, M. B. McCoy, and J. A. Tropp. Living on the edge: phase transitions in convex programs with random data. *Information and Inference*, 3(3):224–294, Jun 2014. ISSN 2049-8772. doi: 10.1093/imai/iau005. URL <http://dx.doi.org/10.1093/IMAI/IAU005>.

Dustin G Mixon. Gordon’s escape through a mesh theorem, Feb 2014. URL <https://dustingmixon.wordpress.com/2014/02/08/gordons-escape-through-a-mesh-theorem/>.

James Bradbury, Roy Frostig, Peter Hawkins, Matthew James Johnson, Chris Leary, Dougal Maclaurin, and Skye Wanderman-Milne. JAX: composable transformations of Python+NumPy programs, 2018. URL <http://github.com/google/jax>.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2016. doi: 10.1109/cvpr.2016.90. URL <http://dx.doi.org/10.1109/cvpr.2016.90>.

J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

Chiyuan Zhang, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals. Understanding deep learning requires rethinking generalization, 2017.

Dong Yin, Raphael Gontijo Lopes, Jon Shlens, Ekin Dogus Cubuk, and Justin Gilmer. A fourier perspective on model robustness in computer vision. In *Advances in Neural Information Processing Systems*, pages 13276–13286, 2019.

Dan Hendrycks and Thomas G Dietterich. Benchmarking neural network robustness to common corruptions and surface variations. *arXiv preprint arXiv:1807.01697*, 2018.

Stanislav Fort, Huiyi Hu, and Balaji Lakshminarayanan. Deep ensembles: A loss landscape perspective, 2020.

Marton Havasi, Rodolphe Jenatton, Stanislav Fort, Jeremiah Zhe Liu, Jasper Snoek, Balaji Lakshminarayanan, Andrew M Dai, and Dustin Tran. Training independent subnetworks for robust prediction. *arXiv preprint arXiv:2010.06610*, 2020.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *CoRR*, abs/1502.03167, 2015. URL <http://dblp.uni-trier.de/db/journals/corr/corr1502.html#IoffeS15>.

Yann LeCun and Corinna Cortes. MNIST handwritten digit database. 2010. URL <http://yann.lecun.com/exdb/mnist/>.

Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms, 2017.

Alex Krizhevsky, Vinod Nair, and Geoffrey Hinton. Cifar-10 (canadian institute for advanced research). URL <http://www.cs.toronto.edu/~kriz/cifar.html>.

Sergey Zagoruyko and Nikos Komodakis. Wide residual networks. *arXiv preprint arXiv:1605.07146*, 2016.

Ekin D Cubuk, Barret Zoph, Dandelion Mane, Vijay Vasudevan, and Quoc V Le. Autoaugment: Learning augmentation policies from data. *arXiv preprint arXiv:1805.09501*, 2018.

Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.

Peiran Gao, Eric Trautmann, Byron Yu, Gopal Santhanam, Stephen Ryu, Krishna Shenoy, and Surya Ganguli. A theory of multineuronal dimensionality, dynamics and measurement. *BioRxiv*, page 214262, 2017.

A Appendix

A.1 Details of networks, datasets and training

In this paper we use two architectures: 1) SimpleCNN, which is a simple 4-layer CNN with 32, 64, 64 and 128 channels, ReLU activations and maxpool after each convolution, followed by a fully-connected layer, and 2) ResNet20v1 as described in He et al. (2015) with batch normalization on (Ioffe and Szegedy, 2015). We also analyze the Vision Transformer Dosovitskiy et al. (2020) pretrained on ImageNet and finetuned on CIFAR-10. We use 5 datasets: MNIST (LeCun and Cortes, 2010), Fashion MNIST (Xiao et al., 2017), CIFAR-10 and CIFAR-100 (Krizhevsky et al.), and ImageNet Deng et al. (2009). The ResNet is trained for 200 epochs using SGD+Momentum at learning rate 0.1, dropping to 0.01 at epoch 80 and 0.001 at epoch 120. The L_2 norm regularization is 10^{-4} . In one experiment, we use data augmentation as described in ¹. For our robustness experiments, we used the Wide-ResNet models (Zagoruyko and Komodakis, 2016) available in ². We trained 11 different sizes of Wide-ResNet models (WRN-28-2 to WRN-28-12) with AutoAugment. Each model was trained from 15 different random weight-initializations for better statistics. We used the following hyperparameters to train each model: a learning decay of 0.1, weight decay of 5e-4, cosine learning rate decay in 200 epochs, and AutoAugment (Cubuk et al., 2018) for data augmentation.

To see what the effect of diverse architectures were on our conclusions, we experimented with the new Vision Transformer (Dosovitskiy et al., 2020) that was pretrained on ImageNet and finetuned on CIFAR-10, as recommended in their published code³.

A.2 Detailed derivation of the closest approach of two affine subspaces

Let us consider a situation in which in a D -dimensional space we have a randomly chosen d -dimensional affine subspace A defined by a point $\vec{X}_0 \in \mathbb{R}^D$ and a set of d orthonormal basis vectors $\{\hat{v}_i\}_{i=1}^d$ that we encapsulate into a matrix $M \in \mathbb{R}^{d \times D}$. Let us consider another random n -dimensional affine subspace B . Our task is to find a point $\vec{X}^* \in A$ that has the minimum L_2 distance to the subspace B , mathematically $\vec{X}^* = \operatorname{argmin}_{\vec{X} \in A} \left| \vec{X} - \operatorname{argmin}_{\vec{X}' \in B} \left| \vec{X} - \vec{X}' \right| \right|$. In words, we are looking for a point in the d -dimensional subspace A that is as close as possible to its closest point in the n -dimensional subspace B . A point within the subspace A is parametrized by a d -dimensional vector $\vec{\theta} \in \mathbb{R}^d$ by $\vec{X}(\vec{\theta}) = \vec{\theta}M + \vec{X}_0 \in A$. This parametrization ensures that for all choices of $\vec{\theta}$ the resulting $\vec{X} \in A$.

Without loss of generality, let us consider the case where the n basis vectors of the subspace B are aligned with the dimensions $D - n, D - n + 1, \dots, D$ of the coordinate system. Let us call the remaining axes $s = D - n$ the *short* directions of the subspace B . A distance from a point \vec{X} to the subspace B now depends only on its coordinates $1, 2, \dots, s$. Therefore $l^2(\vec{X}, B) = \sum_{i=1}^s X_i^2$. This is the case because of our purposeful choice of coordinates.

Given that the only coordinates influencing the distance are the first s values, let us, without loss of generality, consider a \mathbb{R}^s subspace of the original \mathbb{R}^D only including those. Then the distance between a point within the subspace A parametrized by the vector $\vec{\theta}$ is $l^2(\vec{X}(\vec{\theta}), B) = \left| \vec{\theta}M + \vec{X}_0 \right|^2$. Given our restrictions, now the $\vec{\theta} \in \mathbb{R}^d$, $M \in \mathbb{R}^{d \times s}$ and $\vec{X}_0 \in \mathbb{R}^d$. The distance l attains its minimum for $\partial_{\vec{\theta}} l^2 = (\vec{\theta}M + \vec{X}_0) M^T = \vec{0}$, producing the minimality condition $\vec{\theta}^* M = -\vec{X}_0$. There are now 3 cases:

1. The overdetermined case, $d > s$. In case $d > s = D - n$, the optimal $\vec{\theta}^* = -\vec{X}_0 M^{-1}$ belongs to a $(d - s = d + n - D)$ -dimensional family of solutions that attain 0 distance to the plane B . In this case the affine subspaces A and B intersect and share a $(d + n - D)$ -dimensional intersection.

2. A unique solution case, $d = s$. In case of $d = s = D - n$, the solution is a unique $\vec{\theta}^* = -\vec{X}_0 M^{-1}$. After plugging this back to the distance equation, we obtain $\vec{\theta}$ is $l^2(\vec{X}(\vec{\theta}^*), B) = \left| -\vec{X}_0 M^{-1} M + \vec{X}_0 \right|^2 =$

¹https://github.com/keras-team/keras/blob/master/examples/cifar10_resnet.py

²<https://github.com/tensorflow/models/tree/master/research/autoaugment>

³https://github.com/google-research/vision_transformer

$|- \vec{X}_0 + \vec{X}_0|^2 = 0$. The square (in this case) matrix M and its inverse M^{-1} cancel each other out.

3. An underdetermined case, $d < s$. In case of $d < s$, there is generically no intersection between the subspaces. The inverse of M is now the Moore-Penrose inverse M^+ . Therefore the closest distance is $\vec{\theta}$ is $l^2(\vec{X}(\vec{\theta}^*), B) = |- \vec{X}_0 M^+ M + \vec{X}_0|^2$. Before our restriction from $D \rightarrow s$ dimensions, the matrix M consisted of d D -dimensional, mutually orthogonal vectors of unit length each. We will consider these vectors to be component-wise random, each component with variance $1/\sqrt{D}$ to satisfy this condition on average. After restricting our space to s dimensions, M 's vectors got reduced to s components each, keeping their variance $1/\sqrt{D}$. They are still, in expectation, mutually orthogonal, however, their length got reduced to \sqrt{s}/\sqrt{D} . The (transpose) of the inverse M^+ consists of vectors of the same directions, with their lengths scaled up to \sqrt{D}/\sqrt{s} . That means that, in expectation, MM^+ is a diagonal matrix with d diagonal components set to 1, and the remainder being 0. The matrix $(I - M^+M)$ contains $(s-d)$ ones on its diagonal. The projection $|\vec{X}_0(I - M^+M)|^2$ is therefore of the expected value of $|X_0|^2(s-d)^2/D$. The expected distance between the d -dimensional subspace A and the d -dimensional subspace B is, in expectation

$$\mathbb{E}[d(A, B)] \propto \begin{cases} \frac{\sqrt{D-n-d}}{\sqrt{D}} & n+d < D, \\ 0 & n+d \geq D. \end{cases} \quad (6)$$

We ran a numerical experiment using automatic differentiation in JAX (Bradbury et al., 2018) where we generated random affine subspaces of different dimensions and measured their closest approach using optimization to locate the place. The numerical results presented in Figure 5 match the analytic predictions in Equation 6 well.

A.3 Empirical fit function

The empirical fit function we use to extract the critical dimension of the cutting hyperplane $d_{50\%}^*$ is shown in 8.

$$p(d; A, B, C, D) = A + \frac{B}{1 + \exp(-\log(d/C)/D)}. \quad (7)$$

It is a sigmoid function that depends logarithmically on the dimension d and can be offset from $p=0$ at for low d and from $p=1$ for high d . That is the case as sometimes the neural networks we analyzed would not have any regions of a particular class reaching all the way to 100%. In other cases, even optimization in a line $d=1$ would be able to get to a $p > 10\%$ (for 10 class classification).

For fitting the loss $\mathcal{L}(d)$, we utilized the fact that the cross-entropy loss depends logarithmically on p , and therefore used

$$\mathcal{L}(d; A, B, C, D) = -\log \left[A + \frac{B}{1 + \exp(-\log(d/C)/D)} \right]. \quad (8)$$

In both cases A, B, C and D are free fit parameters. We used SciPy optimizer (Virtanen et al., 2020) to find the parameters and their covariance.

A.4 Cutting plane axis-alignment – the effect of sparsity

When choosing the matrix M that defines the span of the subspace in which we are optimizing, we can choose to make the rows of M sparse. On one end, each basis vectors might generically be non-zero in each of its components, while on the other end, a single non-zero element per basis vector is allowed. Geometrically, this corresponds to the alignment of the subspace with the axes (pixels and their channels for images) of the input space. Figure 13 shows the effect of the sparsity of M on the resulting $d_{25\%}^*, d_{50\%}^*, d_{75\%}^*$ and $d_{90\%}^*$. The sparser the M , the higher the dimension needed to reliably reach the 25%, 50%, 75%, and 90% class confidence region respectively. The effect of sparsity is visible, however, it is 1) not very significant (changing the dimension by a small part of the total $D = 3072$ for CIFAR-10), and 2) its effect disappears for even small amounts of non-zero elements in M .

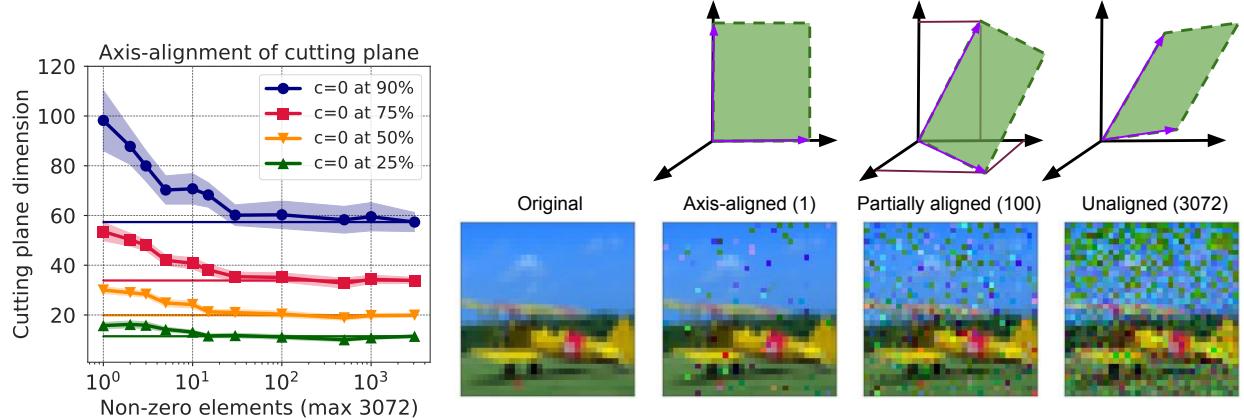


Figure 13: The effect of axis alignment of the cutting planes. The figure shows the cutting plane dimension necessary to reach 4 thresholds levels (the 4 data lines) of class 0 probability (y-axis) from a random starting point for a well trained ResNet20v1 on CIFAR-10. We vary the number of non-zero elements of the basis vectors of the random cutting plane (x-axis). For a small number of non-zero elements, single pixels are varied, while for a 3072 non-zero elements (the maximum value), all pixels are varied jointly. The axis-aligned random cuts require higher dimensions to hit the same accuracy regions of class 0.

A.5 Training on randomly permuted labels

For training on randomly permuted labels of the training set, we observe the critical dimension $d_{50\%}^*$ to rise significantly, meaning that a much higher dimensional cutting plane is needed to reliably intersect a class manifold. The breakdown by class for ResNet20v1 on CIFAR-10 and CIFAR-100 is shown in Figure 14. The comparison to semantically meaningful labels is shown in Figure 6.

A.6 Additional cutting curves for CIFAR-10 and CIFAR-100

Two additional detailed cutting plane results can be found in this subsection: SimpleCNN on CIFAR-10 in Figure 15, and ResNet20v1 on CIFAR-100 in Figure 16.

A.7 Dimension as a function of training stage

While Figure 10 shows the aggregate effect of training epoch on the the critical cutting plane dimension averaged over all single-class regions, the detailed per-class results can be found in Figure 18 for ResNet20v1 on CIFAR-10 (two independently initialized and trained models), in Figure 19 for SimpleCNN on CIFAR-10, and in Figure 20 for ResNet20v1 on CIFAR-100.

A.8 The effect of network width

We found that wider networks have lower class manifold dimensions. Our results for WideResNet-28-K ([Zagoruyko and Komodakis, 2017](#)) (WRN-28-K, where K is specifying the width of the layers) averaged over all 10 classes of CIFAR-10 are shown in Figure 11. The results for individual classes are shown in Figure 21. The trend that with higher width K the $d_{50\%}^*$ goes down and therefore the manifold dimension goes up holds for individual classes as well as their average.

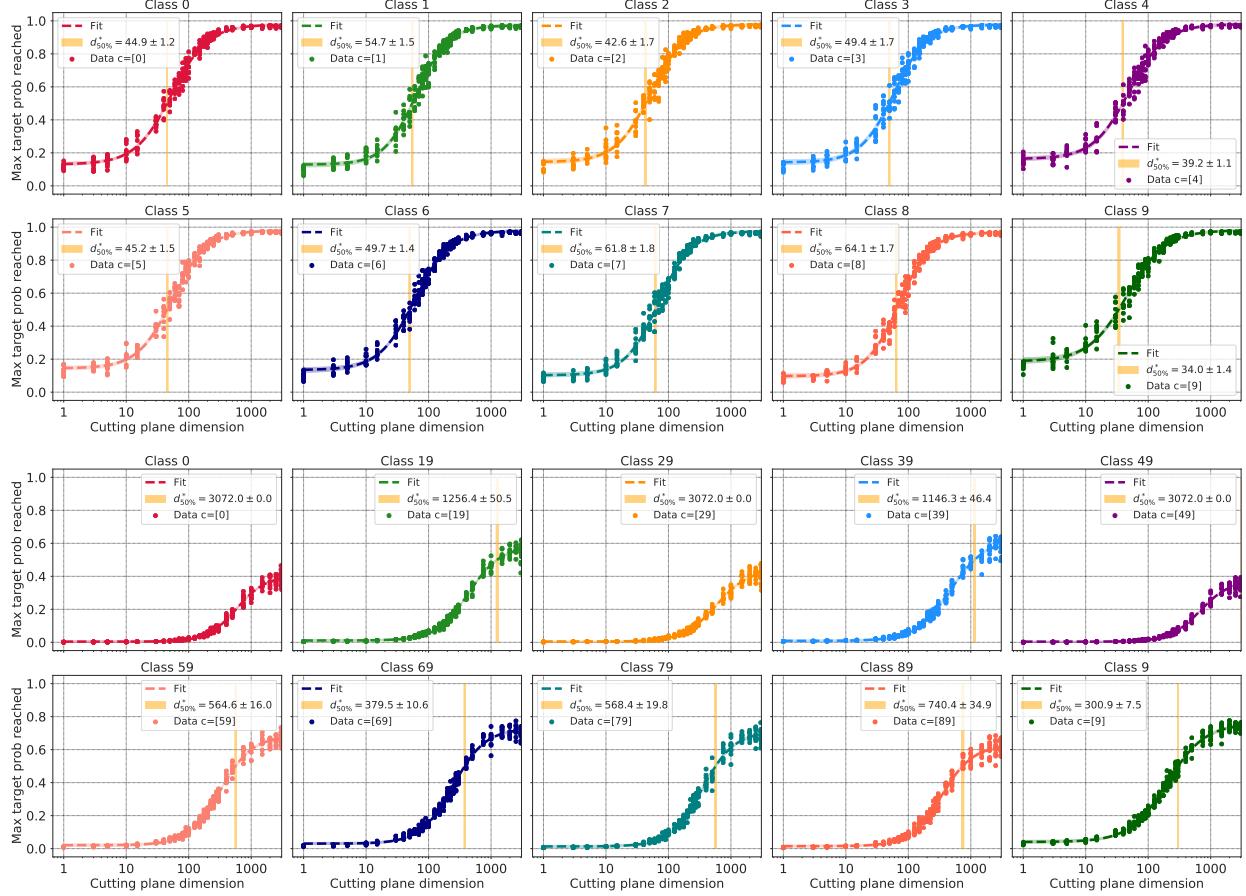


Figure 14: Maximum probability reached on cutting planes of different dimensions for all 10 target classes of CIFAR-10 (top row) and CIFAR-100 (bottom row) for a ResNet20v1 trained to 100% training accuracy on *randomly permuted* class labels. The $d_{50\%}^*$ is consistently higher and therefore the dimension of the high confidence manifolds is lower than for semantically meaningful labels (Figure 2), suggesting geometrically a very different function being learned.

A.9 The effect of training set size

In Fig. 8 we show an example of the training set size dependence of the cutting plane dimension $d_{50\%}^*$ for 3 classes of CIFAR-10. The results for all 10 classes can be found in Fig. 22.

A.10 Simple measures of dataset dimensionality

In this work we focus on measuring the dimension of the learned class manifolds that a trained neural network develops during the course of training on a dataset. Generally, the dimensions we find are very high, for examples look at the summary Figure 6. For CIFAR-10 and 100 we observe class dimension manifolds of even 3000 and above out of 3072. To get a comparison between the learned manifold and the dataset itself, we looked at several simple measures of dimension for the dataset itself:

1. The number of dimensions in the Principle Components Analysis of the images of a particular class that explain 90% of the variance.
2. Participation ratio as described in Gao et al. (2017)

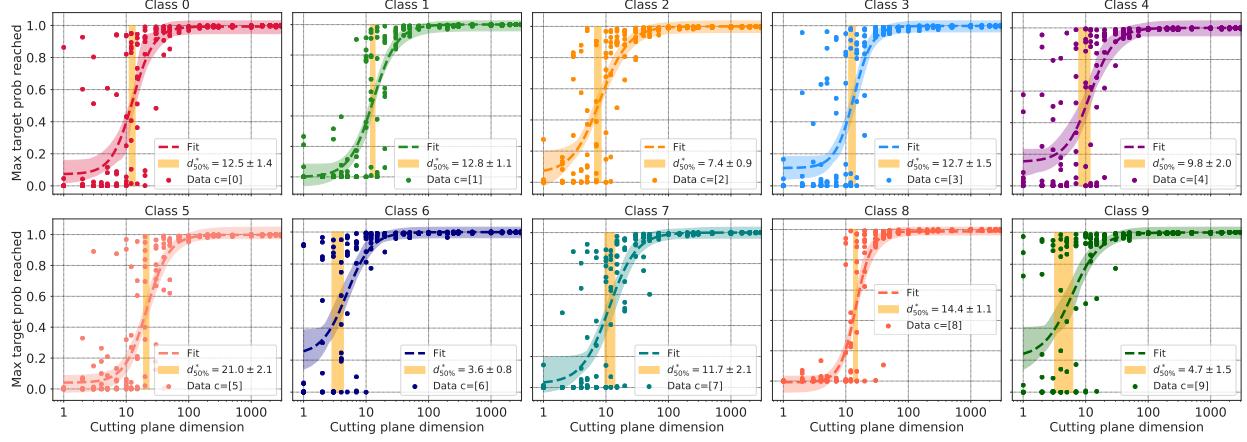


Figure 15: Maximum probability of single classes of CIFAR-10 reached on cutting planes of dimension d . The figure shows the dependence of the probability of a single class of CIFAR-10 (y-axes) reached on random cutting hyperplanes of different dimensions (x-axes). The results shown are for a well-trained ($> 76\%$ test accuracy) SimpleCNN on CIFAR-10. Each dimension d is repeated 10 \times with random planes and offsets.

3. The effective dimension as described in [Vershynin \(2018\)](#) and which we use indirectly to estimate the dimension of the learned manifolds as well.

For the individual classes of CIFAR-10, we get $d_{\text{PCA}} = 98 \pm 20$, $d_{\text{participation}} = 260 \pm 30$, and $d_{\text{effective}} = 4.5 \pm 0.5$. All of these estimates are $\ll 3072$ and \ll the measured dimension of the learned class manifolds.

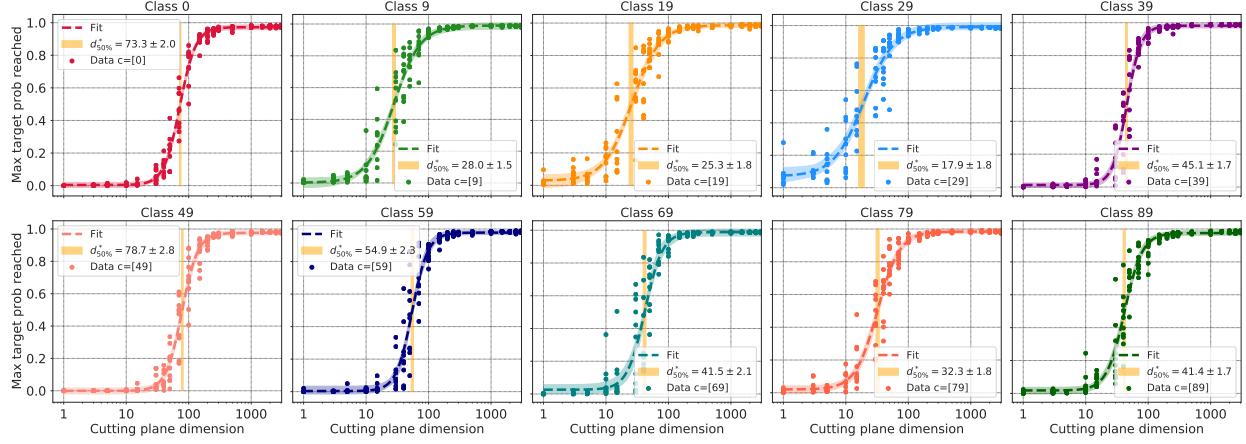


Figure 16: Maximum probability of selected single classes of CIFAR-100 reached on cutting planes of dimension d . The figure shows the dependence of the probability of a single class of CIFAR-100 (y-axes) reached on random cutting hyperplanes of different dimensions (x-axes). The results shown are for a well-trained ($> 67\%$ test accuracy) ResNet20v1 on CIFAR-100. Each dimension d is repeated 10 \times with random planes and offsets.

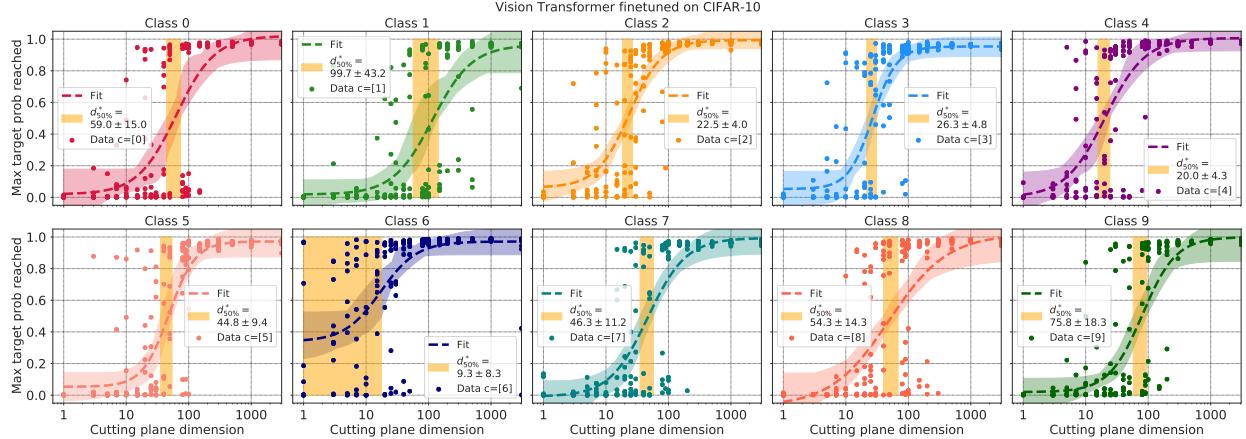


Figure 17: Maximum probability of all classes of CIFAR-10 reached on cutting planes of dimension d . The figure shows the dependence of the probability of a single class of CIFAR-10 (y-axes) reached on random cutting hyperplanes of different dimensions (x-axes). The results shown are for a well-trained Vision Transformer pre-trained on ImageNet and finetuned to CIFAR-10 to test accuracy $> 97\%$. Each dimension d is repeated 10 \times with random planes and offsets.

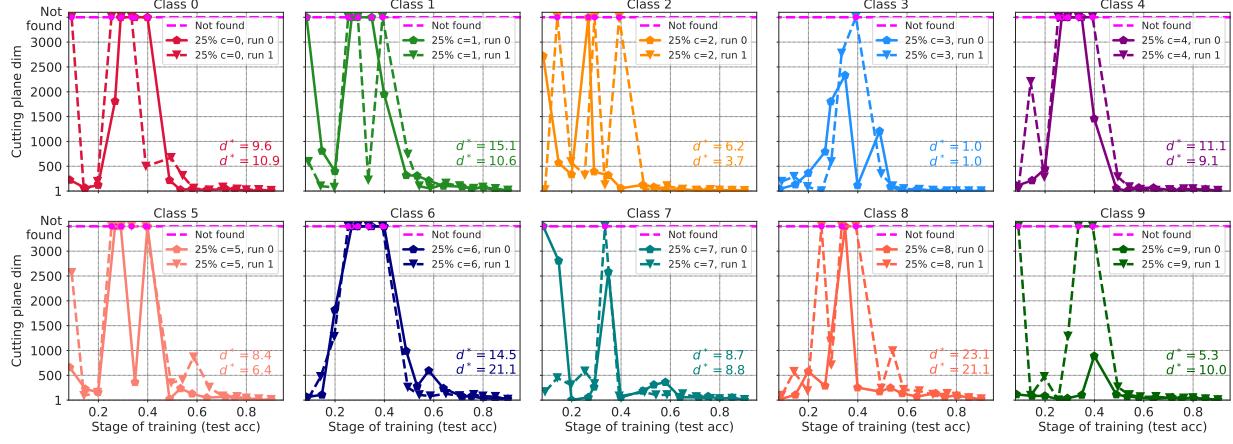


Figure 18: The cutting plane dimension needed to reach 25% probability for the 10 classes of CIFAR-10 as a function of training stage for a ResNet20v1, averaged over two initializations and runs.

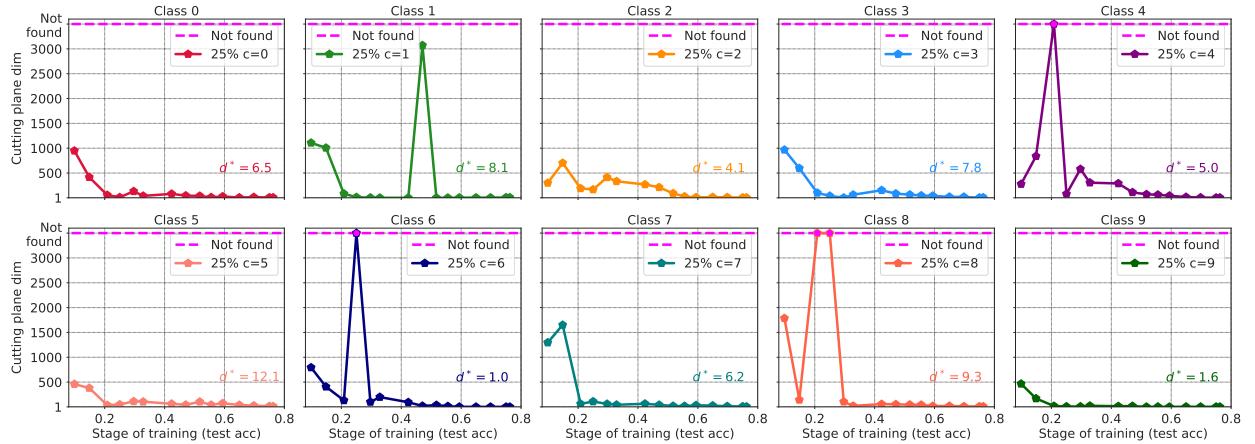


Figure 19: The cutting plane dimension needed to reach 25% probability for the 10 classes of CIFAR-10 as a function of training stage for a SimpleCNN.

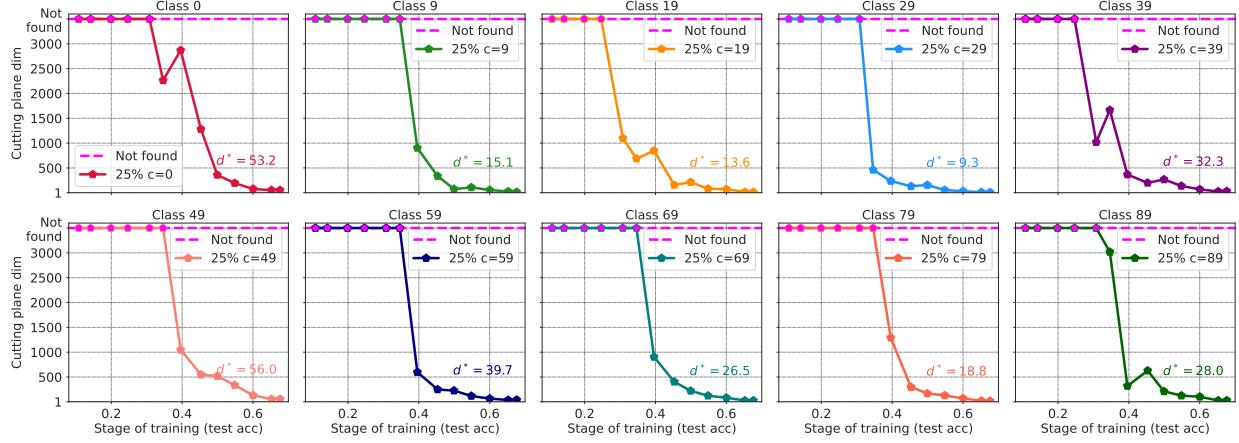


Figure 20: The cutting plane dimension needed to reach 25% probability for 10 randomly selected classes of CIFAR-100 as a function of training stage for a fully trained ResNet20v1.

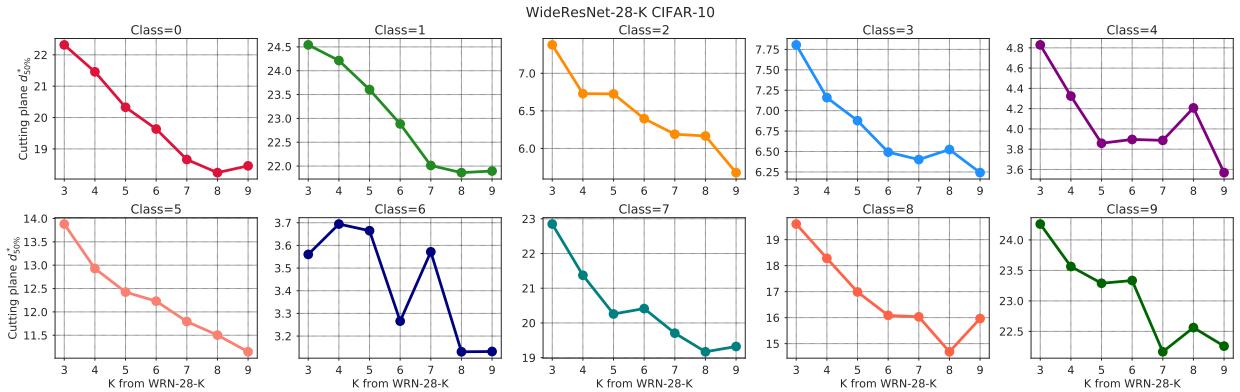


Figure 21: The effect of network width on the dimension of cutting plane necessary to reach a particular probability. The panels shows results for individual classes of CIFAR-10 for WRN-28-K for different values of the width parameter K . $d^*_{50\%}$, the dimension of a cutting plane need to reach the class manifolds, goes down with with K , meaning that the class manifold dimension goes up as $3072 - d^*_{50\%}$. The accuracy and dimension averaged over all 10 classes are shown in Figure 11.

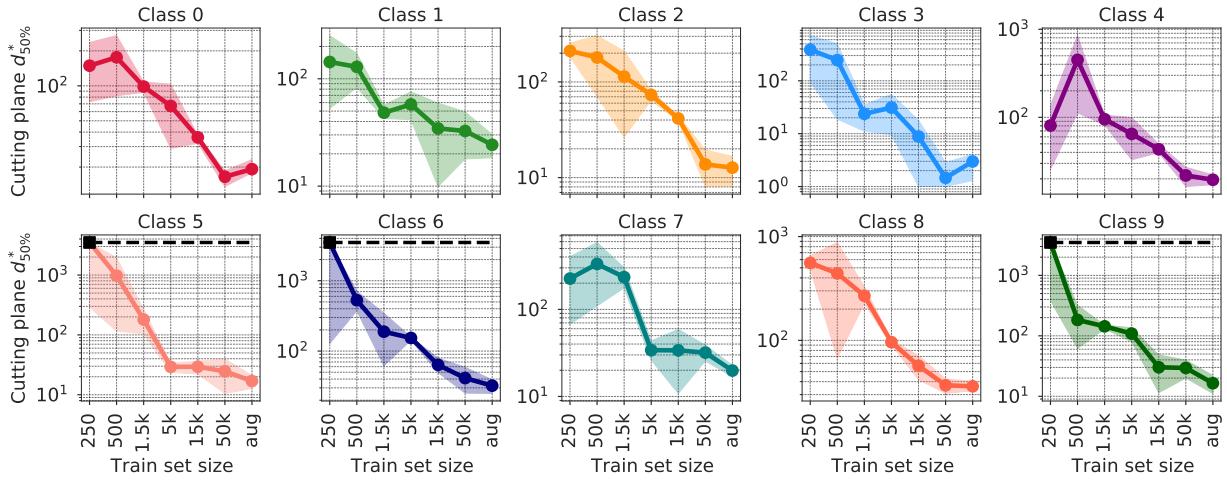


Figure 22: Comparison of the cutting plane dimension needed to get 50% of the target class for ResNets trained to 100% training accuracy on subsets of the training set of CIFAR-10 (mean and standard deviation of 2 networks shown). The bigger the training set, the smaller the $d_{50\%}^*$, therefore the higher the class manifold dimension. The trend continues with the addition of data augmentation (aug), and takes the manifolds from low-D ($\ll D$) for small sets, to high-D ($\approx D$) for large set + augmentation.