NEURAL NETWORK DECISION BOUNDARY VOLUME AND ITS RELATIONSHIP TO GENERALISATION

J. BRODZKI, M. BURFITT AND P. DŁOTKO

Abstract. By combining a Monte Carlo method with procedures to generate adversarial examples for artificial neural networks, we introduce a technique to measure the local volume of the decision boundary of a deep neural network. Our aim is to demonstrate that the complexity of a neural network can be measured through the complexity of the geometry its decision boundary. By targeting measurements of the decision boundary in regions of the input space that correspond to its local and global geometry we are able to give evidence that over variations in hyperparameters optimal network generalisation is achieved at critical points in the geometric structure of the network decision boundaries.

We would expect better generalisation to correspond to a simpler geometric and topological structure in a networks decision boundary, however we reveal that this is not always the case and that the geometric criteria corresponding to better generalisation can vary considerably between data sets and even optimisation procedure.

In the course of the paper, we also explore connections between our methods and a classical tube formulas and examine the consequences of high dimensional geometry on our measurements.

1. Introduction

Deep neural networks have show increasing success over the past several decades with diverse applications application including computer vision [45], natural language processing [14], protein folding [72] and many more.

Despite their success, it is not fully understood why neural networks are so susceptible to small adversarial perturbation unrecognisable to humans perception [28] while still being able to generalise so well to unseen test data or data not even contained within the original training distribution [70]. Here we consider the latter question. It is believed that neural networks generalise so effectively by making use of an inductive bias [63], allowing them to regularise automatically, preferring to converge to a simpler functions regardless of the number of parameters in the network architecture. Despite this apparent intrinsic mechanism, networks have been observed to classify training images assigned random labels without error [86], even if the training images themselves are randomly generated and even in the presence of other explicit regularisation. In these cases there are no underlying pattern in the data which the network could identify and hence no underlying data structures on which generalisation can be achieved with inductive bias. Inductive bias therefore cannot alone explain the unexpectedly small gaps between training and test data achieved with neural networks more generally.

In order to identify good network architectures and hyperparameter choices, generalisation and capacity measures have been proposed. However it has been shown empirically [41] that many generalisation measures work well for tuning specific hyperparameters but generally not for others. In fact for some complexity measures show spurious correlations that do not reflect more causal insights about how generalization can arise and the authors recommend a more rigour approach be taken.

This paper focuses on neural networks trained for classification tasks by developing a geometric complexity measures based on quantifying the volume of a small neighbourhood of the networks decision boundary around chosen points, typically taken from a training set. Through our methods we reveal correlations in the behaviours of decision boundary geometry and the generalisation

performance of the network function. However, the geometric attributes affecting generalisation vary between hyperparameter types, the training data and even the chosen optimisation procedure demonstrating a more complex relationship than might be expected.

It has been shown [51] that the final layer of a deep neural network usually converges to that of a linear support vector machine, for which the decision boundary is a co-dimension one hyperplane. The structure of the decision boundary on the entire network studied here is much less well understood and holds significantly more information on the effectiveness of the network function. With the aim of uncovering structures that explain the generalisation properties of a deep neural network, we approach this problem by measuring directly the geometry of the decision boundary within its input space.

In a classification problem, the decision boundary of a neural network is a finite union of codimension-1 sub-manifolds partitioning the networks decision regions, making it an objects accessible to study from a geometric perspective. Topological arguments have previously be used to study theoretically the restrictions on a network function imposed by a bounded height [42]. A central tool for extracting geometric and topological structure in data is topological data analysis, a central method of which is persistent homology [8, 18]. In recent years geometric and topological properties have been applied directly to enhance and understand neural networks: feature extraction as inputs to neural networks and machine learning [47, 66], adapting network architecture [56] and incorporating prior information and regularisation [13, 12, 22, 77]. In particular studying network weights or activation spaces of networks trained for classification tasks has been a common approach to assessing the geometric structure of the learned network function in order to asses properties including adversarial robustness, interpretability and generalisation [71, 23, 68, 9, 61, 78, 69]. The alternative more intuitive approach we propose here is to study instead the geometry of the decision boundary directly.

A formula for the volume of the tubular neighbourhoods of a manifold embedded in a Euclidean space was first fully formulated by Weyl [80] in 1939, in term of invariants dependent of the curvature of the manifold. Tube formulas has since inspired developments in differential geometry [34] and found applications in statistics and data analysis [46, 54, 50]. The initial motivation for our work is that a tubular neighbourhood is a computationally feasible quantity in high dimensions that for a small enough neighbourhood is directly proportional to the volume of the decision boundary itself, therefore should captures information on the generalisation performance of a neural network, which we demonstrate in Section 3 for low dimensional examples using conventional geometric methods.

Our new procedure which we introduce in Section 6 of this paper provides effectively in a high dimensional setting an estimate of a lower bound of the ε-neighbourhood volume by combining a Monte Carlo method with the construction of adversarial examples, to determine the distance between the network decision boundary and points within the Monte Carlo sample. In particular we are able to make use of computationally efficiently adversarial attack method without compromising the accuracy as our measurements due to the fact that we only require an accurate distances when in far closer proximity to the decision boundary than usually used when forming perturbed adversarial inputs in network security applications. The reasons why convectional methods fail to provide measurements in high dimensional input spaces are detailed in Section 4, Mathematical work on tube formulas and their relationship with our work is discussed in Section 5.

With the aim of revealing the complexity of the decision boundary at different scales, we propose in Section 6.2 three subspace of the input space of a network on which to apply our procedure. The first subspace is the the space of all possible input vectors, the second a neighbourhood of the training data and the third a neighbourhood of points sampled near the decision boundary between training label classes. When combined these measurements give a reading of the network functions performance when compared to the same measures of other similarly trained network functions. Moreover throughout Section 6 we justify using arguments from high dimensional geometry why these measurements should provides good results.

Finally we present in Section 7 an experimental validation of our neighbourhood boundary volume measurements by training fully connected and convolutional neural networks under changes of dropout hyperparameter settings on the MNIST, Fashion MNIST and CIFAR-10 data sets. This allows us to demonstrate the relationship between neighbourhood boundary volume and network function generalisation. In particular we observe that network functions with better generalisation often occur at a critical point in the behaviour of one of our three boundary volumes measurement. However in general the relationship between measurements varies between data sets and even training algorithms, suggesting the relationship is not as simple as we might first expect.

1.1. **Related work.** Many attempts have been made to theoretically derive generalisation bounds [64, 5, 60, 62, 79, 26, 55], however these bound are usually far form tight motivating improved bounds for practical use [16, 3, 40, 49]. Other proposed generalisation measures include tracking the stability of the optimization algorithm [37, 81], measuring the magnitude of the gradient noise [10, 74] and empirically based generalisation measures include encouraging the network to converge to a shallower local minimum of the loss function [73] and an information geometry inspired approach using a Fisher-Rao norm capacity measure [52].

Generalisation with respect to complexity of a decision boundary is considered in [4], where a complexity measure based on approximating the number of hyperplanes used to separate two classes of data is tested on simple low dimensional data distributions. The results show that for a number of machine learning algorithms including simple neural networks, that as the complexity of the decision boundary increases the test error increases alongside reading from a number of other standard error estimation methods.

Three of the most similar studies to our work are [36, 24, 68]. Ramamurthy, Varshney and Mody [68] define a labeled Čech complex which they apply to estimate the topology of a decision boundary of a neural network or data set between two label classes using persistent homology. The persistent homology is computed directly form the data point and corresponding labels, either the true labels or classified labels in each case respectively and a set of persistence statistics is obtained. This enables the authors to make a comparison of the shape of the boundary of the network that and the type of complexity of the data in order to attempt to pick better generalising networks. In our work we propose to measure decision boundary of a neural network more directly and are therefore able to make measurements at more location and scales than just being restricted to a measure directly between label classes.

Georgiev, Franken and Mukherjee [24] propose to study decision boundaries as a heat diffusion process with Brownian motion, by estimating the probabilities that a randomly moving particle crosses the decision boundary after some number of time steps. The Authors also attempt to relate their method and results to generalisation and decision boundary curvature, however their empirical result focus on comparing methods training for adversarial robustness. Other work [25, 59, 58, 19] have also suggested important connections exist between the geometry of the decision boundary and adversarial robustness.

Guan and Loew [36] attempted to connect neural network generalisation to the complexity of the decision boundary by measuring the complexity of decision boundaries between sets of data labels. They achieve this by obtaining a sample of points close to the decision boundary lying linearly between training points and employ an entropy measurement on the eigenvalues of the covariance matrix of the boundary points within local patches of the boundary. The authors use their method to provide evidence that better generalising neural networks network trained with dropout regularisation have simpler decision boundaries than those trained without dropout regularisation, though they train only networks in the two class classification setting.

Information about the shape of decision boundaries and structure of decision regions of neural networks has also been studied empirically with a verity of methods [84, 43, 20], in all cases some form of adversarial attack method is made use of in order to study the properties of the decision boundaries. A study of average distance to the decision boundary during taring was given in [57].

A measurement called boundary thickness that studies the average distance between level sets of two label classes of a network function using Monte Carlo methods is introduced in [83] and considered in relation to adversarial robustness. Also relating the geometry of the and decision regions, Cao and Gong [7] measured using a Monte Carlo method the percentage decision region volumes for each class around benign and adversarial examples in order to distinguish between the them. This work was built on that of He, Li and Song [39] who proposed to compare examples by considering their proximity to the the boundary in a large number of orthogonal direction, both approaches provide evidence that a characterisation of the geometric structure of the decision boundary around examples rather than just the minimal distances between the point and the boundary is important. In our present work we choose to take a more global perspective studying the function generally rather than in the region of single examples. The geometry and topology neural network decision boundaries have also been studied form a theoretical perspective with a view toward generalisation in [1, 53].

2. Background

We set out here the notation used in the rest of this paper and introduce the necessary background on adversarial attacks relevant to developing our neighbourhood boundary volume methods. Adversarial attacks are important to our work as we use them to obtain an upper bound on the distance from a point to the network decision boundary.

2.1. **Notation.** Throughout this work we assume that X is a finite set of n-dimensional training data points in $[0,1]^n$ with corresponding discreet labels Y in the standard basis of $\mathbb{R}^m$, where m is number of classification classes. Let $N\colon \mathbb{R}^n \to \mathbb{R}^m$ be a feed forward neural network function with n inputs and m outputs. Let $d\colon \mathbb{R}^n \times \mathbb{R}^n \to \mathbb{R}$ be a metric on the input space of network N. In particular, a feed froward neural network on k layers of respective sizes $n = l_0, l_1, \ldots, l_k = m$ is defined inductively as the composition of functions

$$\phi_i(W_i x + b_i)\colon \mathbb{R}^{l_{i-1}} \to \mathbb{R}^{l_i} \tag{1}$$

where W_i is an l_{i-1} by l_i matrix and the vector $b_i \in \mathbb{R}^{l_i}$ captures the network parameters for $i = 1, \ldots, k$, with all trainable entries. A fully connected network is defined by having no restrictions on the entries of the W_i and b_i during training. Convolutional network layers are defined on data given as a cubical array such as an image and apply the same trainable operations to each local cubical patch in order to obtain entries of a next cubical layer [27, § 9]. Once a way to flatten the cubical arrays is chosen such an operation can be formatted in the form of equation 1, with the additional possible inclusion of a pooling operation after the application of the activation function.

The network N is trained on data set (X, Y) with respect to a cost function obtained as the mean value of loss functions $\mathcal{L}(N, x, y)\colon \mathbb{R}^n \times \mathbb{R}^m \to \mathbb{R}$ over $x \in X$. There are standard choices of the loss function in the literature, we use here the cross-entropy loss function. For each $x \in \mathbb{R}^n$, denote by $N_j(x) \in \mathbb{R}$ the output of N in the coordinate $j = 1, \ldots m$. Then the prediction of the network function

$$p(N(x))\colon \mathbb{R}^m \to \mathcal{P}(\{1, \ldots, m\})$$

at $x \in \mathbb{R}^n$ is given by the set of coordinates j that maximises $N_j(x)$. Note that if $p(N(x)) = \{i\}$ we just write $p(N(x)) = i$ for short. The decision boundary $B(N) \subseteq \mathbb{R}^n$ of N is a set of point between decision regions of labels classified by N, that is

$$\begin{aligned} B(N) = \{x \in \mathbb{R}^n \mid \text{for every } \varepsilon > 0 \text{ there are } y, z \in \mathbb{R}^n \text{ with} \\ d(x,y) < \varepsilon, d(x,z) < \varepsilon \text{ and } p(N(y)) \neq p(N(z))\} \end{aligned}$$

The decision boundary of a feed forward neural network for a two class classification problem is a manifold of codimension 1 in $\mathbb{R}^n$ with probability 1. In a multiclass classification problem the the decision boundary boundary is the union of the (non-empty) codimension 1 manifold with boundary sitting between the decision regions of each pair of class labels.

2.2. **Adversarial attack methods.** An *adversarial example* at distance $\varepsilon > 0$ to a point $x \in \mathbb{R}^n$ (often chosen from the training set X) is an $a \in \mathbb{R}^n$ such that

$$p(N(x)) \neq p(N(a)) \text{ and } d(x, a) \leq \varepsilon.$$

Geometrically an adversarial example at x is another point lying nearby x at the other side of a decision boundary. An *adversarial attack method* or *adversarial attack* is a procedure for generating adversarial examples for a given trained network and a point in its domain.

The reason for generating adversarial examples in our work is to determine the distance of a given point X to the network decision boundary. It should be noted that these measurements come with no theoretical guarantees on their error and therefore can only be consider a *upper bounds* of those distances. Moreover there is a trade off in the effectiveness with which an adversarial attack finds an adversarial example against the computation time required to execute the attack and the distance ε over which the attack is effective. As, typically for every point x, we need to generate a large number of adversarial examples, our methods should generally be applied using a computationally efficient attack which would results in a less reliable estimate of the distance to the boundary. This is compensated for by the fact that we require our adversarial examples to be generated within a smaller distance ε than would generally be considered for evaluating network robustness for security reasons, for an example with networks tainted on MNIST comparing the FGSM and a strong projected gradient decent adversarial attack over varying ε values see [67]. In the experimental part of this work we apply our methods using the fast gradient sign method adversarial attack, though in practice it will be useful to take measurements across a range of ε values coming from an adapted version of this attack detailed in Section 6.3. This means that ε value may be obtained over a good range of outputs and a specific value may be selected latter.

In this work we consider only white box adversarial attack methods, where adversarial examples are generated with full knowledge of the network function N. In addition we always use the metric d induced by the l_∞-norm when generating adversarial examples. We now detail the fast gradient sign method (FGSM) adversarial attack, for more details on other adversarial attack methods see for example [82, 85].

The fast gradient sign method [28], is obtained using the gradient of the loss function $\mathcal{L}$ locally at a point $x \in \mathbb{R}^n$ with a label y by

$$\text{FGSM}_\varepsilon(N, x, y) = x + \varepsilon \cdot \text{sign}(\nabla_x \mathcal{L}(N, x, y)) \tag{2}$$

where the function sign rounds each non-zero coordinate to ± 1 and ∇_x is the derivative with respect to x. Note that in contrast to the stochastic gradient descent method used to train the neural network, here we aim to maximise the value of the local loss function in order to obtain an adversarial example with a different label to the class of the original point x.

Given a label $i \neq p(N(x))$, we can also perform a FGSM attack targeted at obtaining an adversarial example with label i by subtracting rather than adding the local loss function about x with respect to label i from x. This adversarial attack method is given by

$$\text{FGSM}^i_\varepsilon(N, x) = x - \varepsilon \cdot \text{sign}(\nabla_x \mathcal{L}(N, x, e_i)) \tag{3}$$

where e_i is the standard basis vector with 1 in position i and 0 entries otherwise. See Figure 1 for an illustration of the FGSM and targeted FGSM procedures.

For the purposes of estimating ε-neighbourhood boundary volumes over a range of ε values, we require a more sophisticated adversarial attack that aims to obtain an adversarial examples close to the initial point and returns this distance. By adapting the FGSM we provide such a method in Section 6.3 and this procedure is applied in our experimental work.

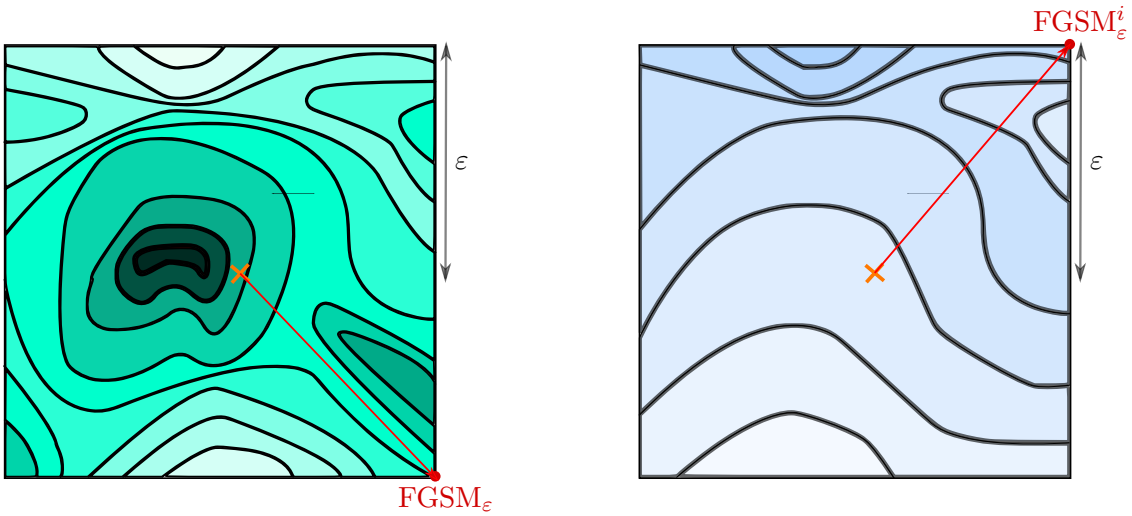

FIGURE 1. Contours of the local loss function within ε of a data point x in orange, where a darker colour indicates a lower value of the loss function. To the left, the FGSM adversarial attack on x is indicated by the red arrow. To the right the targeted FGSM adversarial attack on x is indicated by the red arrow targeted towards target label i. Both adversarial attack methods are used to find examples within ε of x. Note that the direction of arrows must be diagonal inside the ε-box about x due to the application of the sign function. The FGSM aims to maximise loss while the targeted FGSM aims minimise its loss.

3. The role of depth and decision boundary geometry in generalisation of 2-dimensional NN

In order to demonstrate the use of geometric and topological measurements in analysing the generalisation performance of deep neural networks, we present a two class classification experiment in the two dimensional setting. Our aim is to demonstrate that generalisation of a neural network can be measured through complexity in the geometry of its decision boundary. We show that different geometric measures at varying scales reveal distinct aspects of that complexity. In particular, successful generalisation minimises a certain geometric measure of network function complexity.

Though not used to obtain results later in the paper, in this section we make use of methods from topological data analysis including Betti number and persistent homology to study the geometry and topology of neural network decision boundaries. For an elementary introduction to topological data analysis see for example [17].

Here our neural network learns a continuous function $N\colon \mathbb{R}^2 \to [0,1]$ where binary classification labels are determined by rounding a single output to the nearest integer. Our activation functions are $\phi_i = \tanh$ applied component-wise within each layer except the final layer, where the logistic function is applied in the final step. Before training, weights are randomly initialised and distributed according to the He normal conditions [38] with training using the Adam optimisation procedure [44] and the binary cross-entropy cost function

$$\mathcal{L}(N) = -\frac{1}{k}\sum_{i=1}^{k} y_i \log(N(x_i)) + (1-y_i)\log(1-N(x_i)).$$

The network partition of $\mathbb{R}^2$ is then given in terms of its preimage by

$$\begin{aligned} f^{-1}\big([0,0.5)\big) &\quad \text{the decision region of class 0,} \\ f^{-1}\big((0.5,1]\big) &\quad \text{the decision region of class 1 and} \\ f^{-1}(0.5) &\quad \text{the decision boundary.} \end{aligned}$$

Though not conclusively proven, it is widely acknowledged that deeper network architectures show greater functional expressiveness than shallow architectures even when the number of network parameters remains constant. Moreover the optimal network functions in terms of generalisation to the test data should occur at an intermediate architecture within the spectrum between shallower and deeper networks with a fixed number of parameters. We would expect a more expressive network to have more complicated geometric and topological structure in its partition of the input space while a network with better generalisation should be simpler. The relationship between these properties of the network function should be measurable in the structure of the network decision boundary and decision regions.

Here we train several fully connected networks of differing architectures and similar numbers of units on a noisy spiral data set. The 800 data points of either class are randomly sampled by

$$\begin{aligned} \theta &= 4\pi\sqrt{\text{initial}} \\ r &= (-1)^{\text{class}}(2\theta+\pi) \\ \text{point} &= (r\cos(\theta), r\sin(\theta)) + \text{noise} \end{aligned}$$

for class $= 0$ or 1, and random numbers $initial \in [0,2]$ being uniformly sampled. The variable noise is bidimensionaly Gaussian distributed with mean 0 and variance 4 in all directions. Furthermore, the points are then scaled to sit inside $[0,1]^2$. A 1200 point test set for validating the performance of the network function is also sampled from the same distribution.

The first geometric measurement we make on the network functions is the length of the network decision boundary in the unit square $[0,1]^2$. This is obtained by counting the number of disagreements in class label between horizontally and vertically neighbouring points of a 100 by 100 grid sample of $[0,1]^2$ thought of as forming a 2-dimensional binary image on $[0,1]^2$. Explicitly, the grid points are taken as the product of 100 equally spaced points in $[0,1]$ with the smallest point at 0 and the largest at 1. By the Intermediate Value Theorem, a disagreement in network classification between two points implies that the boundary between the two classes is located between them. This measurement is a global summary of the whole network function within its domain contained in $[0,1]^2$.

The second measurement is the sum of the 0^{th} Betti numbers of the homology of the decision regions corresponding to the class 0 and 1, respectively. The measurement is obtained by computing the homology of those points classified by the network as class 0 and 1 separately within the 100 by 100 grid sample regarded as a cubical complex inside $[0,1]^2$. This captures the number of connected components in the two decision regions. However, this measurement has the disadvantage of treating all components equally regardless of size and can be significantly unstable as a consequence.

To overcome these problems, we also consider measurements using persistent homology. We again consider a 100 by 100 grid sample and treat each point in this as the centre of a cube in a 2-dimensional cubical complex C covering $[0,1]^2$. The cubical complex is filtered by the function $F\colon C \to [0,1]$, where $F(x)$ is the minimal l_∞ distance from x to a grid point adjacent to the boundary, that is a point for which the network label disagrees with that of any of its 4 or less adjacent grid points.

The 0-dimensional persistent homology of this filtration corresponds to connected components in the boundary with the death time of a feature describing the minimal distances between boundary components. The 1-dimensional persistence captures disconnected components in the decision region with the significance of these features corresponding to their diameter. In addition, the measurement also captures curvature related information by measuring segments of a decision region separated by

bottlenecks in the decision boundary, that is regions where the decision boundary is narrow. The significance of these features corresponds to the difference of their narrowest diameter and the width of the bottlenecks connecting them. Intuitively this persistence measure provides a summary of more local features of the complexity of the network decision boundary. A demonstration of the persistent homology of the filtration is given in Figure 2.

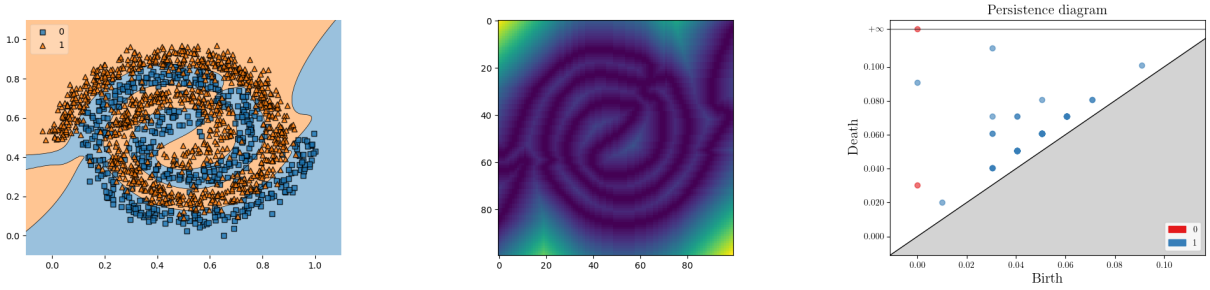

FIGURE 2. The left panel shows the training data and the decision regions of a fully connected network trained on the data. In the centre is a plot of the filtration from the boundary on a 100 by 100 grid sample, where blue represents 0 and yellow the highest value in the filtration. To the right is the persistence diagram containing 0 and 1 dimensional features corresponding to the filtration.

In each dimension we summarise the persistent homology features by taking the norm of finite pertinence features, that is in each dimension the sum of the squared distance of each point in the persistence diagram from the leading diagonal. We call this the *persistence norm* of the diagram. In particulate, the squared distance form the diagonal is used to emphasise the presence of larger more significant features over the smaller ones.

The main observable difference in the network outputs given in Figure 3, are that shallow networks learned functions with jagged decision boundaries finding it slightly harder to fit all the test data, while the deep networks found smoother decision boundaries but over-fit the data by forming more disconnected components in their decision regions.

These observations are verified in Figure 4, where the values of test accuracy, boundary length, Betti numbers and persistence norms are tracked for networks during the training of ten randomly initialised network functions. The networks that perform best with respect to test accuracy are those in the middle of the transition between width and depth. The boundary length shows that the deeper networks achieved longer boundaries after training. The deeper networks also have higher Betti numbers throughout training. The increasing number of disconnected components present in shallower networks is again evident from the size of the norm of the 0-dimensional persistence, this plot additionally demonstrating that once trained the disconnected components are further apart for deep networks. Finally in the plot of the the norm of the 1-dimensional persistence we see that the less well performing networks in terms of test accuracy converge to larger values than those achieving higher test accuracy. This shows that the larger disconnected components for deep networks and segments between bottlenecks for shallower networks both result in higher measurements confirming the explanation for their worse performance due to having more complex boundaries at a local scale.

We conclude that in this case, the measurements of boundary length show that deeper networks learn more complicated decision boundaries overall while the functions that minimise the norm of the 1-dimensional persistence diagram correspond to better generalising network functions with simpler local geometry, seen by their revers ordering when compared to the test accuracies in figure 4. This minimum appears to result as the optimal position in a trade-off between the increased global geometric complexity of deeper networks and the increased local geometric complexity of shallower networks. Hence, the combination of measurements of the topology of the decision regions together

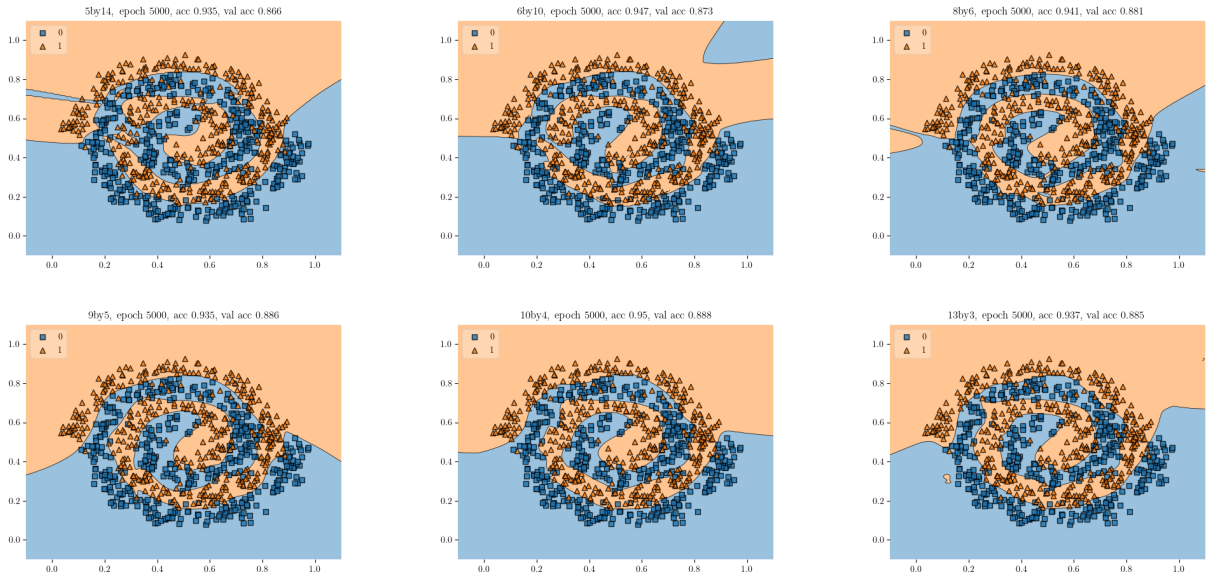

FIGURE 3. Examples of fully-connected neural networks trained on a 800 point spiral data set, with different architectures ranging between 14 hidden layers with 5 hidden units to 3 hidden layers with 13 hidden units. Each network was trained for 5000 epochs. For the deeper networks we see more disconnected components in the decision boundary, while for the shallower networks the boundary is rougher and more jagged.

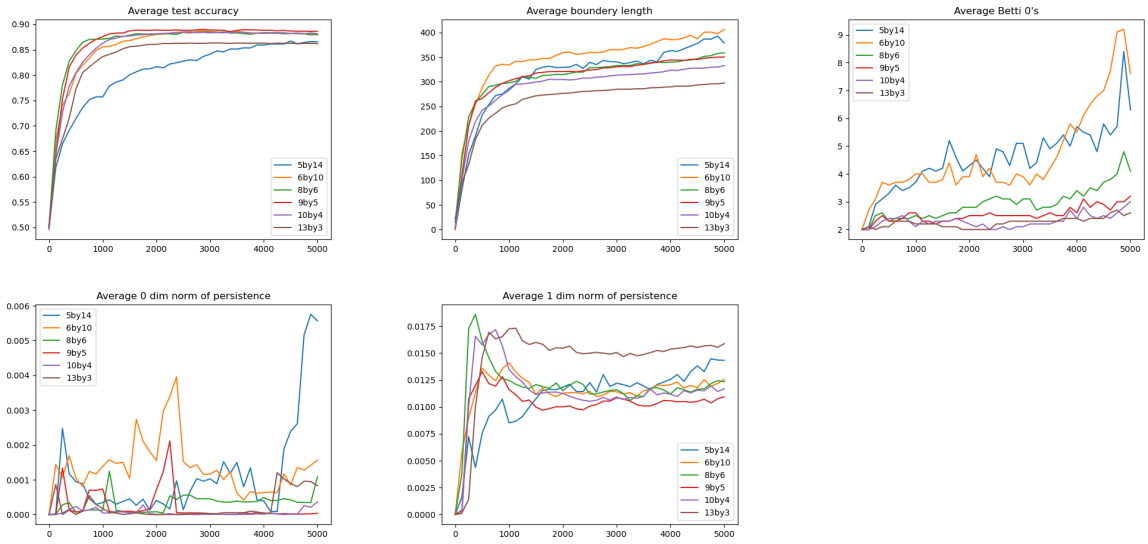

FIGURE 4. Training statistics of fully connected networks trained on a spiral data set each with a similar numbers of hidden weights averaged over 10 training initialisations. The deeper networks generally have longer decision boundaries and higher decision region Betti numbers. The average 1-dimensional normed persistence is after training minimised around the network architectures achieving best test accuracy.

with the length of the decision boundaries provide insight into how the network function has learned to fit the data. We therefore aim to construct methods to translate such geometric measurements

into tools that can be applied to networks trained on higher dimensional data commonly used in practice with deep learning.

4. Obstacles to quantifying the geometry of decision boundaries in high dimensions

Our observations in the previous section demonstrate the useful information available when measuring the geometry of a neural network decision boundaries lying in a low dimensional space. Before proposing a procedure to perform measurements for network with a higher dimensional input space, we first discuss some of the mathematical differences when dealing with geometry in a high dimensional euclidean spaces and how they can be overcome.

The curse of dimensionality is often used to refer to a number of mathematical phenomena that make studying data with a large number of parameters difficult [29]. Perhaps the easiest example of the curse of dimensionality is for grid samples such as those we considered in the previous section. In this case a 100 by 100 grid sample in 2 dimensions requires 10000 points to be sampled. For an arbitrary dimension d to achieve the same precision of coverage we would require 100^d points to be sampled, a quantity that quickly becomes impractical as d grows.

Many properties presenting a curse of dimensionality can be expressed in terms of a concentration of measure around certain set as dimensions increases. An example of this is given in [30], where it is shown that the number N of randomly selected vectors in $\mathbb{R}^d$ that are almost certainly orthogonal grows exponentially with d. More precisely we have the following proposition. Given $\varepsilon > 0$, say two vectors are ε-orthogonal if the dot product of their unit vectors in less than ε.

Proposition 4.1 ([30]). *For $\varepsilon, \theta > 0$, a set of N uniform random vectors chosen in the unit ball in $\mathbb{R}^d$ are pairwise ε-orthogonal with probability greater than θ when*

$$N \leq e^{\frac{\varepsilon^2 d}{4}} \sqrt{\log(1/\theta)}.$$

The above bound can be derived from the property that as d increases the surface volume of a d-sphere is concentrated in a smaller neighbourhood of its equator. Such concentration of measure properties in high dimensional spaces are studied and applied throughout the literature [48, 75, 31, 32, 76], including the more general waist concentration results of Gromov [35].

A manifestation of the curse of dimensionality is perhaps most relevant to measuring geometry of network decision boundaries is the concentration of distance in high dimensional l_p-spaces. Denote by $\| - \|_p$ the p-norm on $\mathbb{R}^d$. For independent randomly sampled points $x_1, \ldots, x_m$ drawn from $[0,1]^d$ we have that

$$\begin{aligned} \text{if} \quad & \lim_{d\to\infty} var\left(\frac{\|x_i\|_p}{\mathbb{E}\|x_i\|_p}\right) = 0 \\ \text{then} \quad & \lim_{d\to\infty} \frac{\max_{i,j} \|x_i^d - x_j^d\|_p - \min_{i,j} \|x_i^d - x_j^d\|_p}{\min_{i,j} \|x_i^d - x_j^d\|_p} = 0 \end{aligned} \tag{4}$$

where $i, j = 1, \ldots, m$, which is a special case of a theorem of Beyer, Goldstein, Amakrishnan and Shaft [6]. The theorem can be interpreted by saying that in high dimensions most points have a similar pairwise distance. It is also clear that if we sample points uniformly in $[0,1]^d$, then as d increases we also increase the number of coordinates and so increase the expected l_p distance between any pair of randomly sampled points. We can see the effect of equation 4 on real data sets. For example, the MNIST and CIFAR-10 training set each contain 50000 images of dimensions 784 and 3072 respectively and the value of

$$\frac{\max_{0\leq i<j\leq 50000} \|x_i - x_j\|_2 - \min_{0\leq i<j\leq 50000} \|x_i - x_j\|_2}{\min_{0\leq i<j\leq 50000} \|x_i - x_j\|_2}$$

for x_i, x_j in the training sets are approximately 19.5360 and 10.0561 for MNIST and CIFAR-10 respectively.

In the case of a Vietoris-Rips filtration of the simplicial complex built on a collection of points lying in a high dimensional space, a consequence of the concentration of distance is that any geometric structure in the data will be obscured by the relatively large distances between points in the sample. So to obtain meaningful measurements of the persistent homology of such an object we are required again to sample an impractically large number of points. Figures 5 and 6 demonstrate that given 1000 points uniformly sampled in dimension d from two parallel planes lying on faces of $[0,1]^d$ or a unit cylinder embedded in $[0,1]^d$, the sample is not sufficient to recover the homology of these sets by dimension $d = 15$. In the parallel planes example we expect to be able to identify a single longest living non-infinite 0-cycle corresponding to the connected component of one of the two planes and in the cylinder case a single longest living 1-cycle corresponding to the cycle in the cylinder. The features corresponding to the connected component and cycle in both cases respectively are hard to distinguish by dimension 12 and completely obscured by noise at dimension 15.

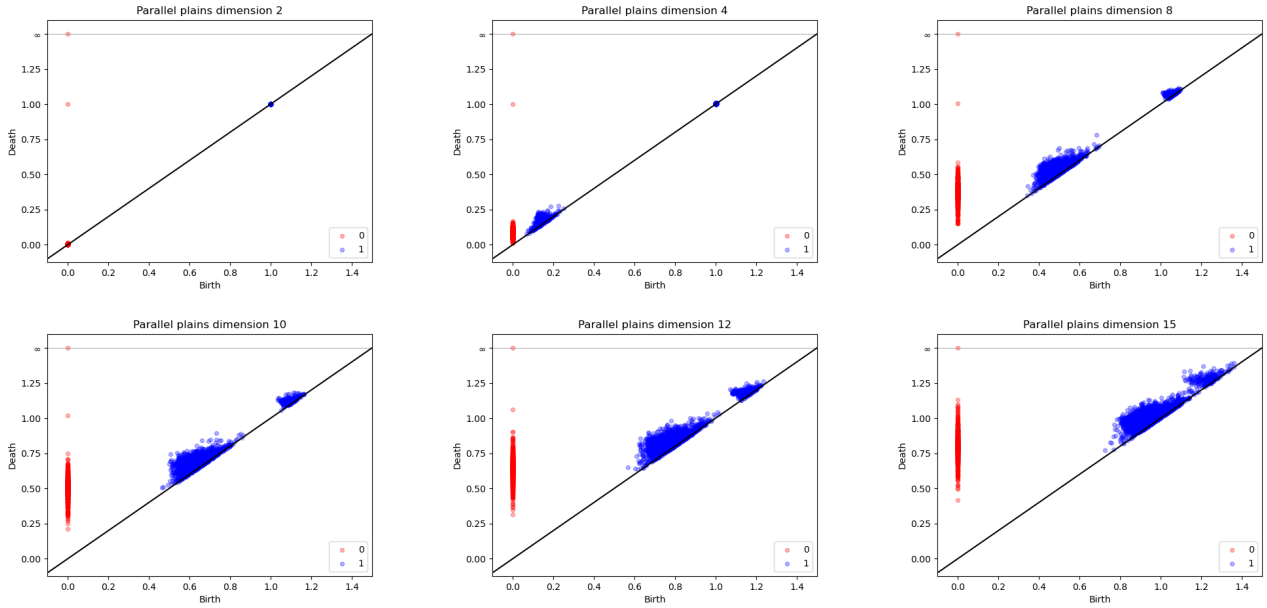

FIGURE 5. Persistence diagrams of a 1000 point sample from an embedding of two parallel planes in $[0,1]^d$ at distance 1 apart, varying dimension d. The presence of the 0-dimensional homology representing the disconnected planes is indistinguishable from the noise by dimension 15.

It is therefore clear that making precise direct measurements of the persistence of decision regions and the volume of the decision boundary similar to the constructions presented in Section 3 becomes impractical in high dimensions. In the boundary length case, this is because network decision boundary are usually locally a co-dimension one hyperspace in higher dimensions, so there will be a combinatorial explosion in the number of required grid points to measure.

The alternative perspective on data from a high-dimensional setting is that while some of our intuitions from low-dimensions no longer apply, the geometry of high-dimensional data is far less variable, as demonstrated by our earlier examples under the name of the curse of dimensionality. This phenomenon is known as the Blessing of Dimensionality [15, 2]. Geometric methods applicable to high-dimensional data analysis might therefore be useful on a wider variety of high dimensional applications providing they are designed to exploit the high dimensional geometry.

One of our main contributions in this paper is to demonstrate a method that is suitable to large deep neural networks, that can explore the shape of network function by describing the size of the boundary within a high dimensional setting. We achieve this by developing a procedure to replicate

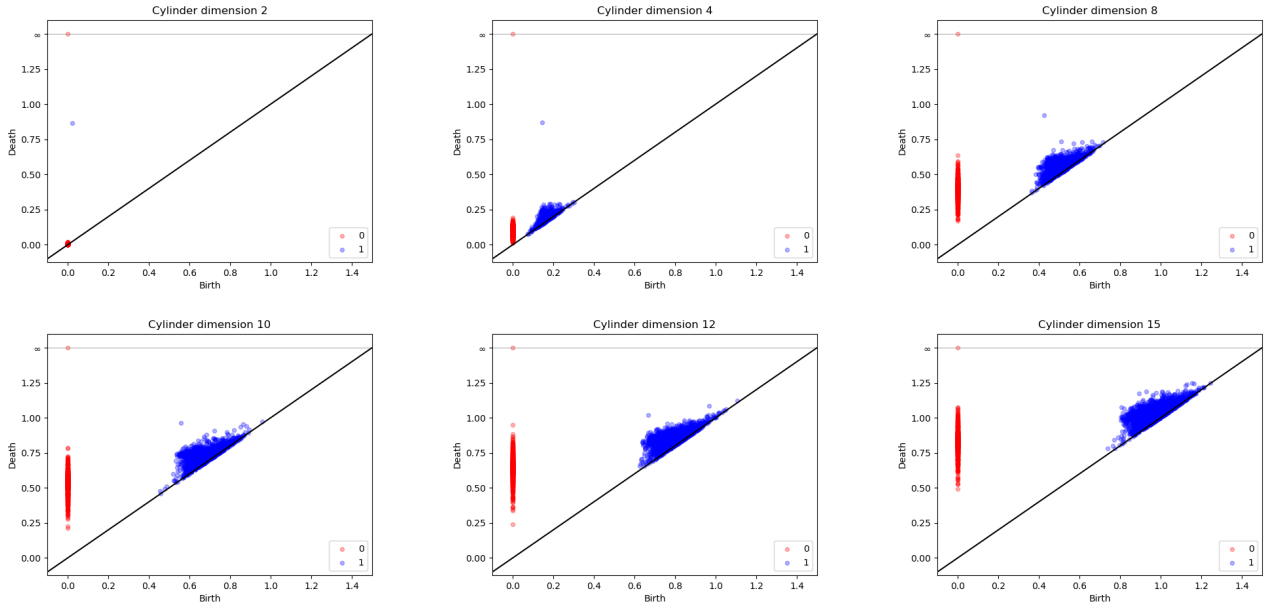

FIGURE 6. Persistence diagrams of a 1000 point sample from an embedding of a cylinder of radius 0.5 in $[0,1]^d$, varying dimension d. The presence of the 1-dimensional homology representing the cycle in the cylinder is indistinguishable from the noise by dimension 15.

the boundary length measurements of Section 3 by considering instead the more accessible volume of a small neighbourhood of the boundary. This measurement is then targeted at particular locations for which it is shown to provide useful information about the effectiveness of the network function.

5. Weyl's tube formula

In this section we discuss the mathematics of tubular neighbourhoods of manifolds and consider how those tubular neighbourhood might affect such measurements when applied to neural network decision boundaries. In particulate we justify that for a small enough neighbourhood volume parameter ε the neighbourhood volume is approximately proportional to the true volume of the decision boundary of the network, which is our main object of interest.

The volume $V_M^{\mathbb{R}^n}$ of an ε-tubular neighbourhood of a q-dimensional manifold M embedded in $\mathbb{R}^n$ (with respect to the l_2 metric) was studied in a classical paper of Weyl [80] in 1939 and has since inspired developments in differential geometry [34]. Weyl's tube formula is a polynomial in the tube radius ε with coefficient containing invariants $k_{2i}(M)$ dependent on the curvature of M. The full expression of Weyl's tube formula is given by

$$V_M^{\mathbb{R}^n}(\varepsilon) = \frac{(\pi\varepsilon^2)^{(n-q)/2}}{((n-q)/2)!} \sum_{i=0}^{[q/2]} \frac{\varepsilon^{2i} k_{2i}(M)}{(n-q+2)(n-q+4)\cdots(n-q+2i)}. \tag{5}$$

In particular, for compact M, the invariant $k_0(M)$ is the volume of the manifold M. When M is closed and even dimensional, the Gauss-Bonnet theorem may be applied to realise $k_{q/2}(M)$ as $(2\pi)^{q/2}$ times the Euler characteristic of M.

In this work we are interested in measuring the ε-neighbourhood volume of the decision boundaries of a deep feed forward neural network. We propose to do this because it is a quantity we are able to measure using Monte Carlo methods even in high dimensions, and it is a useful proxy for the volume of the decision boundary. In a two-class classification setting for a network with smooth activation

functions, equation (5) shows immediately that the volume of an ε-neighbourhood of the decision boundary is approximately proportional to the volume of the decision boundary, providing $\varepsilon > 0$ is small.

More precisely, since equation (5) is a polynomial in ε with lowest degree term being linear in ε, for small ε values $V_M^{\mathbb{R}^n}$ is approximately linear in a scalar multiple of $k_0(M)$, the volume of M. The decision boundary of a feed forward neural network used for classification in a two class classification problem trained with gradient descent is a manifold of codimension 1 in some $\mathbb{R}^n$ with probability almost 1. If the activation functions of the network are smooth such as for tanh activation functions, then the manifold is also smooth. If the activation functions of the network are piecewise linear such as for rectified linear activation functions, then the manifold is a piecewise linear manifold. In this case we would expect network function to be approximating a smooth manifold, however obtaining an explicit tubular formula in this situation turns out to be more a difficult problem than for the smooth case [11, 21].

For a multiclass classification problem the decision boundary consists of the union of manifold boundaries between pairs of classes not contained in another decision region. In this case any intersection between the manifolds will with almost certain probability be a codimension-2 manifold and provided ε is small enough its effect on the neighbourhood boundary volume should generally be negligible in comparison to the total volume of the decision boundary which we justify empirically in Appendix B. In addition we show how to apply our methods to identify the presence of curvature in the decision boundary between pairs of class labels in Appendix C.

6. Methodology

In this section we set out the procedure for measuring the neighbourhood boundary volume of a neural network function. This is carried out in the following steps. First, we select an adversarial attack method; the effectiveness of that method will determine the accuracy of the volume estimate. Secondly, we select regions of the input data space on which to measure the neighbourhood boundary volume; our selection aims to reveal the complexity of the function on different scales.

More precisely measure the neighbourhood boundary volume of a neural network function using a Monte Carlo method combined with the adversarial attack procedure specified at the end of the section, that gives measurements at a range of ε values. We provide three regions on which measurements can be made. We denote the resulting values by **Bvol** a global measurement with respect to possible inputs, **TrainBvol** a global measurement around training data and **LAdvBvol** a local measurement between training label classes. The latter two measurement also depend on an additional δ variable to determine their local search radius, for a sensible choice of which we justify the interpretation of the measures.

6.1. Estimating ε-neighbourhood boundary volume.

Given a region $\mathcal{U} \subseteq \mathbb{R}^n$, $\varepsilon > 0$ and an adversarial attack method A for obtaining adversarial examples at a distance as near as possible to the decision boundary, we propose to obtain an estimate of the volume of the ε-neighbourhood of the decision boundary of network function N in $\mathcal{U}$ through Monte Carlo sampling of $\mathcal{U}$. As a point is determined to be within ε of the decision boundary only when an adversarial example less than a distance of ε away can be generated, our volume estimate is in general a lower bound on the true ε-neighbourhood boundary volume. The quality of this estimate depends on the reliability of the adversarial attack method A and the Monte Carlo error.

By applying the law of large numbers, Monte Carlo methods can be used to estimate almost any quantity that can be expressed probabilistically. In the case of volume, given a large uniformly distributed sample of a subspace of $\mathbb{R}^n$, we can obtain an estimate for the volume of the sub-region $\mathcal{U}$ by counting the proportion of points that fall inside $\mathcal{U}$. More precisely in our case, the ε-neighbourhood boundary volume $\mathrm{Bvol}_\varepsilon(N, U)$ is approximated as the probability

$$\mathrm{Bvol}_\varepsilon(N, \mathcal{U}) = \mathrm{Volume}(\mathcal{U}) \cdot \mathbb{P}\left(d(x, A(x)) \leq \varepsilon \mid x \in \mathcal{U}\right) \tag{6}$$

which may be estimated using a Monte Carlo method by computing the proportion of points satisfying $d(x, A(x)) \leq \varepsilon$ from a large uniform random sample of points in $\mathcal{U}$ with some degree of error depending on the sample size. In practice, and in all our experiments, it is convenient to set $\text{Volume}(\mathcal{U}) = 1$.

As discussed in Section 5, our aim is to use the ε-neighbourhood boundary volume as an alternative to the impractical measurement of the boundary volume alone, and for this we would like to choose ε as small as possible. However the choice of ε must also be large enough to ensure that the Monte Carlo estimation is applied to a substantial enough fraction of $\mathcal{U}$ in order to give a reliable result.

It is not strictly necessary to compute an adversarial example in order to estimate the distance to the boundary. Adversarial examples are suited to studying neural networks and would not necessarily be ideally applicable to other machine learning algorithms. For instance the computation of $d(x, A(x))$ could be replaced with an alternative method such as an upper bound obtained by interval arithmetic on N. Interval arithmetic evaluates a function under study on interval boxes bounded by upper and lower bounds $x, x' \in \mathbb{R}^n$, where an interval is given by

$$[x, x'] = \{z \in \mathbb{R}^n \mid x_i \leq z_i \leq x'_i \text{ for } i = 1, \ldots, n\}.$$

The output of a function on an interval is an interval that contains at least all values of the function on points contained in the input interval. We could measure the boundary volume with interval arithmetic by replacing the condition $d(x, A(x)) \leq \varepsilon$ in equation (6) with an indicator function

$$\mathbb{I}_{x,\varepsilon} = \begin{cases} 1 & \text{if there are } x_1, x_2 \in N([x - \varepsilon, x + \varepsilon]) \text{ such that } p(x_1) \neq p(x_2) \\ 0 & \text{otherwise} \end{cases}$$

where adding and subtracting ε is applied to every component of the vector. However, while the interval bounds can be effectively computed between layers of a feed forward neural network [33, 65], we found them to be too imprecise to use for boundary volume measurements. The error on the interval bound can vary considerably depending on the location of the input interval in $\mathbb{R}^n$, meaning the estimated thickness of the ε-neighbourhood of the boundary would be overestimated more within some regions of the decision boundary than others.

In addition, changes in the network architecture and hyper-parameters such as regularisation cause the scale of the interval error bounds to vary dramatically making comparison between different models difficult. Figure 7 shows the effect of depth and l2-regularisation on the interval error of a grid of intervals evaluated with interval arithmetic on the network function. In the case of the deeper network the interval error is so large that the shape of the boundary is no longer visible. The l2-regularisation improves the accuracy of the interval error, though the width of the error around the boundary varies for different regions of the input space.

6.2. **Sample spaces.** Since we assume all training data lies in $[0, 1]^n$, the most straightforward region to choose as the sample space would be $\mathcal{U} = [0, 1]^n$, which corresponds to measuring the volume of an ε-neighbourhood of the decision boundary on the entire space of possible data points. We use the shorthand

$$\textbf{Bvol} = \text{Bvol}_\varepsilon(N, [0, 1]^n)$$

to denote this volume measurement. However as the training data is usually assumed to be sampled from a much lower dimensional manifold in a selected region of $[0, 1]^n$, measuring ε-neighbourhood boundary volume on the whole of $[0, 1]^n$ would capture structure of the decision boundary unrelated to the classification task.

We therefore propose two additional target sample spaces $\mathcal{U}$ on which to measure the ε-neighbourhood boundary volume. The first will target measurements of the global complexity of decision boundary restricted within the region relevant to training data. The second region will look to analyse the local complexity of the decision boundary between classification classes. All three sample spaces are examined in the experimental part of the paper in Section 7. Both sample spaces are formed by considering a number of cubical regions centered on points lying in $[0, 1]^n$ and whether the overlaps of these cubical regions change the interpretation of measurements. However, we go on to justify in

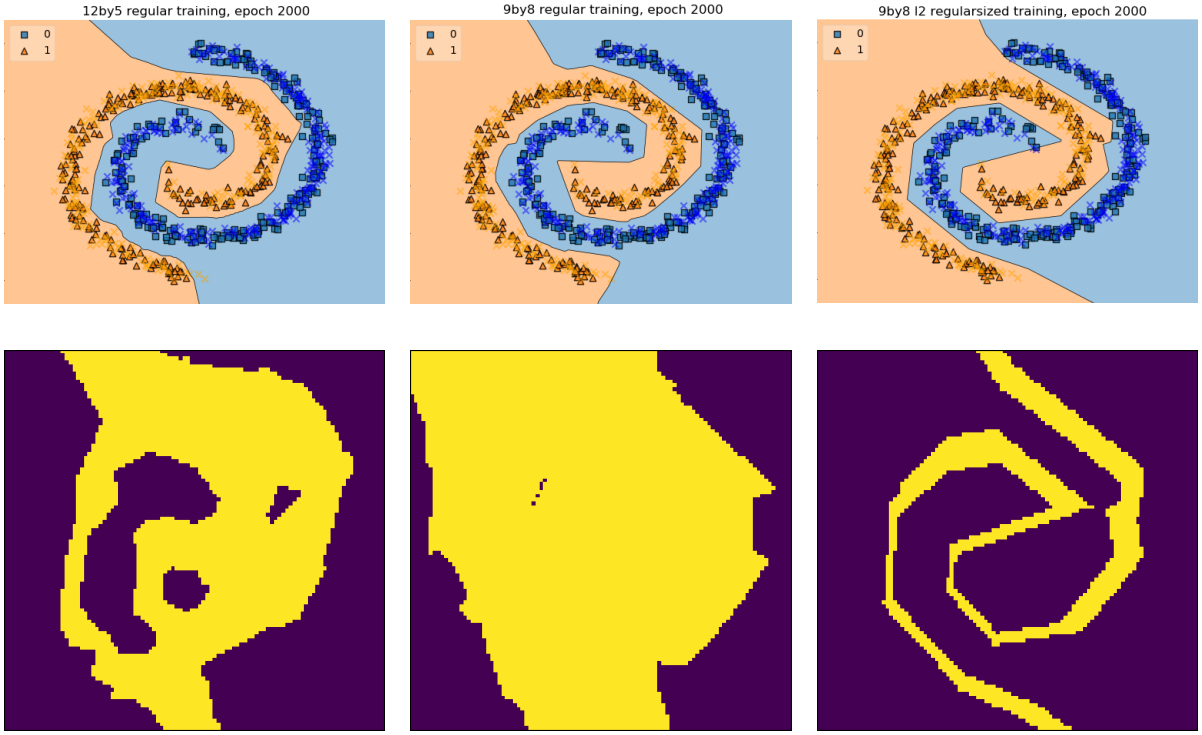

FIGURE 7. Three networks with Relu activation functions are trained on a spiral data set for 2000 epochs with the Adam optimizer. In the first column we show a network with 5 layers of 12 hidden units, the second column 8 layers of 9 hidden units and the last column again 8 layers of 9 hidden units trained with l2-regularisation. The top row shows plots of the network functions, the bottom row shows the evaluation of the network on a 100 by 100 grid of intervals coloured yellow if the output interval contains the boundary and blue otherwise. The interval arithmetic estimate of the neighbourhood boundary volume is usually a large overestimate and varies considerably in thickness around the boundary.

Remark 6.2 that in a high dimensional setting under most reasonable circumstances the two interpretations we set out are in fact almost identical, so it will be unambiguous to refer to them both with the same notation. The two regions are defined as follows.

(1) Given training data X and a $\delta > \varepsilon$, set

$$\mathcal{U} = \bigcup_{x \in X} B_\delta(x)$$

where $B_\delta(x) = \{x' \in \mathbb{R}^n \mid d_\infty(x, x') \leq \delta\}$ is the ball of radius δ around x. We use the shorthand

$$\textbf{TrainBvol} = \mathrm{Bvol}_\varepsilon(N, \cup_{x \in X} B_\delta(x)) \tag{7}$$

to denote this volume measurement. Providing δ is smaller than the minimal distance between points in X the δ-neighbourhoods around points do not overlap, so **TrainBvol** will measure the ε-boundary volume in the δ-neighbourhood of the training set. See Figure 8 for a demonstration of the set $\mathcal{U}$ in this case. We want to highlight that **TrainBVol** depends on the parameters ϵ and δ that need to be selected in sensible way to get a meaningful measurement. Please consult the experimental section for examples.

Otherwise the probabilistic interpretation of the value approximated previously in equation (6) by sampling in $\mathcal{U}$ changes to the expected value of the boundary volume in the

δ-neighbourhoods of the training set multiplied by the number of training points,

$$\textbf{TrainBvol} = |X| \cdot \mathbb{E}(\text{Bvol}_\varepsilon(N, B_\delta(x)) \mid x \in X). \tag{8}$$

In practice measuring the average ε-neighbourhood boundary volume in a δ-neighbourhood of the training set might be interpreted as measuring how much the decision boundary twists around the training data and is a global measurement with respect to the training data.

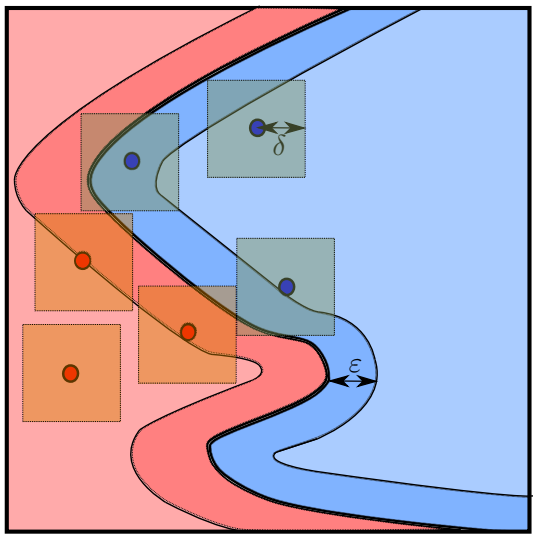

FIGURE 8. Sample space of δ neighbourhoods about training points in which to estimate the **TrainBvol** ε-neighbourhood volume of the network decision boundary.

(2) Our second target for neighbourhood boundary volume obtains measurements in a two stage process. Firstly, following the procedures detailed in [36], we sample points lying linearly between two training examples of opposite classes to obtain a point on the boundary between them. These are the orange points from the example given in Figure 9. This is achieved by repeatedly subdividing the ray between the points in half with further subdivisions applied to a section that contains the boundary and considering the midpoint of the remaining interval after the final subdivision. The proximity of this linear adversarial example to the network decision boundary boundary is up to a degree of accuracy determined by the number α of subdivisions. Assuming that the two points are no more then distance 1 apart, the distance of the adversarial example from the decision boundary is bounded above by $1/2^\alpha$. While such points near the boundary are obtained in a slightly biased fashion the output efficiently provides us with a good sample of points located between label classes.

Secondly, we then measure how much the decision boundary twists between the data classes of distinct classes by making boundary volume measurement in a δ-neighbourhood of the point sampled on the decision boundary between the classes. Again, see Figure 9 for a demonstration of this procedure.

Denote the set of pairs of points from X with different class labels in Y by

$$\text{cp}(X, Y) = \{x_i, x_j \in X \times X \mid \text{ such that the corresponding } y_i, y_j \in Y \text{ satisfy } y_i \neq y_j)\}.$$

Define $LA(x, x')$ to be the closest adversarial example to x lying on the ray between points $x, x' \in \mathbb{R}^n$. If the δ-neighbourhoods are disjoint then we define

$$\textbf{LAdvBvol} = \text{Bvol}_\varepsilon(N, \cup_{(x,x') \in \text{cp}(X,Y)} B_\delta(LA(x, x')).$$

Otherwise, we consider in the region $\mathcal{U} = \coprod_{(x,x')\in \mathrm{cp}(X,Y)} B_\delta(LA(x,x'))$, in the case when the $B_\delta(LA(x,x'))$ overlap. In this case, the probabilistic interpretation of the ε-neighbourhood boundary volume measurement is

$$\textbf{LAdvBvol} = |\mathrm{cp}(X,Y)| \cdot \mathbb{E}\big(\mathrm{Bvol}_\varepsilon(N, B_\delta(LA(x_i,x_j)) \mid (x_i,x_j) \in \mathrm{cp}(X,Y)\big). \tag{9}$$

In practice however, even for a reasonably sized data set X, there would be a very large number of possible pairs of points in $\mathrm{cp}(X,Y)$. Therefore we only take a random sample among all possible pairs (x_i,x_j) with $y_i \neq y_j$ to obtain an estimate of a ε-neighbourhood boundary volume measurement of **LAdvBvol**.

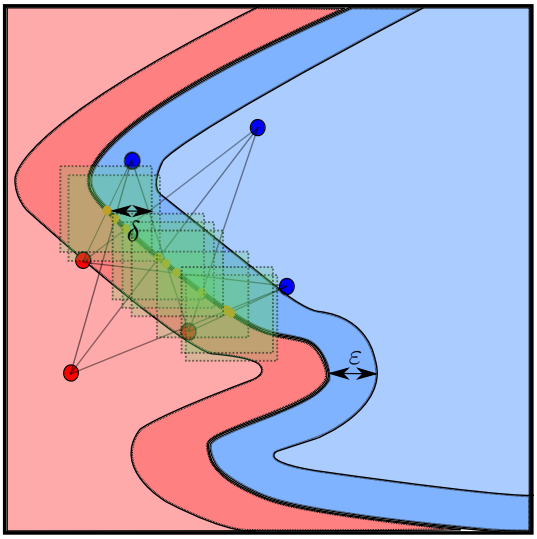

FIGURE 9. The points sampled on the boundary linearly between data classes in the first step (orange) within which the second step estimate of **LAdvBvol**, the ε-neighbourhood volume of the network decision boundary in a sample space of δ neighbourhoods about these boundary points is made.

Remark 6.1. The ε-boundary volume measurements of the decision boundary we proposed are confined to the volume within the union of cubes in $\mathbb{R}^n$. In the case of this intersection being a manifold with boundary, the Weyl tube formula discussed in Section 5 determines the boundary of the tube to be given by the direction of the normal vector on the boundary of the manifold. It is clear that in low dimensions there would usually be a difference in the ε-boundary volume considered in the two cases as the normal vector of the network decision boundary at a point intersecting a face of the cube need not be perpendicular to the normal vector of the face of the cube. However, in high dimensions Proposition 4.1 tells us that randomly chosen normal vectors on the manifold will be almost perpendicular to all faces of the cube with high probability, making the volume of regions inside the cubes approximately identical to Weyl's notion of a tubular neighbourhood. In practice the axes of the cube are related to the data as they represent the data features, though this possible correlation could be removed by randomising the coordinates before taking a neighbourhood boundary volume measurement.

Remark 6.2. If two unit cubes aligned with coordinate axes in $\mathbb{R}^n$ overlap by d_i in each coordinate axis for $i = 1,\ldots,n$ then the volume of the overlap region is $\prod_{i=1,\ldots,n} d_i$. Hence given m axis-aligned unit cubes in $\mathbb{R}^n$ of equal size with overlap in each axis bounded above by $0 \leq \zeta < 1$, then

$$\text{cube overlap volume } \leq \binom{m}{2}\zeta^n < \frac{m^2\zeta^n}{2} \tag{10}$$

Therefore even if the number of cubes m grows proportionally to the dimension n, the percentage volume of cube overlap vanishes as $n \to \infty$. So in high dimensions the probabilistic interpretations presented above in equations (7) and (8) for **TrainBvol** and equations (2) and (9) for **LAdvBvol**, converge to the same value as dimensions increase even when the δ-neighbourhoods overlap. The value of the bound given in equation (10) is plotted against the number of dimension in the case of 100 cubes for each dimension in Figure 10, here we see that even for very large ζ values the size of the bound reduces to almost zero in a relatively small number of dimensions.

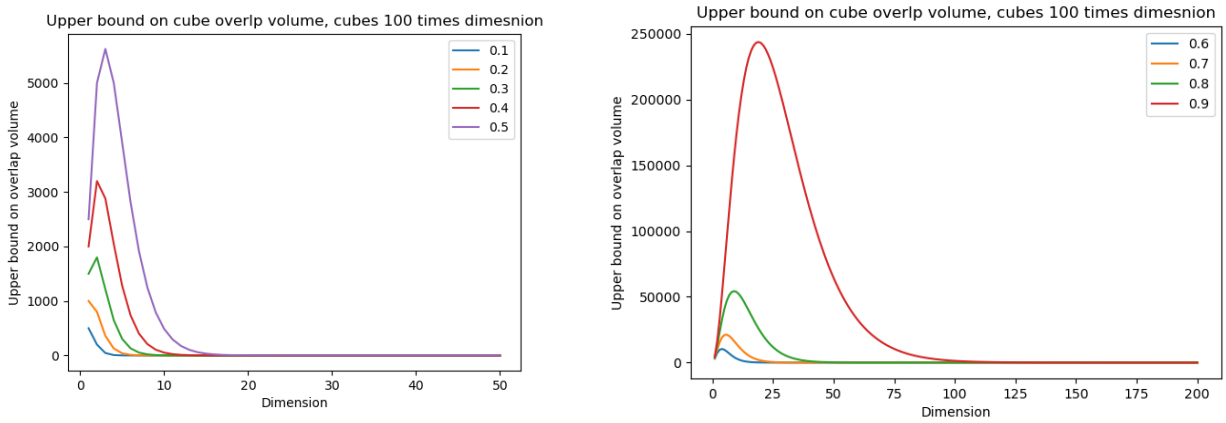

FIGURE 10. Value of the bound given in equation (10) over varying dimensions in the case of 100 cubes for each dimension, with curves for varying sizes of ζ between 0.1 to 0.9. The bound consistently vanishes for even modestly high dimensions.

In the two dimensional case of a curve in the plane the Weyl tube formula (5) simplifies to depend only on ε and the length of the curve. As discussed in Remark 6.1, in low dimensions the correspondence between Weyl's tube formula and the ε-boundary volume measurements **Bvol**, **TrainBvol** and **LAdvBvol** inside cubes can break down. In addition, as discussed in Remark 6.2 any overlap between cubes could cause differences between **Bvol** and **TrainBvol**. However, we see in Figure 11 that the **Bvol** and **TrainBvol** measurements taken during the experiment presented in Section 3 are almost identical in shape to the boundary length measurements. In this case **Bvol** and **TrainBvol** are similar as the spiral data set is well dispersed throughout $[0, 1]^2$. We also note that in the **LAdvBvol** plot in Figure 11, the narrowing of the **LAdvBvol** measurements shows that despite the shorter length of the shallower network architectures the local volume measurements are similar, indicating the increasing roughness of the boundaries obtained from training shallower networks when compared to the smoother higher curvature disconnected boundary components of deeper networks. This demonstrates that our proposed methods developed for higher dimensions reveal the same information we were able to obtains about the geometry of the decision boundary using other techniques for 2-dimensions in Section 3.

6.3. **A simple near boundary adversarial attack method.** As outlined in the procedure from Section 6.1, to compute the ε-neighbourhood boundary volume we require a method that returns an approximate distance from a point to the decision boundary of the neural network function. We propose to do this by performing an adversarial attack to obtain an adversarial example as close as possible to the boundary near the initial point. This allows us to measure the neighbourhood boundary volume over a range of ε values so we do not need to select an optimal ε in advance.

The attack method should not be computationally expensive as the Monte Carlo method requires that we take a large number of samples. However, as our aim is only to approximate the boundary

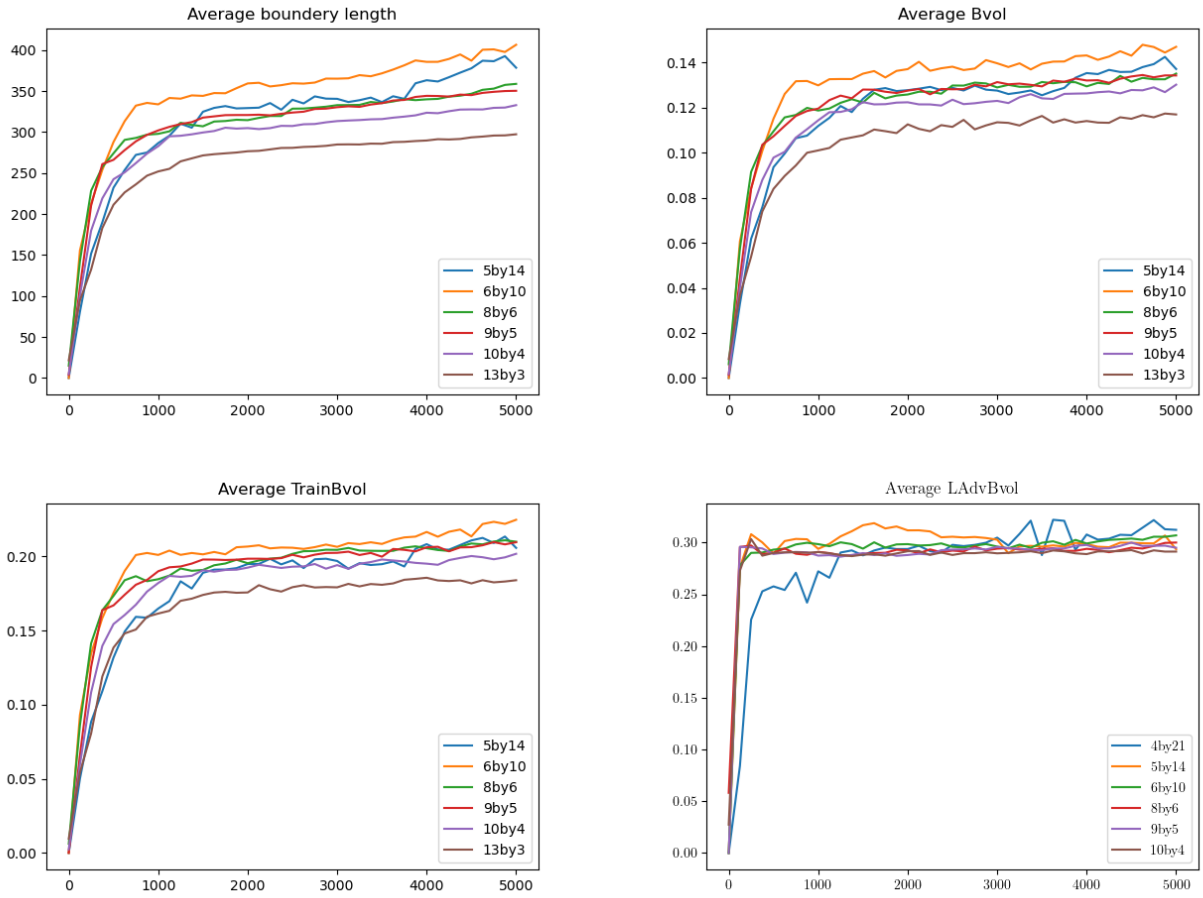

FIGURE 11. Comparison between the average boundary lengths during training of networks on 2-dimensional data presented in Figure 3 compared to average ε-neighbourhood boundary volume measurements **Bvol**, **TrainBvol** and **LAdvBvol** taken during the training of the same network. For the neighbourhood boundary volume measurements $\varepsilon = 0.01$, $\delta = 0.05$ and 1000 points were used for the Monte Carlo estimation. The **Bvol**, **TrainBvol** measurements are very similar to boundary length with higher values for deeper networks. The **LAdvBvol** narrows after training, suggesting similar complexity of network decision boundaries between between data classes. We see that the the trends in the **Bvol** and **TrainBvol** plots are the same as the total boundary length plot in 2-dimensions, while **LAdvBvol** measurements contain different information about the decision boundary more locally.

volume in a small neighbourhood of the boundary, the adversarial attack method need only give accurate distance measurements when in close proximity to the decision boundary. For these purposes we introduce an adapted version of the FGSM (see equation (2)) to find adversarial examples close to the decision boundary, an example demonstrating that the FGSM performs as well as projected gradient decent attack at distances less than 0.1 on a network trained on MNIST is given in [67].

Given a point $x \in \mathbb{R}^n$ we look for an adversarial example in the direction

$$v = \text{sign}(\nabla_x \mathcal{L}(N, x, y))$$

to a minimal accuracy determined by parameter $\alpha \in \mathbb{N}$, that is a point $A(x)$ (if it exists) the closest point to x in the direction v whose predicted label differs from x up to an error in distance of $\frac{1}{\alpha}$

Since we wish to apply this to measuring the boundary volume of a ε-neighbourhood, we may also put an upper bound on our search distance of ε. Therefore in practice α should be chosen so that $\frac{1}{\alpha}$ is considerably smaller than ε. The adversarial example $A(x)$ is then determined by normalizing the direction vector v to $\hat{v}$ and evaluating in the ascending order of k the network predictions $p(N(\frac{k}{\alpha} \cdot \hat{v}))$ for $k = 1, \ldots, k_{\max}$ such that $k_{\max}$ is the maximal integer for which $\frac{k_{\max}}{\alpha} < \varepsilon$.

Remark 6.3. It should be noted that the FGSM is more effective against a network with rectified linear activation functions due to the linear nature of the adversarial attack method. As a consequence, ε-neighbourhood boundary volume measurements using the method we just set out should therefore provide a more reliable bound for a network with rectified linear activation functions.

Remark 6.4. Referring back to Section 5 and the motivation for the ε-neighbourhood boundary volume measurement using Weyl's tube formula, the ε value for the adversarial attack method presented in this section is an l_∞ metric distance not an l_2 metric distance. However, the result of applying our adversarial attack method to compute boundary volume on a subspace of $\mathbb{R}^n$ can still be interpreted as a lower bound on the l_2 metric tube if we take the tube radius to be $\sqrt{n}$ times the l_∞ value of ε. This is because the l_∞ ball of radius ε is a cube with side lengths 2ε, which has a maximal radius of $\varepsilon\sqrt{n}$ in the l_2 metric.

7. Experimental results

In this section we present experimental results for neighbourhood boundary volume measurements with networks trained on fully connected and convolutional neural networks under changes of dropout hyperparameter settings in their final hidden layer on the MNIST, Fashion MNIST and CIFAR-10 data sets in order to demonstrate the relationship between neighbourhood boundary volume and network function generalisation.

7.1. Summary of results.

In Section 7.3 For each data set and network architecture we vary the the dropout rate [1] between 0 and 0.5, presenting the neighbourhood boundary volumes of the network for **Bvol**, **TrainBvol** and **LAdvBvol** over 10 random network initializations for appropriately chosen ε and δ values and observe the average trend in measurements. For the convolutional neural networks we observe a straightforward relationship where we see an increase in **LAdvBvol** values as dropout rates increase and a local minimum in the **TrainBvol** measurements roughly coinciding with the dropout rate of highest test accuracy. For the fully connected networks we observe a more complex trade-off between the local geometry of **LAdvBvol** and global geometry of **TrainBvol** where the behaviour of one appears to drive the behaviour in the other and for any given data set the local maxima in the test accuracy correspond to the critical points in either **LAdvBvol** or **TrainBvol** measurements.

The reliability and error margin in the boundary volume measurements in terms of chosen ε value and number of Monte Carlo samples is considered along side the selection on network parameter settings thought Section 7.2 with reference to additional empirical validation presented as part of Appendixes B and C.

7.2. Experimental setup.

For MNIST and Fashion MNIST we trained fully connected networks with a single hidden layer of 100 units and for CIFAR-10 a fully connected network with 3 hidden layers of 1024 units. The architecture for the hidden layers of the convolutional neural networks is given in Table 1. All networks used rectified linear activation functions and a softmax activation on their output layer.

Networks were trained with the cross-entropy loss function using randomly initialised weights distributed according to He normal conditions [38]. Unless otherwise stated the networks trained on MNIST used the stochastic gradient descent (SGD) optimizer at a learning rate of 0.01 and the networks trained on Fashion MNIST and CIFAR-10 were trained with the Adam optimizer [44] at

[1] Defined as the probability of removing a neuron in the network during each batch of training.

MNIST	Fashion MNIST	CIFAR-10
Conv 16, $5 \times 5 + 1$	Conv 16, $5 \times 5 + 1$	Conv 16, $3 \times 3 + 1$
Pool $2 \times 2 + 2$	Pool $2 \times 2 + 2$	Pool $2 \times 2 + 2$
Conv 32, $5 \times 5 + 1$	Conv 32, $5 \times 5 + 1$	Conv 32, $3 \times 3 + 1$
Pool $2 \times 2 + 2$	Pool $2 \times 2 + 2$	Pool $2 \times 2 + 2$
FC 100	FC 100	Conv 64, $3 \times 3 + 1$
		Pool $2 \times 2 + 2$
		FC 256

TABLE 1. Convolutional architectures used for training on data sets MNIST, Fashion MNIST and CIFAR-10. The notation Conv a, $b \times b + 1$ means a 2-dimensional convolutional layer with a square filters of size b, all with a stride of 1 and same padding. All pooling layers Pool $2 \times 2 + 2$ used square filters of size 2 with a stride of 2 and had no padding. The notation FC c means a fully connected layer with c hidden units.

a learning rate of 0.0001. All training was completed with constant batch sizes of 32. The number of training iterations varies considerably between experiments and were chosen to coincide with the epoch at which the networks approximately first achieve their optimal training accuracy. We therefore detail the number of training epochs individually later.

Unless otherwise stated, in all experiments with ε-neighbourhood boundary volume measurements we used ε and δ parameter settings detailed in Table 2 and the Monte Carlo sample size was always 10^4. In all cases the adversarial attack method presented in Section 6.3 was used. For **LAdvBvol** measurements the number of subdivisions α used in the first step of the procedure was 10 and the number of these adversarial examples near the decision bounder sampled was 10^3.

Data (Nework)	Method	ε	δ
MNIST (FC)	**Bvol**	0.001	-
	TrainBvol	0.003	0.2
	LAdvBvol	0.001	0.2
Fashion MNIST (FC)	**Bvol**	0.001	-
	TrainBvol	0.003	0.2
	LAdvBvol	0.0008	0.2
CIFAR-10 (FC)	**Bvol**	0.002	-
	TrainBvol	0.003	0.2
	LAdvBvol	0.0008	0.2
MNIST (Conv)	**Bvol**	0.0005	-
	TrainBvol	0.002	0.2
	LAdvBvol	0.0003	0.2
Fashion MNIST (Conv)	**Bvol**	0.001	-
	TrainBvol	0.003	0.2
	LAdvBvol	0.001	0.2
CIFAR-10 (Conv)	**Bvol**	0.001	-
	TrainBvol	0.0025	0.05
	LAdvBvol	0.0005	0.05

TABLE 2. The parameter values of ε and δ for each neighbourhood boundary volume measurement using the notation and methods presented in Section 6.1.

The choice of δ in Table 2 was made by considering the maximal distance between points in training set classes with the l_∞ metric. For MNIST, the minimal distance between two classes was approximately 0.737 in the case of classes 1 and 7. For Fashion MNIST, all distances were above 0.4 except for 5 pairs $(0,2),(1,6),(2,4),(2,6),(4,6)$ the smallest of which was for $(2,4)$ at a distance of approximately 0.318. For CIFAR-10, the minimal distance between classes was 0.22. For most experiments, we chose a value of $\delta = 0.2$ as this was always less than the distance between classes. The only exception is for training convolutional neural networks on CIFAR-10 where we found a smaller δ was necessary for clean results and here we used $\delta = 0.05$ Applying the findings of Remark 6.2 we can conclude that in all cases **TrainBvol** is almost identical to the ε-neighbourhood boundary volume of the union of regions around the training data. Meanwhile, as can be seen in Table 3, these choice of δ values were also reasonable for **LAdvBvol**, as despite a few close by points very few of the 10000 sampled points lying close to the boundary are ever less than 0.2 away from each other.

Data (network)	Minimal distance	Average distance	Fraction < 0.2	Fraction < 0.1	Fraction < 0.05
MNIST (Fc)	0.0102	0.587	0.0034	0.0008	0.0004
Fashion MNIST (Fc)	0.000400	0.4601	0.0269	0.0181	0.0145
CIFAR-10 (Fc)	0.000270	0.388	0.0669	0.039	0.0313
MNIST (Conv)	0.148	0.577	0.0002	0	0
Fashion MNIST (Conv)	0.000463	0.465	0.0215	0.0057	0.0026
CIFAR-10 (Conv)	0.000216	0.385	0.0774	0.0538	0.0473

TABLE 3. Statistics on the the minimal l_∞ distances between any pairs of 10000 points sampled using the method described in Section 6.2, for constructing points lying linearly between two training examples as part of the **LAdvBvol** method on a trained neural network. For a network trained with each combination of data sets and network architecture the minimum distance and average minimum distances between pairs of points are given along with the fraction of those pairs of points which were less than 0.2, 0.1 or 0.05 apart.

To test the size of the Monte Carlo error at our size of 10^4 point samples, we took a fully connected network trained on MNIST and looked at the mean and range of values for each of the three ε-neighbourhood boundary volume measurements which are presented in Table 4. In all cases the range of estimates vary by less than 0.004.

	Bvol	**TrainBvol**	**LAdvBvol**
Mean	0.27737	0.06626	0.16829
Range	0.00237	0.00193	0.00395

TABLE 4. Mean values and ranges over 10 trials of the three ε-neighbourhood boundary volume measurements to demonstrate the size of the Monte Carlo error on a fully connected network trained for 100 epochs on MNIST with the SGD optimizer for 100 epochs using previously outlined sampling settings and ε and δ values as presented in the first row Table 2.

Both the number of training examples and the linear adversary sample size in are 10^3 while the total number of Monte Carlo samples within a δ-neighbourhood of the points used to approximate **TrainBvol** and **LAdvBvol** is 10^4, only a factor of 10 bigger. We found that this was in fact a large enough sample to obtain a stable estimate to an accurate value as is demonstrated in Table 4. An

explanation of this is that in both cases the measurement can be seen as the volume of the union of δ-neighbourhoods of the sampled points as discussed in Remark 6.2, so it is not necessary to have a very a large sample size per neighbourhood.

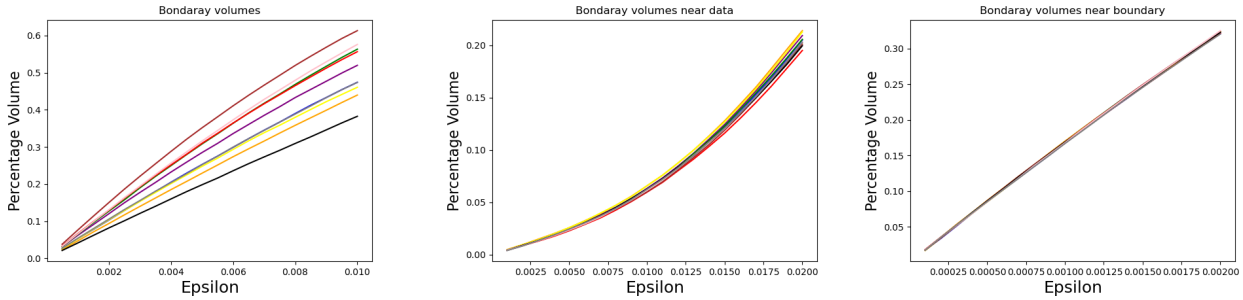

FIGURE 12. Values of ε-neighbourhood boundary volume measurements over a range of ε values for 10 networks trained on MNIST. The stability of the measurements appears good over the entire range of ε. The rate or growth with increasing ε of neighbourhood boundary volume measurements is largely dependent on the accuracy of the adversarial attack and curvature of the decision boundary which we explore in more detail in Appendix C.

In Figure 12 we show the three ε-neighbourhood boundary volume measurements on 10 networks trained with MNIST over a range of ε values. In addition to remaining smooth over a range of values each network is distinguishable with measurement values keeping the same order. This demonstrates that the choice of ε value does not effect much the measurements stability. We chose to use ε values towards the lower end of the value ranges used in Figure 12 separately in each case satisfying a number of conditions in order to guarantee that the volume measurements were proportional to the true volume of the decision boundary as we initially discussed in Section 5. Firstly, in accordance with the results of Appendix B, we chose ε values so that as far as possible ε-neighbourhood boundary volume measurements were well below 0.2 to avoid any effect of intersections between different class boundary manifolds. Secondly, particularly in the case of networks trained on MNIST we chose ε small enough in order to avoid the effects of curvature on measurements as discussed further in Appendix C, that is we chose ε small enough so that as a function of ε the neighbourhood boundary volume measurements are approximately locally linear.

7.3. **Effect of dropout on the decision boundary.** We now investigate the effect of regularisation on the shape of the decision boundary of a trained neural network. In the experiments we vary the values of dropout regularisation applied to the final hidden layers of the networks varying between 0 and 0.5. In each case we train 10 randomly initialised networks which we present in the columns of a graph indicating the mean and 1 standard deviation either side. The number of training epochs selected for each experiment are given in Figure 5, these were varied to keep them in line with roughly the correct epoch number in terms of first obtaining the optimal training accuracy for each individual data set and dropout rate. We present here the boundary volume results for **trainBvol** and **LAdvBvol** as these give the most information. The **Bvol** measurements are presented in Appendix A and are omitted here as we found that these results were generally noisier and gave less information than the other neighbourhood boundary volume measurements.

We present first the results on convolutional neural networks in Table 13. As a consequence of their more sophisticated architecture they provided a simpler structure in the behaviour of their decision boundaries. After this we then present the results on fully connected networks in Figure 14.

Data	Architecture	Optimiser	Epochs	+epochs per 0.1 dropout
MNIST	Fc	SGD	100	0
MNIST	Conv	SGD	200	50
MNIST	Fc	Adam	70	30
MNIST	Conv	Adam	50	15
Fashion MNIST	Fc	Adam	100	0
Fashion MNIST	Conv	Adam	50	15
CIFAR-10 MNIST	Fc	Adam	25	5
CIFAR-10 MNIST	Conv	Adam	30	10

TABLE 5. Number of training epochs and increase in the number of training epochs per increase in 0.1 of dropout rate on the final hidden layer. Epochs are given for fully connected and convolution neural networks for each data set used in experiments in this section.

In Table 13 we see that the effect of varying the dropout rate on the final layer of a convolution neural networks trained on MNIST with the SGD optimizer and Fashion MNIST with the Adam optimizer. In this case, the test accuracy continues to increase with higher dropout rates. The **TrainBvol** measurements decrease as the dropout rate is increased even up to the point where the dropout rate becomes too high to stably train with, so the local minimum is the global minimum. Meanwhile, the **LAvBvol** measurements increase monotonically as dropout rate increases.

With the networks trained on MNIST with the Adam optimizer, we see the test accuracy is highest around a dropout rate of 0.4 lowering slightly on average by 0.5. Similarly to the SGD case the **LAvBvol** measurements generally increased monotonically with dropout rate, however this time a minimum in the **TrainBvol** measurements appears at a dropout rate of 0.3 just before the optimal dropout rate in the test accuracy.

For the networks trained on Fashion MNIST in the third row of Table 13 , the test accuracy increases up to dropout rates of 0.3 and 0.4 before lowering slightly. The **TrainBvol** measurements decrease to a local minimum as dropout increases around 0.3 corresponding to the best performing networks in term of test accuracy. The **LAvBvol** measurements vary little with a slight increase as dropout rate increases.

Finally in the last row of 13, the results for networks trained on CIFAR-10 show that the test accuracy increased with increased dropout rate leveling off at a rate of 0.4 and 0.5. As for the other data sets the **LAvBvol** measurements can be seen to increase slightly with dropout rate, though in this case the results are considerably more noisy with a less steep slope. Meanwhile, the **TrainBvol** measurements fall to local minima around dropout rates of 0.2 and 0.3 slightly preceding the leveling off in test accuracy.

Figure 14 presents the values of the test accuracy and ε-neighbourhood boundary volume measurements for fully connected networks. The result for training on MNIST with the SGD and Adam optimizers over varying dropout rate in the first and second rows of Figure 14, show that on average the best test accuracies were obtained with a dropout rate around 0.1. The **TrainBvol** measurements are minimised also around dropout values of 0.1 and 0.15 while, the **LAdvBvol** values in the SGD case decrease monotonically and increase monotonically in the Adam case.

An explanation for all the observations so far would be that increasing the dropout rate drives primarily a change in local behavior of **LAdvBvol**, while there is then a trade off in the global behavior of **TrainBvol**. It appears that the values at which this trade off finds network functions with better generalising properties is at the tuning point in **TrainBvol**. However, it is surprising in the Adam case for fully connected networks that an increasing regularisation hyperparameter such as dropout would lead to an increase in complexity in both the local **LAdvBvol** and global **LAdvBvol** measurements.

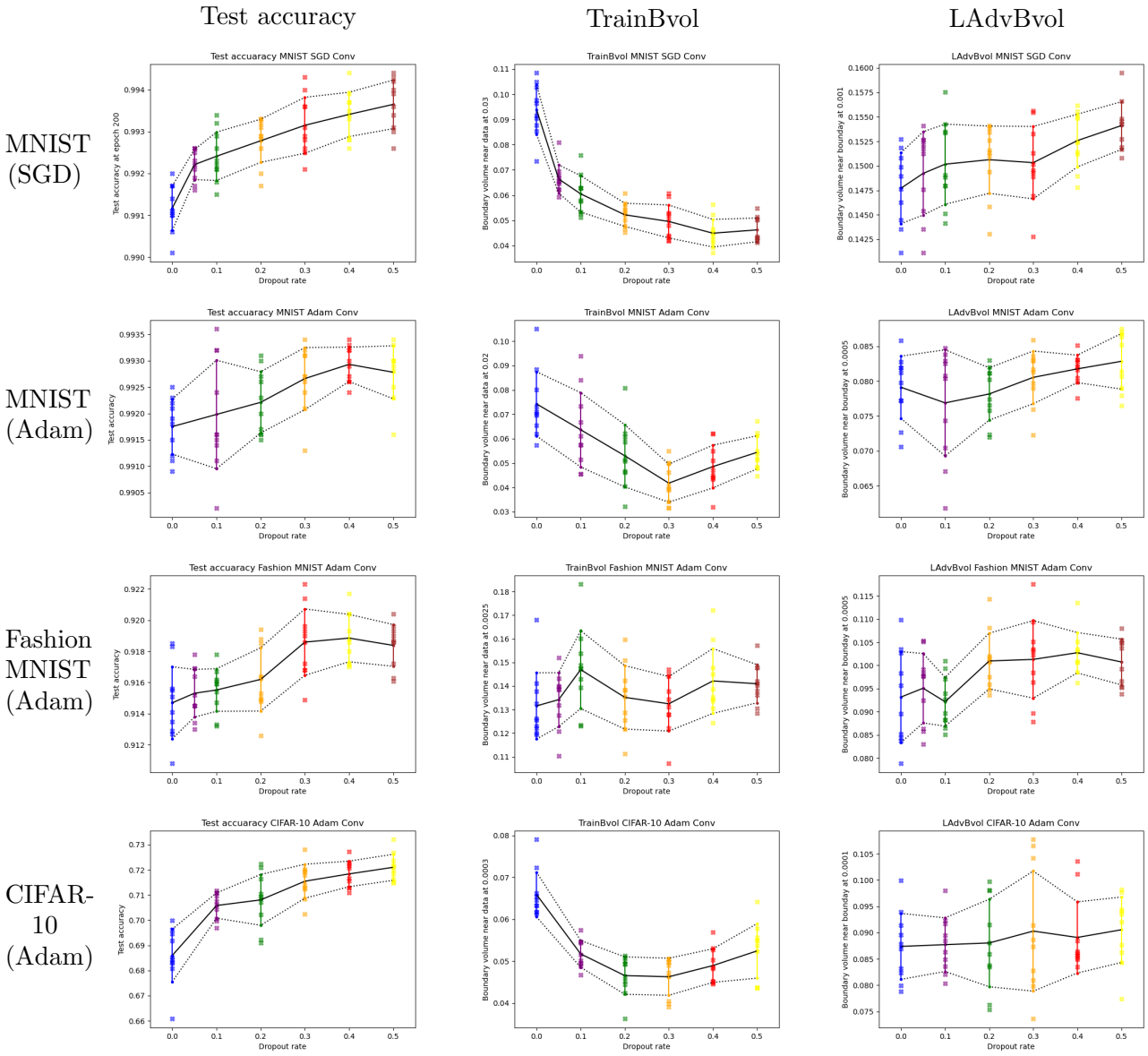

FIGURE 13. Boundary volume over varying dropout rate in the final fully connected layer applied to convolutional neural networks. The top two rows are trained on MNIST with the SGD and Adam optimizers respectively, the third row Fashion MNIST and bottom row CIFAR-10. We see a local minima in **TrainBvol** around or just before the optimal test accuracy, while the **LAdvBvol** measurements generally increase.

It should be noted that for results on MNIST the **TrainBvol** measurements grow non-linearly in ε (see Figure 12), which is not the case for other data sets due curvature as discussed in Appendix C. It was therefore necessary to ensure that the values of ε were sufficiently small so that they lay in a range that was approximately linear. We observed that the effect of curvature decreased as the dropout rate increased as we would expect, this results in a shift to larger values of the **TrainBvol** local minima for larger ε values.

In the third row of Figure 14 are the results from the networks trained on Fashion MNIST with varying dropout rate. In this case the **TrainBvol** decreases monotonically as the dropout rate

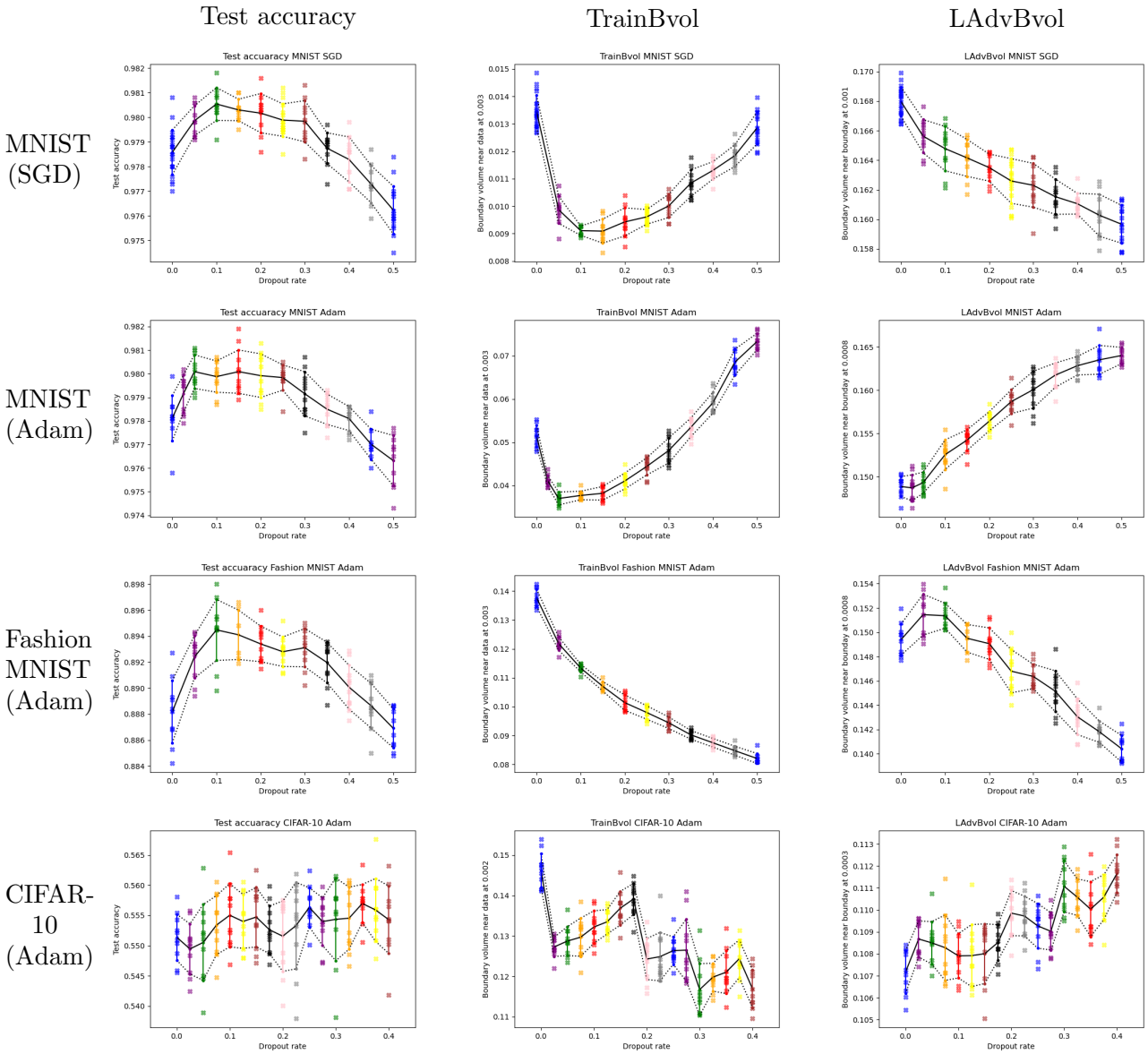

FIGURE 14. Boundary volume and test accuracy over varying dropout rate applied to fully connected neural networks. The top two rows are trained on MNIST with the SGD and Adam optimizers respectively, the third row Fashion MNIST and bottom row CIFAR-10. Optimal test accuracy is achieved around a critical point of either **TrainBvol** or **LAdvBvol**, while the other of the two measurements changes monotonically.

increases, while **LAdvBvol** increases to a maximum before decreasing. This time the optimal test accuracy coincides with the change in behavior of the **LAdvBvol** measurements. So in this case the roles of the **TrainBvol** and **LAdvBvol** measurements are reversed and the change in behaviour occurs around a local maximum rather than a local minimum. suggesting a simplification in the complexity of the network function generally as the dropout rate is increased.

The final row of Figure 14 contains the results from networks trained on CIFAR-10 and here we observe behaviour similar to previous cases but this time repeated multiple times in the boundary volume measurements over changes in dropout rate. Though noisy there there appear to be three

local peaks in the average test accuracy occurring around dropout rates of 0.1, 0.25 and 0.35 and these coincide with three local minima in the **LAdvBvol** plot. The values of **TrainBvol** also appear related to **LAdvBvol** measurements as their behaviour changes between increasing and decreasing shortly after the local minima in **LAdvBvol**, in particular increasing when **TrainBvol** decreases and decreasing when **TrainBvol** increases.

8. Conclusion

In this paper we present a method that can be applied to measure the boundary volume of a small neighbourhood of the decision boundary of a neural network. We identify several regions of the input space where measurements allow us to deduce the properties of the network functions at varying scales and hence provide a geometric explanation of network properties. Using ideas from differential and high dimensional geometry, we justified that for small neighbourhoods these measurements are directly related to the volume of the decision boundary itself. We then also show that our boundary volume measurements can be efficiently computed for networks trained on large high dimensional data sets.

To demonstrate our procedure we apply our method to studying generalisation in deep learning by measuring the the decision boundaries of networks trained over changes in dropout regularisation hyperparameters. Here we observe that the generalisation properties of the network function are related to the structure of the decision boundary and that networks with better generalisation often occur at a critical point in the behaviour of local or global boundary volumes. This observation appear to be a consequence of a trade off between complexity at different scales. For convolutional neural networks the optimal generalisation consistently occurs around a local minimum of the neighbourhood boundary volume measurements within a neighbourhood of the training set. However for fully connected neural networks the nature of the connection between generalisation and decision boundary structures varies between data set and even training algorithms, suggesting a complex relationship between the two in general.

References

1. M. Alfarra, A. Bibi, H. Hammoud, M. Gaafar, and B. Ghanem, *On the Decision Boundaries of Neural Networks: A Tropical Geometry Perspective*, arXiv e-prints (2020), arXiv:2002.08838.
2. J. Anderson, M. Belkin, N. Goyal, L. Rademacher, and J. Voss, *The more, the merrier: the blessing of dimensionality for learning large gaussian mixtures*, Proceedings of The 27th Conference on Learning Theory (Barcelona, Spain) (Maria Florina Balcan, Vitaly Feldman, and Csaba Szepesvári, eds.), Proceedings of Machine Learning Research, vol. 35, PMLR, 13–15 Jun 2014, pp. 1135–1164.
3. S. Arora, R. Ge, B. Neyshabur, and Y. Zhang, *Stronger generalization bounds for deep nets via a compression approach*, Proceedings of the 35th International Conference on Machine Learning (Jennifer Dy and Andreas Krause, eds.), Proceedings of Machine Learning Research, vol. 80, PMLR, 10–15 Jul 2018, pp. 254–263.
4. E. Atashpaz-Gargari, C. Sima, U. M. Braga-Neto, and E. R. Dougherty, *Relationship between the accuracy of classifier error estimation and complexity of decision boundary*, Pattern Recognition **46** (2013), no. 5, 1315–1322.
5. P. L. Bartlett, D. J. Foster, and M. J Telgarsky, *Spectrally-normalized margin bounds for neural networks*, Advances in Neural Information Processing Systems (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.
6. K. Beyer, J. Goldstein, R. Ramakrishnan, and U. Shaft, *When is "nearest neighbor" meaningful?*, Database Theory — ICDT'99 (Berlin, Heidelberg) (Catriel Beeri and Peter Buneman, eds.), Springer Berlin Heidelberg, 1999, pp. 217–235.
7. X. Cao and N. Z. Gong, *Mitigating evasion attacks to deep neural networks via region-based classification*, Proceedings of the 33rd Annual Computer Security Applications Conference (New York, NY, USA), ACSAC 2017, Association for Computing Machinery, 2017, p. 278–287.
8. G. Carlsson, *Topology and data*, Bull. Amer. Math. Soc. (N.S.) **46** (2009), no. 2, 255–308. MR 2476414 (2010d:55001)
9. G. Carlsson and R. B. Gabrielsson, *Topological approaches to deep learning*, Topological Data Analysis (Cham) (Nils A. Baas, Gunnar E. Carlsson, Gereon Quick, Markus Szymik, and Marius Thaule, eds.), Springer International Publishing, 2020, pp. 119–146.

10. P. Chaudhari and S. Soatto, *Stochastic gradient descent performs variational inference, converges to limit cycles for deep networks*, International Conference on Learning Representations, 2018.
11. J. Cheeger, W. Muller, and R. Schrader, *Kinematic and tube formulas for piecewise linear spaces*, Indiana University Mathematics Journal **35** (1986), no. 4, 737–754 (English).
12. C. Chen, X. Ni, Q. Bai, and Y. Wang, *A topological regularizer for classifiers via persistent homology*, Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics (Kamalika Chaudhuri and Masashi Sugiyama, eds.), Proceedings of Machine Learning Research, vol. 89, PMLR, 04 2019, pp. 2573–2582.
13. J. R. Clough, I. Oksuz, N. Byrne, J. A. Schnabel, and A. P. King, *Explicit topological priors for deep-learning based image segmentation using persistent homology*, Information Processing in Medical Imaging (Cham) (Albert C. S. Chung, James C. Gee, Paul A. Yushkevich, and Siqi Bao, eds.), Springer International Publishing, 2019, pp. 16–28.
14. J. Devlin, M. Chang, K. Lee, and K. Toutanova, *BERT: Pre-training of deep bidirectional transformers for language understanding*, Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers) (Minneapolis, Minnesota), Association for Computational Linguistics, June 2019, pp. 4171–4186.
15. D. Donoho, *High-dimensional data analysis: The curses and blessings of dimensionality*, AMS Math Challenges Lecture (2000), 1–32.
16. G. K. Dziugaite and D. M. Roy, *Computing Nonvacuous Generalization Bounds for Deep (Stochastic) Neural Networks with Many More Parameters than Training Data*, arXiv e-prints (2017), arXiv:1703.11008.
17. H. Edelsbrunner and J. Harer, *Computational topology - an introduction.*, American Mathematical Society, 2010.
18. H. Edelsbrunner and D. Morozov, *Persistent homology: theory and practice*, European Congress of Mathematics (2014), 31–50.
19. A. Fawzi, S. Moosavi-Dezfooli, and P. Frossard, *Robustness of classifiers: from adversarial to random noise*, Advances in Neural Information Processing Systems (D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, eds.), vol. 29, Curran Associates, Inc., 2016.
20. A. Fawzi, S. Moosavi-Dezfooli, P. Frossard, and S. Soatto, *Empirical study of the topology and geometry of deep networks*, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June 2018.
21. H. Federer, *Geometric measure theory*, Classics in Mathematics, Springer Berlin Heidelberg, 2014.
22. R. B. Gabrielsson, B. J. Nelson, A. Dwaraknath, P. Skraba, J. L. Guibas, and G. Carlsson, *A topology layer for machine learning*, Proceedings of the Twenty Third International Conference on Artificial Intelligence and Statistics (Silvia Chiappa and Roberto Calandra, eds.), Proceedings of Machine Learning Research, vol. 108, PMLR, 08 2020, pp. 1553–1563.
23. T. Gebhart, P. Schrater, and A. Hylton, *Characterizing the shape of activation space in deep neural networks*, 2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA), 2019, pp. 1537–1542.
24. B. Georgiev, L. Franken, and M. Mukherjee, *Heating up decision boundaries: isocapacitory saturation, adversarial scenarios and generalization bounds*, International Conference on Learning Representations, 2021.
25. J. Gilmer, N. Ford, N. Carlini, and E. Cubuk, *Adversarial examples are a natural consequence of test error in noise*, Proceedings of the 36th International Conference on Machine Learning (Kamalika Chaudhuri and Ruslan Salakhutdinov, eds.), Proceedings of Machine Learning Research, vol. 97, PMLR, 06 2019, pp. 2280–2289.
26. N. Golowich, A. Rakhlin, and O. Shamir, *Size-independent sample complexity of neural networks*, Information and Inference: A Journal of the IMA **9** (2020), no. 2, 473–504.
27. I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*, MIT Press, 2016, http://www.deeplearningbook.org.
28. I. Goodfellow, J. Shlens, and C. Szegedy, *Explaining and harnessing adversarial examples*, International Conference on Learning Representations, 2015.
29. A. Gorban and I. Tyukin, *Blessing of dimensionality: Mathematical foundations of the statistical physics of data*, Philosophical Transactions of The Royal Society A Mathematical Physical and Engineering Sciences **376** (2018), –.
30. A. Gorban, I. Tyukin, D. Prokhorov, and K. Sofeikov, *Approximation with random bases: Pro et contra*, Information Sciences **364–365** (2016), 129–145.
31. A. N. Gorban, I. T. Yu, and I. Romanenko, *The blessing of dimensionality: Separation theorems in the thermodynamic limit*, IFAC-PapersOnLine **49** (2016), no. 24, 64–69, 2th IFAC Workshop on Thermodynamic Foundations for a Mathematical Systems Theory TFMST 2016.
32. A.N. Gorban and I.Y. Tyukin, *Stochastic separation theorems*, Neural Networks **94** (2017), 255–259.
33. S. Gowal, K. Dvijotham, R. Stanforth, R. Bunel, C. Qin, J. Uesato, R. Arandjelovic, T. Mann, and P. Kohli, *On the Effectiveness of Interval Bound Propagation for Training Verifiably Robust Models*, arXiv e-prints (2018), arXiv:1810.12715.
34. A. Gray, *An introduction to weyl's tube formula*, pp. 1–12, Birkhäuser Basel, Basel, 2004.
35. M. Gromov, *Isoperimetry of waists and concentration of maps*, Geometric and Functional Analysis **13** (2003), 178–215.

36. S. Guan and M. Loew, *Analysis of generalizability of deep neural networks based on the complexity of decision boundary*, 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), 2020, pp. 101–106.
37. M. Hardt, B. Recht, and Y. Singer, *Train faster, generalize better: Stability of stochastic gradient descent*, Proceedings of The 33rd International Conference on Machine Learning (New York, New York, USA) (Maria Florina Balcan and Kilian Q. Weinberger, eds.), Proceedings of Machine Learning Research, vol. 48, PMLR, 20–22 Jun 2016, pp. 1225–1234.
38. K. He, X. Zhang, S. Ren, and J. Sun, *Delving deep into rectifiers: Surpassing human-level performance on imagenet classification*, Proceedings of the IEEE International Conference on Computer Vision (ICCV), Dec 2015.
39. W. He, B. Li, and D. Song, *Decision boundary analysis of adversarial examples*, International Conference on Learning Representations, 2018.
40. D. Hsu, Z. Ji, M. Telgarsky, and L. Wang, *Generalization bounds via distillation*, International Conference on Learning Representations, 2021.
41. Y. Jiang, B. Neyshabur, H. Mobahi, D. Krishnan, and S. Bengio, *Fantastic generalization measures and where to find them*, International Conference on Learning Representations, 2020.
42. J. Johnson, *Deep, skinny neural networks are not universal approximators*, International Conference on Learning Representations, 2019.
43. H. Karimi, T. Derr, and J. Tang, *Characterizing the decision boundary of deep neural networks*, WSDM '20: Proceedings of the 13th International Conference on Web Search and Data Mining, 01 2020.
44. D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings (Yoshua Bengio and Yann LeCun, eds.), 2015.
45. A. Krizhevsky, I. Sutskever, and G. E. Hinton, *Imagenet classification with deep convolutional neural networks*, Commun. ACM **60** (2017), no. 6, 84–90.
46. S. Kuriki and A. Takemura, *Application of tube formula to distributional problems in multiway layouts*, Applied Stochastic Models in Business and Industry **18** (2002), no. 3, 245–257.
47. R. Kwitt, C. Hofer, A. Uhl, and M. Niethammer, *Deep learning with topological signatures*, Advances in Neural Information Processing Systems 30 (NIPS), 2017, p. 1633–1643 (English).
48. M. Ledoux, *The concentration of measure phenomenon*, vol. 89, AMS, 01 2001.
49. Y. Lei and Y. Ying, *Sharper generalization bounds for learning with gradient-dominated objective functions*, International Conference on Learning Representations, 2021.
50. R. Lev, E. Saucan, and G. Elber, *Curvature estimation over smooth polygonal meshes using the half tube formula*, Mathematics of Surfaces XII (Berlin, Heidelberg) (Ralph Martin, Malcolm Sabin, and Joab Winkler, eds.), Springer Berlin Heidelberg, 2007, pp. 275–289.
51. Y. Li, L. Ding, and X. Gao, *On the Decision Boundary of Deep Neural Networks*, arXiv e-prints (2018), arXiv:1808.05385.
52. T. Liang, T. Poggio, A. Rakhlin, and J. Stokes, *Fisher-rao metric, geometry, and complexity of neural networks*, Proceedings of the Twenty-Second International Conference on Artificial Intelligence and Statistics (Kamalika Chaudhuri and Masashi Sugiyama, eds.), Proceedings of Machine Learning Research, vol. 89, PMLR, 04 2019, pp. 888–896.
53. B. Liu and M. Shen, *Some Geometrical and Topological Properties of DNNs' Decision Boundaries*, arXiv e-prints (2020), arXiv:2003.03687.
54. C. Loader, *The Volume-of-Tubes Formula: Computational Methods and Statistical Applications*, arXiv Mathematics e-prints (2005), math/0511502.
55. P. M. Long and H. Sedghi, *Generalization bounds for deep convolutional neural networks*, International Conference on Learning Representations, 2020.
56. E. R. Love, B. Filippenko, V. Maroulas, and G. Carlsson, *Topological Deep Learning*, arXiv e-prints (2021), arXiv:2101.05778.
57. D. Mickisch, F. Assion, F. Greßner, W. Günther, and M. Motta, *Understanding the Decision Boundary of Deep Neural Networks: An Empirical Study*, arXiv e-prints (2020), arXiv:2002.01810.
58. S. Moosavi-Dezfooli, A. Fawzi, O. Fawzi, P. Frossard, and S. Soatto, *Robustness of classifiers to universal perturbations: a geometric perspective*, arXiv e-prints (2017), arXiv:1705.09554.
59. S. Moosavi-Dezfooli, A. Fawzi, J. Uesato, and P. Frossard, *Robustness via curvature regularization, and vice versa*, 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2019, pp. 9070–9078.
60. V. Nagarajan and Z. Kolter, *Deterministic PAC-bayesian generalization bounds for deep networks via generalizing noise-resilience*, International Conference on Learning Representations, 2019.
61. G. Naitzat, A. Zhitnikov, and L. Lim, *Topology of deep neural networks*, Journal of Machine Learning Research **21** (2020), no. 184, 1–40.
62. B. Neyshabur, S. Bhojanapalli, and N. Srebro, *A PAC-bayesian approach to spectrally-normalized margin bounds for neural networks*, International Conference on Learning Representations, 2018.

63. B. Neyshabur, R. Tomioka, and N. Srebro, *In Search of the Real Inductive Bias: On the Role of Implicit Regularization in Deep Learning*, arXiv e-prints (2014), arXiv:1412.6614.
64. B. Neyshabur, R. Tomioka, and N. Srebro, *Norm-based capacity control in neural networks*, Proceedings of The 28th Conference on Learning Theory (Paris, France) (Peter Grünwald, Elad Hazan, and Satyen Kale, eds.), Proceedings of Machine Learning Research, vol. 40, PMLR, 07 2015, pp. 1376–1401.
65. L. Oala, C. Heiß, J. Macdonald, M. März, W. Samek, and G. Kutyniok, *Interval Neural Networks: Uncertainty Scores*, arXiv e-prints (2020), arXiv:2003.11566.
66. C. S. Pun, K. Xia, and S. Xian Lee, *Persistent-Homology-based Machine Learning and its Applications – A Survey*, arXiv e-prints (2018), arXiv:1811.00252.
67. A. Raghunathan, J. Steinhardt, and p. Liang, *Certified defenses against adversarial examples*, International Conference on Learning Representations, 2018.
68. K. N. Ramamurthy, K. Varshney, and K. Mody, *Topological data analysis of decision boundaries with application to model selection*, Proceedings of the 36th International Conference on Machine Learning (Kamalika Chaudhuri and Ruslan Salakhutdinov, eds.), Proceedings of Machine Learning Research, vol. 97, PMLR, 07 2019, pp. 5351–5360.
69. A. Rathore, N. Chalapathi, S. Palande, and B. Wang, *Topoact: Visually exploring the shape of activations in deep learning*, Computer Graphics Forum **40** (2021), 1–14.
70. B. Recht, R. Roelofs, L. Schmidt, and V. Shankar, *Do ImageNet classifiers generalize to ImageNet?*, Proceedings of the 36th International Conference on Machine Learning (Kamalika Chaudhuri and Ruslan Salakhutdinov, eds.), Proceedings of Machine Learning Research, vol. 97, PMLR, 09–15 Jun 2019, pp. 5389–5400.
71. B. Rieck, M. Togninalli, C. Bock, M. Moor, M. Horn, T. Gumbsch, and K. Borgwardt, *Neural persistence: A complexity measure for deep neural networks using algebraic topology*, International Conference on Learning Representations, 2019.
72. Andrew W. Senior, Richard Evans, J. Jumper, J. Kirkpatrick, L. Sifre, T. Green, C. Qin, A. Žídek, A. Nelson, A. Bridgland, H. Penedones, S. Petersen, K. Simonyan, S. Crossan, P. Kohli, D. Jones, D. Silver, K. Kavukcuoglu, and D. Hassabis, *Improved protein structure prediction using potentials from deep learning*, Nature **577** (2020), no. 7792, 706–710.
73. N. Shirish Keskar, D. Mudigere, J. Nocedal, M. Smelyanskiy, and P. Tang, *On Large-Batch Training for Deep Learning: Generalization Gap and Sharp Minima*, arXiv e-prints (2016), arXiv:1609.04836.
74. S. L. Smith and Q. V. Le, *A bayesian perspective on generalization and stochastic gradient descent*, International Conference on Learning Representations, 2018.
75. M. Talagrand, *Concentration of measure and isoperimetric inequalities in product spaces*, Publications Mathématiques de l'Institut des Hautes Études Scientifiques **81** (1995), 73–205.
76. I. Tyukin, A. Gorban, K. Sofeykov, and I. Romanenko, *Knowledge transfer between artificial intelligence systems*, Frontiers in Neurorobotics **12** (2018), 49.
77. F. Wang, H. Liu, D. Samaras, and C. Chen, *Topogan: A topology-aware generative adversarial network*, Computer Vision – ECCV 2020 (Cham) (Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, eds.), Springer International Publishing, 2020, pp. 118–136.
78. S. Watanabe and H. Yamana, *Topological measurement of deep neural networks using persistent homology*, Annals of Mathematics and Artificial Intelligence, 07 2021.
79. C. Wei and T. Ma, *Data-dependent sample complexity of deep neural networks via lipschitz augmentation*, Advances in Neural Information Processing Systems (H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, eds.), vol. 32, Curran Associates, Inc., 2019.
80. H. Weyl, *On the volume of tubes*, American Journal of Mathematics **61** (1939), no. 2, 461–472.
81. A. C. Wilson, R. Roelofs, M. Stern, N. Srebro, and B. Recht, *The marginal value of adaptive gradient methods in machine learning*, Advances in Neural Information Processing Systems (I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, eds.), vol. 30, Curran Associates, Inc., 2017.
82. H. Xu, Y. Li, W. Jin, and J. Tang, *Adversarial attacks and defenses: Frontiers, advances and practice*, Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (New York, NY, USA), KDD '20, Association for Computing Machinery, 2020, p. 3541–3542.
83. Y. Yang, R. Khanna, Y. Yu, A. Gholami, K. Keutzer, J. E. Gonzalez, K. Ramchandran, and M. W. Mahoney, *Boundary thickness and robustness in learning models*, Advances in Neural Information Processing Systems (H. Larochelle, M. Ranzato, R. Hadsell, M. F. Balcan, and H. Lin, eds.), vol. 33, Curran Associates, Inc., 2020, pp. 6223–6234.
84. R. Yousefzadeh and D. O'Leary, *Investigating decision boundaries of trained neural networks*, arXiv e-prints (2019), arXiv:1908.02802.
85. X. Yuan, P. He, Q. Zhu, and X. Li, *Adversarial examples: Attacks and defenses for deep learning*, IEEE Transactions on Neural Networks and Learning Systems **30** (2019), no. 9, 2805–2824.
86. C. Zhang, S. Bengio, . Hardt, B. Recht, and O. Vinyals, *Understanding deep learning requires rethinking generalization*, International Conference on Learning Representations, 2017.

Appendix A. Neighbourhood boundary volume on the entire parameter space

In this section we present results for **Bvol** measurements with varying dropout parameter on the final layer of fully connected and convolutional neural networks corresponding the results presented in Section 7.3 for **TrainBvol** and **LAdvBvol**. Unless otherwise stated all hyperparameter setting used are the same as those given at the beginning of Sections 7 and 7.3.

The **LAdvBvol** results over varying dropout rate are presented in Figure 15. In general most of the measurements are too noisy to be of practical use, with large overall between the standard deviation bars, suggesting that much of the structure of the decision boundary ate this large a scale away form the training data is unrelated to its behaviour near to the training data.

For the convolutions examples the **Bvol** measurement appear to somewhat decrease for higher dropout rates with the exception of the networks trained on CIFAR-10. In the fully connected examples the **Bvol** measurement appear increase somewhat as the dropout rate increases with the exception of the networks trained on Fashion MNIST where size of **Bvol** rapidly decrease to almost zero. These observations suggest that in general the total size of the boundary in $[0,1]^n$ increases as the the network boundary has to become more complicated to accommodate the requirements of the increased dropout regularisation. However, for the network trained on Fashion MNIST the opposite happened and the networks vastly oversimplified to accommodate the dropout regularisation.

Appendix B. Intersections between boundary manifolds of pairs of classes

Following the discussion on tube formulas in Section 5 we consider here neighbourhood boundary volume measurements using targeted FGSM adversarial attacks (equation (3)) in order to analyse the effective size of intersection between decision boundary manifolds separating pairs of classification regions. The main observation that we verify empirically here is that provided ε is chosen small enough, the ε-neighbourhood boundary volume of the co-dimension 2 intersection of the decision boundaries separating different pairs of classes is negligible compared to the total boundary volume. Therefore in this case for measurement purposes the ε-neighbourhood boundary volume is effectively the volume of a tubular neighbourhood around a co-dimension 2 manifold. Our results also allow us to compare the effect on neighborhood boundary volume measurements of the stronger targeted FGSM adversarial attack to those using the untargeted FGSM (equation (2)) used to obtain our results in Section 7.3. Unless otherwise stated all hyperparameter setting used are the same as those given at the beginning of Section 7 and no regularisation was applied during the training of the networks in this section.

Given a point $x \in \mathbb{R}^n$ in the input space of a network N, if a targeted adversarial attacks for k or more labels $i = 1, \ldots, m$ and $i \neq p(N(x))$ succeed in constructing a class i adversarial example $A_i(x)$ within ε of x, then x lies within ε of decision boundaries of at least k distinct classes. Similarly to the definition of $\mathrm{Bvol}_\varepsilon(N,\mathcal{U})$ in Section 6.1, assuming A_i is a adversarial attack method used to obtain adversarial examples at a distance as near as possible to the decision boundary of label i, we may measure the k-intersection ε-neighbourhood volume of the decision boundary by setting

$$TA_\varepsilon(x) = |\{d(x, A_i(x)) \leq \varepsilon \mid 1 \leq i \leq m,\ i \neq p(N(x))\}|$$

then

$$\mathrm{Bvol}^k_\varepsilon(N,\mathcal{U}) = \mathbb{P}\left(k \leq TA_\varepsilon(x) \mid x \in \mathcal{U}\right). \tag{11}$$

As before a lower bound on this volume can be measured using Monte Carlo sapling in $\mathcal{U}$. Moreover by taking the sample spaces $\mathcal{U}$ considered in Section 6.2 using the definition of $\mathrm{Bvol}^k_\varepsilon(N,\mathcal{U})$ given in equation (11) instead of $\mathrm{Bvol}_\varepsilon(N,\mathcal{U})$, we obtain corresponding definitions of $\mathbf{Bvol}^k$, $\mathbf{TrainBvol}^k$ and $\mathbf{LAdvBvol}^k$.

Tables 6 and 7 present the results from experiments with $\mathrm{Bvol}^k_\varepsilon(N,\mathcal{U})$ on fully connected and convolutional neural networks respectively trained on MNIST, Fashion MNIST and CIFAR-10. In each case, the optimisation procedure either SGD or Adam is given along with the data set on which the network was trained. In both tables the fist column method, determines which sample space $\mathcal{U}$

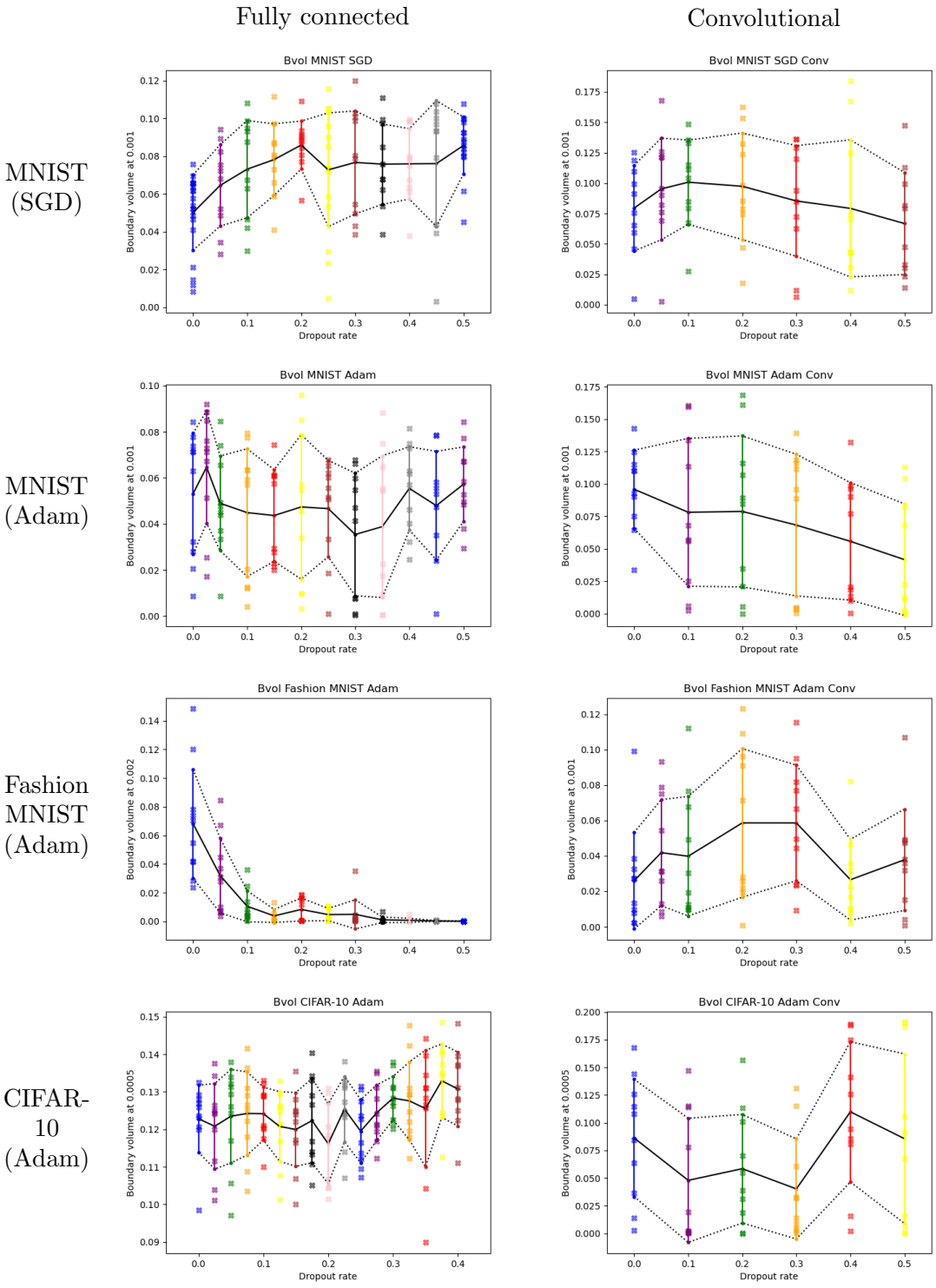

FIGURE 15. Global boundary volume measurements **Bvol** on $[0,1]^n$ for fully connected and convolutional neural networks trained on MNIST, Fashion MNIST and CIFAR-10. In general the measurements are too noisy to be practical useful with large overall between the standard deviation bars.

is used in the row. The second column gives the ε value, we use same values for each method and for

Method	ε	Untargeted volume	Intersection volume ≥ 1	Intersection volume ≥ 2	Intersection volume ≥ 3
MNIST (SGD)					
Bvol	0.001	0.05671	0.05693	0.00044	0
	0.005	0.27005	0.27225	0.01546	0.00054
	0.01	0.50467	0.50996	0.05623	0.00495
TrainBvol	0.0005	0.00219	0.00222	0.00002	0
	0.0025	0.012	0.01221	0.00063	0.00003
	0.005	0.02517	0.02571	0.00286	0.00038
LAdvBvol	0.0001	0.01728	0.01736	0.00007	0
	0.0005	0.08457	0.08518	0.00183	0.00006
	0.001	0.16794	0.17038	0.00759	0.00029
MNIST (Adam)					
Bvol	0.001	0.03868	0.03878	0.00064	0
	0.005	0.18669	0.1879	0.01416	0.00035
	0.01	0.36131	0.36447	0.03778	0.00205
TrainBvol	0.0005	0.0084	0.00845	0.00015	0
	0.0025	0.04052	0.04137	0.00421	0.00051
	0.005	0.08559	0.08712	0.01529	0.0032
LAdvBvol	0.0001	0.01874	0.0188	0.00014	0
	0.0005	0.09365	0.0948	0.00352	0.00008
	0.001	0.18191	0.18603	0.01439	0.00074
Fashion MNIST (Adam)					
Bvol	0.001	0.04252	0.04303	0.00161	0.00003
	0.005	0.19937	0.20356	0.03829	0.006
	0.01	0.36927	0.37593	0.09736	0.02579
TrainBvol	0.0005	0.02513	0.02538	0.0005	0.00003
	0.0025	0.11733	0.11965	0.01394	0.00121
	0.005	0.22245	0.22756	0.04786	0.00873
LAdvBvol	0.0001	0.01899	0.01909	0.00022	0
	0.0005	0.09434	0.09591	0.00421	0.0001
	0.001	0.1853	0.19001	0.01611	0.00069
CIFAR-10 (Adam)					
Bvol	0.001	0.17817	0.18413	0.01567	0.00101
	0.005	0.66255	0.71992	0.32805	0.1044
	0.01	0.90855	0.95437	0.76752	0.48579
TrainBvol	0.0005	0.04016	0.04071	0.00141	0.00001
	0.0025	0.18153	0.19079	0.03515	0.00496
	0.005	0.32964	0.34935	0.12075	0.03567
LAdvBvol	0.0001	0.03629	0.03632	0.00039	0
	0.0005	0.17516	0.17806	0.00896	0.00032
	0.001	0.20793	0.21185	0.0129	0.00047

TABLE 6. The ε-neighbourhood boundary volumes of intersection between manifolds of decision boundary separating pairs of class of fully connected networks.

each network. The third column give the ε-neighbourhood boundary volume measurements obtained

Method	ε	Untargeted volume	Intersection volume ≥ 1	Intersection volume ≥ 2	Intersection volume ≥ 3
	MNIST (SGD)				
Bvol	0.001	0.08157	0.082	0.0015	0
	0.005	0.37202	0.37584	0.03992	0.00002
	0.01	0.65104	0.65762	0.11456	0.0003
TrainBvol	0.0005	0.00012	0.00012	0	0
	0.0025	0.00094	0.00095	0.00001	0
	0.005	0.00235	0.00237	0.00006	0.00001
LAdvBvol	0.0001	0.01495	0.01496	0.00002	0
	0.0005	0.07441	0.07485	0.001	0
	0.001	0.14427	0.14551	0.00467	0.00007
	MNIST (Adam)				
Bvol	0.001	0.10504	0.10688	0.00627	0.00018
	0.005	0.44266	0.45329	0.12922	0.02275
	0.01	0.71518	0.72887	0.29169	0.08892
TrainBvol	0.0005	0.00042	0.00047	0.00001	0
	0.0025	0.00226	0.00226	0.00008	0
	0.005	0.00485	0.00489	0.00031	0.00001
LAdvBvol	0.0001	0.01595	0.01596	0.00005	0
	0.0005	0.07853	0.07904	0.00154	0.00004
	0.001	0.15479	0.15664	0.00653	0.00021
	Fashion MNIST (Adam)				
Bvol	0.001	0.01094	0.01099	0.00012	0
	0.005	0.06914	0.06979	0.00462	0.00011
	0.01	0.21612	0.21999	0.02023	0.00115
TrainBvol	0.0005	0.0361	0.03664	0.00225	0.00008
	0.0025	0.16828	0.17101	0.01097	0.00077
	0.005	0.3051	0.3094	0.06932	0.01287
LAdvBvol	0.0001	0.0199	0.02	0.00019	0
	0.0005	0.09916	0.10068	0.00524	0.00008
	0.001	0.19111	0.19587	0.0214	0.00143
	CIFAR-10 (Adam)				
Bvol	0.001	0.19441	0.19723	0.0121	0.00005
	0.005	0.75878	0.77226	0.22996	0.00383
	0.01	0.97722	0.98336	0.44805	0.03472
TrainBvol	0.0005	0.10809	0.11205	0.020325	0.00293
	0.0025	0.44748	0.471925	0.201795	0.07493
	0.005	0.6927	0.72011	0.45247	0.26346
LAdvBvol	0.0001	0.04441	0.04495	0.00117	0.00004
	0.0005	0.20575	0.21325	0.02924	0.0029
	0.001	0.37443	0.39481	0.10744	0.02002

TABLE 7. The ε-neighbourhood boundary volumes between intersections of manifolds of decision boundary separating pairs of class of convolutional neural networks.

with the untargeted FGSM adversarial attack and the final three columns give results of using the targeted FGSM to obtain $\mathbf{Bvol}_\varepsilon^k$, $\mathbf{TrainBvol}_\varepsilon^k$ and $\mathbf{LAdvBvol}_\varepsilon^k$ for $k = 1$, 2 and 3 respectively.

We first note that the intersection ε-neighbourhood boundary volume for $k \geq 1$ provides a measurement of **Bvol**, **TrainBvol** and **LAdvBvol** using a stronger adversarial attack then the untargeted FGSM. Comparing the third and fourth columns of both Tables 6 and 7, we see that in all rows the value in the fourth column are larger and that the difference in values of the columns is always a small value, negligible in comparison to the total values of either column changing at most the second significant figure. In particular this justifies that it would be computational inefficient to used the target FGSM adversarial attacks for neighbourhood boundary volume measurements more generally and suggest that the **Bvol**, **TrainBvol** and **LAdvBvol** are reasonably accurate at the ε values presented.

By comparing the fourth and fifth columns of Tables 6 and 7 we see that the fraction of the neighbourhood boundary volume contained in the Intersection volume for $k \geq 2$ increases as a percentage of the total as ε increases in size. Providing ε is chosen to be small enough the percentage of volume in an intersection volume of more than 1 boundary manifold of class pairs can be ensured to be negligible. We used these results to inform the ε hyperparameter choices in Table 2, in order to ensure that the interaction volume ≥ 2 should never be more than 10% of the total ε-neighbourhood boundary volume measurements.

It can also been seen from the final column of both Tables 6 and 7, that the ε-neighbourhood volume of the intersection of 3 or more is usually negligible in compassion to the the other measurements at the ε values we consider, with the exception of for example the **Bvol** value on CIRAR-10 at $\varepsilon = 0.01$ in Table 6, in which case even the untargeted neighbourhood boundary volume measurements contains over 90% of the volume making the measurement unreasonably large for piratical use.

Appendix C. Decision boundary curvature

Motivated by the curvature invariants present in the Weyl tube formula discussed in Section 5, we consider in this section curvature information available from neighbourhood boundary volume measurements over a range of ε values. However the only fact we make use of is that a flat hypersurface has a ε-neighbourhood boundary volume linear in ε.

More precisely, considering the neighbourhood boundary volume as a function of $\varepsilon \geq 0$ we would expect to see the derivative of this function decrease for lager ε. This is because in practice our adversarial attack method becomes less effective for larger ε values and because any self intersections in the tubular neighbourhood of the boundary also decrease the total volume as ε increases. However if we observe an increase of in the first derivative of the function this can only be explained by the presence of curvature in decision boundary. Unless otherwise stated all hyperparameter setting used in this section are the same as those given at the beginning of Section 7 and no regularisation was applied during the training of the networks in this section.

In order to restrict ourselves to measuring decision boundaries of manifolds and to gain more detailed information, we measure the neighbourhood boundary volume of the decision boundary between a given pair of class labels. In order to achieve this we again make use of a targeted adversarial attack method $A_i(x)$ aiming to construct an adversarial example for x close to the decision boundary of class i. To obtain our results we use the targeted FGSM adversarial attack as given in equation 3. Given a network N and a pair $i < j$ of class labels $i, j = 1, \ldots, m$ we define the ε-neighbourhood boundary volume of the pair (i, j), by

$$\begin{aligned}\mathrm{Bvol}_{\varepsilon}^{(i,j)}(N,\mathcal{U}) = \mathbb{P}(d(x, A_k(x)) \leq \varepsilon \mid x \in \mathcal{U} \text{ and } (p(N(x)) = i,\ k = j) \\ \text{or } (p(N(x)) = j,\ k = i))).\end{aligned} \tag{12}$$

A data set with m label therefore has $\binom{m}{2}$ possible $\mathrm{Bvol}_{\varepsilon}^{(i,j)}(N,\mathcal{U})$ measurements to make. Moreover by taking the sample spaces $\mathcal{U}$ considered in Section 6.2 using the definition of $\mathrm{Bvol}_{\varepsilon}^{(i,j)}(N,\mathcal{U})$ given in equation (12) instead of $\mathrm{Bvol}_{\varepsilon}(N,\mathcal{U})$ we obtain corresponding definitions of $\mathbf{Bvol}^{(i,j)}$, $\mathbf{TrainBvol}^{(i,j)}$ and $\mathbf{LAdvBvol}^{(i,j)}$.

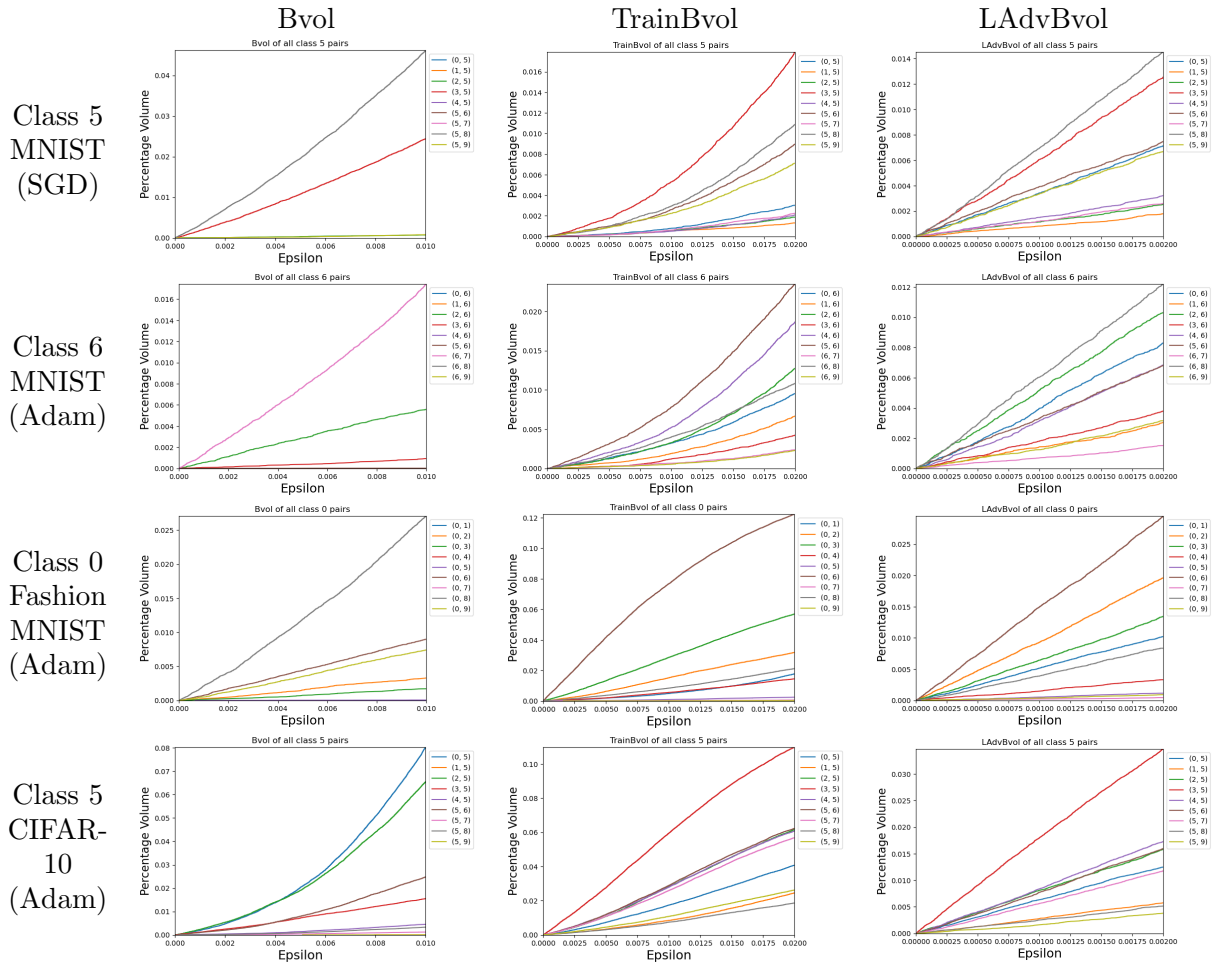

FIGURE 16. Neighbourhood boundary volume measurements of the boundary manifolds between a single class and all other classes trained on fully connect neural networks over varying ε-neighbourhood values. There is an increase in the slope of some reading hence evidence of curvature in some reading particularly in those for **TrainBvol** on MNIST.

Figures 16 and 17 show plot of $\mathbf{Bvol}^{(i,j)}$, $\mathbf{TrainBvol}^{(i,j)}$ and $\mathbf{LAdvBvol}^{(i,j)}$ for fully connected and convolutional neural networks respectively trained on MNIST, Fastion MNIST and CIFAR-10 data sets with $\varepsilon \in [0, 0.01]$, $\varepsilon \in [0, 0.02]$ and $\varepsilon \in [0, 0.002]$ for each of $\mathbf{Bvol}^{(i,j)}$, $\mathbf{TrainBvol}^{(i,j)}$ and $\mathbf{LAdvBvol}^{(i,j)}$ respectively and all examples presented are for pairs containing some specified class as one member of each pair.

We are interested in cases where there is an increase in the first derivative of these curves and the most striking feature of the results is that this is consistently the case for $\mathbf{TrainBvol}^{(i,j)}$ on all pairs form the MNIST data set only and not to any extent in any other case. The presence of curvature in $\mathbf{TrainBvol}^{(i,j)}$ for MNIST in the fully connected and convolutional cases with networks trained with both the SGD and Adam optimisation procedures. Moreover the rate of increase in the slope of the functions appears to be steeper in the convolution case suggesting that there might be more curvature present in the decision boundaries trained by the convolution neural networks on MNIST then the fully connected network decision boundaries.

The main other place we see the evidence of curvature is in the case of many for the $\mathbf{Bvol}^{(i,j)}$ particularly for networks trained on CIFAR-10 and the convolutional network trained on Fashion

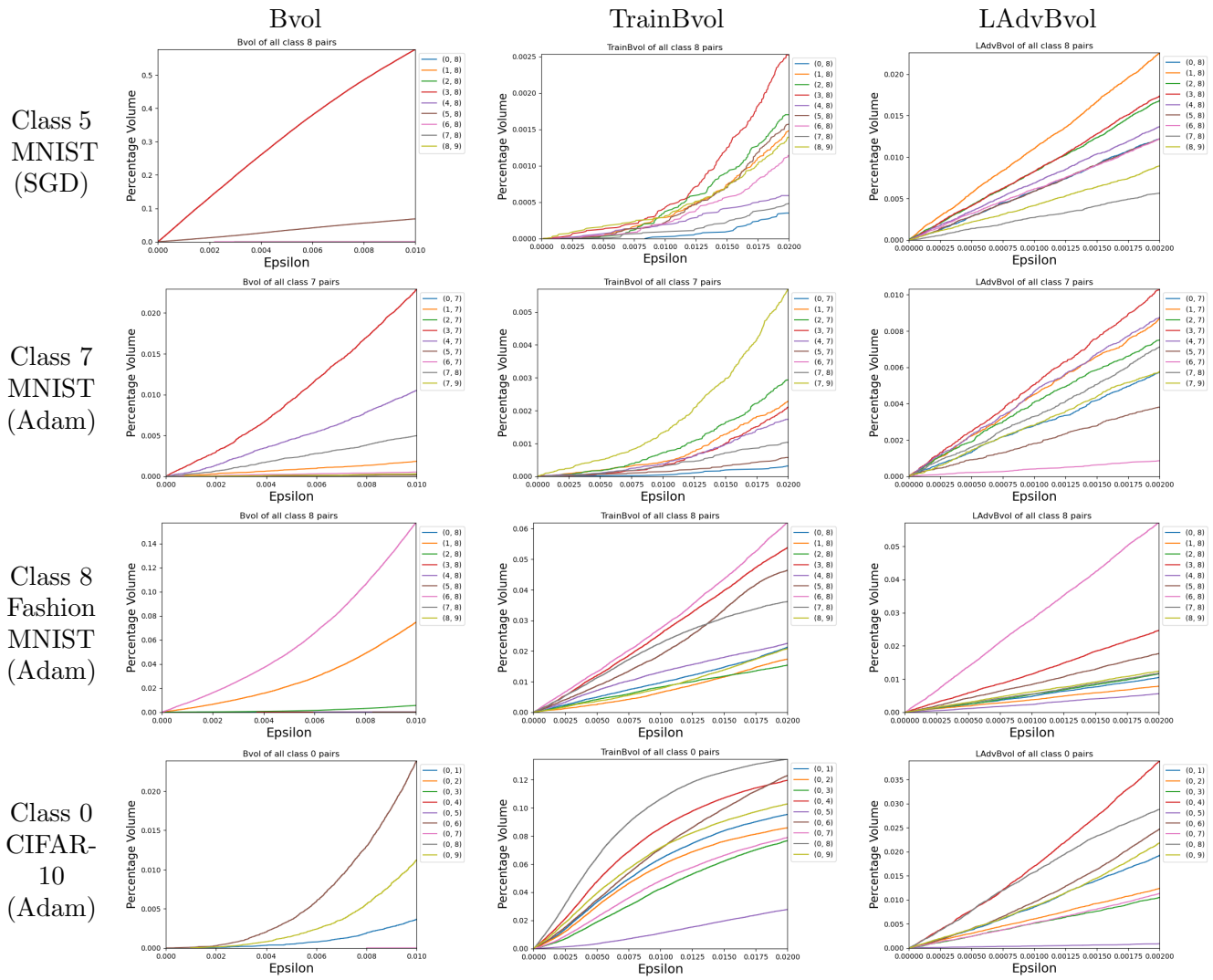

FIGURE 17. Neighbourhood boundary volume measurements of the boundary manifolds between a single class and all other classes trained on convolutional neural networks over varying ε-neighbourhood values. There is an increase in the slope of some reading hence evidence of curvature in some reading particularly in those for **TrainBvol** on MNIST.

MNIST suggesting it is a large sale feature in these cases. otherwise there is is only the occasional evidence of curvature between boundary class pairs featuring in $\mathbf{TrainBvol}^{(i,j)}$ for convolutional networks trained n CIFAR-10 and Fashion MNIST and for **LAdvBvol** in the convolutional case trained on CIFAR-10.

The fact that we observe curvature for the $\mathbf{Bvol}^{(i,j)}$ and $\mathbf{TrainBvol}^{(i,j)}$ measurements and almost never for the $\mathbf{LAdvBvol}^{(i,j)}$ measurements directly between the data manifolds suggests that the location of the curvature is around the edges of the data manifold suggests and that the decision boundary is generally flat directly between label classes.