Exact Visualization of Neural Network Geometry and Decision Boundary

Ahmed Imtiaz Humayun, *Rice University*
Randall Balestriero, *Meta AI Research*
Richard Baraniuk, *Rice University*

NEURAL INFORMATION PROCESSING SYSTEMS

Challenge: Current methods for visualizing the decision boundary of deep neural networks or the zero level sets of individual neurons require sampling, dichotomic search or gradients. Such methods are neither exact nor efficient.

Solution: We provide a fast and scalable exact visualization method for neural network geometry (level sets of neurons) and decision boundary of Deep Neural Networks with continuous piecewise linear (CPWL) non-linearities.

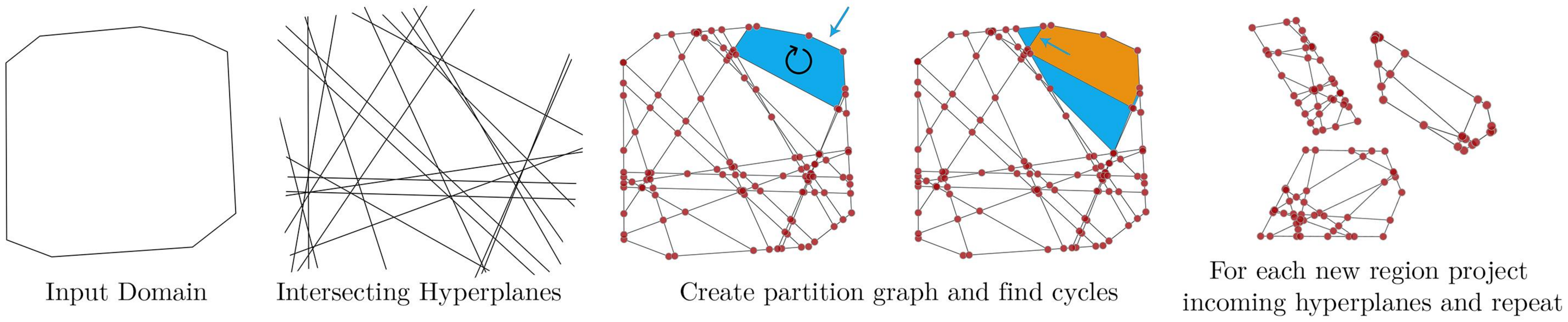
Input Domain
Intersecting Hyperplanes
Create partition graph and find cycles
For each new region project incoming hyperplanes and repeat

Fig 1: Given an input domain as a polygonal region and a set of hyperplanes, SplineCam (proposed method) first produces a graph using all the edge-hyperplane and hyperplane-hyperplane intersections. To find all the convex cycles in the graph, we select a boundary edge (blue arrow), do a breadth first search to find the shortest path through the graph between the two nodes and obtain the adjacent region (blue). While performing the traversal we enqueue the traversed edges for repetition. For each of the enqueued edges, we repeat the process to obtain the neighboring regions. Each non-boundary edge is allowed to be traversed twice, once from either direction. Once regions are found, we obtain a new set of hyperplanes corresponding to deeper layers and create partition graphs for each region.

Deep Neural Networks are Affine Spline Operators

Deep Neural Networks with CPWL non-linearities are Affine Spline Operators i.e., their input to output mapping is expressed as

$$\boldsymbol{S}(\boldsymbol{x}) = \sum_{\omega\in\Omega}(\boldsymbol{A}_\omega\boldsymbol{x}+\boldsymbol{b}_\omega)\mathbb{1}_{\{\boldsymbol{x}\in\omega\}}$$

Here, Ω is the partitioning induced by the network in the input space, $\boldsymbol{A}_\omega$ and $\boldsymbol{b}_\omega$ are affine parameters for the region ω and x are input vectors.

Visualizing 2D slices of a DNN input space partition analytically

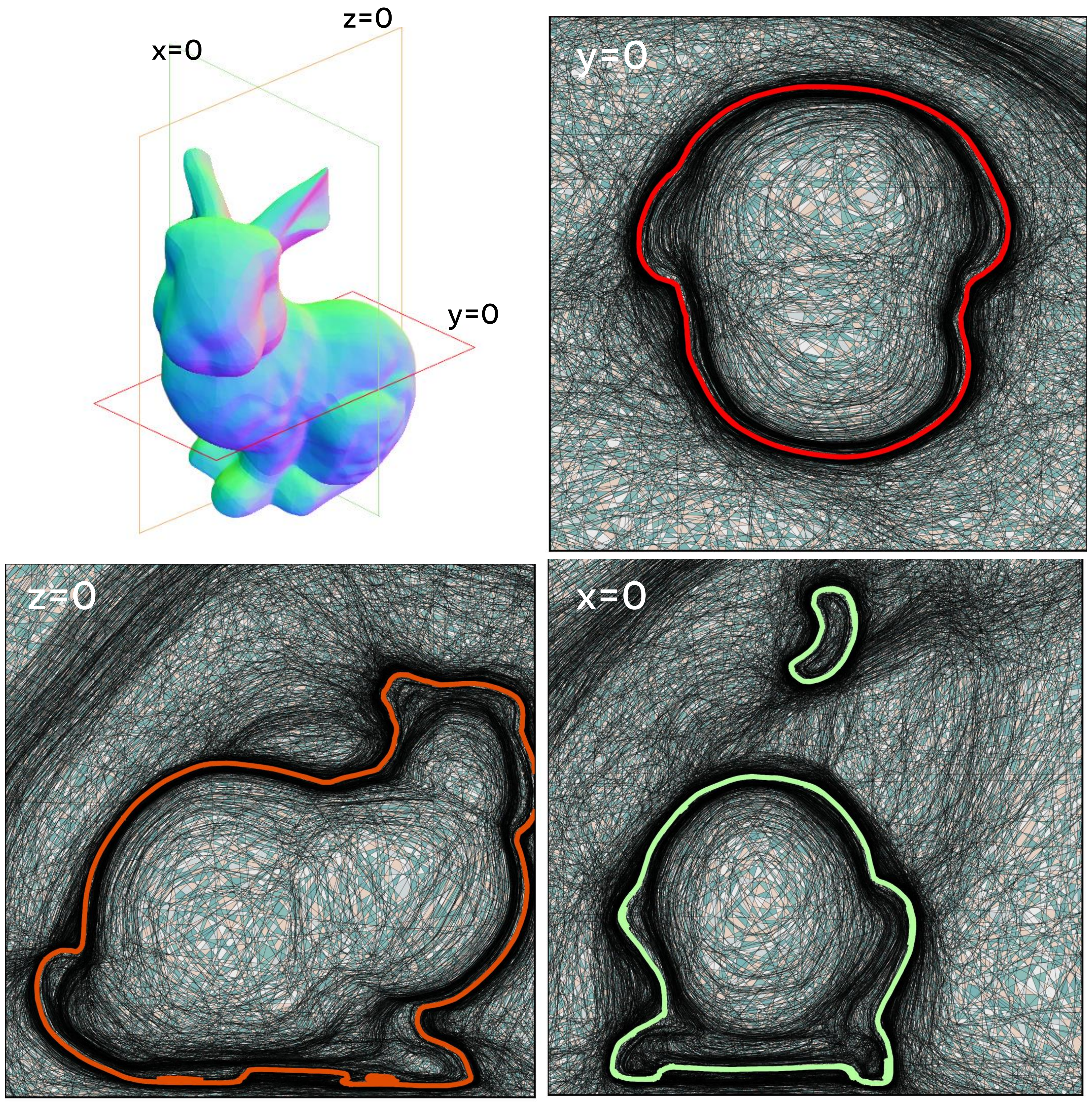

Fig 2: Neural networks with CPWL non-linearities can be exactly visualized as affine spline operators. Here we present exact visualization of the decision boundary and partition geometry of a 3D neural signed distance field (SDF). **(Top left)** Surface normals obtained from the learned signed distance field with annotations indicating slices used for visualization. For each of the slices **(Rest)**, we can see the spline partition geometry of the learned SDF- each contiguous line represents a neuron, on either side of which it gets activated/deactivated. Neurons from different depths of the network create a partitioning of the input space into 'linear regions'. Here the colored lines (red, orange, green) represent the decision boundary learned by the SDF. Note that while the final neuron obtains the decision boundary, many neurons place their boundaries close to the ground truth surface to obtain the final SDF representation.

Fast Computation of Partition Geometry

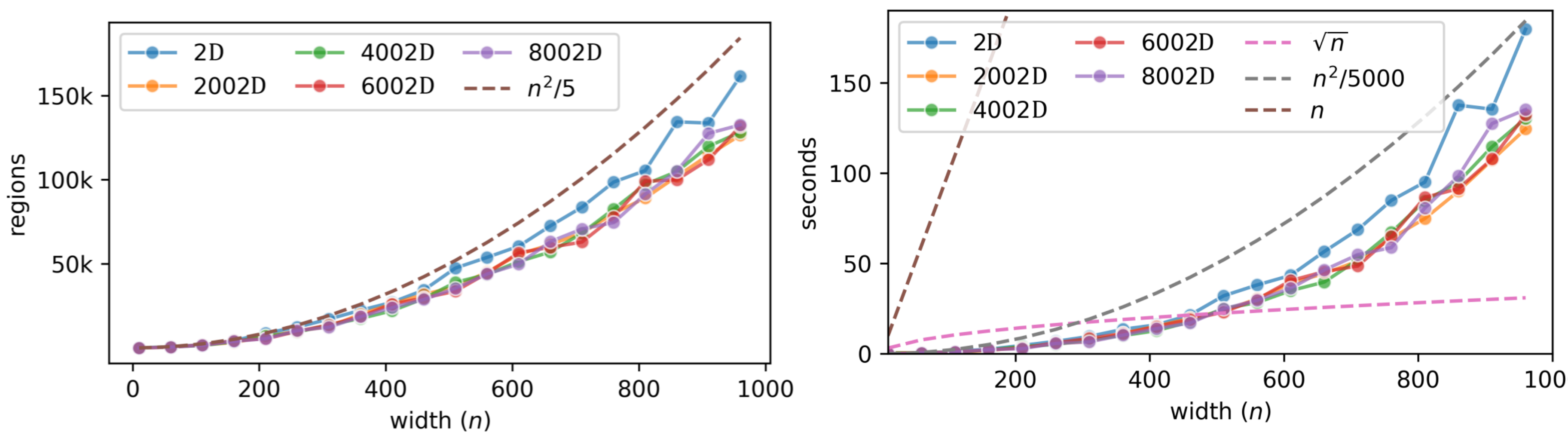

Fig 3: Growth of the number of regions with width **(Left)** and runtime of our algorithm **(right)** for a single layer randomly initialized ReLU neural network with variable width (n). Solid lines represent different input space dimensionality. For all the input dimensions, we take a randomly oriented square 2D domain centered on the origin and compute the input space partitioning on this domain. With increased input dimensionality, we see a slight reduction in number of regions and runtime.

Visualizing and sampling the decision boundary

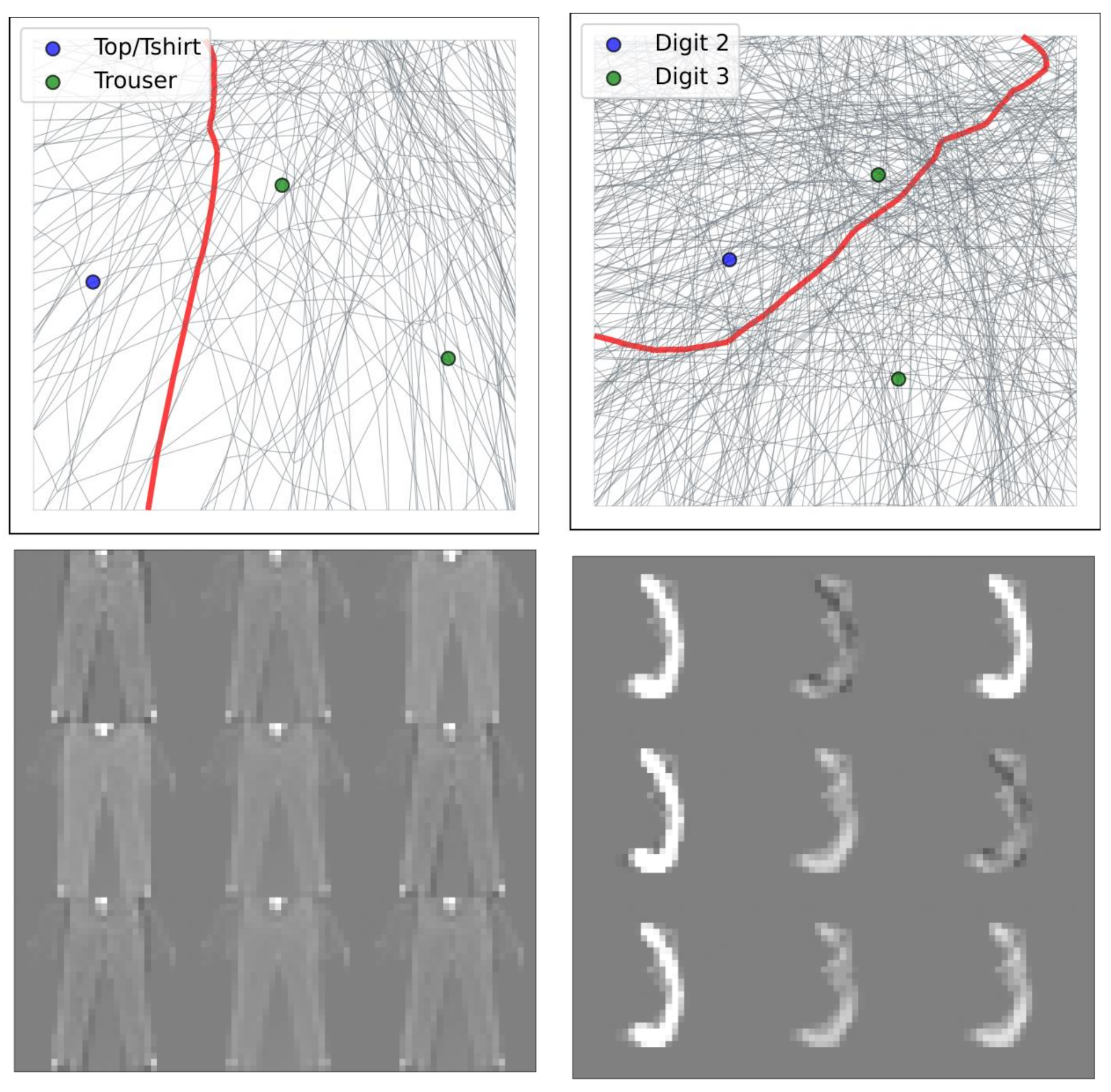

Fig 4: **(Top Left)** Decision boundary (red) visualization for an MLP with width 50 and depth 3, trained on fashion-MNIST. Black lines represent the spline partition of the network. Three correctly classified samples from training are used to define a 2D plane in the input space for visualization. **(Top Right)** Decision boundary and partition visualization of a convolutional neural network trained on MNIST, with two convolutional layers and one hidden fully connected layer of width 50. One of the digit 3 samples is misclassified by the network as digit 2. **(Bottom)** Random samples from the decision boundary.

Local characterization of the input space via partition statistics

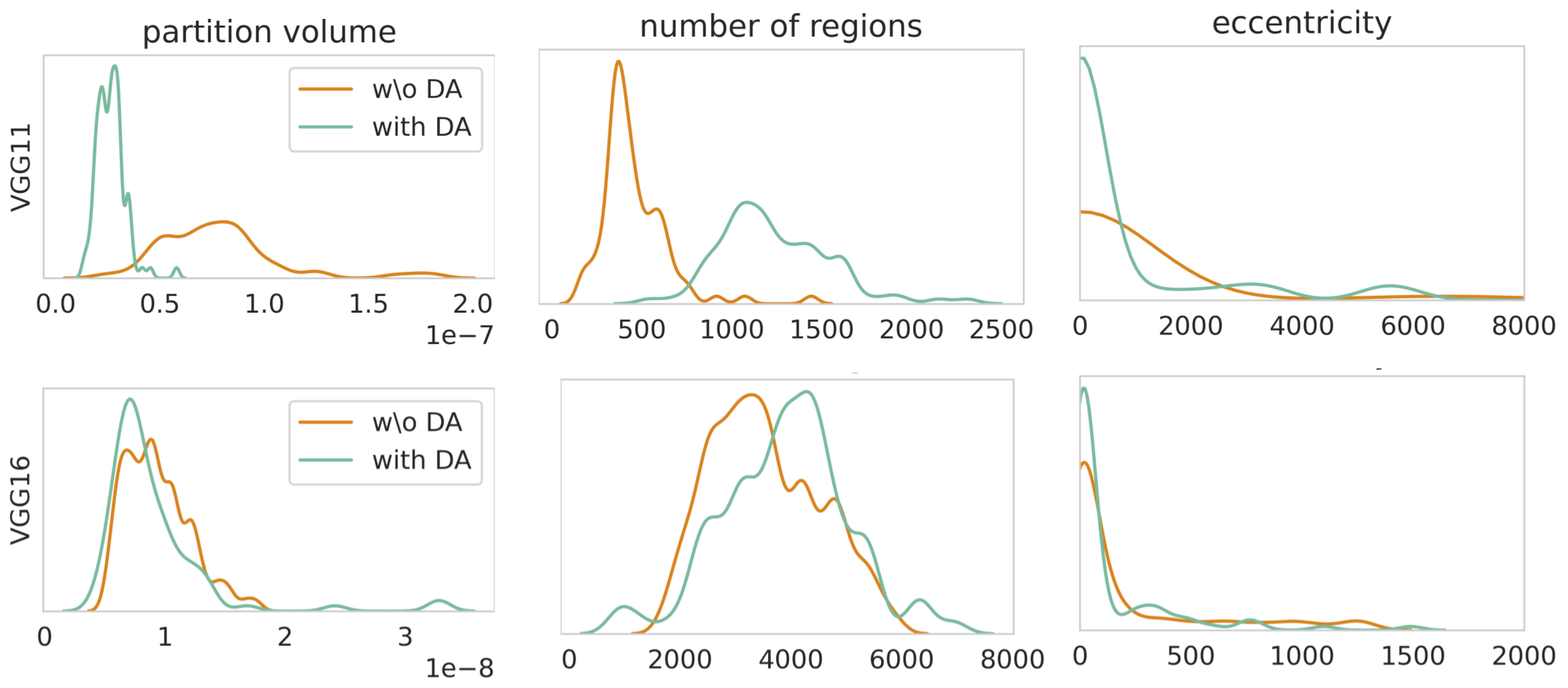

Fig 5:Average partition statistics around 90 TinyImageNet test samples with and without data augmentation (DA) training for VGG11 and VGG16. The average volume and number of regions are indicative of partition density whereas eccentricity is indicative of the shape of the regions. Both DA and increasing VGG depth, increases region density around test points.