Experimental Observations of the Topology of Convolutional Neural Network Activations*

Emilie Purvine,[1†] Davis Brown,[1] Brett Jefferson,[1] Cliff Joslyn,[1] Brenda Praggastis,[1] Archit Rathore,[2] Madelyn Shapiro,[1] Bei Wang,[2] Youjia Zhou[2]

[1] Pacific Northwest National Laboratory
[2] Scientific Computing and Imaging (SCI) Institute and School of Computing, University of Utah
{emilie.purvine, davis.brown, brett.jefferson, cliff.joslyn, brenda.praggastis, madelyn.shapiro}@pnnl.gov,
architrathore1@gmail.com, {beiwang, zhou325}@sci.utah.edu

Abstract

Topological data analysis (TDA) is a branch of computational mathematics, bridging algebraic topology and data science, that provides compact, noise-robust representations of complex structures. Deep neural networks (DNNs) learn millions of parameters associated with a series of transformations defined by the model architecture, resulting in high-dimensional, difficult-to-interpret internal representations of input data. As DNNs become more ubiquitous across multiple sectors of our society, there is increasing recognition that mathematical methods are needed to aid analysts, researchers, and practitioners in understanding and interpreting how these models' internal representations relate to the final classification. In this paper, we apply cutting edge techniques from TDA with the goal of gaining insight into the interpretability of convolutional neural networks used for image classification. We use two common TDA approaches to explore several methods for modeling hidden-layer activations as high-dimensional point clouds, and provide experimental evidence that these point clouds capture valuable structural information about the model's process. First, we demonstrate that a distance metric based on persistent homology can be used to quantify meaningful differences between layers, and we discuss these distances in the broader context of existing representational similarity metrics for neural network interpretability. Second, we show that a mapper graph can provide semantic insight into how these models organize hierarchical class knowledge at each layer. These observations demonstrate that TDA is a useful tool to help deep learning practitioners unlock the hidden structures of their models.

Introduction

Convolutional neural networks (CNNs) are a class of deep learning (DL) models that have been widely used for image classification tasks with great success, but the reasoning behind their decisions is often difficult to determine. Recent work has established an active field of explainable DL to tackle this problem. There are tools that highlight areas of the images most influential to the classification (Selvaraju et al. 2017), or reconstruct idealized input images for each output class (Mahendran and Vedaldi 2015; Wei et al. 2015). There are even tools that try to impose human concepts on the DL model (Kim et al. 2018). The complexity and dependencies present within these trained models demand methods in explainable DL that can summarize complex data without losing critical structures, producing features of internal representations that are both stable and persistent with respect to changing inputs and noise, and significant with respect to representing meaningful features of the input data.

Topological data analysis (TDA) is an emerging field that bridges algebraic topology and computational data science. One of the hallmarks of TDA is its ability to provide compact, noise-robust representations of complex structures within data. These are exactly the kind of representations that are needed in the DL space where different training runs or noisy input data may result in slightly different hidden activations but in no change in the ultimate classification. In other well-documented cases, slight changes in input, perhaps unseen to the human eye, result in misclassifications. We believe TDA can help us understand these cases as well by recognizing changes in the compact representations of the complex structures of hidden activation layers.

In this paper, we build upon others' recent work in using TDA to understand various aspects of machine learning (ML) and DL models. We provide experimental results that show how a topological viewpoint of hidden-layer activations can summarize and compare the complex structures within them and how the conclusions align with our human understanding of the image classification task. We begin by providing some preliminaries on CNNs and TDA and summarize related work. We then show our experiments, which use two tools from TDA: persistent homology and mapper. Finally, we conclude with a discussion and our directions for future work.

Preliminaries

Convolutional Neural Networks

CNNs are a type of deep neural network that respects the spatial information existing in the input data. They use shared weights to provide translation invariant measures of

*Version including technical appendix can be found on ArXiv.
†Primary; all other authors listed in alphabetical order

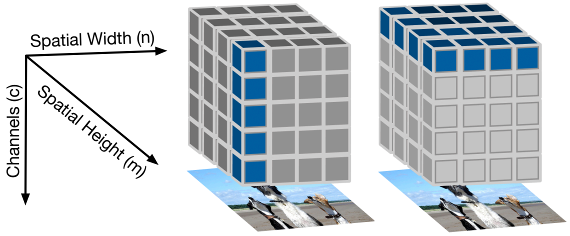

Figure 1: Visualization of spatial activation (middle) and channel activation (right) within an activation tensor.

correlation across an input, which makes them ideal for image classification tasks, where objects requiring identification might be found anywhere in an image.

Mathematically, a trained neural network used for classification is best described as the composition of linear and non-linear *tensor maps* called *layers*, where a *tensor* is a multi-dimensional real-valued array. The input to a neural network is a tensor, and the output of the network is a probability vector indicating the likelihood the input belongs to each class. The intermediate outputs from each layer of the composition are called feature maps or *activation tensors*. Linear layers use tensor maps that respect element-wise addition and scalar multiplication, and can be either fully connected or convolutional.

Convolutional layers use cross correlation, also known as a sliding dot product, to map 3D tensors to 3D tensors. If the activation tensor from a convolutional layer has dimensions $c \times n \times m$, we say the tensor has c *channels* and nm *spatial dimensions*. Activation tensors may be sliced into spatial and channel activations, as shown in Figure 1, and then reshaped to obtain vector representations of their values.

Persistent Homology

One of the two topological tools that we use in our work is persistent homology (PH). At a high level, PH is a method for understanding the topological structure of a space that data are sampled from. We typically have access only to the sample, in the form of a point cloud, and use PH to infer large-scale structures of the unknown underlying space. Here, we provide a brief overview of PH and point readers to Edelsbrunner and Harer (2008); Ghrist (2008) for more details.

The theoretical basis for persistent homology lies in the concept of *homology* from algebraic topology. Given a topological object, e.g., a surface or the geometric realization of a *simplicial complex* (a collection of finite sets, Σ, such that if $\tau \subset \sigma$ and $\sigma \in \Sigma$ then $\tau \in \Sigma$), its homology is an algebraic representation of its cycles in all dimensions. In dimensions 0, 1, and 2, the cycles have simple interpretations as connected components, loops, and bubbles, respectively. Higher dimensional interpretations exist but are less intuitive.

Given a single point cloud, $S \subset \mathbb{R}^k$, we can construct a family of associated simplicial complexes on which to compute homology. In this paper, we use the Vietoris-Rips (VR) complex given a scale parameter ϵ, $VR(S, \epsilon)$. In short, $VR(S, \epsilon)$ is a simplicial complex where each collection of points in S whose pairwise distances are all at most ϵ is a set in $VR(S, \epsilon)$. We show examples of two VR complexes (just the 1-skeleton, the pairwise edges) of the same point cloud at two scale parameters in Figure 2.

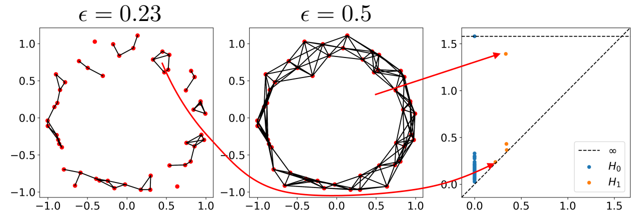

Figure 2: VR complexes at two ϵ values and the PD of the point cloud. In the PD, orange (resp. blue) points represent 1D (resp. 0D) persistent features. Points on the horizontal dotted line are those that persist through the entire filtration and have no death threshold.

Finally, we can describe the motivation and concept of PH. A single point cloud technically is a simplicial complex, but it is not interesting homologically. Whereas constructing a VR complex at a single scale parameter does provide an interesting topological object, it does not capture the multiscale phenomena of the data. PH is a method that considers all VR scale parameters together to identify at which ϵ a cycle is first seen (is "born") and at which ϵ' the cycle is fully triangulated ("dies"). This set of birth and death values for a sequence of simplicial complexes of a given point cloud provides a topological fingerprint for a point cloud often summarized in a *persistence diagram* (PD) as a set of (b, d) coordinates. Figure 2 also shows the point cloud's PD from the full sequence of ϵ thresholds.

PDs form a metric space under a variety of distance metrics. In this paper, we will use *sliced Wasserstein (SW) distance* introduced by Carrière, Cuturi, and Oudot (2017). Given two PDs, the SW distance is computed by integrating the Wasserstein distances for all projections of the PD onto lines through the origin at different angles.

Mapper

The mapper algorithm was first introduced by Singh, Memoli, and Carlsson (2007). It is rooted in the idea of "partial clustering of the data guided by a set of functions defined on the data" (2007). On a high level, the mapper graph captures the global structure of the data.

Let $S \subset \mathbb{R}^k$ be a high-dimensional point cloud. A *cover* of S is a set of open sets in $\mathbb{R}^k$, $\mathcal{U} = \{U_i\}$ such that $S \subset \cup_i U_i$. In the classic mapper construction, obtaining a cover of S is guided by a set of scalar functions defined on S, referred to as *filter functions*. For simplicity, we describe the mapper construction using a single filter function $f : S \to \mathbb{R}$. Given a cover $\mathcal{V} = \{V_\ell\}$ of $f(S) \subset \mathbb{R}$ where $f(S) \subseteq \cup_\ell V_\ell$, we can obtain a cover $\mathcal{U}$ of S by considering as cover elements the clusters (for a choice of clustering algorithm) induced by $f^{-1}(V_\ell)$ for each V_ℓ.

Then, the 1D *nerve* of any cover $\mathcal{U}$ is a graph and is denoted as $\mathcal{N}_1(\mathcal{U})$. Each node i in $\mathcal{N}_1(\mathcal{U})$ represents a cover

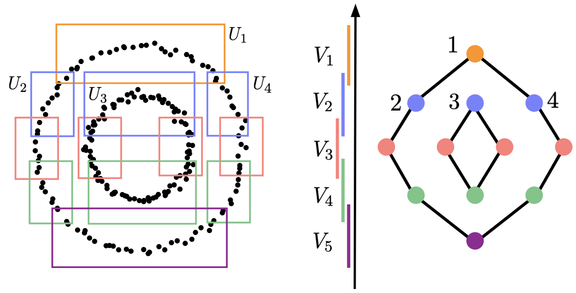

Figure 3: A mapper graph of a point cloud containing two nested circles.

element U_i, and there is an edge between nodes i and j if $U_i \cap U_j$ is non-empty. If $\mathcal{U}$ is constructed as above, from a clustering of preimages of a filter function f, then its 1D nerve, denoted as $\mathcal{M} = \mathcal{M}(S, f) := \mathcal{N}_1(\mathcal{U})$, is the *mapper graph* of (S, f).

Consider the point cloud in Figure 3 as an example containing two nested circles. It is equipped with a height function $f : S \to \mathbb{R}$. A cover $\mathcal{V} = \{V_1, \cdots, V_5\}$ of $f(S)$ is formed by five intervals (see Figure 3 middle). For each ℓ $(1 \leq \ell \leq 5)$, $f^{-1}(V_\ell)$ induces a number of clusters that are subsets of S. Such clusters form the elements of a cover $\mathcal{U}$ of S. As shown in Figure 3 (left), the cover elements of $\mathcal{U}$ are contained within the 12 rectangles on the plane. The mapper graph of S is shown in Figure 3c. For instance, cover $f^{-1}(V_1)$ induces a single cover element U_1 of S, and it becomes node 1 in the mapper graph of S. $f^{-1}(V_2)$ induces 3 cover elements U_2, U_3 and U_4, which become nodes 2, 3 and 4. Since $U_1 \cap U_2 \neq \emptyset$, an edge exists between node 1 and node 2. The two circular structures in Figure 3 (left) are captured by the mapper graph in Figure 3 (right).

Related Work

The value of TDA to organize, understand, and interpret various aspects of ML and DL models has been recognized in several current research directions. Much of this research has focused on model parameters, structure, and weights. Guss and Salakhutdinov (2018) examine model architecture selection by defining the "topological capacity" of networks, or the ability for the network to capture the true topological complexity of the data. They explore the learnability of model architectures in the face of increasing topological complexity of data. Gabrielsson and Carlsson (2019) build the mapper graph of a point cloud of learned weights from convolutional layers within a simple CNN and find that the weights of different CNN model architectures trained on the same data set have topological similarities. "Neural persistence", developed by Rieck et al. (2019), is a topological measure of complexity of a fully connected deep neural network that depends on learned weights and network connectivity. They find networks that use best practices such as dropout and batch normalization have statistically higher neural persistence, and define a stopping criterion to speedup the training of such a network.

Other studies, like that of Wheeler, Bouza, and Bubenik (2021) use TDA to study activation tensors of simple multilayer perceptron networks to discover how the topological complexity, as measured by a property of persistence landscapes, changes through the layers. Gebhart, Schrater, and Hylton (2019); Lacombe, Ike, and Umeda (2021) investigate the topology of neural networks via "activation graphs," which model the natural graphical structure of the network. Finally, most closely related to our work is that of Rathore et al. (2021), which describes TopoAct, a visual platform to explore the organizational principle behind neuron activations. TopoAct displays the mapper graph of activation vectors for a single layer at a time in a CNN to show how the model organizes its knowledge via the branching structures. The authors consider a point cloud formed by randomly sampling a single spatial activation in a given layer for each image in a corpus. We extend this work by using a larger and more data-driven sample of spatial activations to build our mapper graphs, quantifying the intuition of "pure" and "mixed" mapper nodes, considering the effect of noisy input on the resulting graph, and showing how our results generalize to multiple common model architectures.

Point Cloud Summaries of Activations

Following the approach of Rathore et al. (2021), we model each convolutional layer of a CNN as an $Np \times c$ point cloud by sampling p spatial activation vectors from the $c \times n \times m$ activation tensors produced by N images in a dataset. This gives us a collection of point clouds that can be used to study the evolution of the activation space (i.e., the space of spatial activations), as the complexity of features learned by each layer increases as we move deeper into the model (Zhou et al. 2015; Olah et al. 2020). We introduce several data-driven sampling methods with the goal of improving upon the quality of the sampled point cloud representation.

Random and full activations. In our mapper experiments, for a fixed layer, we construct a high-dimensional point cloud by *randomly sampling* a single ($p = 1$) spatial activation from each input image, as in Rathore et al. (2021). We additionally experiment with *full activation sampling* ($p = nm$) by including all spatial activations of a given layer for each image in the point cloud construction.

Top l^2-norm activations. In our PH experiments, for a fixed layer we construct a point cloud with *top l^2-norm sampling* ($p = 1$) by selecting the spatial activation with the strongest l^2-norm from each image.

Foreground and background activations. For a fixed convolutional layer, each spatial position in the activation tensor can be traced back to its *effective receptive field*, which is the region of the input image that the network has "seen" via contributions from previous layers. Naturally, each spatial activation corresponds to the subset of the foreground and background pixels in its effective receptive field. To investigate how foreground and background information of an input image manifests in the activation space, we first use `cv2.grabCut` from the OpenCV library (Bradski 2000) to perform image segmentation and identify the foreground and background pixels in the images. We then assign

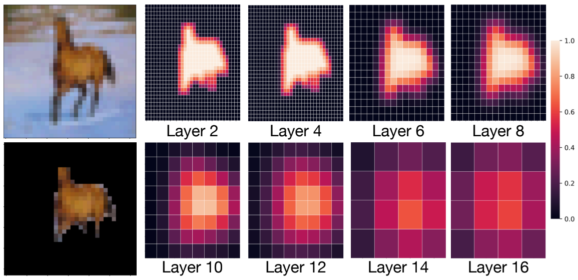

Figure 4: Spatial positions whose effective receptive field contains primarily foreground pixels are highly weighted in foreground sampling.

a weight to each spatial activation according to the number of foreground or background pixels in its effective receptive field, as illustrated in Figure 4. The spatial activations with the greatest weight are selected to represent each image in the point cloud construction, referred to as *foreground* or *background sampling*. In our mapper experiments, we study the "top p" foreground and background activations for $p = 1$ and $p = 5$.

Reproducibility Details

The following two sections outline our experiments using PH and mapper graphs to study the standard benchmark dataset CIFAR-10 (Krizhevsky and Hinton 2009) on a ResNet-18 architecture (He et al. 2016). We perform standard preprocessing to normalize the images by the mean and variance from the full training set. Code for the models and additional details regarding the dataset, as well as the parameters and computing infrastructure specific to each set of experiments, are provided in the arXiv technical appendix.

Experiments with PH

Using the top l^2-norm sampling method, we construct point cloud summaries of activations from the CIFAR-10 dataset on a ResNet-18 model to study the PH of the activation space. The SW distance between PDs of these point cloud summaries — which we will refer to from now on as the *SW distance between layers* — proves to be an interesting topological metric for capturing similarity between layers; it exhibits some of the fundamental qualities of strong representation similarity metrics for neural networks but fails to be sensitive to others (Ding, Denain, and Steinhardt 2021).

Relationships Between Layers

In Figure 5, we observe a grid-like pattern in the SW distances between layers of ResNet-18 similar to the results found in Kornblith et al. (2019), which the authors attribute to the residual architecture. This observation supports our belief that meaningful qualities of the model and its architecture can be uncovered by studying the topology of the activation space with PH.

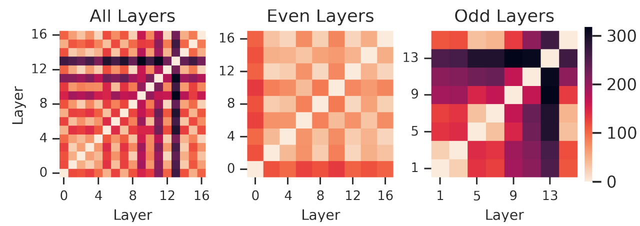

Figure 5: SW distances between convolutional layers of ResNet-18; results averaged over 10 random batches of 1000 CIFAR-10 test set images (CV < 0.17).

Representation Similarity Metrics & Intuitive Tests

Metrics such as canonical correlation analysis (CCA) (Morcos, Raghu, and Bengio 2018; Raghu et al. 2017), centered kernel alignment (CKA) (Kornblith et al. 2019), and orthogonal Procrustes distance (Ding, Denain, and Steinhardt 2021) provide dissimilarity measures that can be used to compare layers of neural networks. Recent work has demonstrated the value of topological approaches to representation similarity such as Representation Topology Divergence (Barannikov et al. 2022). These methods operate on an $N \times cnm$ matrix representation of a convolutional layer, where the $c \times n \times m$ activation tensors produced by each of the N inputs from the dataset are normalized and unfolded into vectors in $\mathbb{R}^{cnm}$. Here we note this as a key difference from our $N \times c$ point cloud representation obtained through top l^2-norm sampling but leave a more thorough comparison to future work.

We apply the intuitive specificity and sensitivity tests outlined by Ding, Denain, and Steinhardt (2021) to probe the utility of the SW distance between layers as a representation similarity metric for neural networks. In comparison to the intuitive test results shown for CCA, CKA, and orthogonal Procrustes distance from Ding, Denain, and Steinhardt (2021), this metric exhibits some non-standard behavior, for which we provide some speculative explanations but further work is needed to fully understand such a metric.

Specificity. To measure the impact of model initialization seed on the SW distance between layers, we trained 100 ResNet-18 models with different initialization seeds on CIFAR-10, and constructed top l^2-norm point cloud representations of the layers of each model from $N = 1000$ test set images. Figure 6 shows SW distances for two of the models "A" and "B", comparing pairs of layers in Model A (left) as well as pairs of layers between Model A and Model B (right). We find that variation in model seed has almost no impact on the SW distances, as shown by the near-identical heatmaps and highlighted for layer 9 (bottom row). The internal and cross-model SW distances relative to Model A layer 9 are highly correlated, with $\rho \approx 0.907$ computed by averaging correlation with fixed Model A over the 99 remaining randomly initialized models as Model B. Averaging internal and cross-model correlation relative to each layer of Model A, we find $\rho \approx 0.910$. We conclude that SW distance between layers is highly specific and robust to variation in initialization seed.

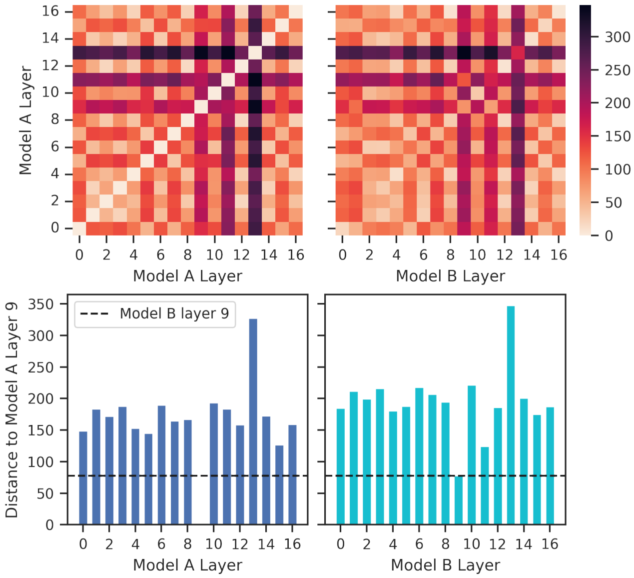

Figure 6: Intuitive specificity test of SW distance between convolutional layers of two ResNet-18 models initialized with different random seeds, for 1000 CIFAR-10 test set images.

Sensitivity. A representation similarity metric should be robust to noise without losing sensitivity to significant alterations. We apply the intuitive sensitivity test of Ding, Denain, and Steinhardt (2021) by taking the SW distance between each layer and its low-rank approximations as we delete principal components from the $N \times c$ point cloud. The SW distance to the corresponding layer in another model is averaged over the remaining 99 randomly initialized models to compute a baseline SW distance for each layer. This baseline defines a threshold of *detectable* SW distance, above which distance cannot be solely attributed to different initialization. In Figure 7, we see the sensitivity of this metric is heavily dependent on layer depth.

Experiments with Mapper Graphs

In this section, we explore how the topology of the activation space changes across layers by constructing mapper graphs from spatial activations from $N = 50\text{k}$ CIFAR-10 training images on a ResNet-18 model. The mapper graph filter function is the l^2-norm of each spatial activation. We employ and extend *MapperInteractive* (Zhou et al. 2021), an open-source web-based toolbox for analyzing and visualizing high-dimensional point cloud data via its mapper graph. Because of the visual nature of mapper graphs, our experiments will largely be evaluated by exploring and comparing the *qualitative* properties of the visualizations rather than quantitative comparisons of structures. The exception will be our purity measures, introduced in a later subsection.

Random and Full Activations

In Figure 8, we compare the mapper graphs generated from a point cloud of random activations ($50\text{k} \times c$) against those

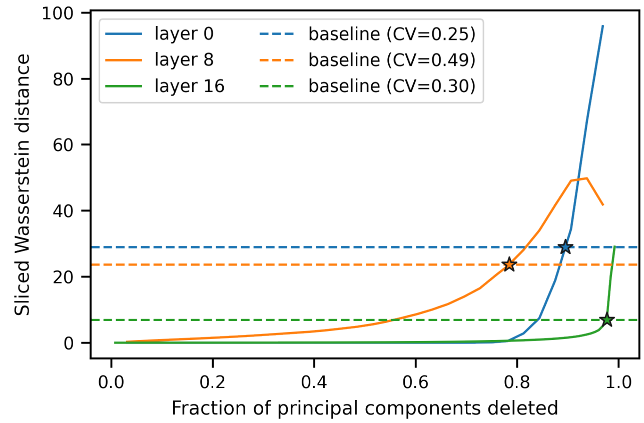

Figure 7: Intuitive sensitivity test of SW distance for the first (0), middle (8), and last (16) convolutional layers of ResNet-18, for 1000 CIFAR-10 test set images.

generated from the full activations ($50\text{k} \cdot nm \times c$) across different convolutional layers, where c is the number of dimensions of each activation, and nm is the total number of spatial activation vectors per image. The glyph for each node of the mapper graph is a pie chart showing the composition of class labels in that node. It can be seen that at layer 16, the mapper graphs of the random and full activations clearly capture the separation among class labels; there is a central region in the graph where nodes with mixed labels (with lower l^2-norm) separate out into branches with single labels (with higher l^2-norm). As we move toward earlier layers, the ability of the mapper graphs to show class separation gradually deteriorates. In addition, both random and full activations show similar bifurcation patterns, indicating robustness with respect to the sampled activations.

Foreground and Background Activations

Next, we study whether branching structures emerge at earlier layers if we use top foreground or background activations. Figure 9 shows the evolution of mapper graphs using the foreground and background activations across layers. We observe that the mapper graph of foreground activations at layer 15 already shows notable class bifurcations. Such early separations are less obvious for random and full activations. The mapper graphs of background activations also show clear class separations at layer 15 and 16, indicating that background pixels likely play an important role in class separation as well. Mapper graphs for the top 5 foreground and background activations are provided, along with similar observations in the technical appendix.

Activations with Gaussian Noise

To explore the stability of mapper graphs to noise in the input data, we injected pixel-wise Gaussian noise to all 50k images with different standard deviations (σ). Examples of how the images change as the standard deviation increases are shown in Figure 10, and the corresponding mapper graphs at layer 16 are shown in Figure 11. It can be seen

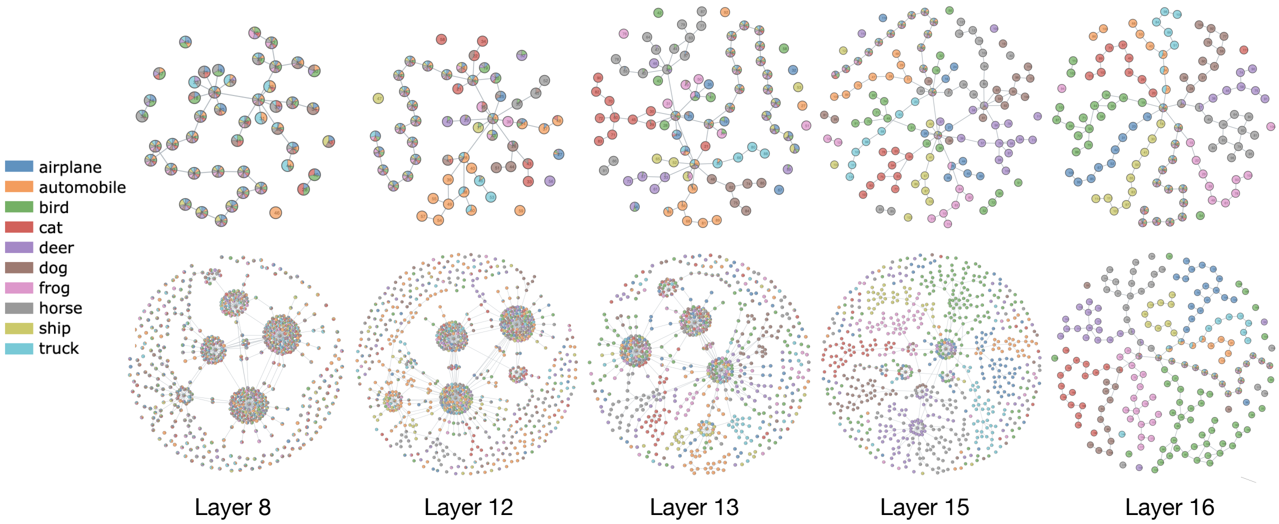

Figure 8: Mapper graphs from random (top) and full (bottom) activations from ResNet-18 using the CIFAR-10 dataset.

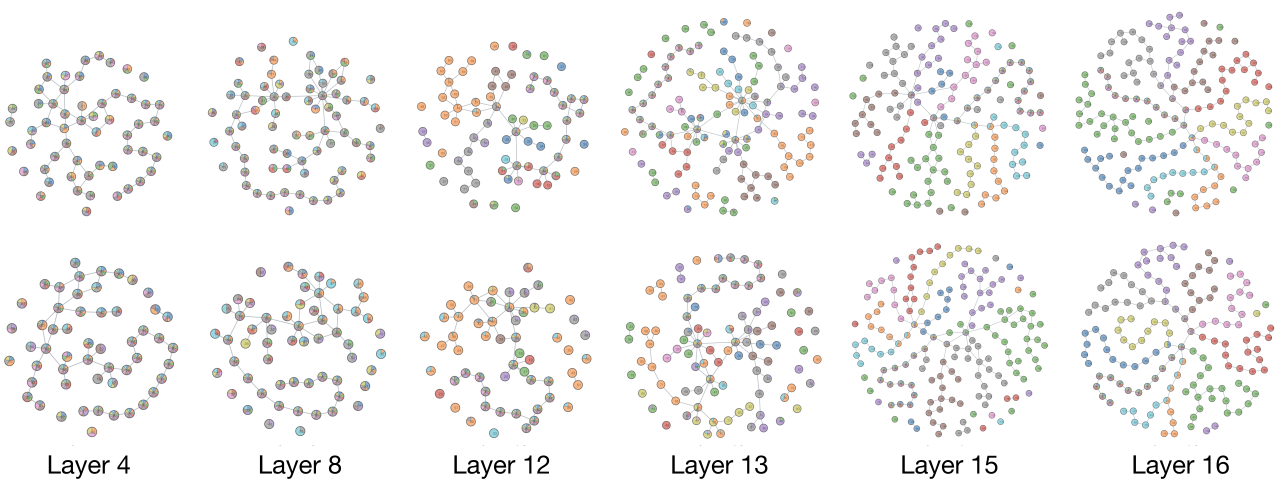

Figure 9: Mapper graphs generated from the foreground (top) and background (bottom) activations with the largest weights.

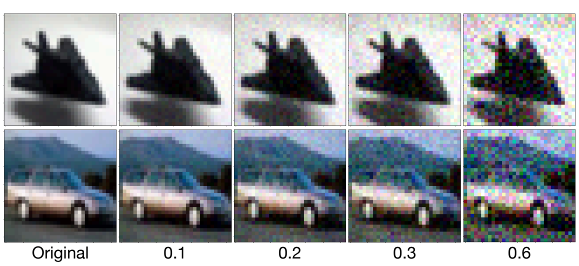

Figure 10: Examples of CIFAR-10 images with perturbations. Column 1 contains the original, and columns 2-4 contain images perturbed with different standard deviations.

that the mapper graphs are stable for small perturbations ($\sigma = 0.1$). As σ increases, mapper graphs illustrate that the model's ability to differentiate different classes decreases. This observation aligns with the intuition that increasing the noise level will decrease prediction accuracy.

Mapper Graph Purity Measures

For an image classification task, each point (i.e., a spatial activation) $x \in S$ is assigned a class label (inherited from the class label of its corresponding input image). We introduce three quantitative measures to quantify how well a mapper graph of the activation space separates the points from different classes.

Node-wise purity. Given a mapper graph $\mathcal{M}$, the node-wise purity of a node i is defined as $\alpha_i = \frac{1}{c_i}$, where c_i is the number of class labels in node i: the more classes in node i, the less pure node i is. Figure 12 (bottom) shows the node-wise purity of mapper graphs for foreground (top 1 and 5), random, and full activations at a variety of layers (aligning with the layers seen in Figures 8 and 9). We observe that node-wise purity is larger in deeper layers, indicating that

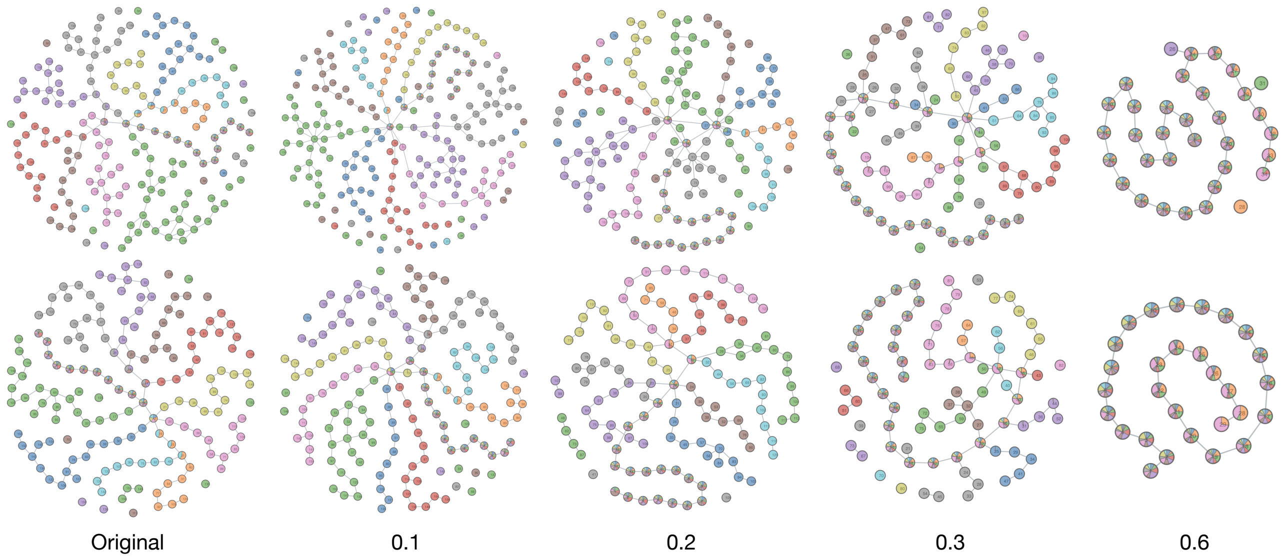

Figure 11: Perturbed mapper graphs generated from the full activations (top) and the foreground activations (bottom) at the last convolutional layer.

the underlying model gets better at separating the classes the deeper we go. However, the type of sampling seems not to influence the purity as much. Top 5 foreground sampling tends to have slightly higher purity, whereas random sampling has lower purity.

Point-wise purity. For a point $x \in S$, the point-wise purity is defined as

$$\beta_x = \frac{\sum_{i=1}^{n_x} \alpha_i}{n_x},$$

where n_x is the number of nodes containing point x. It is the average node-wise purity of all nodes containing x.

Class-wise purity. For a class k, the class-wise purity is defined as

$$\gamma_k = \frac{\sum_{i=1}^{N_c} \beta_i}{N_c},$$

where N_c is the number of points in class k. It is the average value of point-wise purity for all points in class k. Figure 12 (top) shows the class-wise purity of the deer class for foreground (top 1 and 5), random, and full activations at the same set of layers as node-wise purity. As was the case for the node-wise purity, we observe a general trend of increased class-wise purity of mapper graphs in deeper layers of the neural network.

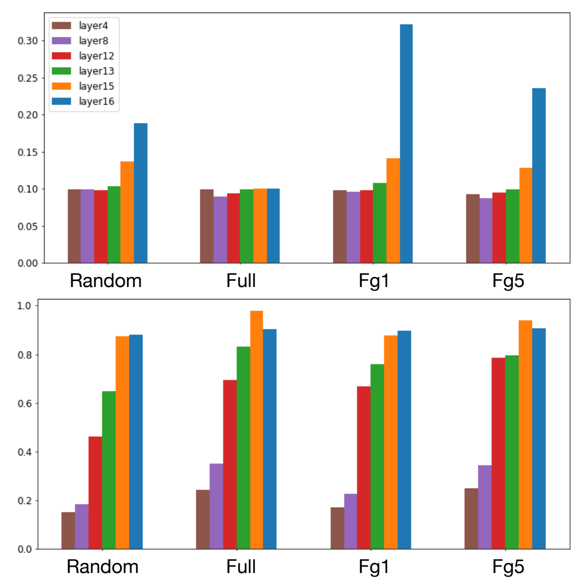

Figure 12: Top: class-wise purity of the deer class for random, full activations, and foreground (top 1 and 5) at a variety of layers; bottom: node-wise purity for random, full activations, and foreground (top 1 and 5) at a variety of layers, and the legend is the same as that of the top plot.

Generalization of Mapper Experiments to Additional Models

In order to show that our mapper graph observations are not dependent on the ResNet-18 architecture or CIFAR-10 data set we also perform these experiments using a different model-data pair. To compare with the prior experiments which use the lower resolution CIFAR-10 data set, the experiments in this section use a subset of 10 classes from the ImageNet dataset (Deng et al. 2009), as shown in the legend of Figure 13. There are 1300 images per class, resulting in a set of $N = 13$k images. The images have varying resolutions with an average resolution of 469×378. The data is pre-processed by first resizing each image to 256 pixels and center cropping to a patch of size 224×224, followed by a normalization with mean and variance of the original ImageNet training set images. For foreground ex-

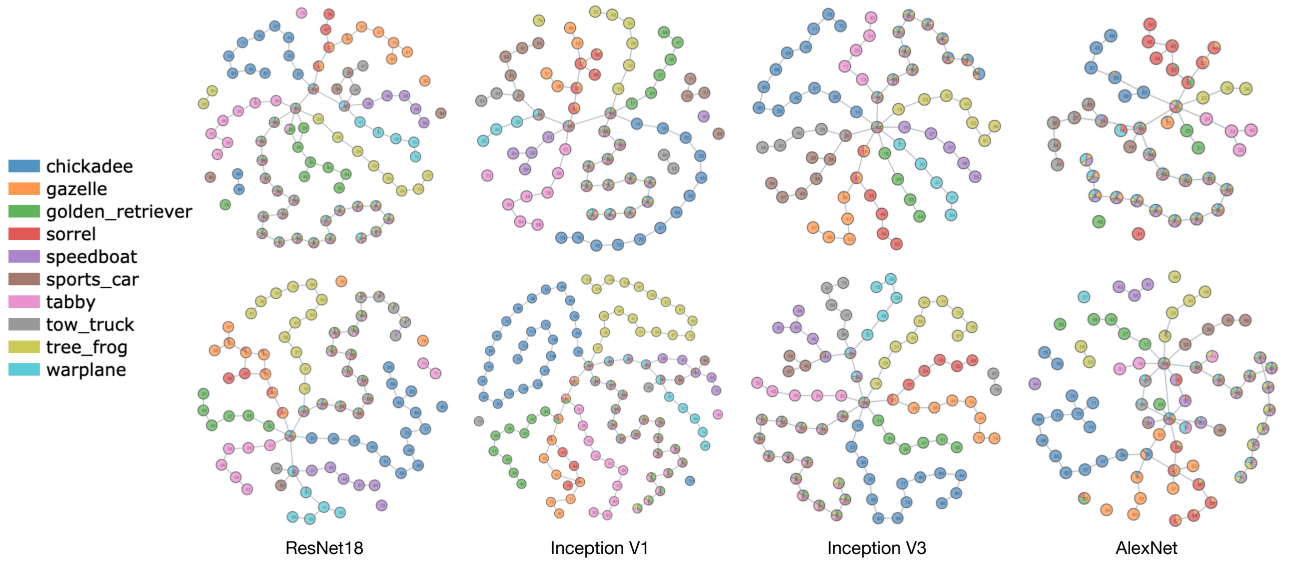

Figure 13: Mapper graphs of random (top) and foreground (bottom) activations for models trained on the ImageNet dataset.

traction, we apply a different strategy than previously used since `cv2.grabCut` does not work as well with the ImageNet dataset due to the large amount of high frequency details in the image backgrounds. Instead we use a pre-trained DeepLabV3 semantic segmentation model (Chen et al. 2017) to obtain the foreground mask which is then applied to the images to get the foreground pixels.

The models that we use for the generalization experiments include ResNet-18, Inception_v1 (Szegedy et al. 2015), Inception_v3 (Szegedy et al. 2016) and AlexNet (Krizhevsky, Sutskever, and Hinton 2012). The number of parameters of each model is 11.6M, 6.6M, 27.2M and 61.1M respectively.

Figure 13 shows the resulting mapper graphs generated from the last layer of each model. Through these experiments, we demonstrate that the structures and insights we observe on ResNet-18 applied to CIFAR-10 are applicable to a wide range of other image recognition models as well.

Discussion and Future Work

Our experiments using PH and mapper to study activation tensors of CNNs add to the growing body of literature to suggest that TDA provides useful summaries of DL models and hidden representations. The ability of mapper graphs to summarize point clouds from activation tensors and identify branching structures was previously shown in (Rathore et al. 2021). In our paper, we go beyond the random activations of that prior work to build mapper graphs of foreground, background, and full activation point clouds. These mapper graphs exhibit branching structures at earlier layers and show robustness with respect to image noise. Our new purity measures further quantify the observation that mapper graphs' branching structures align with class separations, and improve as we go deeper into the layers. Moreover, we also show that the mapper graph branching structures are present not just in ResNet-18 applied to CIFAR-10 but also to ImageNet studied using ResNet-18, InceptionV1 and V3, and AlexNet.

Although the mapper graphs we study come from a single trained model, our PH experiments show that the topological structures of the point clouds from which the mapper graphs are built are independent of the training run. Work has yet to be done to characterize those topological structures for CNNs beyond mapper graphs, but the fact that the distances are training-invariant indicates that such structures are indeed present and thus likely relevant to model interpretation. Although SW distance does pass the specificity test, we observed that, like the widely-cited CKA, it does not pass the sensitivity test of Ding, Denain, and Steinhardt (2021). We expect this is in part due to the previously noted differences between the standard representation and our sampled point cloud; however, our sampling approach is needed to mitigate the computational costs of PH, which scale with dimensionality of the underlying space.

In future work, we plan to further characterize the types of topological structures present in hidden layers of CNNs, explore theoretical justifications for the success of our experiments, and complete a more thorough analysis of the sensitivity of the SW distance via principal component removal. Finally, in order to aid DL practitioners in unlocking the hidden structures of their models, we plan to implement our methods into user-friendly tools.

Acknowledgements

MS, BW, and YZ are key contributors to this work. BW was partially funded by NSF DMS 2134223 and IIS 2205418.

References

Barannikov, S.; Trofimov, I.; Balabin, N.; and Burnaev, E. 2022. Representation Topology Divergence: A Method for Comparing Neural Network Representations. In Chaudhuri, K.; Jegelka, S.; Song, L.; Szepesvari, C.; Niu, G.; and

Sabato, S., eds., *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, 1607–1626. PMLR.

Bradski, G. 2000. The OpenCV Library. *Dr. Dobb's Journal of Software Tools*.

Carrière, M.; Cuturi, M.; and Oudot, S. 2017. Sliced Wasserstein Kernel for Persistence Diagrams. In Precup, D.; and Teh, Y. W., eds., *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, 664–673. PMLR.

Chen, L.-C.; Papandreou, G.; Schroff, F.; and Adam, H. 2017. Rethinking Atrous Convolution for Semantic Image Segmentation. arXiv preprint arXiv:1706.05587.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. ImageNet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255.

Ding, F.; Denain, J.-S.; and Steinhardt, J. 2021. Grounding Representation Similarity Through Statistical Testing. In Ranzato, M.; Beygelzimer, A.; Dauphin, Y.; Liang, P.; and Vaughan, J. W., eds., *Advances in Neural Information Processing Systems*, volume 34, 1556–1568.

Edelsbrunner, H.; and Harer, J. 2008. Persistent homology—a survey. In *Surveys on discrete and computational geometry*, volume 453, 257–282. American Mathematical Society.

Gabrielsson, R. B.; and Carlsson, G. 2019. Exposition and Interpretation of the Topology of Neural Networks. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 1069–1076.

Gebhart, T.; Schrater, P.; and Hylton, A. 2019. Characterizing the Shape of Activation Space in Deep Neural Networks. *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, 1537–1542.

Ghrist, R. 2008. Barcodes: the persistent topology of data. *Bulletin of the American Mathematical Society (New Series)*, 45(1): 61–75.

Guss, W. H.; and Salakhutdinov, R. 2018. On Characterizing the Capacity of Neural Networks using Algebraic Topology. arXiv preprint arXiv:1802.04443.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep Residual Learning for Image Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

Kim, B.; Wattenberg, M.; Gilmer, J.; Cai, C.; Wexler, J.; Viegas, F.; et al. 2018. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (tcav). In *International conference on machine learning*, 2668–2677. PMLR.

Kornblith, S.; Norouzi, M.; Lee, H.; and Hinton, G. 2019. Similarity of Neural Network Representations Revisited. In Chaudhuri, K.; and Salakhutdinov, R., eds., *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, 3519–3529. PMLR.

Krizhevsky, A.; and Hinton, G. 2009. Learning multiple layers of features from tiny images. Technical Report TR-2009, University of Toronto, Toronto, Ontario.

Krizhevsky, A.; Sutskever, I.; and Hinton, G. E. 2012. Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

Lacombe, T.; Ike, Y.; and Umeda, Y. 2021. Topological Uncertainty: Monitoring trained neural networks through persistence of activation graphs. In *Proceedings of the 30th International Joint Conference on Artificial Intelligence*, 2666–2672.

Mahendran, A.; and Vedaldi, A. 2015. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 5188–5196.

Morcos, A. S.; Raghu, M.; and Bengio, S. 2018. Insights on representational similarity in neural networks with canonical correlation. In Bengio, S.; Wallach, H. M.; Larochelle, H.; Grauman, K.; Cesa-Bianchi, N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, 5732–5741.

Olah, C.; Cammarata, N.; Schubert, L.; Goh, G.; Petrov, M.; and Carter, S. 2020. Zoom In: An Introduction to Circuits. *Distill*. Https://distill.pub/2020/circuits/zoom-in.

Raghu, M.; Gilmer, J.; Yosinski, J.; and Sohl-Dickstein, J. 2017. SVCCA: Singular Vector Canonical Correlation Analysis for Deep Learning Dynamics and Interpretability. In Guyon, I.; von Luxburg, U.; Bengio, S.; Wallach, H. M.; Fergus, R.; Vishwanathan, S. V. N.; and Garnett, R., eds., *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 6076–6085.

Rathore, A.; Chalapathi, N.; Palande, S.; and Wang, B. 2021. TopoAct: Visually Exploring the Shape of Activations in Deep Learning. *Computer Graphics Forum*, 40(1): 382–397.

Rieck, B. A.; Togninalli, M.; Bock, C.; Moor, M.; Horn, M.; Gumbsch, T.; and Borgwardt, K. 2019. Neural Persistence: A Complexity Measure for Deep Neural Networks Using Algebraic Topology. In *International Conference on Learning Representations (ICLR 2019)*.

Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization. In *Proceedings of the IEEE international conference on computer vision*, 618–626.

Singh, G.; Memoli, F.; and Carlsson, G. 2007. Topological Methods for the Analysis of High Dimensional Data Sets and 3D Object Recognition. In Botsch, M.; Pajarola, R.; Chen, B.; and Zwicker, M., eds., *Eurographics Symposium on Point-Based Graphics*.

Szegedy, C.; Liu, W.; Jia, Y.; Sermanet, P.; Reed, S.; Anguelov, D.; Erhan, D.; Vanhoucke, V.; and Rabinovich, A. 2015. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 1–9.

Szegedy, C.; Vanhoucke, V.; Ioffe, S.; Shlens, J.; and Wojna, Z. 2016. Rethinking the inception architecture for computer vision. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2818–2826.

Wei, D.; Zhou, B.; Torrabla, A.; and Freeman, W. 2015. Understanding intra-class knowledge inside CNN. arXiv preprint arXiv:1507.02379.

Wheeler, M.; Bouza, J.; and Bubenik, P. 2021. Activation Landscapes as a Topological Summary of Neural Network Performance. In *2021 IEEE International Conference on Big Data (Big Data)*, 3865–3870. IEEE.

Zhou, B.; Khosla, A.; Lapedriza, À.; Oliva, A.; and Torralba, A. 2015. Object Detectors Emerge in Deep Scene CNNs. In Bengio, Y.; and LeCun, Y., eds., *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Zhou, Y.; Chalapathi, N.; Rathore, A.; Zhao, Y.; and Wang, B. 2021. Mapper Interactive: A Scalable, Extendable, and Interactive Toolbox for the Visual Exploration of High-Dimensional Data. In *2021 IEEE 14th Pacific Visualization Symposium (PacificVis)*, 101–110.