Visualizing and Analyzing the Topology of Neuron Activations in Deep Adversarial Training

Youjia Zhou [1] **Yi Zhou** [1] **Jie Ding** [2] **Bei Wang** [1]

Abstract

Deep models are known to be vulnerable to data adversarial attacks, and many adversarial training techniques have been developed to improve their adversarial robustness. While data adversaries attack model predictions through modifying data, little is known about their impact on the neuron activations produced by the model, which play a crucial role in determining the model's predictions and interpretability. In this work, we aim to develop a topological understanding of adversarial training to enhance its interpretability. We analyze the topological structure—in particular, mapper graphs—of neuron activations of data samples produced by deep adversarial training. Each node of a mapper graph represents a cluster of activations, and two nodes are connected by an edge if their corresponding clusters have a nonempty intersection. We provide an interactive visualization tool that demonstrates the utility of our topological framework in exploring the activation space. We found that stronger attacks make the data samples more indistinguishable in the neuron activation space that leads to a lower accuracy. Our tool also provides a natural way to identify the vulnerable data samples that may be useful in improving model robustness.

1. Introduction

Despite the great success of deep learning, in the past decade, researchers have found that overparameterized deep neural network models are ubiquitously vulnerable to data adversarial attacks. Specifically, Szegedy et al. (Szegedy et al., 2013) and Goodfellow et al. (Goodfellow et al., 2014) showed that we can find and apply imperceptible perturbations to the data samples such that the perturbed samples are almost indistinguishable from normal examples and can fool neural network models that are trained on normal samples. In particular, various types of data adversarial attack methods have been developed for deep learning, including the projected gradient descent (PGD) type attack (Madry et al., 2018), single pixel attack for images (Su et al., 2019), and physical patch attack that covers an image with a printed adversarial patch to attack the model (Brown et al., 2017). All these research findings have raised much concern about the safety and robustness of deep learning in the community, and have attracted a lot of attention on understanding and improving the adversarial robustness of deep models.

To enhance the robustness of machine learning (ML) models against data adversaries, a standard and popular approach is adversarial training (Sinha et al., 2018; Goodfellow et al., 2014), whose main idea is to generate adversarial samples using a certain attack method and then use them to train the model. This empirical approach has been shown to be able to substantially improve the robustness of the model against adversarial attacks.

In this paper, we are interested in forming a topological understanding of adversarial training to increase its interpretability. A number of previous works study the geometry of adversarial training, most of which focus on a geometric understanding of the decision boundaries, e.g., (He et al., 2018; Khoury & Hadfield-Menell, 2018; Zhang et al., 2021; Rade & Moosavi-Dezfooli, 2022; Liang et al., 2022; Xu et al., 2023). However, to the best of our knowledge, few works exist that study the topology of adversarial training. In particular, little is known about the *topological structure of the neuron activations* of adversarially-trained deep networks, which constitutes the main goal of this work.

1.1. Our Contribution

We propose a topological framework for summarizing and analyzing the space of neuron activations in adversarial training. We use *neuron activations* to refer to the high-dimensional vector representations (i.e., the intermediate outputs of neurons) produced by a particular layer of a neural network, and *activation space* to refer to the space

[1]University of Utah, Salt Lake City, USA. [2]University of Minnesota-Twin Cities, Minneapolis, USA. Correspondence to: Bei Wang <beiwang@sci.utah.edu>.

Proceedings of the 2^{nd} Annual Workshop on Topology, Algebra, and Geometry in Machine Learning (TAG-ML) at the 40^{th} International Conference on Machine Learning, Honolulu, Hawaii, USA. 2023.

of these activations. We leverage the *mapper graph* (Singh et al., 2007) to summarize the topological structure of the activation space. Each node of the mapper graph represents a cluster of activations, and two nodes are connected by an edge if their corresponding clusters have a nonempty intersection. Our contributions are as follows:

- We provide an open source interactive visualization tool (Zhou, 2023) that demonstrates the utility of our topological framework in exploring the activation space of adversarial training.
- We study the evolution of topology of activation spaces across different levels of adversarial attacks.
- We analyze the weak regions of the activation space, that is, topological neighborhoods with low prediction accuracy vulnerable to adversarial attacks.
- We work toward model refinement by leveraging the identified weak regions to improve robust test accuracy.

Our work is at the intersection of topological data analysis (TDA) and visualization. It also contributes towards the visualization of neuron activations, as an example of visual analytics systems that support model explanation, interpretation, debugging, and improvement for deep learning (Hohman et al., 2018).

1.2. Related Work

Adversarial attack. Various types of data adversarial attack methods have been developed for deep learning. For example, the Fast Gradient Sign Method (FGSM) (Goodfellow et al., 2014) generates adversarial examples based on the sign of gradients. The projected gradient descent (PGD) attack is another popular attack (Madry et al., 2018), which finds the adversarial perturbation using gradient updates followed by projection onto certain constraints, e.g., ℓ_∞ or ℓ_2 ball constraints. There are other types of attacks that target specific scenarios, such as the one-pixel attack that fools the model by changing only one pixel of an image (Su et al., 2019), physical patch attack that covers a part of an image with a printed and adversarially designed patch to attack the model (Brown et al., 2017), and backdoor poisoning attack (Xian et al., 2023) that creates a few backdoor data inputs with a specific trigger (e.g., a patch (Gu et al., 2017) or watermark (Chen et al., 2017)) and target labels.

Adversarial training. A standard and popular adversarial training approach is to train the model using adversarial samples generated by a certain attack method (Sinha et al., 2018; Goodfellow et al., 2014). In addition, adversarial training has been formulated as a nonconvex minimax optimization problem (Madry et al., 2018), and convergence of gradient-based algorithms has been established under certain assumptions (Gao et al., 2019). The performance of adversarial training was further improved (Ding et al., 2020) by combining the usual cross-entropy loss with a margin maximization loss term applied to the correctly classified examples. It was shown that misclassified examples have more impact on the final robustness than correctly classified examples, and incorporating misclassified examples in adversarial training as a regularizer (Wang et al., 2020) improves the adversarial robustness.

Visualization of neuron activations. A number of approaches have been proposed in recent years to explain the behavior of deep learning by visualizing the features learned by hidden units of the neural networks (Hohman et al., 2018). *Activation maximization* (Erhan et al., 2009) finds the input images that maximize the neuron activations, whereas *salience maps* are obtained by projecting neuron activations from hidden layers back onto the input space (Simonyan et al., 2014). *DeepVis* (Yosinski et al., 2015) visualizes the neuron activations produced on each layer of a CNN as it processes images/videos live. *Multifaceted feature visualization* synthesizes a visualization of input image that activates a neuron (Nguyen et al., 2016). *TCAV* (Testing with Concept Activation Vectors) uses directional derivatives of activations to quantify the sensitivity of model predictions to a concept (Kim et al., 2018). Dimensionality reduction (DR) has been applied to neuron activations (Karpathy, 2014; Nguyen et al., 2016). In particular, *activation atlas* (Carter et al., 2019) combines feature visualization with DR to visualize averaged activations, whereas *SUMMIT* (Hohman et al., 2020) not only computes aggregated activations but also captures relationships between neurons across layers.

Topology of deep leaning. Exploring the geometry and topology of deep learning models, in particular, understanding their decision boundaries/regions is an active area of research, e.g., (Fawzi et al., 2018; Liu & Shen, 2022). A number of prior works use tools from TDA to study topological complexity of deep learning. *Topological capacity* (Guss & Salakhutdinov, 2018) and *neural persistence* (Rieck et al., 2019) were introduced to quantify the learnability and complexity of neural networks, respectively. Persistent homology (Edelsbrunner & Harer, 2007) of activations was used to study how model complexity changes across layers (Wheeler et al., 2021). Mapper graphs of learned weights from convolutional layers were used to quantify topological similarities among different CNN model architectures (Gabrielsson & Carlsson, 2019). *Activation graphs* were used to investigate the topology of neural networks (Gebhart et al., 2019; Lacombe et al., 2021). The work that is most relevant to ours is *TopoAct* (Rathore et al., 2021), which explores mapper graphs of neuron activations from image classifiers. TopoAct was further extended to study activations of images with Gaussian noise (Purvine et al., 2023) and to explore the topology of word embeddings from large language models (LLMs) during fine-tuning (Rathore et al., 2023). Different from previous work, we apply the mapper graph to study activations from adversarial training.

2. Preliminaries

2.1. Mapper Graph

We capture the topology of neuron activations via a graphical representation called the *mapper graph*, which arises from a "partial clustering of the data guided by a set of functions defined on the data" (Singh et al., 2007).

Let $\mathbb{X}$ be a high-dimensional point cloud. A *cover* of $\mathbb{X}$ is a set of open sets in $\mathbb{R}^d$, $\mathcal{U} = \{U_i\}_{i\in I}$ such that $\mathbb{X} \subset \cup_{i\in I} U_i$. The one-dimensional nerve of $\mathcal{U}$ is a graph and is denoted as $\mathcal{N}_1(\mathcal{U})$. Each node i in $\mathcal{N}_1(\mathcal{U})$ represents a cover element U_i, and there is an edge between nodes i and j if $U_i \cap U_j$ is not empty.

In the classic mapper construction (Singh et al., 2007), obtaining a cover of $\mathbb{X}$ is guided by a set of scalar functions defined on $\mathbb{X}$, referred to as *filter* functions. In our setting, we define a mapper graph with a single filter function $f : \mathbb{X} \to \mathbb{R}$. A cover $\mathcal{V} = \{V_k\}_{k=1}^n$ of $f(\mathbb{X}) \subset \mathbb{R}$ can be obtained such that $f(\mathbb{X}) \subseteq \cup_k V_k$, and the cover $\mathcal{U}$ of $\mathbb{X}$ can then be obtained by considering the clusters induced by points in $f^{-1}(V_k)$ for each V_k as cover elements. The one-dimensional nerve of $\mathcal{U}$, denoted as $\mathcal{M} = \mathcal{M}(\mathbb{X}, f) := \mathcal{N}_1(\mathcal{U})$, is the *mapper graph* of $(\mathbb{X}, f)$.

Take Figure 1 as an example. A point cloud $\mathbb{X}$ is sampled from the silhouette of a butterfly and equipped with a height function $f : \mathbb{X} \to \mathbb{R}$. A cover $\mathcal{V} = \{V_1, \cdots, V_6\}$ of $f(\mathbb{X})$ is formed by six intervals (see Figure 1 middle). For each k ($1 \le k \le 6$), $f^{-1}(V_k)$ induces a number of clusters that are subsets of $\mathbb{X}$. These clusters (enclosed by rectangles) form the elements of a cover $\mathcal{U}$ of $\mathbb{X}$ (see Figure 1 left). The mapper graph of $\mathbb{X}$ is shown in Figure 1 (right). For instance, $f^{-1}(V_1)$ induces four cover elements of $\mathbb{X}$, including U_1, whereas $f^{-1}(V_2)$ induces three cover elements, including U_2. U_1 and U_2 become nodes 1 and 2 in the mapper graph, respectively. Since $U_1 \cap U_2 \neq \emptyset$, an edge connects node 1 and node 2 in the mapper graph.

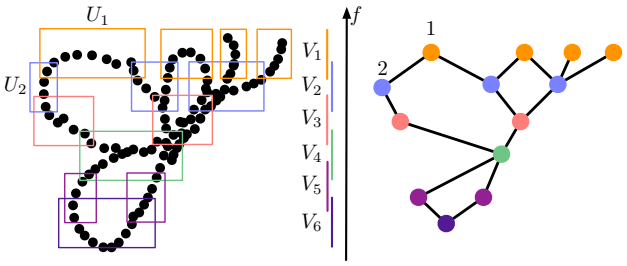

Figure 1. A mapper graph (right) of a point cloud sampled from the silhouette of a butterfly (left).

Here, we construct a mapper graph using a *uniform cover*, that is, all intervals covering $f(\mathbb{X})$ are of the same length. Several parameters are needed, including the filter function f, the number of cover elements n, and their percentage of overlap p, the metric $d_{\mathbb{X}}$ on $\mathbb{X}$, and the clustering method. As shown in Figure 1, $n = 6$, $p = 30\%$, $d_{\mathbb{X}}$ is the Euclidean distance, and the clustering method is the density-based DBSCAN (Ester et al., 1996). With an appropriate choice of parameters, the mapper graph captures the shape of the input data. Another less widely used strategy is the *balanced cover* implemented in *giotto-tda* (Tauzin et al., 2020), where the inverse image of each interval contains an equal number of points. For a discussion on parameter tuning, see (Zhou et al., 2021; Chalapathi et al., 2021).

2.2. Adversarial Machine Learning

To enhance the robustness of ML models against data adversaries, a standard and popular approach is to perform adversarial training (Sinha et al., 2018; Goodfellow et al., 2014), which generates adversarial samples using a certain attack method and then uses them to train the model. Consider a standard classification problem with a given set of training samples $\{x_i, y_i\}_{i=1}^n$, where x_i is the i-th sample and y_i denotes the corresponding label. We aim to train a classifier $h_\theta : \mathbb{X} \to \mathbb{Y}$ parameterized by $\theta \in \mathbb{R}^d$ to solve this classification problem, where h_θ typically corresponds to a deep neural network. In traditional deep learning, we train the classifier by solving the empirical risk minimization problem $\min_{\theta\in\mathbb{R}^d} \frac{1}{n}\sum_{i=1}^n \ell(h_\theta(x_i), y_i)$, where ℓ is usually chosen to be the cross-entropy loss.

To perform adversarial training against the ℓ_p-PGD attack, we aim to solve the following optimization problem,

$$\min_{\theta\in\mathbb{R}^d} \max_{\{\xi_i\}_i:\|\xi_i - x_i\|_p \le \epsilon} \frac{1}{n}\sum_{i=1}^n \ell(h_\theta(\xi_i), y_i). \quad (1)$$

To elaborate, for any fixed model θ, the inner maximization part of (1) aims to find data samples $\{\xi_i\}_{i=1}^n$ that (i) are ϵ-close to the original samples $\{x_i\}_{i=1}^n$ in terms of the ℓ_p norm, and (ii) achieve very high classification loss. These samples are called *adversarial samples*, and the ℓ_p constraint controls the search space of data adversary. On the other hand, the outer minimization part of (1) aims to train a robust model θ that achieves low classification loss on these advesarial samples $\{\xi_i\}_{i=1}^n$. In practice, we solve the above adversarial training problem via an alternating approach, i.e., we first apply PGD to generate adversarial data samples ξ for a fixed model θ, and then train the model using these adversarial data samples.

2.3. Neuron Activations

For an image classification task, the input to a neural network is a tensor, and the output is a probability vector showing the likelihood the input belongs to each class. The intermediate outputs of neurons from each layer are called *activation tensors*. Activation tensors may be sliced into spatial activations or channel activations. We work with spatial activations (referred to as activations or activation vectors)

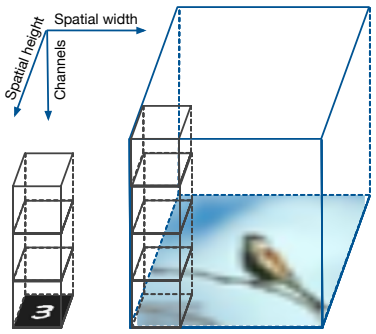

Figure 2. Illustration of a spatial activation within an activation tensor for a MNIST (left) and a CIFAR-10 image (right).

in this paper, shown in Figure 2. For each image from the MNIST dataset, a single spatial activation is generated at the last layer of an MLP model across 10 channels, forming a 10-dimensional point cloud with 60K points. For each image from the CIFAR-10 dataset, we randomly sample a single spatial activation from 4×4 images patches, generated at the last convolutional layer of ResNet-18, forming a 512-dimensional point cloud with 50K points. We focus on the last layer as those are the features linearly separated by the classifier; see (Purvine et al., 2023) for examples across layers (without adversarial attacks).

2.4. Mapper Graphs of Neuron Activations

Given a high-dimensional point cloud $\mathbb{X}$ formed by activation vectors, we construct a mapper graph of $\mathbb{X}$ that captures its topological structure, i.e., how the activation vectors are organized with a dataset-model pair. We use ℓ_2-norm of activation vectors as the filter function, which has been shown to produce meaningful results in studying image activations (Rathore et al., 2021). We define a cluster of points associated with a node in the mapper graph a *topological neighborhood* following (Rathore et al., 2023). Therefore, each node in the mapper graph is a topological neighborhood, and the edges between these nodes encode the overlaps (or connectivity) between these neighborhoods. In particular, two points x and y in $\mathbb{X}$ are in the same topological neighborhood if they are close to each other in terms of a metric $d_{\mathbb{X}}$ (e.g., a Euclidean metric), and their function values $f(x)$ and $f(y)$ fall in the same interval of $f(\mathbb{X})$.

We also review the notion of *purity* of a topological neighborhood (Rathore et al., 2023), defined to be $p(X) := 1 - \frac{H(D_X)}{H(D)}$. Here, D_X is the observed distribution of labels for points in X and D is a uniform distribution of all labels; H denotes the Shannon entropy of a distribution. $p(X)$ reaches the highest value of 1 when all points in X are from the same class, and the lowest value of 0 when the points are uniformly distributed over all classes (Rathore et al., 2023); *pure* neighborhoods have a purity of 1; otherwise they are *impure*.

3. Visualizing the Topology of Neuron Activations in Adversarial Training

Experimental setup. We conduct adversarial training experiments using various standard image classification datasets and deep learning models, including the MNIST (LeCun et al., 2010) and CIFAR-10 (Krizhevsky, 2009) datasets, and the multi-layer perceptron (MLP) and ResNet-18 models (He et al., 2015). For each dataset-model pair, we train the model with the dataset using two different approaches: (i) standard training using the original clean data (i.e., without adversarial training), and we denote the trained model as M_{clean}; and (ii) adversarial training with different types of attack, and we denote the adversarially-trained model as M_{adv}. We explore two types of attack, ℓ_∞- and ℓ_2-PGD attack where ℓ_∞ attack is considered a stronger attack than a ℓ_2 attack; see the supplementary material for details.

Interactive visualization tool. We present an interactive visualization tool, referred to as *Mapper Interactive Adversarial Training* (or MIAT) to explore how the topological structure of neuron activations change under adversarial training. The tool is an extension of Mapper Interactive (Zhou et al., 2021), available open source. The tool takes as input a high-dimensional point cloud of activations and computes its mapper graph on-the-fly with user-defined parameters. Specifically, it allows the exploration of image samples associated with selected mapper nodes for both M_{clean} and M_{adv}. Its frontend is implemented using HTML/CSS/JavaScript stack with D3.js and JQuery JavaScript libraries, while its backend uses Python via a Flask server.

Exploratory visual analysis. Using our interactive tool, we mainly perform two exploratory analysis tasks. First, we explore topological neighborhoods with high and low prediction accuracy to better understand input images that are vulnerable to adversarial attacks across different levels (Section 3.1). Second, we study the evolution of topology as we increase the attack level (Section 3.2).

3.1. Training MLP with MNIST

Topology of a clean model. We start with the MNIST-MLP dataset-model pair. Figure 3 (top left) shows the mapper graph generated from M_{clean} using a uniform cover. The pie chart on each node shows the composition of class labels in that node. Figure 3 (top right) shows the node-wise prediction accuracy. The overall prediction accuracy of M_{clean} is 97.48%, therefore, we observe that most of the nodes have a very high node-wise prediction accuracy.

Using the interactive tool, we first explore pure and impure nodes (topological neighborhoods) in the mapper graph. Intuitively, impure nodes may be considered as being centered around the decision boundaries of the model where samples from two or more classes have similar neuron activations.

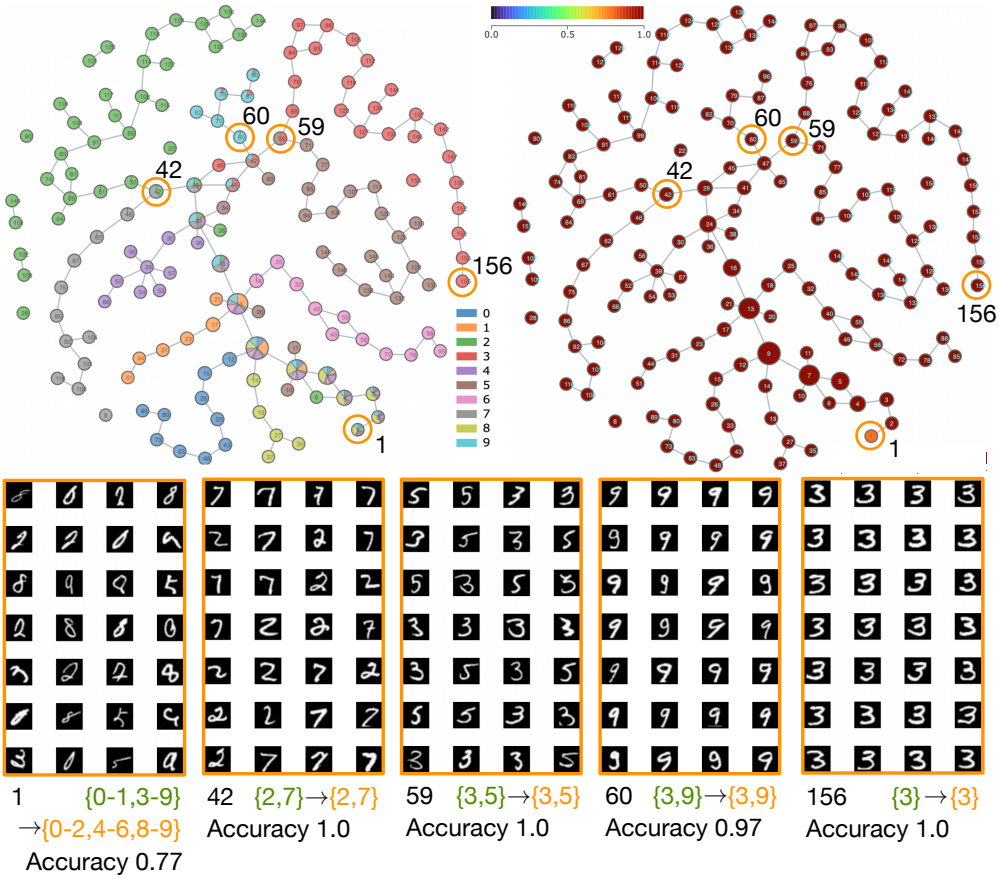

Figure 3. Mapper graph of M_{clean} visualized by pie charts of true labels (top left) and colored by node-wise prediction accuracy (top right).

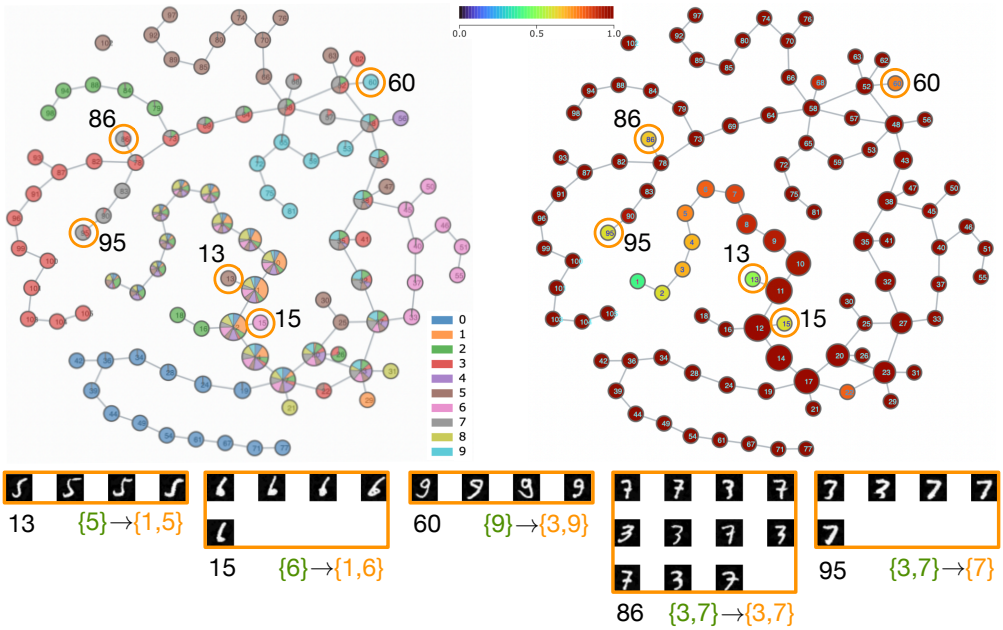

Figure 4. Mapper graph of M_{adv} under ℓ_∞-PGD attack visualized by pie charts of true labels (top left) and colored by node-wise prediction accuracy (top right). Attack level $\epsilon = 0.05$. Nodes circled in orange are examples of weak regions.

The average node purity has been shown to be correlated with model performance on the unseen data (Rathore et al., 2023). As shown in Figure 3, node 156 is a pure node with a prediction accuracy of 1.0: all images in the node have a

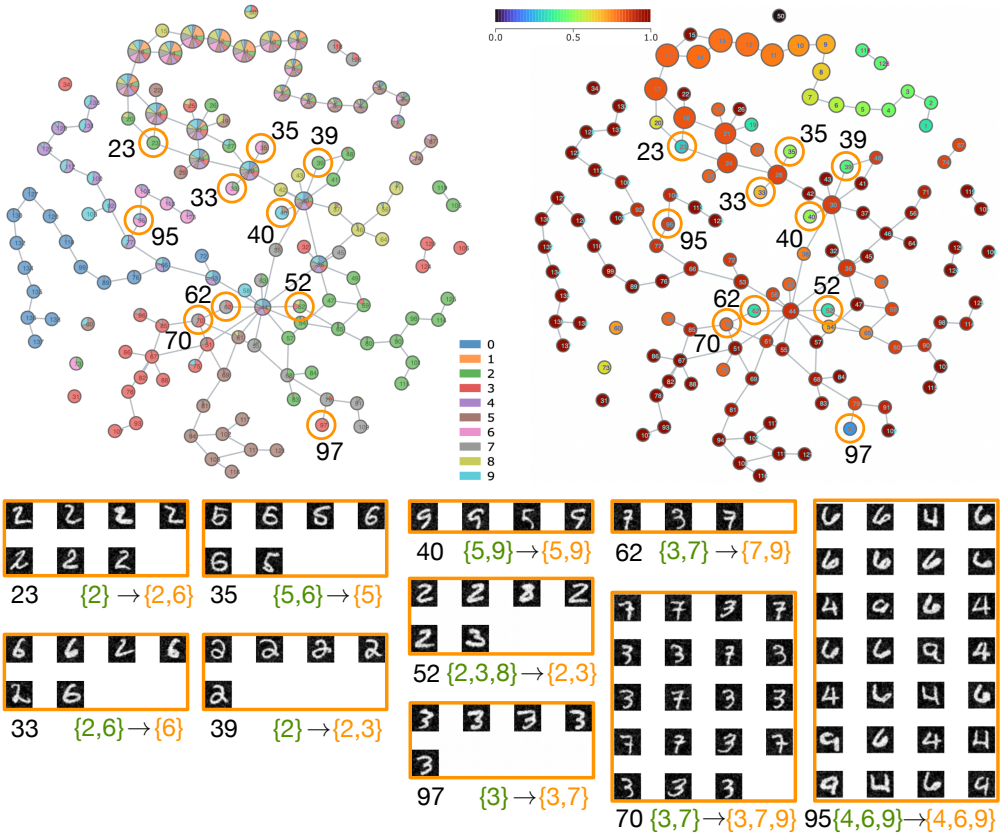

Figure 5. Mapper graph of M_{adv} under ℓ_∞-PGD attack visualized by pie charts of true labels (top left) and colored by node-wise prediction accuracy (top right). Attack level $\epsilon = 0.30$. Nodes circled in orange are examples of weak regions.

true label of digit 3 and a predicted label of digit 3. Nodes 42, 59, and 60 are all impure nodes with a high prediction accuracy: they highlight the decision boundary among digits $\{2, 7\}, \{3, 5\}, \{3, 9\}$, respectively. Node 60 has a slightly imperfect prediction accuracy as a digit 9 is predicted as 3.

Second, we explore the so-called *weak topological regions* (weak regions for short), which are topological neighborhoods with low prediction accuracy. We consider weak regions to be candidates vulnerable to adversarial attacks. As shown in Figure 3 (bottom), we explore the weak regions such as node 1 in the mapper graph, which has a node-wise prediction accuracy of 77%. We explore the images associated with the points (activation vectors) in node 1, with true labels {0-1,3-9} (in green) and predicted labels {0-2,4-6,8-9} (in orange). In particular, node 1 contains images that are harder to identify, compared with the images of nodes with higher accuracy, such as node 156.

Topology of adversarial models. We now explore mapper graphs generated from adversarial models across different attack levels. Figure 4 shows the mapper graph generated from M_{adv} with a small ℓ_∞-PGD attack level ($\epsilon = 0.05$) using a uniform cover. We focus on exploring the weak regions (highlighted inside orange circles) and study where and how the node-wise prediction accuracy changes with adversarial attack, indicating model confusion.

We first observe that some pure nodes become weak regions even with a small attack level. Node 13 (with a prediction accuracy of 0.5) shows an example of the model confusion, where samples in the node have a true label of $\{5\}$ (in green) and predicted labels of $\{1, 5\}$ (in orange); two of the perturbed images of digit 5 are misclassified as digit 1. Node 15 contains digits 6 which are misclassified as digits 1; this is interpretable as these digit 6 images contain tiny loops that may be easily mistaken as digit 1 with small perturbations. Furthermore, a few digits 9 in node 60 are misclassified as digits 3. Next, some impure nodes become weak regions, e.g., some digits 3 in nodes 86 and 95 are misclassified as digits 7.

Figure 5 shows the mapper graph generated from M_{adv} with a large ℓ_∞-PGD attack level ($\epsilon = 0.30$). Compared with Figure 4, we observe more weak region candidates with lower prediction accuracy. Specifically, we observe that digits 3 and 7 become further mixed as more adversarial noises are added to the images (see nodes 62 and 70). Furthermore, the model becomes more confused with a larger attack level: digits 3 and 7 are further confused not only among themselves (in node 97) but also with digits 9 (in nodes 62 and 70). In addition, the same class may be confused with dif-

ferent classes in different parts of the activation space. For example, digits 2 are misclassified as digits 6 in nodes 23 and 33, whereas digits 2 are mistaken as digits 3 in node 39. Digits 5 are confused with digits 6 in node 35, whereas they become confused with digits 9 in node 40. Node 95 highlights model confusion among digits 4, 6 and 9.

Remark. We have a few takeaways from the above exploration. For the clean model M_{clean}, impure nodes in the mapper graph capture decision boundaries. For the adversarial model M_{adv}, weak regions highlight data samples that are vulnerable to adversarial attacks thus relevant to model confusion. First, the same class may be confused with different classes in different parts of the activation space. Second, as we increase the attack level, classes that are easily confused at a lower attack level remain confused at a higher attack level, with more images joining the weak regions. Furthermore, new classes become confused with one another with increased perturbation to the images.

3.2. Training ResNet-18 with CIFAR-10

Training ResNet-18 model with the CIFAR-10 dataset, we now explore how the mapper graphs evolve as we increase the attack level ϵ. Figure 6 demonstrates the mapper graph of M_{clean} (using clean images) and the evolution of mapper graphs of M_{adv} (using perturbed images) across five different attack levels: $\epsilon \in \{0.01, 0.05, 0.10, 0.20, 0.30\}$. We observe that as ϵ increases, the number of impure nodes in the mapper graph increases; at the same time, the test accuracy decreases dramatically, whereas the training accuracy remains high (see the supplementary material). For instance, at $\epsilon = 0.30$, the test accuracy is at 79.41% with a training accuracy of 99%, indicating that the underlying model is over-fitting the training samples. Our observation that impure nodes dominate the mapper graphs under adversarial training aligns with the adversarial model being overfitted, that is, even though most topological neighborhoods contain highly mixed labels (low purity), the model still achieves high training accuracy by over-fitting the decision boundary.

We further compute a *weighted average purity* of the nodes for each mapper graph. It is computed by multiplying the node purity with the number of samples within the node and dividing it by the total number of samples across all nodes. As illustrated in Figure 7, as the attack level increases, the weighted average purity decreases.

4. Towards Model Refinement

Following the experimental setup in Section 3, we work toward model refinement by leveraging the identified weak regions and obtain mixed results, which we detail below. We are interested in the following question: under what conditions can we leverage the topology of the mapper graph to improve robust accuracy?

Recall that we define weak regions to be topological neighborhoods with low prediction accuracy. In practice, to include points from the weak regions for refinement, we first rank the topological neighborhoods based on their node-wise prediction accuracy in an ascending order, and then select misclassified points from these neighborhoods until the total number of selected points reaches around 20% of the training data size, e.g., 10K. For MLP with MNIST data, we observe a marginal improvement of robust accuracy when the adversarially-trained model M_{adv} is further refined using samples in the weak regions. However, for ResNet-18 with CIFAR-10, we observe no clear model improvement following the same procedure.

We conjecture that the purity of topological neighborhoods associated with adversarial training is correlated with the utility of weak regions and plays an important role in addressing the above question. Specifically, if purity is relatively high, then it is meaningful to identify weak regions for refinement; otherwise the mapper graph is too noisy to take advantage of weak regions in model refinement.

Training MLP with MNIST. We first train an MLP using the MNIST dataset, the resulting M_{clean} has a test accuracy of 97.48%. We then perform standard adversarial training to obtain the robust model M_{adv}, based on which we further perform model refinement by leveraging the misclassified samples in the weak regions, and we denote the refined model as M_{refine}.

Table 1 compares the robust (test) accuracy achieved by the models M_{adv} and M_{refine} under different levels of ℓ_∞-PGD attack. In particular, we train the refined model M_{refine} using mapper graphs with a uniform cover and a balanced cover, respectively. It can be seen that the two refined models consistently achieve higher robust accuracy than those achieved by M_{adv} over different levels of ℓ_∞-PGD attack, and their robust accuracies are comparable to each other. Similarly, under different levels of ℓ_2-PGD attack, we observe marginal improvement of robust accuracy as shown in Table 2. This shows that for MLP with MNIST, the topological structure in the mapper graph can effectively help identify the weak regions containing samples vulnerable to adversarial attack.

Training ResNet-18 with CIFAR-10. On the other hand, we repeat the same experiment by training a ResNet-18 with CIFAR-10. The resulting mapper graph appears to be ineffective in identifying weak regions that are useful for model improvement. In particular, topological neighborhoods have low purity and weak region identification is not as effective as MLP with MNIST; see the supplementary material.

Remark. We provide an explanation for the above experimental results on model refinement. Specifically, images in

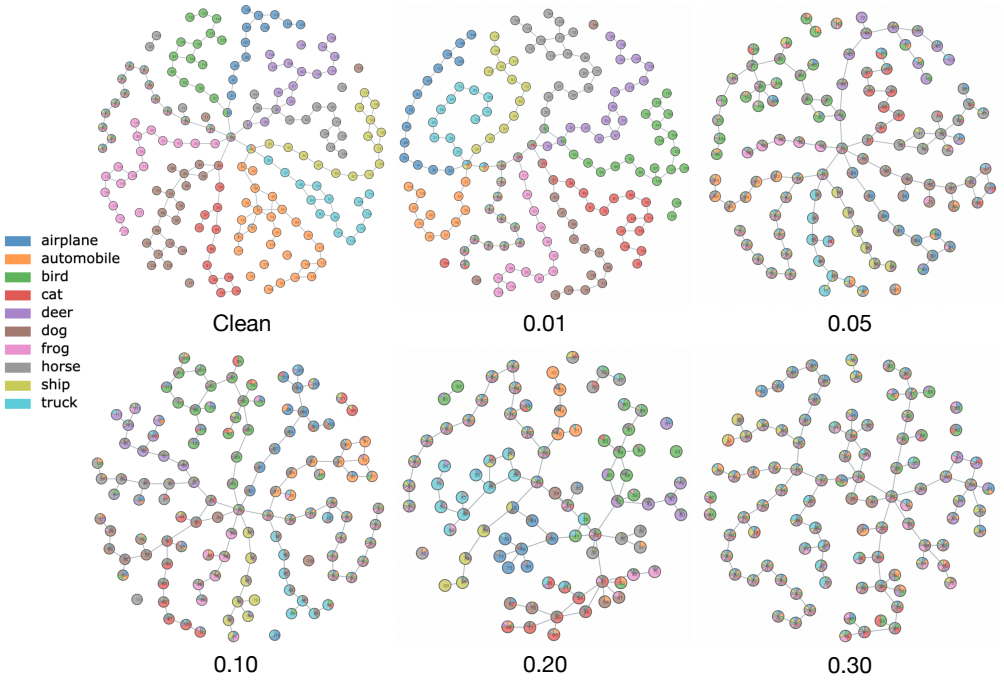

Figure 6. The mapper graph of M_{clean} and the mapper graphs of M_{adv} under ℓ_2-PGD attack with different ϵ values for CIFAR-10 data and ResNet-18 model.

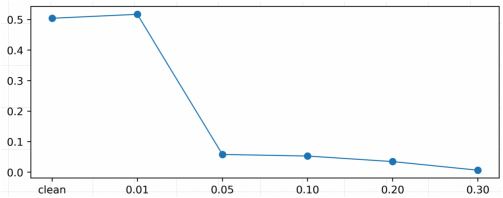

Figure 7. Weighted average node purity of mapper graphs across different attack levels.

Attack level ϵ	0.05	0.1	0.15	0.2	0.25	0.3
M_{adv}	93.240 (0.354)	90.175 (1.704)	88.730 (1.513)	86.505 (2.864)	84.250 (3.663)	81.675 (5.211)
M_{refine} (uniform cover)	93.490 (0.240)	90.440 (1.400)	88.985 (1.336)	86.655 (2.977)	84.760 (3.762)	82.040 (4.936)
Improvement	**0.250** (0.113)	**0.265** (0.304)	**0.255** (0.177)	**0.150** (0.113)	**0.510** (0.099)	**0.365** (0.276)
M_{refine} (balanced cover)	93.445 (0.332)	90.530 (1.499)	89.010 (1.699)	86.540 (2.871)	84.755 (4.179)	82.015 (5.084)
Improvement	**0.205** (0.021)	**0.355** (0.205)	**0.280** (0.156)	**0.035** (0.007)	**0.505** (0.516)	**0.340** (0.127)

Table 1. Robust accuracy: average and standard deviation of robust test accuracy for refined models. MNIST with L_∞ PGD attack. Standard deviations (in parentheses) are obtained using two different seeds in the initialization.

the MNIST dataset have only a few modalities, i.e., they are highly similar within each class. Consequently, we observe in Figure 4 and Figure 5 that the topological structure (i.e., bifurcations) of its associated mapper graph is highly robust to adversarial attacks, leading to a high robust test accuracy. As a comparison, the images in the complex CIFAR-10 dataset have more modalities, i.e., they are highly diverse even within each class. Consequently, we observe in Figure 6 that the topological structure of its associated mapper graph is highly vulnerable to adversarial attacks. In fact,

Attack level ϵ	0.05	0.1	0.2	0.3
M_{adv}	97.160 (0.297)	96.745 (0.064)	94.150 (2.857)	92.145 (4.830)
M_{refine} (balanced cover)	97.175 (0.191)	96.830 (0.269)	94.370 (2.390)	92.255 (4.589)
Improvement	**0.015** (0.106)	**0.085** (0.205)	**0.220** (0.467)	**0.110** (0.240)

Table 2. Robust accuracy: average and standard deviation of robust test accuracy for refined models. MNIST with L_2 PGD attack.

under a small ℓ_2-PGD attack level $\epsilon = 0.05$, the nodes in the mapper graph of CIFAR-10 are already highly impure, while the training robust accuracy is as high as 99%. This shows that the adversarially-trained model basically overfits all the adversarial samples, whose neuron activations are actually hard to distinguish in the activation space.

5. Conclusion

We developed an interactive visualization tool to analyze the topological structures of neuron activations in adversarial training. Our analysis showed that stronger attacks on more complex datasets make the neuron activations more indistinguishable, reducing the purity of topological neighborhoods in the activation space. We expect that our tool can provide an effective way to diagnose the adversarial robustness of a model in the activation space and inspire new approaches to quantify the closeness of trained models to the theoretical limit (Zhang et al., 2022). We also envision that a topological understanding of weak regions will lead to new topologically inspired attacks and defenses.

Acknowledgement

We thank Archit Rathore for his work during the early stage of this project. This work was partially funded by NSF grants DMS-2134223, DMS-2134148, and IIS-2205418.

The work of Yi Zhou was supported in part by U.S. National Science Foundation under the grants CCF-2106216, DMS-2134223 and CAREER-2237830.

References

Brown, T., Mane, D., Roy, A., Abadi, M., and Gilmer, J. Adversarial patch. In *Workshop in Neural Information Processing Systems*, 2017.

Carter, S., Armstrong, Z., Schubert, L., Johnson, I., and Olah, C. Activation Atlas. *Distill*, 4(3):e15, 2019.

Chalapathi, N., Zhou, Y., and Wang, B. Adaptive covers for mapper graphs using information criteria. *IEEE International Conference on Big Data (IEEE BigData)*, 2021.

Chen, X., Liu, C., Li, B., Lu, K., and Song, D. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Ding, G. W., Sharma, Y., Lui, K. Y. C., and Huang, R. Mma training: Direct input space margin maximization through adversarial training. In *International Conference on Learning Representations*, 2020.

Edelsbrunner, H. and Harer, J. Persistent homology - a survey. In *Surveys on Discrete and Computational Geometry: Twenty Years Later*. American Mathematical Society, 2007.

Erhan, D., Bengio, Y., Courville, A., and Vincent, P. Visualizing higher-layer features of a deep network. Technical report, Univeristy of Montreal, 2009.

Ester, M., Kriegel, H.-P., Sander, J., and Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. *International Conference on Knowledge Discovery and Data Mining*, pp. 226–231, 1996.

Fawzi, A., Moosavi-Dezfooli, S.-M., Frossard, P., and Soatto, S. Empirical study of the topology and geometry of deep networks. *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2018.

Gabrielsson, R. B. and Carlsson, G. Exposition and interpretation of the topology of neural networks. In *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 1069–1076, 2019. doi: 10.1109/ICMLA.2019.00180.

Gao, R., Cai, T., Li, H., Hsieh, C.-J., Wang, L., and Lee, J. D. Convergence of adversarial training in overparametrized neural networks. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Gebhart, T., Schrater, P., and Hylton, A. Characterizing the shape of activation space in deep neural networks. *2019 18th IEEE International Conference On Machine Learning And Applications (ICMLA)*, pp. 1537–1542, 2019.

Goodfellow, I. J., Shlens, J., and Szegedy, C. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*, 2014.

Gu, T., Dolan-Gavitt, B., and Garg, S. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Guss, W. H. and Salakhutdinov, R. On characterizing the capacity of neural networks using algebraic topology. arXiv preprint arXiv:1802.04443, 2018.

He, K., Zhang, X., Ren, S., and Sun, J. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*, 2015.

He, W., Li, B., and Song, D. Decision boundary analysis of adversarial examples. *6th International Conference on Learning Representations*, 2018.

Hohman, F., Kahng, M., Pienta, R., and Chau, D. H. Visual analytics in deep learning: An interrogative survey for the next frontiers. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 2018.

Hohman, F., Park, H., Robinson, C., and Chau, D. H. P. Summit: Scaling deep learning interpretability by visualizing activation and attribution summarizations. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):1096–1106, 2020.

Karpathy, A. t-SNE visualization of CNN codes. https://cs.stanford.edu/people/karpathy/cnnembed/, 2014.

Khoury, M. and Hadfield-Menell, D. On the geometry of adversarial examples. *arXiv preprint arXiv:1811.00525*, 2018.

Kim, B., Wattenberg, M., Gilmer, J., Cai, C., Wexler, J., Viegas, F., and Sayres, R. Interpretability beyond feature attribution: Quantitative testing with concept activation vectors (TCAV). *International Conference on Machine Learning*, 2018.

Krizhevsky, A. Learning multiple layers of features from tiny images. Technical report, University of Toronto, 2009.

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

LeCun, Y., Cortes, C., and Burges, C. Mnist handwritten digit database. *ATT Labs [Online]. Available: http://yann.lecun.com/exdb/mnist*, 2, 2010.

Liang, X., Qian, Y., Huang, J., Ling, X., Wang, B., Wu, C., and Swaileh, W. Towards the desirable decision boundary by moderate-margin adversarial training. *arXiv preprint arXiv:2207.07793*, 2022.

Liu, B. and Shen, M. Some geometrical and topological properties of dnns' decision boundaries. *Theoretical Computer Science*, 908:64–75, 2022.

Madry, A., Makelov, A., Schmidt, L., Tsipras, D., and Vladu, A. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Nguyen, A., Yosinski, J., and Clune, J. Multifaceted feature visualization: Uncovering the different types of features learned by each neuron in deep neural networks. arXiv:1602.03616, 2016.

Purvine, E., Brown, D., Jefferson, B., Joslyn, C., Praggastis, B., Rathore, A., Shapiro, M., Wang, B., and Zhou, Y. Experimental observations of the topology of convolutional neural network activations. *Proceedings of the 37th AAAI Conference on Artificial Intelligence (AAAI)*, 2023.

Rade, R. and Moosavi-Dezfooli, S.-M. Reducing excessive margin to achieve a better accuracy vs. robustness trade-off. In *International Conference on Learning Representations*, 2022.

Rathore, A., Chalapathi, N., Palande, S., and Wang, B. TopoAct: Visually exploring the shape of activations in deep learning. *Computer Graphics Forum (CGF)*, 40(1): 382–397, 2021.

Rathore, A., Zhou, Y., Srikumar, V., and Wang, B. TopoBERT: Exploring the topology of fine-tuned word representations. *Information Visualization*, 2023.

Rieck, B. A., Togninalli, M., Bock, C., Moor, M., Horn, M., Gumbsch, T., and Borgwardt, K. Neural persistence: A complexity measure for deep neural networks using algebraic topology. In *International Conference on Learning Representations (ICLR 2019)*, 2019.

Simonyan, K., Vedaldi, A., and Zisserman, A. Deep inside convolutional networks: Visualising image classification models and saliency maps. *Workshop at International Conference on Learning Representations,*, 2014.

Singh, G., Mémoli, F., and Carlsson, G. Topological methods for the analysis of high dimensional data sets and 3D object recognition. *Eurographics Symposium on Point-Based Graphics*, 22, 2007.

Sinha, A., Namkoong, H., and Duchi, J. Certifiable distributional robustness with principled adversarial training. In *International Conference on Learning Representations*, 2018.

Su, J., Vargas, D. V., and Sakurai, K. One pixel attack for fooling deep neural networks. *IEEE Transactions on Evolutionary Computation*, 23(5):828–841, 2019.

Szegedy, C., Zaremba, W., Sutskever, I., Bruna, J., Erhan, D., Goodfellow, I., and Fergus, R. Intriguing properties of neural networks. *arXiv preprint arXiv:1312.6199*, 2013.

Tauzin, G., Lupo, U., Tunstall, L., Pérez, J. B., Caorsi, M., Medina-Mardones, A., Dassatti, A., and Hess, K. giotto-tda: A topological data analysis toolkit for machine learning and data exploration. *Journal of Machine Learning Research*, 22:1–6, 2020.

Wang, Y., Zou, D., Yi, J., Bailey, J., Ma, X., and Gu, Q. Improving adversarial robustness requires revisiting misclassified examples. In *International Conference on Learning Representations*, 2020.

Wheeler, M., Bouza, J., and Bubenik, P. Activation landscapes as a topological summary of neural network performance. In *IEEE International Conference on Big Data (Big Data)*, pp. 3865–3870, 2021.

Xian, X., Wang, G., Srinivasa, J., Kundu, A., Bi, X., Hong, M., and Ding, J. Understanding backdoor attacks through the adaptability hypothesis. *International Conference on Machine Learning (ICML)*, 2023.

Xu, Y., Sun, Y., Goldblum, M., Goldstein, T., and Huang, F. Exploring and exploiting decision boundary dynamics for adversarial robustness. *arXiv preprint arXiv:2302.03015*, 2023.

Yosinski, J., Clune, J., Nguyen, A., Fuchs, T., and Lipson, H. Understanding neural networks through deep visualization. *Deep Learning Workshop at the 31st International Conference on Machine Learning*, 2015.

Zhang, J., Zhu, J., Niu, G., Han, B., Sugiyama, M., and Kankanhalli, M. Geometry-aware instance-reweighted adversarial training. *International Conference on Learning Representations*, 2021.

Zhang, J., Ding, J., and Yang, Y. Is a classification procedure good enough?—a goodness-of-fit assessment tool for classification learning. *Journal of the American Statistical Association*, pp. 1–11, 2022.

Zhou, Y. Source Code for Visualizing and Analyzing the Topology of Neuron Activations in Deep Adversarial Training. https://github.com/MapperInteractive/MapperInteractive/tree/MIAT, 2023.

Zhou, Y., Chalapathi, N., Rathore, A., Zhao, Y., and Wang, B. Mapper Interactive: A scalable, extendable, and interactive toolbox for the visual exploration of high-dimensional data. *Proceedings of IEEE 14th Pacific Visualization Symposium (PacificVis)*, 2021.

A. Supplementary Material

To compute the mapper graphs, we hand-tuned the mapper parameters. For MNIST data, $n = 40$ and $p = 50\%$. For CIFAR-10 data, $n = 40$ and $p = 25\%$. For DBSCAN, $minPts = 5$, and the size of the neighborhood ϵ is determined by the "elbow" approach (see (Zhou et al., 2021) for its details).

A.1. Details on Adversarial Training

In experiments, we train the standard adversarial model M_{adv} with the projected gradient descent (PGD) type attack (Madry et al., 2018). We explore the performance of M_{adv} against two variations of perturbation bound: ℓ_∞ and ℓ_2. For MNIST data, we set the learning rate to be 0.01, and for CIFAR-10 data, the learning rate is 0.1. We train each instance of M_{adv} for 200 epochs.

To perform the model refinement, we first identify weak regions in the mapper graph of activation vectors from the training dataset. We then train M_{refine} using only the training images that belong to these weak regions. We train each instance of M_{refine} for 50 epochs. To compute the test accuracy, we compare each activation vector from the test dataset with its nearest neighbor of the activations from the training data. If the nearest neighbor belongs to a weak region, we identify the test activation vector as in the weak region as well. For test images belonging to weak regions, we use M_{refine} to predict their labels. For test images not belonging to weak regions, we use M_{adv} to predict their labels. The overall prediction accuracy is then calculated by adding the number of correctly predicted images from both M_{refine} and M_{adv}, and dividing it by the total number of images in the test dataset.

A.2. Additional Experiments on Training MLP with MNIST for Model Refinement

Figure 8 shows the mapper graphs of neuron activations constructed using a uniform cover (left) and a balanced cover (right), respectively. In a standard training with the clean data, the model's corresponding mapper graphs contain clear bifurcations, where different classes of images are located in separate branches of the mapper graph.

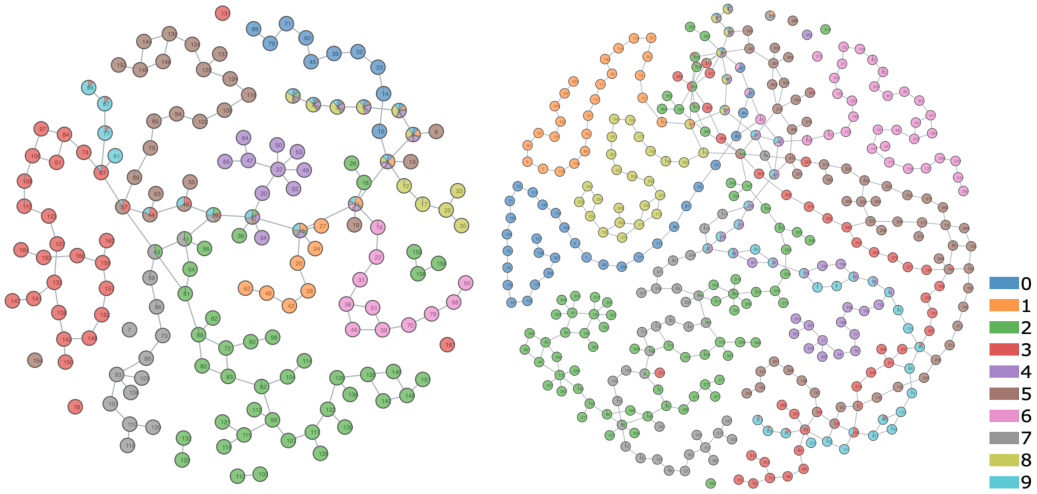

Figure 8. Mapper graphs generated with a uniform cover (left) and a balanced cover (right). MNIST with MLP M_{clean}.

When we subject the model to ℓ_∞-PGD attacks across different levels, the corresponding mapper graphs are shown in Figure 9 with a uniform cover and Figure 10 with a balanced cover, where weak regions are highlighted by red circles.

When we subject the model to ℓ_2-PGD attacks across different levels, the corresponding mapper graphs are shown in Figure 11 with a balanced cover, where weak regions are highlighted by red circles.

A.3. Training ResNet-18 with CIFAR-10

We train a ResNet-18 with the CIFAR-10 dataset, where the clean model M_{clean} has a test accuracy of 93.28%. Again, we perform model refinement by leveraging the misclassified samples in the weak regions.

Figure 12 shows the mapper graphs of neuron activations from M_{clean} constructed using a uniform cover (left) and a balanced

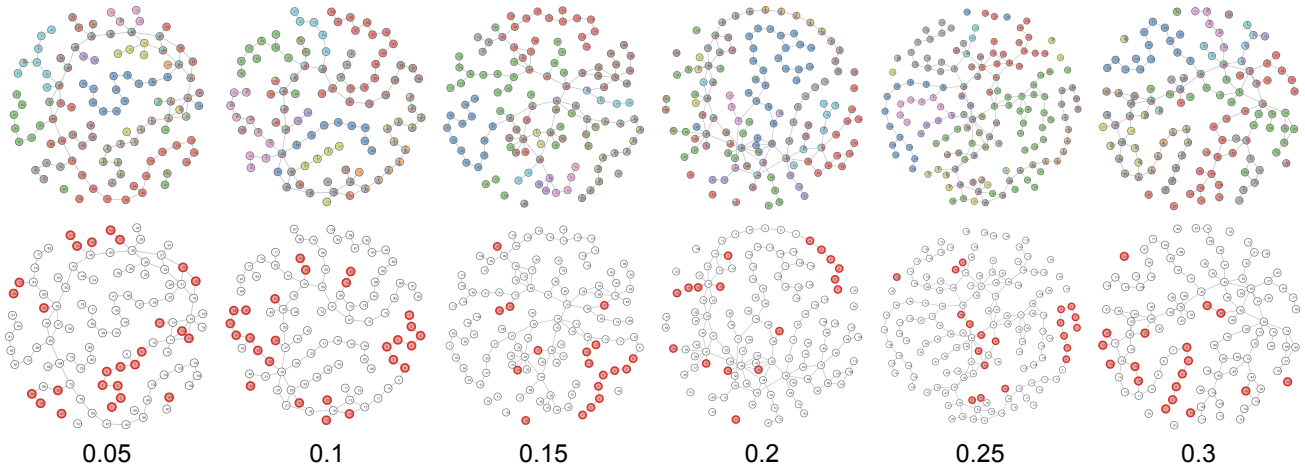

Figure 9. Mapper graphs using a uniform cover, MNIST with MLP under ℓ_∞-PGD attack (top). Weak regions are highlighted in red circles (bottom).

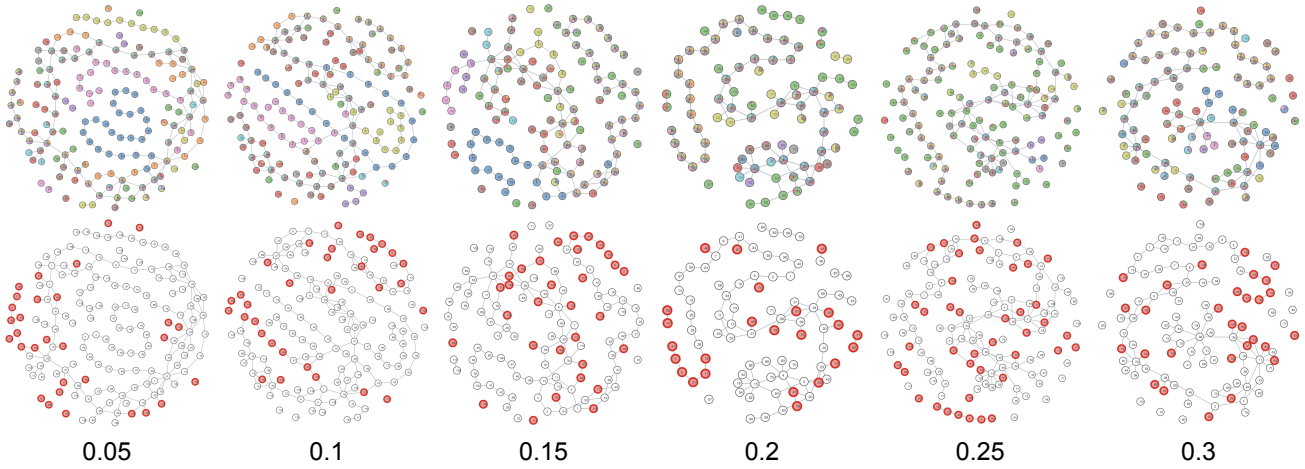

Figure 10. Mapper graphs using a balanced cover, MNIST with MLP under ℓ_∞-PGD attack (top). Weak regions are highlighted in red circles (bottom).

cover (right), respectively. They are shown to contain clear bifurcations that separate different image classes into branches.

Table 3 compares the (robust) test accuracy achieved by the models M_{adv} and M_{refine}, respectively, under different levels of ℓ_∞-PGD attack. We do not observe improved robust accuracy. Similar observations can be obtained under different levels of ℓ_2-PGD attacks shown in Table 4. This observation shows that for ResNet-18 with the CIFAR-10, the topological structure in the mapper graph is not effective in helping identify the weak regions containing samples vulnerable to adversarial attacks.

To dive deeper into the structures of the mapper graphs when the model is under ℓ_∞- or ℓ_2-PGD attacks, we highlight the mapper graphs with identified weak regions in Figure 13 and Figure 14. Topological neighborhoods under such a setting typically have low purity but high training accuracy, indicating that the model is overfitting, thus making it difficult to identify weak regions useful for model refinement.

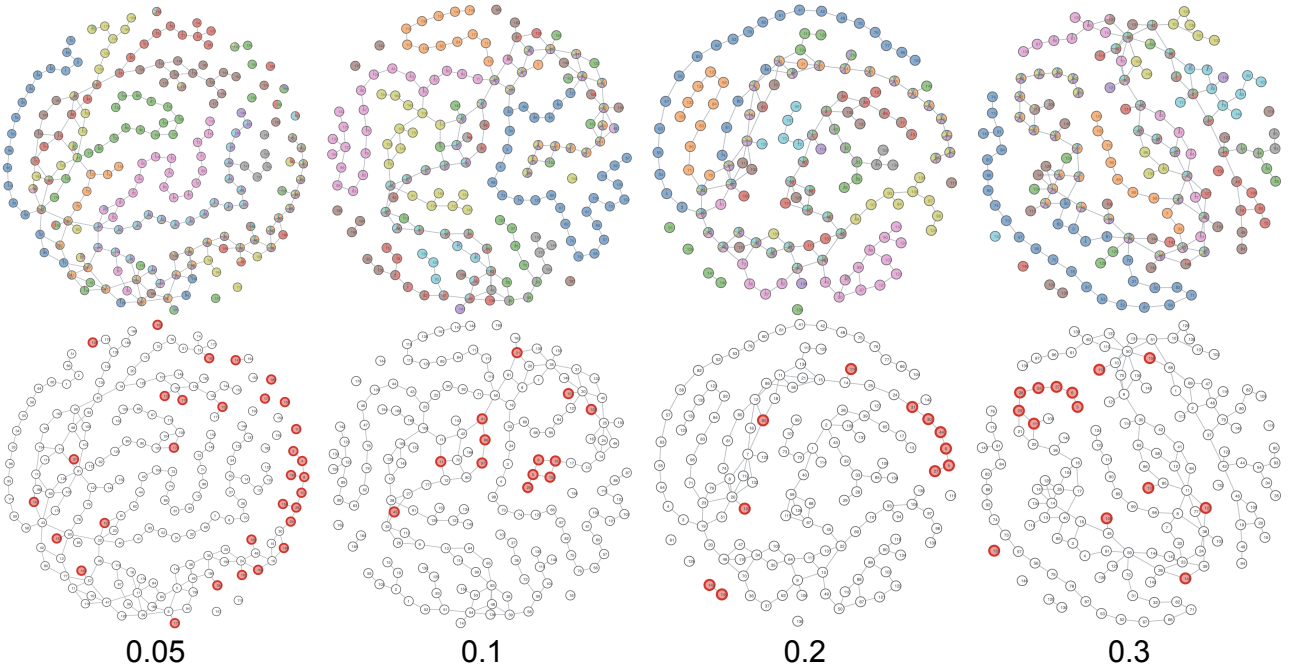

Figure 11. Mapper graphs using a balanced cover, MNIST with ℓ_2-PGD attack (top). Weak regions are highlighted in red circles (bottom).

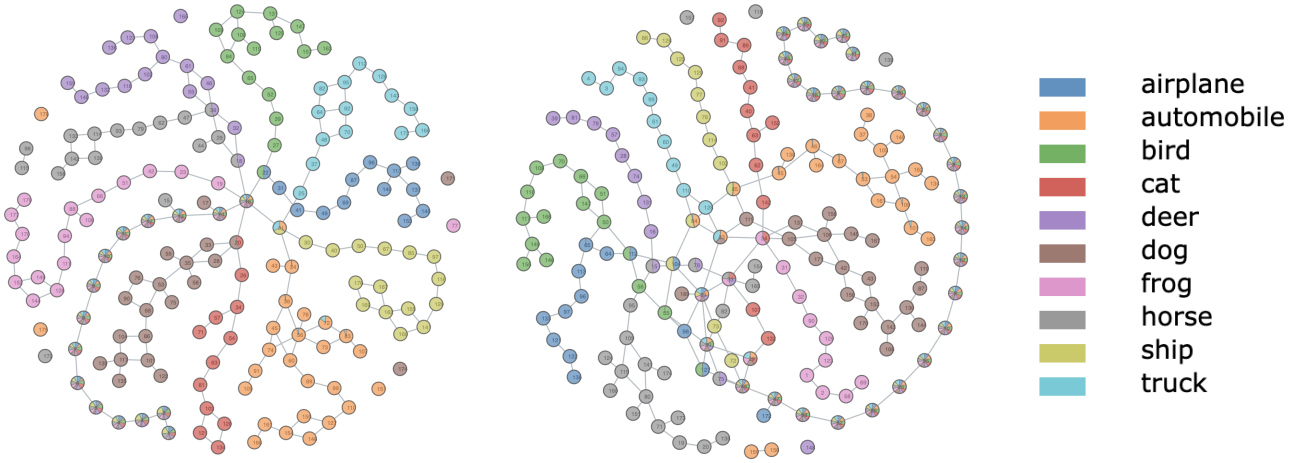

Figure 12. Mapper graphs generated with a uniform cover (left) and a balanced cover (right). CIFAR-10 with ResNet-18 M_{clean}.

Attack level ϵ	0.01	0.02	0.03	0.05
M_{adv}	75.210 (0.156)	58.535 (0.049)	46.585 (0.092)	39.060 (0.240)
M_{refine} (balanced cover)	75.170 (0.085)	58.250 (0.339)	46.415 (0.120)	38.920 (0.113)
Improvement	-0.040 (0.071)	-0.285 (0.290)	-0.170 (0.212)	-0.140 (0.354)

Table 3. Robust accuracy: average and standard deviation of robust test accuracy for refined models. CIFAR-10 with ResNet-18 under ℓ_∞-PGD attack. Standard deviations (in parentheses) are obtained using two seeds in the initialization.

Attack level ϵ	0.01	0.05	0.1	0.2	0.3
M_{adv}	94.015 (0.163)	91.150 (0.156)	88.415 (0.092)	83.365 (0.233)	79.410 (0.226)
M_{refine} (balanced cover)	93.990 (0.170)	91.085 (0.148)	88.225 (0.078)	83.250 (0.184)	78.470 (0.891)
Improvement	-0.025 (0.007)	-0.065 (0.007)	-0.190 (0.014)	-0.115 (0.049)	-0.940 (0.665)

Table 4. Robust accuracy: average and standard deviation of robust test accuracy for refined models. CIFAR-10 with ResNet-18 under ℓ_2-PGD attack. Standard deviations (in parentheses) are obtained using two seeds in the initialization.

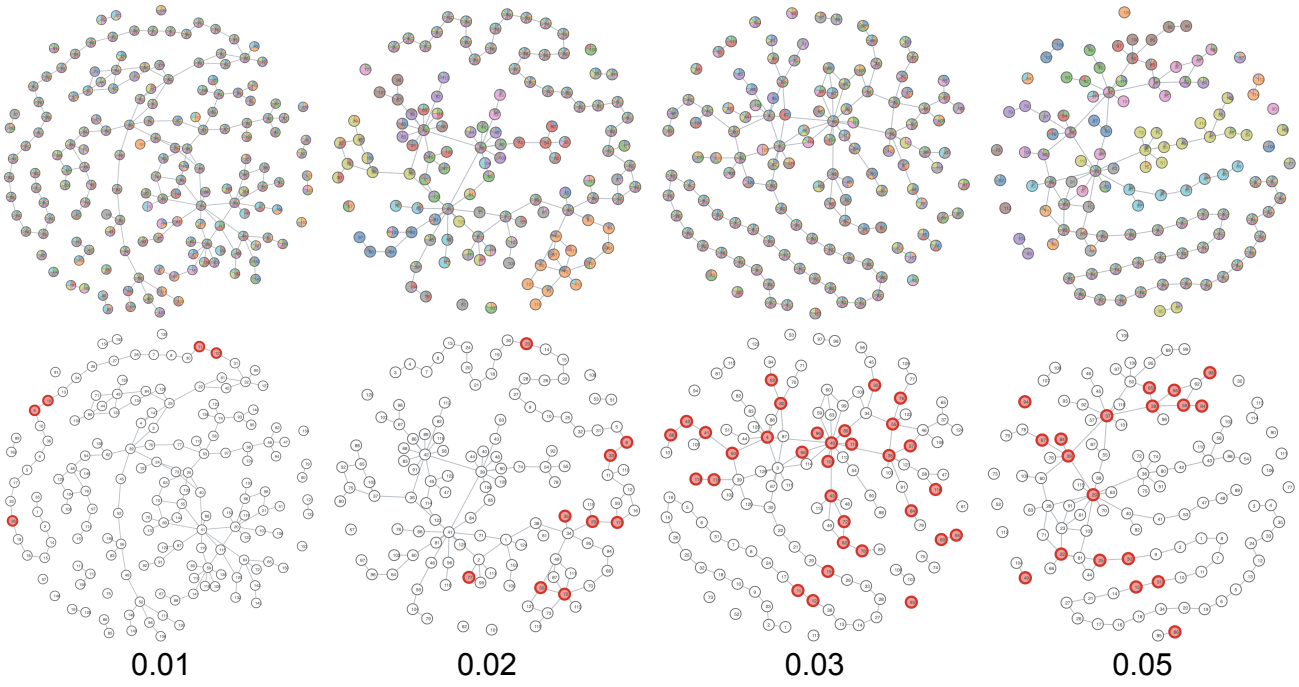

Figure 13. Mapper graphs using a balanced cover, CIFAR-10 with ResNet-18 under ℓ_∞ PGD attacks (top). Weak regions are highlighted in red circles (bottom).

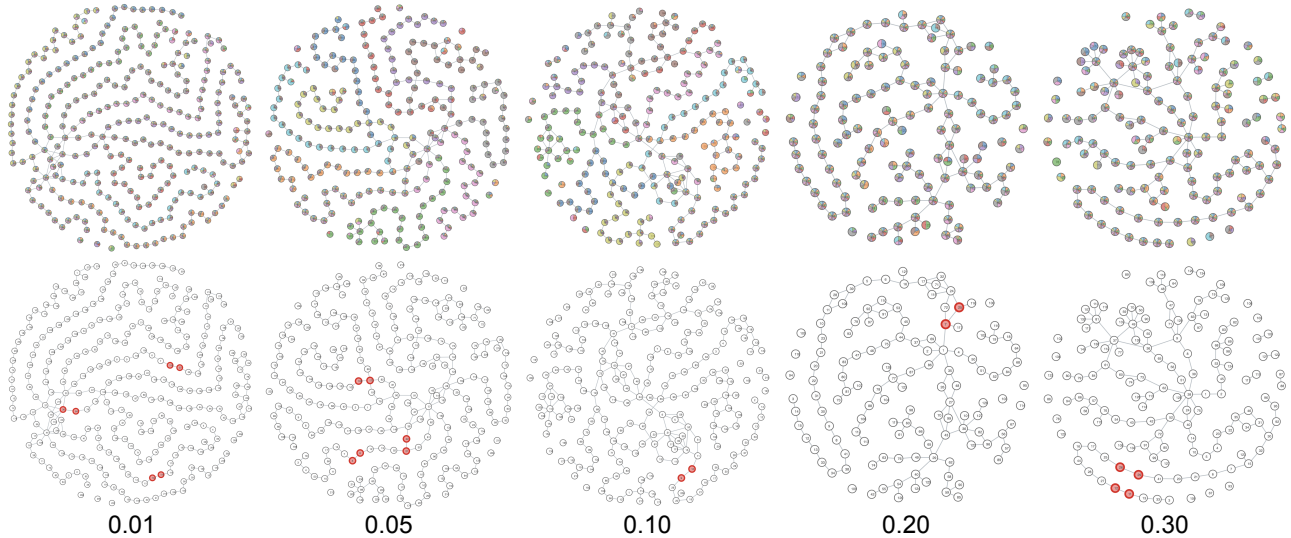

Figure 14. Mapper graphs using a balanced cover, CIFAR-10 with ResNet-18 under ℓ_2-PGD attacks (top). Weak regions are highlighted in red circles (bottom).