TACO-Net: Topological Signatures Triumph in 3D Object Classification

Anirban Ghosh and Ayan Dutta

School of Computing
University of North Florida
Florida, USA
Email: anirban.ghosh, a.dutta@unf.edu

Abstract. 3D object classification is a crucial problem due to its significant practical relevance in many fields, including computer vision, robotics, and autonomous driving. Although deep learning methods applied to point clouds sampled on CAD models of objects and/or captured by LiDAR or RGBD cameras have achieved remarkable success in recent years, achieving high classification accuracy remains a challenging problem due to the unordered point clouds and their irregularity and noise. To this end, we propose a novel state-of-the-art (SOTA) 3D object classification technique that combines topological data analysis with various image filtration techniques to classify objects when they are represented using point clouds. We transform every point cloud into a voxelized binary 3D image to extract distinguishing topological features. Next, we train a lightweight one-dimensional Convolutional Neural Network (1D CNN) using the extracted feature set from the training dataset. Our framework, `TACO-Net`, sets a new state-of-the-art by achieving 99.05% and 99.52% accuracy on the widely used synthetic benchmarks ModelNet40 and ModelNet10. It also achieved one of the highest accuracies among all the from-scratch supervised learning techniques when tested on the hardest variant of the ScanObjectNN dataset. When tested with ten different kinds of corrupted ModelNet40 inputs, the proposed `TACO-Net` demonstrates strong resiliency overall. The code can be found in the supplementary material.

1 Introduction

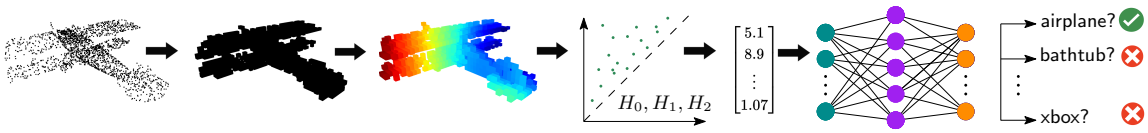


Fig. 1: An airplane point cloud (from ModelNet40) is converted into a 3D binary image, which is then transformed into a set of 3D grayscale images (just one is shown) using different filtration techniques. Every grayscale image admits a separate cubical persistence. A feature vector of length 36 is obtained for every persistence. The vectors are then concatenated to form the final feature vector for the plane. The feature vectors are finally trained using a 1D CNN for object classification.

Semantic object recognition is one of the most fundamental capabilities that modern-day autonomous systems, such as robots and cars, demand to operate in dynamic real-world environments. Given an input point cloud, the objective is to classify the input into one of the known categories [31,

33,52]. Such object classification is critical in numerous real-world applications, including autonomous driving, robotics, augmented reality, and 3D scene understanding. In recent years, deep learning techniques have achieved remarkable success in the 3D object classification task by using various modes of inputs such as voxels [31], multi-views [23], and raw point clouds [33], to name a few.

Unlike 2D images, point clouds provide rich geometric and spatial information in three dimensions, enabling more precise object recognition in complex environments [37]. Further, unlike 2D images, point clouds enable a mobile robot, for example, to recognize objects in various ambient light and weather conditions efficiently. However, point cloud-based object classification remains challenging due to the unstructured and sparse nature of point cloud data, which lacks a regular grid structure and often suffers from noise, occlusion, and varying point densities [2, 42, 46]. Furthermore, the permutation invariance of points and the need to capture both local and global geometric features add layers of complexity to model design [60]. These factors demand innovative methodologies that effectively learn from unordered and irregular 3D geometric data, driving continued research and development in this field.

Most of the existing approaches use a deep machine learning framework where the input is either a set of pictures of the object, a raw set of points, or volumetric shape of the object [13, 23, 31, 33–35, 41]. Unlike these, we voxelize any n-point 3D object cloud once to compute persistent-homology signatures via cubical persistence, reframing topological descriptors as a universal substrate for deep representation learning. From these fixed, pre-computed signatures, our lightweight 1D CNN learns a highly structured, class-disentangled latent space for 3D shape recognition, yielding strong separability without requiring gradients to flow through the front-end topological data analysis (TDA) computations. The 1D CNN is then trained end-to-end on the resulting topological feature vectors for the final classification task. See Fig. 1 for a system-level overview of our approach. Although TDA has a direct connection with shapes, surprisingly, TDA has not been successfully used for large-scale 3D object classification. Similar to the challenges associated with designing deep learning models using existing network layers, finding an effective TDA pipeline presents a challenge. First, we test the proposed **t**opological data **a**nalysis-based obje**c**t classificati**o**n framework, named `TACO-Net`, on ModelNet40 and ModelNet10 datasets. Our experiments show that we achieve SOTA accuracy for both these datasets. Furthermore, we have chosen ten common corruptions to test the robustness of `TACO-Net`. The corrupted dataset is tested on the model trained with the uncorrupted ModelNet40 dataset.

Results show that the proposed `TACO-Net` yields high accuracy for all but one corruption type at a low level while achieving moderate accuracy with highly corrupted test data. Further, when tested on a real-world dataset, namely OmniObject3D [51], `TACO-Net` again achieved the highest accuracy. We tested `TACO-Net` on the six popular variants of ScanObjectNN. It achieved the second-highest accuracy of 91.87% among all the existing supervised learning techniques for the hardest variant PB_T50_RS. For the OBJ_BG and OBJ_ONLY variants, it achieved the SOTA accuracies of 95.08% and 94.01%, respectively. To show the generalizability of `TACO-Net`, we tested it on two 3D medical object datasets, namely VesselMNIST3D [56] and AdrenalMNIST3D [55], where `TACO-Net` surpassed the highest accuracies and F1-scores of numerous existing techniques such as PointNet [33], PointNet++ [34], and DGCNN [49], among others. The main contributions of our paper are as follows.

- To our knowledge, this is the first 3D classification framework that learns a discriminative latent space over fixed cubical persistent-homology signatures using a lightweight 1D CNN that effectively internalizes and amplifies non-learned topological descriptors.

- Our proposed `TACO-Net` framework achieves state-of-the-art accuracy on ModelNet10/40, and two challenging real-world benchmarks - OBJ_BG and OBJ_ONLY of ScanObjectNN - demonstrating that it effectively learns class-disentangled structure over topological inputs, and while performing competitively on the hardest variant of ScanObjectNN.
- `TACO-Net` demonstrates strong robustness to severe point-cloud corruptions and generalizes across multiple real-world benchmarks and their variants, a scale of cross-dataset validation rarely reported in 3D object-classification literature.

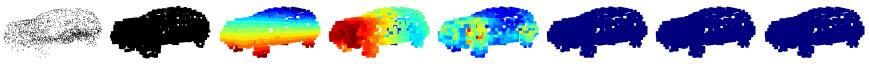
Related Works. Three main types of approaches are prevalent in the 3D object classification literature: voxel-based, multi-view imaging-based, and raw point-based [37]. Many hybrid methods combine one or more of the above-mentioned techniques. In voxel-based methods, features of the volumetric representation of input point clouds are learned and classified [31]. One of the earliest approaches in this direction is 3D Shapenets [52]. Although the objects to be classified are in 3D, taking 2D pictures of them from various angles and classifying those 2D pictures instead has gained attention through MVCNN by Su et al. [41], where they used 80 pictures of each 3D object. GVCNN [13] improved upon MVCNN by using only 8 images. MHBN [57], on the other hand, used only 6 views of the object, but managed to achieve high mean class accuracy. Usually, convolutional neural networks are used for these 2D image classification techniques. In [29], the authors have used a recurrent neural module along with CNNs. Hypergraph learning has been highly effective for object classification, as shown in [12]. One of the highest accuracy yielding approaches, RotationNet [23], also uses multiple views of the objects, albeit these are unsupervised viewpoints. Point-based approaches are most popular - they take raw point clouds as inputs and learns from their unstructured format, which makes them robust against corruption [27, 33, 34]. One of the pioneering works in this direction is PointNet [33]. PointNet++ [34] improved upon PointNet by capturing local geometric structures. Transformer-based learning strategies have received attention as well [61]. Graph neural networks have been successful in classifying point cloud objects [32, 49]. Compared to these, our novel methodology extracts topological features, which are learned by a 1D CNN for object classification and achieves SOTA accuracy.

This paper uses TDA features extracted from the point clouds for object classification. Such a TDA-based approach has been previously used for MNIST data classification [15]. TDA has recently been used to solve a diverse range of problems, such as in medical imaging [39], biomedicine [40], oncology [6], and cybersecurity [1].

2 Description of `TACO-Net`

The TDA front-end of `TACO-Net` is a *universal, non-learning, dataset-agnostic pipeline* that can process any input point cloud by voxelizing it within its own axis-aligned bounding box (AABB), producing a binary cubical complex independent of global pose or scale normalization. The voxel grid is then transformed via a symmetric family of task-agnostic 3D filtrations (height, radial, density, erosion, dilation, and signed distance), each designed to probe complementary aspects of the cubical topology, yielding grayscale volumes on which cubical persistent homology is computed. From each persistence diagram, we extract stable topological summary statistics, including persistent entropy and multiple amplitudes (Wasserstein, Bottleneck, Betti curve, persistence landscape, and heat kernels), whose stability and injectivity properties ensure robust, shape-sensitive encodings. The pipeline, therefore, acts as a universal topological *shape sensor*, while the subsequent lightweight 1D CNN learns higher-order interactions over these fixed signatures to produce highly class-separable representations. We present a diagram of the `TACO-Net` pipeline in Fig. 4.

Filtration types [15]. Let $\mathcal{B} : I \subseteq \mathbf{Z}^3 \to \{0, 1\}$ be a 3D binary image, where every $p \in I$ is a voxel. A voxel is *activated* if its value is 1; otherwise, it is *deactivated*. A grayscale filtration converts $\mathcal{B}$ into a grayscale 3D image $\mathcal{G} : I \subseteq \mathbf{Z}^3 \to \mathbf{R}$. Such filtrations can highlight different topological features in the binary image, even visually. We briefly describe the six kinds of filtrations used in `TACO-Net` to obtain a set of grayscale 3D images for every 3D binary image (constructed for every point cloud) for extracting topological feature vectors. Owing to the difference in the filtration functions, every filtration tends to highlight different features of $\mathcal{B}$. See Fig. 2 for an illustration.



(a) (b) (c) (d) (e) (f) (g) (h)

Fig. 2: A 2048-element point cloud of a car from ModelNet40 is shown in (a) and its 3D binary image with voxel size 0.05 in (b). The grayscale images obtained after height using $v : (-1, 0, 0)$, radial using $c : (4, 4, 10)$, density, dilation, erosion, and signed distance filtrations, are shown in (c), (d), (e), (f), (g), (h), respectively. Hotter voxels have higher grayscale values. For brevity, voxels outside the shape are not shown; consequently, (f), (g), and (h) appear almost the same.

Height filtration. It needs a direction vector v in 3-space. Every activated voxel $p \in \mathcal{B}$ is assigned a grayscale value that equals the distance between p and the hyperplane defined by v. Every deactivated voxel is assigned the maximum distance between any voxel of $\mathcal{B}$ and the hyperplane defined by v, plus one. For `TACO-Net`, we have considered the 26 direction vectors in $\{\{0, 1, -1\}^3\} \setminus \{(0, 0, 0)\}$.

Radial filtration. A reference voxel c, called *center*, is supplied. Every activated voxel $p \in \mathcal{B}$ is assigned the distance between p and c. The deactivated voxels are assigned the maximum distance between c and any voxel, plus one. For `TACO-Net`, 27 centers $c_1, c_2, \ldots, c_{27}$ have been considered, chosen as the 27 vertices of a $3 \times 3 \times 3$ grid $\Xi_{\mathcal{B}}$ inside $\mathcal{B}$. We note that $\mathcal{C}_1 := [c_1, \ldots, c_9]$ belong to the first vertical slice of $\Xi_{\mathcal{B}}$ having the lowest x-coordinate, $\mathcal{C}_2 := [c_{10}, \ldots, c_{18}]$ the median, and $\mathcal{C}_3 := [c_{19}, \ldots, c_{27}]$ the highest. The centers are sorted lexicographically in every $\mathcal{C}_i$. Refer to Fig. 3 for an example.

The strength of our 26 height directions is that directional height filtrations discretize the Persistent Homology Transform [45], which is injective on a broad class of shapes; hence, in principle, the family of height persistence diagrams determines the underlying shape with high accuracy, as shown later empirically. Our cube-symmetric set of 26 directions provides an efficient spherical sampling that harmonizes with cubical complexes, producing salient and stable topological events that differ across object categories. By the stability of persistent homology, these diagrams are robust to noise. Complementing height with a set of 27 radial filtrations, from carefully placed centers, and the following four filtrations, injects information about interior organization, yielding consistent gains. This is corroborated by the high accuracy achieved in shape classification (see Sec. 3).

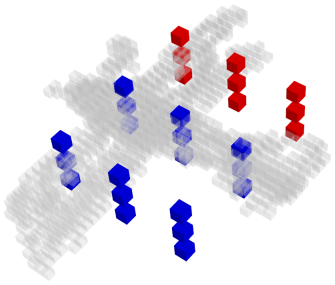


Fig. 3: The 27 radial centers are shown for an airplane 3D binary image. For ModelNet40 and ModelNet10, we have used the centers $c_1, \ldots, c_{18}$, as shown in blue.

Density filtration. Every voxel $p \in \mathcal{B}$ is assigned a grayscale value equal to the number of activated voxels within a ball centered at p having radius r. We fixed r to 1 in our experiments.

Dilation filtration. Every voxel $p \in \mathcal{B}$ is assigned a grayscale value equal to the smallest Manhattan distance to an activated voxel in $\mathcal{B}$. Consequently, active voxels are assigned a 0 grayscale value.

Erosion filtration. It does the opposite of the dilation filtration. The dilation filtration is applied to the binary image $\mathcal{B}'$, obtained from $\mathcal{B}$ by changing activated voxels to deactivated and deactivated ones to activated. Deactivated voxels are assigned a 0 grayscale value.

Signed distance filtration. For every activated voxel $p \in \mathcal{B}$, its grayscale value is the minimum Manhattan distance between p and any deactivated voxel in $\mathcal{B}$ minus 1. For every deactivated voxel, its grayscale value is the negative of the minimum Manhattan distance between p and any activated voxel in $\mathcal{B}$.

Synergy between the filtrations. Rather than treating filtrations as isolated tools, `TACO-Net` leverages their complementary strengths to build a unified pipeline. Height filtrations encode global shape cues by discretizing the Persistent Homology Transform, providing directional sensitivity and robustness to pose variations. Radial filtrations complement this by capturing interior organization and symmetry, injecting structural information that height alone cannot provide. Further shape understanding is achieved through four additional filtrations: density, dilation, erosion, and signed distance (collectively DEDS), each contributing unique geometric insights. Density filtration emphasizes local clustering by assigning grayscale values based on neighborhood occupancy, making it effective for distinguishing compact versus elongated shapes. Dilation filtration measures proximity to active voxels, highlighting external boundaries and object thickness, while erosion filtration does the opposite, revealing hollow regions and internal voids. Finally, signed distance filtration encodes both interior and exterior distances with positive and negative values, producing a nuanced representation of boundary smoothness and depth. This interplay enables TACO-Net to learn a holistic representation of 3D objects, making it robust to noise and generalizable across diverse datasets.

TDA [7, 50] can extract topological information and geometric patterns from datasets using algebraic topology. Persistent homology [10, 62] is a popular tool in TDA, applied to obtain different kinds of topological feature vectors of point clouds. It helps to understand the shape of a point cloud by tracking the birth and death of various topological features (different from feature vectors), such as connected components, holes, and higher-dimensional voids that persist at different scales during an iterative process known as *filtration* (distinct from the six types of filtration mentioned above). A series of nested geometric structures is obtained at various scales during filtration. The topological features that persist (survive) across several iterations of a filtration, inside various sequences of such nested structures, can be used as topological descriptors to compute different topological feature vectors of a point cloud. For a comprehensive overview on persistent homology and various filtration techniques, we urge interested readers to refer to [7, 10, 50, 62]. We use cubical homology and persistence, which are meant for extracting topological information from cubical complexes.

Cubical homology and persistence. [22, 47] A finite *cubical complex* in 3-space is a union of points, line segments, squares, 3D cubes, aligned on the grid $\mathbf{Z}^3$. We leverage *cubical homology*, a variant of persistent homology meant for cubical complexes, to obtain topological feature vectors of grayscale 3D images, which are obtained from 3D binary images, constructed for every point cloud. Any grayscale 3D image can be perceived as a cubical complex K, where every voxel (a pixel

in 3D) is a cube with an intensity value. The voxels, square faces of voxels, edges, and vertices are $3, 2, 1, 0$-cube, respectively. Hence, cubical homology can be applied directly to 3D grayscale images because of their natural grid-like structures. During filtration, the voxels are added in order of increasing intensity, forming a sequence of nested cubical subcomplexes. Starting with the lowest voxel intensity, all voxels with intensity at most t are added to the current cubical complex along with their faces, edges, and vertices, where t is the current voxel intensity being considered. A cubical complex obtained at step i is a subcomplex of the complex obtained at step $i+1$. Thus, we get a sequence of nested subcomplexes $K_0 \subseteq K_1 \subseteq \ldots \subseteq K_m$. Every K_i is called a *sublevel set* of K, the cubical complex built from a given 3D image, since $K_i \subseteq K$.

As voxels are added, the topological features, connected components of voxels (homology group H_0), tunnels or loops (homology group H_1), and enclosed cavities (homology group H_2), take birth or die. Every birth and death of a feature introduces a new birth-death pair in the *cubical persistence*, a multiset of points in $\mathbf{R} \times (\mathbf{R} \cup \{+\infty\})$, where every pair (b, d) in the multiset denotes the birth of a topological feature at time b and its death at time d. Long-surviving features are likely the significant features that can be used in classification tasks. Persistence, represented using a 2D scatter plot, is known as a *persistence diagram* (refer to Fig. 1). In a persistence diagram, every birth-death pair corresponds to a point in the diagram. The cubical persistence of a cubical complex is its *topological signature*. Next, we discuss the topological vectorization methods.

Persistent entropy. [9] Given a cubical persistence, $X = \{(b_i, d_i)\}$, its persistent entropy, denoted by $\rho(X)$, is a real number defined by as, $\rho(X) = -\sum_i p_i \log(p_i)$, where $p_i = \frac{d_i - b_i}{\ell(X)}$, and $\ell(X) = \sum_i (d_i - b_i)$. Having its roots in information theory, it gives an intuitive sense of disorder or complexity in the topological structure. We note that 3 real numbers are obtained for the 3 homology groups, $H_0, H_1,$ and H_2.

Amplitude. Introduced in [15], *amplitude* of a cubical persistence is defined as its distance to the empty persistence (devoid of birth-death pairs). It is used to compare two cubical persistences, obtained from two different 3D grayscale images. `TACO-Net` uses five types of amplitudes with varied parameters. Out of the five, two are metric-based (the Wasserstein and Bottleneck distances), and the remaining three are kernel-based (Betti curve, persistence landscape, and heat). For the Betti curve, and persistence landscape, the diagrams are sampled using 100 filtration values, whereas for the heat kernel, 20 are used. Let $X = \{(b_i, d_i)\}$ be a cubical persistence. For insight into the different kinds of amplitudes, we refer the reader to the Appendix.

p-Wasserstein [44]. The *half-lifetime* of a pair $(b_i, d_i) \in X$ is defined as $\frac{d_i - b_i}{2}$. The Wasserstein amplitude of order p, denoted by $W(X, p)$, is defined as the L_p norm of the vector of half-lifetimes of the birth-death pairs in X. Hence, $W(X, p) = (\sum_i \left(\frac{d_i - b_i}{2}\right)^p)^{1/p}$. For `TACO-Net`, we have used $p = 1, 2$. We obtain 6 real numbers for this metric, since there are 3 homology groups and 2 values of p.

Bottleneck [44]. The Bottleneck amplitude is denoted by $B(X) = W(X, \infty)$. We obtain 3 real numbers for this metric due to the three homology groups.

Betti curve. [44] The Betti curve of X is the function $B_C : \mathbf{R} \to \mathbf{N}$, such that $B_C(s)$ gives the number of birth-pairs in X that contains s when every pair (b_i, d_i) in X is treated as an interval. Two amplitudes are obtained using the L_1 and L_2 norms. We obtain 6 real numbers for this metric, since there are three homology groups and two norms.

$$f_{(b_i,d_i)}(x) = \begin{cases} 0 & \text{if } x \notin (b_i, d_i) \\ x - b_i & \text{if } x \in (b_i, \frac{b_i + d_i}{2}] \\ -x + d_i & \text{if } x \in (\frac{b_i + d_i}{2}, d_i) \end{cases} \quad (1)$$

Landscape [4, 5]. For a birth-death pair $(b_i, d_i) \in X$, let $f_{(b_i,d_i)} : \mathbf{R} \to [0, \infty]$, be a piecewise linear function given in Eq. 1. The *persistence landscape* of X is the sequence of functions $\lambda_k : \mathbf{R} \to [0, \infty], k = 1, 2, 3, \ldots$ where $\lambda_k(x)$ is the k-th largest value of $\{f_{(b_i,d_i)}(x)\}_i$. Further, $\lambda_k(x)$ is set to 0 if the k-th largest value does not exist. The parameter k is called the *layer*. For `TACO-Net`, we have used $k = 1, 2$. Four amplitudes are obtained using L_1 and L_2 norms for both the values of k. We get 12 real numbers for this metric, since there are three homology groups, two norms, and two distinct values of k.

Heat kernel [36]. Gaussians of standard deviation σ are placed over every point in X and a negative Gaussian of σ on the mirror point across the diagonal line in the persistence diagram. Thus, a real-valued function is obtained on $\mathbf{R}^2$. For `TACO-Net`, we have used $\sigma = 0.15$. We get 6 real numbers for this metric, since there are three homology groups and two norms, L_1, L_2.

Hence, for a given 3D grayscale image, obtained by using a filtration, we get a feature vector of length $3 + 33 = 36$, wherein 3 numbers are obtained using persistent entropy and the remaining 33 using amplitude.

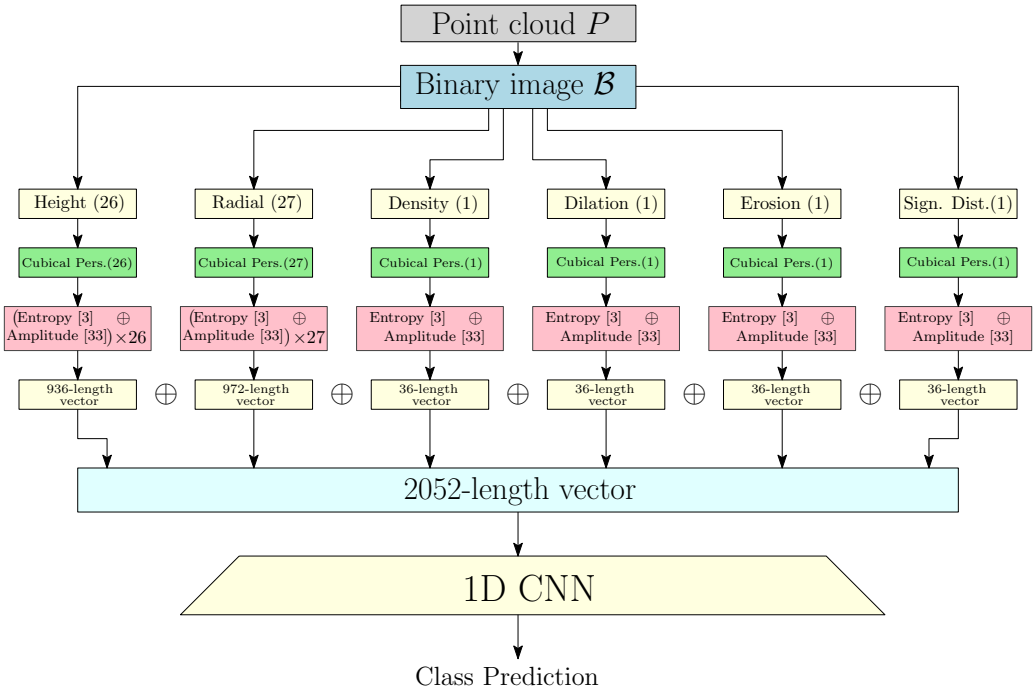


Fig. 4: An illustration of the novel pipeline of `TACO-Net`. The numbers in the parentheses denote the number of variants. For instance, 'Height (26)' implies that 26 variants of the height filtration have been used. The integers inside the square braces denote vector length. For example, Entropy[3] implies that applying entropy to cubical persistence yields a vector of length 3. Further, $\oplus$ denotes vector concatenation.

Feature selection and generation. Refer to Fig. 4 for an illustration. Let P be a point cloud describing some 3D object. We convert P into a voxelized 3D binary image $\mathcal{B}$ such that every active voxel contains at least one point from P. The volume of $\mathcal{B}$ is roughly equal to that of the axis-parallel

bounding box of P. We run 57 filtrations on $\mathcal{B}$, yielding a set of 57 grayscale images. Out of 57 filtrations, there are 26 height filtrations for the 26 direction vectors in $\{\{0,1,-1\}^3\} \setminus \{(0,0,0)\}$; 27 radial filtrations for the 27 centers $c_1, \ldots, c_{27}$, as described in Section 2; one each for the four types: density, dilation, erosion, and signed distance. As explained before, we extract 36 features from a grayscale image. Hence, due to the 57 filtrations used, which resulted in 57 binary images, the length of the final feature vector for P is $57 \cdot 36 = 2052$. However, our experiments found that depending on the dataset, we must discard some of the radial filtrations from the initial 27 centers, as shown in Fig. 3 to achieve the highest possible accuracy.

Classification using a 1D CNN. We use a lightweight 1D CNN deep neural network to classify the features extracted from the point clouds of the objects. In recent years, 1D CNN has been used extensively for such feature and sequence classification with high success [24]. Our network has three 1D CNN layers, each followed by batch normalization and ReLU layers. The three CNN layers after the input have filter sizes of $3, 5$, and 7 respectively, whereas the number of filters in the first two layers is 128 and 64, respectively. In the third CNN layer, the filters are set to the class count for ModelNet40 and ModelNet10 datasets, 40 for ScanObjectNN, and 32 for the VesselMNIST3D and AdrenalMNIST3D datasets.

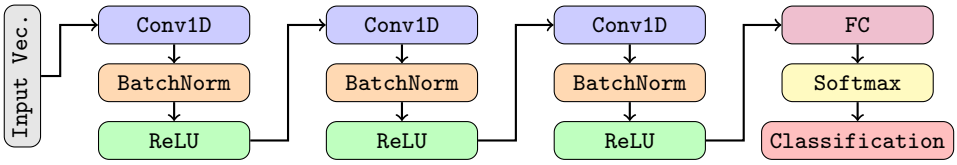


Fig. 5: The architecture of the 1D CNN used in `TACO-Net`. Convolutional layers process the input feature vector before final classification via fully connected (FC) and softmax layers.

After the three consecutive 1D CNN layers, we have a fully connected layer of size equal to the number of classes. Next, the classification is done by applying a softmax function on the outputs of the fully connected layer. Refer to Fig. 5. The time taken for classification is $\mathcal{O}(n + v^3 + v/\rho^3)$, where n is the size of the point cloud, v is the number of voxels in $\mathcal{B}$, and ρ is the voxel-size used. See Theorem 1.

Theorem 1. *Let P be an n-element point cloud that needs to be classified by `TACO-Net`. Then, the time taken for classification is $\mathcal{O}(n + v^3 + v/\rho^3)$, where v is the number of voxels in $\mathcal{B}$ and ρ is the voxel-size used.*

Proof. Initializing all voxels in $\mathcal{B}$ as inactive requires $\mathcal{O}(v)$ time. For each point in P, locating the corresponding voxel in $\mathcal{B}$ takes $\mathcal{O}(1)$ time. Since $|P| = n$, the total time to prepare $\mathcal{B}$ is $\mathcal{O}(v + n)$.

From $\mathcal{B}$, we generate 57 grayscale images using six filtration types: height, radial, density, dilation, erosion, and signed distance. For the 26 height and 27 radial filtrations, each voxel requires a constant-time distance computation, resulting in $\mathcal{O}(v)$ time per filtration. Thus, the total time for generating these 53 grayscale images is $\mathcal{O}(57 \cdot v) = \mathcal{O}(v)$, including initialization. For the density filtration, each voxel must inspect its neighborhood within a ball of radius r, which contains $\mathcal{O}(r^3/\rho^3)$ voxels. In `TACO-Net`, we set $r = 1$, yielding a per-voxel cost of $\mathcal{O}(1/\rho^3)$, and a total cost of $\mathcal{O}(v/\rho^3)$. The remaining three filtrations, dilation, erosion, and signed distance, can be computed in $\mathcal{O}(v)$

time each using efficient distance transform algorithms [11]. Therefore, the total time for generating all 57 grayscale images is $\mathcal{O}(v(1 + 1/\rho^3))$.

Cubical persistence for a single grayscale image can be computed in $\mathcal{O}(v^3)$ time using standard matrix reduction techniques [47]. For 57 images, the total cost is $\mathcal{O}(57 \cdot v^3) = \mathcal{O}(v^3)$. Each voxel generates up to 27 cells (1 cube, 6 faces, 12 edges, 8 vertices), the building blocks of a cubical complex. Each cell can belong to at most one persistence pair. Hence, the worst-case number of birth–death pairs is at most $27v = \mathcal{O}(v)$.

Feature extraction from each cubical persistence involves computing persistent entropy and amplitude metrics. Persistent entropy requires $\mathcal{O}(v)$ time per image. Wasserstein and Bottleneck amplitudes also take $\mathcal{O}(v)$ time. The Betti curve kernel, evaluated over 100 filtration values, requires $\mathcal{O}(v)$ time per image. The persistence landscape kernel, which involves sorting at each of 100 sampled values, incurs $\mathcal{O}(v \log v)$ time. The heat kernel, evaluated over 20 filtration values, takes $\mathcal{O}(v)$ time. Thus, the total time to generate the topological features for one grayscale image is $\mathcal{O}(v \log v)$. For the 57 images, time taken is $\mathcal{O}(57 \cdot v \log v) = \mathcal{O}(v \log v)$.

Since the 1D CNN model is fixed during inference, classification of the topological vector takes constant time, i.e., $\mathcal{O}(1)$.

Combining all components, the overall time complexity of the pipeline is:

$$\mathcal{O}(v + n) + \mathcal{O}\left(v\left(1 + \frac{1}{\rho^3}\right)\right) + \mathcal{O}(v^3) + \mathcal{O}(v \log v) + \mathcal{O}(1) = \mathcal{O}\left(n + v^3 + \frac{v}{\rho^3}\right).$$

3 Experiments

Settings. We have used 7 datasets to validate the efficacy of `TACO-Net` and they are ModelNet10/40 (by far the most popular benchmark for this problem), OmniObject3D (a real-world dataset consisting of 190 classes), ScanObjectNN (a real-world noisy dataset with 15 classes), 3DGrocery100 (a large dataset of grocery items), and two real-world binary medical datasets for further generalizability, namely VesselMNIST3D and AdrenalMNIST3D. More details about these datasets are provided in the Appendix (Sec. **??**).

We have implemented all TDA-related portions of `TACO-Net` in Python using the `giotto-tda` package [44]. The experiments were run on a machine equipped with an Intel `i9-12900K` processor, 32-GB of main memory, and a NVIDIA RTX 3060 GPU. For all the datasets, we evaluate the performance on two main metrics: overall accuracy (OA - average accuracy % across all test cases) and mean class accuracy (mAcc - mean accuracy % across all classes). These are the most common evaluation metrics in object classification. For VesselMNIST3D, we also present the F1-score metric due to the imbalance.

For ModelNet40, we have found the vector length 1728 to be the optimum in terms of OA. The exact number of filtrations that corresponds to this length is 48, out of which 26 are height, one each from the four types: density, dilation, erosion, and signed distance, and 18 are radial corresponding to the 18 centers $c_1, \ldots, c_{18}$, as described in Section 2 and shown in Fig. 3. Therefore, all the results presented below are with 1728-length feature vectors. We have used the same vector length input for ModelNet10 as well. The optimum length for VesselMNIST is 1152 using the two centers c_1, c_2, and for AdrenalMNIST it is 1584 using $c_1, \ldots, c_{14}$. Other relevant parameters and their values for TDA-related experiments are mentioned in Section 2.

We emphasize that `TACO-Net` does *not* perform any dataset-specific tuning of topological descriptors or learning architecture. The same family of filtrations, feature vectorization procedures, and

1D CNN architecture are used across all datasets. Hyperparameters such as voxel resolution and the number of radial centers are selected *only on the training split* of each dataset and are not adjusted based on test-set performance. No dataset-specific heuristics, alignment cues, or class-dependent tuning are employed.

We used MATLAB to implement the 1D CNN. We stopped our training early if the training loss had reached 0.005. Each configuration has been trained 5 times, and the average results are presented in the paper unless specified otherwise. The learnable parameters for ModelNet40, ModelNet10, VesselMNIST, and AdrenalMNIST are 0.72M, 0.71M, 0.50M, 0.66M, respectively. The 1D CNN model was trained using the Adam optimizer with an initial learning rate of 10^{-3} and a minibatch size of 128. Training was performed for up to 1000 epochs and terminated early once the training loss dropped below 0.005. The code can be found in the supplementary material.

3.1 Results

ModelNet10/40. First, we present the results of testing `TACO-Net` on ModelNet40 and 10 datasets. To begin with, we first illustrate the empirical reason behind choosing 18 radial filtration centers along with DEDS and height filters for the ModelNet40 dataset. This result is presented in Fig. **??**(a). As can be seen, with feature vector length 1728, i.e., 18 radial filtration filters, the OA is the highest. Although with the different other feature lengths, the OA is close, but lower than the one with length = 1728.

Table 1: Accuracy (%) comparison across representative methods on ModelNet40.

Method	OA	mAcc	Method	OA	mAcc
3DShapeNets [52]	84.7	77.3	PointMamba [28]	93.6	–
PointNet [33]	89.2	86.2	Point-Transformer [61]	93.7	90.6
MVCNN [41]	90.1	–	Point-Bert [58]	93.8	–
Ma et al. [29]	91.05	–	PointMLP [30]	94.5	91.4
KD-Network [25]	91.8	88.5	MHBN [57]	94.7	93.1
PointNet++ [34]	91.9	–	PointGPT [8]	94.9	–
PointCNN [27]	92.5	88.1	Mamba3D [19]	95.1	–
OctFormer [48]	92.7	–	PointView-GCN [32]	95.4	–
GVCNN [13]	93.1	–	VRN Ensemble [3]	95.54	–
PointNeXt [35]	93.2	90.8	HGNN [12]	96.6	–
PCM [59]	93.4	90.7	RotationNet [23]	97.37	96.29
			`TACO-Net`(Ours)	**99.05**	**97.97**

Next, we have tested `TACO-Net` with different voxel sizes to create the 3D binary image from the given point cloud. We have noticed that with a voxel size 0.05, the OA is the highest. With 0.03 and 0.07, the accuracy values decrease to 98.61 (OA), 96.96 (mAcc), and 98.30 (OA) and 96.16 (mAcc), respectively.

Next, the benchmark results for both 40- and 10-class variants of the ModelNet dataset are presented in Tables 1 and **??** (in the Appendix), respectively. These results prove that our proposed `TACO-Net` framework achieves state-of-the-art OA and mAcc accuracies for both these datasets.

Notably, TACO-Net achieves 1.68% higher OA than RotationNet, the current highest OA-achieving method on the ModelNet40 dataset. Further, TACO-Net comprehensively outperformed the recent transformer-based models, e.g., PointMamba [28] and PointGPT [8], among others. Our macro-averaged precision-recall curve (Fig. **??**(b)) stays tightly clustered near $(1, 1)$, showcasing near-perfect precision and recall across every class. This level of consistency, including on minority classes, indicates strong and balanced multi-class performance. To further assess robustness, we perform 5-fold cross-validation on ModelNet40 using identical hyperparameters. TACO-Net achieves a mean OA of 97.49% with a standard deviation of 0.32, demonstrating stable performance across different data partitions.

Similarly, TACO-Net outperforms RotationNet in OA on the ModelNet10 dataset. Given the lower number of classes available, it was expected that the proposed TACO-Net framework would achieve higher accuracies in ModelNet10 than in ModelNet40. Not only was TACO-Net successful in meeting that expectation, but it also yielded 99.52% OA and mAcc values. Most importantly, to the best of our knowledge, ours is the first approach to push the classification accuracy beyond 99% on both ModelNet40 and ModelNet10. Altogether, these results highlight the strong performance and generalizability of the proposed approach across a wide range of datasets.

Comparison with TopoRec [16]. TopoRec is a TDA-based method designed for large-scale point cloud recognition (different from classification). So, it is natural to compare TACO-Net against it. We found that using TopoRec's closest-vector approach, the OA on ModelNet40 is $\approx 36\%$. However, when using our 1D CNN on vectors generated by TopoRec, we found OA to be around 69%, corroborating the learning power of our 1D CNN.

Real-world Datasets. To further validate real-world applicability, we evaluate TACO-Net on OmniObject3D [51], a challenging benchmark featuring thousands of everyday objects captured under realistic conditions. Despite its complexity and large class diversity, TACO-Net delivers the highest accuracy of 58.90%, decisively outperforming heavyweight baselines including CurveNet, PointNet, PointNet++, and PCT: Point cloud transformer (see Table **??**).

We next evaluate TACO-Net on ScanObjectNN (all six popular variants), a notoriously challenging real-world benchmark characterized by heavy occlusions and background clutter-conditions where many point-based methods struggle [46]. We compare our numbers only against from-scratch supervised learning methods. We acknowledge that there are 'training from pre-training' methods that have achieved slightly better numbers than ours (e.g., PointMamba [28]). Still, to maintain fairness, we have only compared against similar methods that do not use pre-trained models. First, we compare the OBJ_BG, OBJ_ONLY, and the hardest variant, PB_T50_RS, against existing techniques. These results are presented in Table 2. For the OBJ_BG and OBJ_ONLY variants, we achieved SOTA accuracy and second highest for PB_T50_RS, among from-scratch supervised learning methods as shown in Table 2 while outperforming seminal works such as PointNet [33], PointNet++ [34], DGCNN [49], and PointNeXt [35], among others, also outperforming some of the most recent studies such as ADS [20] and X-3D [43]. Further, when tested on the other three variants (PB_T25, PB_T25_R, PB_T50_R) of ScanObjectNN, TACO-Net yields $\geq 91\%$ accuracy (see Table 3). When compared against XGBoost, TACO-Net achieves $\sim 30\%$ higher accuracy, which indicates the topological features are powerful, but the proposed 1D CNN captures descriptive patterns from them, compared to other approaches such as XGBoost (Table 3). Overall, the ScanObjectNN results underscore TACO-Net's ability to remain yield SOTA accuracy in highly cluttered, real-world scenarios with backgrounds and transformations.

Table 2: Comparison of OBJ_BG, OBJ_ONLY, and PB_T50_RS variants of the ScanObjectNN dataset against different *from-scratch supervised learning* methods.

	ScanObjectNN		
Methods	**OBJ_BG**	**OBJ_ONLY**	**PB_T50_RS**
PointNet [33]	73.3	79.2	68.0
PointNet++ [34]	82.3	84.3	77.9
DGCNN [49]	82.8	86.2	78.1
PointCNN [27]	86.1	85.5	78.5
MVTN [18]	-	-	82.8
PointNeXt [35]	-	-	87.7
PointMLP [30]	-	-	85.4
ADS [20]	-	-	87.5
X-3D [43]	-	-	90.7
Mamba3D [19]	94.49	92.43	**92.64**
`TACO-Net`(Ours)	**95.08**	**94.01**	91.87

Table 3: Comparison of our proposed `TACO-Net` and XGBoost on the six popular variants of the ScanObjectNN dataset [46].

Methods	**OBG_ONLY**	**OBJ_BG**	**PB_T25**
XGBoost	68.85	66.95	64.17
`TACO-Net`(Ours)	94.01	95.08	93.39

Methods	**PB_T25_R**	**PB_T50_R**	**PB_T50_RS**
XGBoost	61.74	58.16	57.84
`TACO-Net`(Ours)	93.69	91.60	91.87

`TACO-Net` achieved a competitive OA of 40.18% on the recent and challenging 3DGrocery100 dataset [38], outperforming strong state-of-the-art methods, PointNet [33] and Point cloud transformer [17], which achieved OAs of 37.61% and 39.67%, respectively.

Real-world Medical Data. For VesselMNIST [56], we compared against some of the current SOTA benchmarks for this dataset as shown in Table 4. When compared against the benchmarks presented in [56], `TACO-Net` performed better in terms of both F1 and mAcc. For example, the previous highest F1-score of 0.90 for VesselMNIST was achieved by PointNet++ and PointCNN, whereas our mean F1-score is 0.94 - an improvement of 4.44%. Similarly, for the mAcc metric, our average result is 95.28%, whereas the prior best was 93.52 achieved by PointNet++ - an improvement of 1.88%.

For the AdrenalMNIST dataset, we used the benchmark provided in [55] as our baseline. The comparison results are presented in Table 5. The authors in [55] have used different variations of ResNet along with medical image-specific variants such as ACS [54]. Our proposed `TACO-Net` outperformed all these benchmarks in the OA metric, as shown in the table. Notably, for the VesselMNIST3D dataset, the difference in OA between ours and the current SOTA achieved using ResNet-18 + ACS is 4.58%.

Table 4: VesselMNIST3D [56]: The best mAcc (%) and F1-scores from [56] are reported along with our average results at the bottom.

Network	mAcc	F1
PointNet++ [34]	93.52	0.90
SpiderCNN [53]	92.59	0.87
SO-Net [26]	91.50	0.88
PointCNN [27]	92.38	0.90
DGCNN [49]	90.67	0.86
PointNet [33]	81.62	0.69
`TACO-Net` (Ours)	**95.28**	**0.94**

Table 5: Overall accuracies (OA) in % for AdrenalMNIST3D (A3D) and VesselMNIST3D (V3D) across different methods as benchmarked in [55] compared with `TACO-Net`.

Methods	A3D	V3D
ResNet-18 + 2.5D	77.2	84.6
ResNet-18 + 3D	72.1	87.7
ResNet-18 + ACS	75.4	92.8
ResNet-50 + 2.5D	76.3	87.7
ResNet-50 + 3D	74.5	91.8
ResNet-50 + ACS	75.8	85.8
auto-sklearn [14]	80.2	91.5
AutoKeras [21]	70.5	89.4
`TACO-Net` (Ours)	**80.54**	**97.38**

Resiliency Against Corrupted Test Data. Noise resiliency in `TACO-Net` is achieved because topological features, extracted via persistent homology, capture the global shape characteristics of objects rather than relying on exact point positions. These features remain stable under small perturbations or noise, as persistent homology emphasizes long-lived topological structures while ignoring short-lived, noise-induced artifacts. Consequently, `TACO-Net` maintains high accuracy in most cases even when point clouds are corrupted, as discussed below. To test the robustness of `TACO-Net`, we have used the standardized approach of ModelNet40-C [42], where the test set of ModelNet40 is perturbed by adding various common types of corruptions. Note that the 'uniform downsampling' perturbation was not part of ModelNet40-C.

Two main differences between our implementation and ModelNet40-C are 1) we use 2048×3 size point clouds, and 2) we do not apply any normalization after the perturbation is incorporated. The results for this study are presented in Table **??**. The first row presents the test results found by the model with the highest test accuracy for clean ModelNet40 without any corruption. We use this saved model for inference on the corrupted test dataset and report the results in Table **??**. Two severity levels of data corruption are chosen from [42]: 1 and 5, named Low and High, respectively, in our paper. For 'uniform downsampling', we have removed 10 and 30% random points uniformly for low and high severity, respectively.

We see that `TACO-Net` is very robust against low-level perturbations - always achieves $\geq 94\%$ OA except for 'impulse', where the OA falls to 52.88%. As expected with high-level perturbations, `TACO-Net` shows resiliency. In case of 'impulse', the OA more than halves, but for the others, it performs reasonably well - the average OA being 68.65%. If the underlying shape changes significantly, then topological features become very different from clean class signatures, and `TACO-Net` struggles to recognize corrupted point clouds. Under the high-severity corruption, augmenting the training set with corrupted copies of randomly selected 20% of training instances lifts overall accuracy to 96.11% in the case of rotation, for example, substantially outperforming the non-augmented model (49.39%). This suggests that a task-aligned augmentation confers significant robustness to extreme pose variations without altering the core architecture.

Shape Retrieval. Our method achieves a new state-of-the-art retrieval performance with an mAP of 99.33 on ModelNet40, surpassing the strongest baseline Latformer (97.4) and all prior approaches

(see summary results in Table **??** in the Appendix). Beyond accuracy, the framework demonstrates strong generalizability, showing its effectiveness not only for classification but also for challenging tasks such as shape retrieval.

Note on Efficiency. With voxel size $\rho = 0.05$ (our optimal configuration), the throughput of the TDA feature generation pipeline is 5.3 point clouds/second, when the feature vector length is set to 1728 for the ModelNet40 dataset. This number increased to 8.2 and decreased to 1.4 when ρ was increased to 0.07 or decreased to 0.03, respectively. However, as mentioned earlier, the test accuracy decreases in both cases. However, on the bright side, due to the lightweight 1D CNN of `TACO-Net`, the training time is short (2.50 mins.) and class prediction is lightning-fast, achieving a throughput rate of $16,454$ point clouds/second. Thus, our current full pipeline's overall throughput is approximately 5.3 clouds/second (and 8.2 clouds/second at $\rho = 0.07$). Furthermore, Table **??** indicates that our proposed `TACO-Net`model achieves state-of-the-art accuracy with the very few parameters (0.72M), highlighting its superior efficiency compared to prior methods.

Ablation Study. Our approach to studying the effect of ablation is two-fold. First, using our proposed network (Fig. 5), we test different sets of topological features extracted from a point cloud, i.e., using density, erosion, dilation, and signed distance filtration (DEDS) only, height (H) filtration only, and finally DEDS + H only (i.e., without any radial filtration).

The effect of using different radial filtration along with DEDS and H together is already illustrated in Fig. **??**(a) and discussed earlier. In the next set of ablation studies, we use the best topological features for ModelNet40, i.e., a vector length of 1728, while testing the effect of deleting one CNN layer at a time. The results are listed in Table 6. This study suggests that the features extracted via the H filtration are the most effective, yielding over 98% OA, which outperforms all baselines while being 3.5x faster than the whole pipeline. Similarly, when just entropy is used (without amplitude), vector generation is 2.2x faster while maintaining 96.16% OA. This shows `TACO-Net` is not only accurate but also tunable for resource-constrained scenarios. The finding further supports our rationale for employing 26 directional vectors, as outlined in Sec. 2. Incorporating the DEDS features provides a slight improvement, but the gain is marginal. In contrast, removing two CNN layers leads to a moderate decrease in 4.29% in accuracy. To highlight the effectiveness of our 1D CNN for feature vector classification, we replace it with a heavier 2.2M-parameter transformer that incorporates feature and positional embeddings, mixed self-attention, and a fully connected classifier. Despite its complexity, this transformer yields only 62.20% accuracy on ModelNet40, underscoring the superiority of our lightweight CNN design. On the other hand, XGBoost, a non-deep learning method, also yielded substantially lower OA of 81.35% (see Table **??**). Taken together, these results provide strong justification for our feature selection and network design choices.

Table 6: Ablation study on ModelNet40.

	Variant	OA	mAcc
features	DEDS only	96.52	93.76
	Density only	80.75	75.59
	E only	52.26	43.14
	Dilation only	85.68	81.61
	S only	89.35	81.10
	R only	95.47	91.52
	H only	98.29	96.38
	DEDS + H only	98.82	97.33
	Entropy only	96.16	92.79
net	First 2 Conv1D	98.18	95.93
	First Conv1D	94.76	90.70

t-SNE Analysis on ModelNet40. Using only non-learned topological descriptors as input, the proposed 1D CNN model produces t-SNE embeddings with clear class-wise separation, indicating that the network effectively distills these signatures into robust, discriminative feature representations (Fig. 6). This shows the strong compatibility of topological features with modern deep architectures.

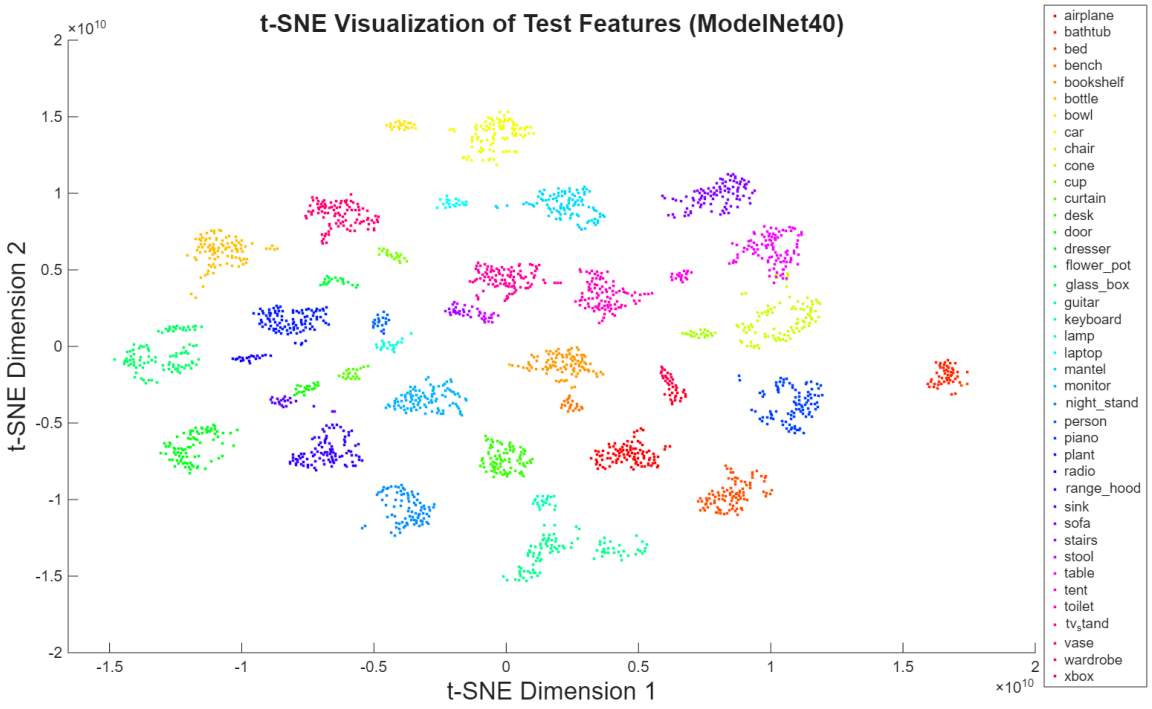


Fig. 6: t-SNE analysis of ModelNet40 using the activations from the last RELU layer of our proposed network.

4 Conclusion and Future Work

Computer vision researchers have made significant advancements in 3D object classification using various deep learning techniques in recent years. However, there are still some challenges to address and open directions to explore. To this end, we have proposed a novel framework, named `TACO-Net`, for 3D object classification. Our proposed approach takes a point cloud of the object as input, converts it into a voxelized 3D binary image, extracts topological signatures from it through various filtration techniques, and finally learns these features using a lightweight 1D CNN. Results show that our proposed technique achieves near-perfect overall accuracy in popular 3D object classification benchmark datasets, namely ModelNet40 and ModelNet10, while outperforming the current SOTA. Further, when tested on two 3D medical datasets consisting of brain MRA and abdominal CT scan data, `TACO-Net`, outperforms all the benchmarks provided in the literature, showcasing its strong generalizable qualities. To enhance real-time performance, future work will focus on exploiting GPU parallelism to increase throughput for the topological feature generation pipeline.

References

1. Akcora, C.G., Li, Y., Gel, Y.R., Kantarcioglu, M.: Bitcoinheist: Topological data analysis for ransomware prediction on the bitcoin blockchain. In: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (2020)
2. Ben-Shabat, Y., Lindenbaum, M., Fischer, A.: 3dmfv: Three-dimensional point cloud classification in real-time using convolutional neural networks. IEEE Robotics and Automation Letters **3**(4), 3145–3152 (2018)
3. Brock, A., Lim, T., Ritchie, J.M., Weston, N.: Generative and discriminative voxel modeling with convolutional neural networks. arXiv preprint arXiv:1608.04236 (2016)
4. Bubenik, P., Dłotko, P.: A persistence landscapes toolbox for topological statistics. Journal of Symbolic Computation **78**, 91–114 (2017)
5. Bubenik, P., et al.: Statistical topological data analysis using persistence landscapes. J. Mach. Learn. Res. **16**(1), 77–102 (2015)
6. Bukkuri, A., Andor, N., Darcy, I.K.: Applications of topological data analysis in oncology. Frontiers in artificial intelligence **4**, 659037 (2021)
7. Chazal, F., Michel, B.: An introduction to topological data analysis: fundamental and practical aspects for data scientists. Frontiers in artificial intelligence **4**, 667963 (2021)
8. Chen, G., Wang, M., Yang, Y., Yu, K., Yuan, L., Yue, Y.: Pointgpt: Auto-regressively generative pre-training from point clouds. Advances in Neural Information Processing Systems **36**, 29667–29679 (2023)
9. Chintakunta, H., Gentimis, T., Gonzalez-Diaz, R., Jimenez, M.J., Krim, H.: An entropy-based persistence barcode. Pattern Recognition **48**(2), 391–401 (2015)
10. Edelsbrunner, Letscher, Zomorodian: Topological persistence and simplification. Discrete & computational geometry **28**, 511–533 (2002)
11. Fabbri, R., Costa, L.D.F., Torelli, J.C., Bruno, O.M.: 2d euclidean distance transform algorithms: A comparative survey. ACM Computing Surveys (CSUR) **40**(1), 1–44 (2008)
12. Feng, Y., You, H., Zhang, Z., Ji, R., Gao, Y.: Hypergraph neural networks. In: Proceedings of the AAAI conference on artificial intelligence. vol. 33, pp. 3558–3565 (2019)
13. Feng, Y., Zhang, Z., Zhao, X., Ji, R., Gao, Y.: Gvcnn: Group-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 264–272 (2018)
14. Feurer, M., Klein, A., Eggensperger, K., Springenberg, J., Blum, M., Hutter, F.: Efficient and robust automated machine learning. Advances in neural information processing systems **28** (2015)
15. Garin, A., Tauzin, G.: A topological "reading" lesson: Classification of mnist using tda. In: 2019 18th IEEE international conference on machine learning and applications (ICMLA). pp. 1551–1556. IEEE (2019)
16. Ghosh, A., Kulbaka, I., Dahlin, I., Dutta, A.: Toporec: Point cloud recognition using topological data analysis. In: Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision. pp. 7544–7553 (2026)
17. Guo, M.H., Cai, J.X., Liu, Z.N., Mu, T.J., Martin, R.R., Hu, S.M.: Pct: Point cloud transformer. Computational visual media **7**(2), 187–199 (2021)
18. Hamdi, A., Giancola, S., Ghanem, B.: Mvtn: Multi-view transformation network for 3d shape recognition. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1–11 (2021)
19. Han, X., Tang, Y., Wang, Z., Li, X.: Mamba3d: Enhancing local features for 3d point cloud analysis via state space model (2024), `https://arxiv.org/abs/2404.14966`
20. Hong, C.Y., Chou, Y.Y., Liu, T.L.: Attention discriminant sampling for point clouds. In: Proceedings of the IEEE/CVF International Conference on Computer Vision. pp. 14429–14440 (2023)
21. Jin, H., Song, Q., Hu, X.: Auto-keras: An efficient neural architecture search system. In: Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining. pp. 1946–1956 (2019)

22. Kaczynski, T., Mischaikow, K., Mrozek, M.: Computational homology, vol. 157. Springer Science & Business Media (2006)
23. Kanezaki, A., Matsushita, Y., Nishida, Y.: Rotationnet for joint object categorization and unsupervised pose estimation from multi-view images. IEEE transactions on pattern analysis and machine intelligence **43**(1), 269–283 (2019)
24. Kiranyaz, S., Avci, O., Abdeljaber, O., Ince, T., Gabbouj, M., Inman, D.J.: 1d convolutional neural networks and applications: A survey. Mechanical systems and signal processing **151**, 107398 (2021)
25. Klokov, R., Lempitsky, V.: Escape from cells: Deep kd-networks for the recognition of 3d point cloud models. In: Proceedings of the IEEE international conference on computer vision. pp. 863–872 (2017)
26. Li, J., Chen, B.M., Lee, G.H.: So-net: Self-organizing network for point cloud analysis. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 9397–9406 (2018)
27. Li, Y., Bu, R., Sun, M., Wu, W., Di, X., Chen, B.: Pointcnn: Convolution on x-transformed points. Advances in neural information processing systems **31** (2018)
28. Liang, D., Zhou, X., Xu, W., Zhu, X., Zou, Z., Ye, X., Tan, X., Bai, X.: Pointmamba: A simple state space model for point cloud analysis. Advances in neural information processing systems **37**, 32653–32677 (2024)
29. Ma, C., Guo, Y., Yang, J., An, W.: Learning multi-view representation with lstm for 3-d shape recognition and retrieval. IEEE Transactions on Multimedia **21**(5), 1169–1182 (2018)
30. Ma, X., Qin, C., You, H., Ran, H., Fu, Y.: Rethinking network design and local geometry in point cloud: A simple residual mlp framework. arXiv preprint arXiv:2202.07123 (2022)
31. Maturana, D., Scherer, S.: Voxnet: A 3d convolutional neural network for real-time object recognition. In: 2015 IEEE/RSJ international conference on intelligent robots and systems (IROS). pp. 922–928. IEEE (2015)
32. Mohammadi, S.S., Wang, Y., Del Bue, A.: Pointview-gcn: 3d shape classification with multi-view point clouds. In: 2021 IEEE International Conference on Image Processing (ICIP). pp. 3103–3107. IEEE (2021)
33. Qi, C.R., Su, H., Mo, K., Guibas, L.J.: Pointnet: Deep learning on point sets for 3d classification and segmentation. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 652–660 (2017)
34. Qi, C.R., Yi, L., Su, H., Guibas, L.J.: Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems **30** (2017)
35. Qian, G., Li, Y., Peng, H., Mai, J., Hammoud, H., Elhoseiny, M., Ghanem, B.: Pointnext: Revisiting pointnet++ with improved training and scaling strategies. Advances in neural information processing systems **35**, 23192–23204 (2022)
36. Reininghaus, J., Huber, S., Bauer, U., Kwitt, R.: A stable multi-scale kernel for topological machine learning. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4741–4748 (2015)
37. Sarker, S., Sarker, P., Stone, G., Gorman, R., Tavakkoli, A., Bebis, G., Sattarvand, J.: A comprehensive overview of deep learning techniques for 3d point cloud classification and semantic segmentation. Machine Vision and Applications **35**(4), 67 (2024)
38. Sheshappanavar, S.V., Anvekar, T., Kundargi, S., Wang, Y., Kambhamettu, C.: A benchmark grocery dataset of realworld point clouds from single view. In: 2024 International Conference on 3D Vision (3DV). pp. 516–527. IEEE (2024)
39. Singh, Y., Farrelly, C.M., Hathaway, Q.A., Leiner, T., Jagtap, J., Carlsson, G.E., Erickson, B.J.: Topological data analysis in medical imaging: current state of the art. Insights into Imaging **14**(1), 58 (2023)
40. Skaf, Y., Laubenbacher, R.: Topological data analysis in biomedicine: A review. Journal of Biomedical Informatics **130**, 104082 (2022)
41. Su, H., Maji, S., Kalogerakis, E., Learned-Miller, E.: Multi-view convolutional neural networks for 3d shape recognition. In: Proceedings of the IEEE international conference on computer vision. pp. 945–953 (2015)

42. Sun, J., Zhang, Q., Kailkhura, B., Yu, Z., Xiao, C., Mao, Z.M.: Benchmarking robustness of 3d point cloud recognition against common corruptions. arXiv preprint arXiv:2201.12296 (2022)
43. Sun, S., Rao, Y., Lu, J., Yan, H.: X-3d: Explicit 3d structure modeling for point cloud recognition. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 5074–5083 (2024)
44. Tauzin, G., Lupo, U., Tunstall, L., Pérez, J.B., Caorsi, M., Medina-Mardones, A.M., Dassatti, A., Hess, K.: giotto-tda:: A topological data analysis toolkit for machine learning and data exploration. Journal of Machine Learning Research **22**(39), 1–6 (2021)
45. Turner, K., Mukherjee, S., Boyer, D.M.: Persistent homology transform for modeling shapes and surfaces. Information and Inference: A Journal of the IMA **3**(4), 310–344 (2014)
46. Uy, M.A., Pham, Q.H., Hua, B.S., Nguyen, T., Yeung, S.K.: Revisiting point cloud classification: A new benchmark dataset and classification model on real-world data. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 1588–1597 (2019)
47. Wagner, H., Chen, C., Vuçini, E.: Efficient computation of persistent homology for cubical data. In: Topological methods in data analysis and visualization II: theory, algorithms, and applications, pp. 91–106. Springer (2011)
48. Wang, P.S.: Octformer: Octree-based transformers for 3d point clouds. ACM Transactions on Graphics (TOG) **42**(4), 1–11 (2023)
49. Wang, Y., Sun, Y., Liu, Z., Sarma, S.E., Bronstein, M.M., Solomon, J.M.: Dynamic graph cnn for learning on point clouds. ACM Transactions on Graphics (tog) **38**(5), 1–12 (2019)
50. Wasserman, L.: Topological data analysis. Annual review of statistics and its application **5**(2018), 501–532 (2018)
51. Wu, T., Zhang, J., Fu, X., Wang, Y., Ren, J., Pan, L., Wu, W., Yang, L., Wang, J., Qian, C., et al.: Omniobject3d: Large-vocabulary 3d object dataset for realistic perception, reconstruction and generation. In: Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. pp. 803–814 (2023)
52. Wu, Z., Song, S., Khosla, A., Yu, F., Zhang, L., Tang, X., Xiao, J.: 3d shapenets: A deep representation for volumetric shapes. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1912–1920 (2015)
53. Xu, Y., Fan, T., Xu, M., Zeng, L., Qiao, Y.: Spidercnn: Deep learning on point sets with parameterized convolutional filters. In: Proceedings of the European conference on computer vision (ECCV). pp. 87–102 (2018)
54. Yang, J., Huang, X., He, Y., Xu, J., Yang, C., Xu, G., Ni, B.: Reinventing 2d convolutions for 3d images. IEEE Journal of Biomedical and Health Informatics **25**(8), 3009–3018 (2021)
55. Yang, J., Shi, R., Wei, D., Liu, Z., Zhao, L., Ke, B., Pfister, H., Ni, B.: Medmnist v2-a large-scale lightweight benchmark for 2d and 3d biomedical image classification. Scientific Data **10**(1), 41 (2023)
56. Yang, X., Xia, D., Kin, T., Igarashi, T.: Intra: 3d intracranial aneurysm dataset for deep learning. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 2656–2666 (2020)
57. Yu, T., Meng, J., Yuan, J.: Multi-view harmonized bilinear network for 3d object recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 186–194 (2018)
58. Yu, X., Tang, L., Rao, Y., Huang, T., Zhou, J., Lu, J.: Point-bert: Pre-training 3d point cloud transformers with masked point modeling. In: Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. pp. 19313–19322 (2022)
59. Zhang, T., Yuan, H., Qi, L., Zhang, J., Zhou, Q., Ji, S., Yan, S., Li, X.: Point cloud mamba: Point cloud learning via state space model. In: Proceedings of the AAAI Conference on Artificial Intelligence. vol. 39, pp. 10121–10130 (2025)
60. Zhao, C., Yang, J., Xiong, X., Zhu, A., Cao, Z., Li, X.: Rotation invariant point cloud analysis: Where local geometry meets global topology. Pattern Recognition **127**, 108626 (2022)
61. Zhao, H., Jiang, L., Jia, J., Torr, P.H., Koltun, V.: Point transformer. In: Proceedings of the IEEE/CVF international conference on computer vision. pp. 16259–16268 (2021)

62. Zomorodian, A., Carlsson, G.: Computing persistent homology. Discrete & Computational Geometry **33**(2), 249–274 (2004)