

Euler Characteristic Transform Based Topological Loss for Reconstructing 3D Images from Single 2D Slices

Kalyan Varma Nadimpalli
 IIIT Bangalore

kalyan.varma@iiitb.ac.in

Amit Chattopadhyay
 IIIT Bangalore

a.chattopadhyay@iiitb.ac.in

Bastian Rieck
 Helmholtz Munich

bastian.rieck@helmholtz-muenchen.de

Abstract

*The computer vision task of reconstructing 3D images, i.e., shapes, from their single 2D image slices is extremely challenging, more so in the regime of limited data. Deep learning models typically optimize geometric loss functions, which may lead to poor reconstructions as they ignore the structural properties of the shape. To tackle this, we propose a novel topological loss function based on the Euler Characteristic Transform. This loss can be used as an inductive bias to aid the optimization of any neural network toward better reconstructions in the regime of limited data. We show the effectiveness of the proposed loss function by incorporating it into **SHAPR**, a state-of-the-art shape reconstruction model, and test it on two benchmark datasets, viz., Red Blood Cells and Nuclei datasets. We also show a favourable property, namely injectivity and discuss the stability of the topological loss function based on the Euler Characteristic Transform.*

1. Introduction

Our brains possess the amazing ability to be able to reconstruct 3D shapes from single 2D images by leveraging prior knowledge and inductive biases about the shapes and sizes of objects based on the information captured from previously observed objects [7]. However, for a computer this inverse problem is ill-posed and extremely challenging. This is because for a single 2D image the space of possible 3D reconstructions is very large and often ambiguous.

There have been prior deep learning-based attempts to solve this challenge, but most of them rely on large datasets and/or 3D models of the shape [4, 11, 18]. The biomedical setting in which we consider this problem, unfortunately, does not provide large labeled datasets and it is too expen-

sive to construct them. The sizes of the datasets available in the biomedical domain are orders of magnitude smaller than the ones available in other domains. To this end, we focus on improving reconstruction performance not by using 3D models or large datasets but instead by adding additional inductive biases in the form of a topology-based regularization to the optimization process. Most models typically optimize geometry-based loss functions that work on a per-pixel basis, such as the DICE loss. We improve the performance of an existing neural network by adding a novel complementary topology-based loss that considers more global topological features, such as connectivity, tunnels, or voids. Specifically, we design a novel regularization term based on the Euler Characteristic Transform [19], that is computationally efficient, can work with any image size and can be plugged into any neural network. An overview of how our loss function can be used can be seen in Fig 2. We demonstrate the efficacy of the proposed loss function by plugging it into the SHAPR model and testing it on two bio-medical datasets used in the prior work [20, 21]. In the current paper, our key contributions are as follows:

- We adapt the Euler Characteristic Transform (ECT), obtaining a novel topological loss function for 3D shape reconstruction that is compatible with any neural network architecture.
- We prove conditions for our proposed ECT-based loss to be injective as well as discuss stability results of ECT on binary images.
- We show the effectiveness of the proposed method by training the SHAPR model [21] with our proposed loss on two benchmark datasets. We see significant improvements in almost all metrics compared to the prior work on both datasets.

Outline. In Section 2 we go over the literature relevant to

our work. In Section 3 we then briefly explain the mathematical background required to understand our work. Subsequently, in Section 4 we describe our proposed loss functions in detail as well as how they fit into the overall training of a neural network. In Section 5 we then prove and discuss favourable mathematical properties of the ECT. In Section 6 we then demonstrate the efficacy of our model and discuss the significance of our results. Finally in Section 7 we summarize our work and list some potential future work.

2. Prior Work

Multiple variants of the problem of 2D to 3D image reconstruction have been studied by various communities for different applications like scene understanding, medical, robot navigation, etc. [7]. The tasks considered differ in their input type, some variants consider multiple slices as the input while some consider a single image like in our formulation. Among the models that only take a single image as an input, most of them require a synthetic 3D model of the output or very large datasets [4, 11, 18].

The application of computational topology to machine learning is an emerging field that has shown promise in various applications [8]. It has recently been used extensively in computer vision tasks like segmentation, image generation, etc. [9, 13, 23]. In the current paper, we improve the performance of image reconstruction models using tools from topology, namely, the Euler Characteristic Transform [19].

SHAPR [21] is the first machine learning model that considers the problem of 2D to 3D reconstruction in the case of biomedical images. This model proved to be significantly better than standard synthetic models like a cylindrical fit and ellipsoid fit. They also showed that features extracted from the 3D reconstruction helped to improve accuracy in downstream classification tasks on the 2D images. Recently, a diffusion-based model DISPR [22] has been introduced that outperforms the GAN-based SHAPR model.

Waibel *et al.* [20] extend the SHAPR model by training the model on a combined loss function of both the DICE loss as well as a regularization term defined by the Wasserstein distance between the persistence diagrams—topological descriptors—of the predicted shape and the ground truth. This model outperforms the SHAPR model and provides much better reconstructions than the vanilla SHAPR model. However, it has been shown by Oner *et al.* [13] that such persistence diagram based loss functions are not optimal for the following reasons:

- Since the ground truth images are binary images, calculating the persistence diagrams over the filtration of pixel values degenerates to calculating the Betti numbers, which is a topological measure of limited expressivity.
- Persistence diagrams throw away location information

and are generally not injective mappings, thus potentially leading to erroneous matchings, which in turn may lead to wrong reconstructions.

To overcome these drawbacks, we exploit the injectivity property and low computational cost of the ECT, developing a novel ECT-based loss function that is more expressive and serves as a better optimization term for a neural network.

3. Mathematical Background

In this section, we briefly introduce the mathematical background required for our work, for a more detailed explanation we refer the reader to Edelsbrunner *et al.* and Turner *et al.* [6, 19].

3.1. Simplicial and Cubical Complex

A simplicial complex is the fundamental building block of algebraic topology, comprised of simplices. A k -simplex σ can be understood as the convex hull of $k + 1$ affinely independent points. A 0-simplex is a point, a 1-simplex is an edge, a 2-simplex is a triangle and a 3-simplex is a tetrahedron. A face τ of a simplex is the convex hull of a subset of the $k + 1$ points. It is often represented as a face by the notation $\tau \preceq \sigma$. A simplicial complex K is a finite collection of simplices satisfying two conditions:

1. $\sigma \in K$ and $\tau \preceq \sigma$ implies that $\tau \in K$
2. $\sigma, \sigma_0 \in K$ implies $\sigma \cap \sigma_0$ is either empty or a face of both.

The dimension of the simplicial complex is the dimension of the largest simplex in the complex, denoted by $\text{Dim}(K)$. A subcomplex L is a subset of a simplicial complex K . K^d is a particular subcomplex that is defined as a subcomplex consisting of all simplices of dimension d from K , that is, $K^d = \{\sigma \in K \mid \dim(\sigma) = d\}$.

A cubical complex is a special variant of a simplicial complex that is particularly useful in representing grid-like shapes. It has recently caught traction in applications for image processing due to the fact that it is better aligned to the grid-like structure of images [1, 15]. Informally, a cubical complex is identical to a simplicial complex except that n -simplices are replaced with n -cubes. For example, the triangles (2-simplices) are replaced by squares (2-cubes), and tetrahedra (3-simplices) by cubes (3-cubes) and so on. Note that all definitions in this section hold for both simplicial and cubical complexes.

Given a d -dimensional binary image, a natural way to convert it to a cubical complex is by defining the 0-cubes as the set of voxels. Then an i -dimensional cube is formed by connecting a set of 2^i adjacent voxels whose voxel values are 1. Note that two d -dimensional voxels are adjacent if they share a $(d - 1)$ -dimensional face. Thus 1-cubes are

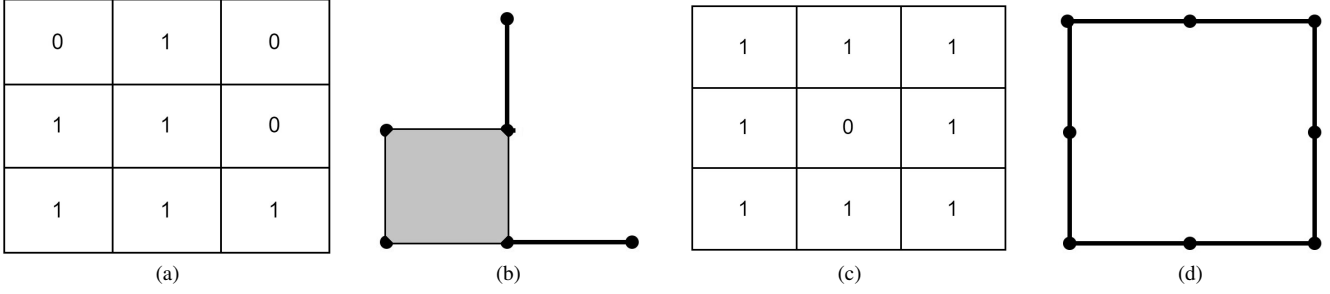


Figure 1. Examples of cubical complex construction from binary images. (b) and (d) are the cubical complexes corresponding to the binary images (a) and (c), respectively.

the edges corresponding to two adjacent voxels with values 1. Similarly, the 2-cubes are the squares corresponding to four adjacent voxels with values 1 and so on. We illustrate this construction by example in Figure 1. Figures 1b and 1d are the cubical complexes corresponding to Figures 1a, 1c when converted by the above procedure.

3.2. Sublevel Sets and Filtrations

Consider a simplicial or cubical complex K and a monotonic function $f : K \rightarrow \mathbb{R}$. By f being monotonic, we mean $f(\sigma) \leq f(\tau)$ whenever $\sigma \preceq \tau$. For such monotonic functions, the sublevel set $K(a)$ corresponding to a real value a is defined by

$$K(a) = f^{-1}(-\infty, a],$$

which is a subcomplex of K . If there are m simplices in K , as we increase a , we get $r + 1 \leq m + 1$ different subcomplexes which can be arranged in an increasing sequence,

$$\emptyset = K_0 \subseteq K_1 \dots \subseteq K_r = K$$

where $K_i = K(a_i)$ and $a_1 < a_2 < \dots < a_r$ are the distinct function values of f at the simplices of the simplicial complex K . This sequence of complexes is called the filtration of K with respect to f . A common filtration we consider is the height filtration. Given a height h and a particular direction \vec{u} , we define the sub-complex $K_{\vec{u},h}$ consists of all the simplices of K whose vertices have height $\leq h$ along the direction \vec{u} . We can naturally define a filtration by increasing the value of h along the direction \vec{u} .

3.3. Euler Characteristic Curve

Given a d -dimensional simplicial complex K , the Euler Characteristic Curve of K along a direction \vec{u} is a function $EC_{\vec{u},K} : \mathbb{R} \rightarrow \mathbb{Z}$ defined by

$$h \mapsto \chi(K_{\vec{u},h}), \quad (1)$$

where $\chi(K_{\vec{u},h})$ is the Euler characteristic of the simplicial complex $K_{\vec{u},h}$, which is defined as

$$\chi(K_{\vec{u},h}) = \sum_{i=0}^d (-1)^i \text{Card } K_{\vec{u},h}^i, \quad (2)$$

where $\text{Card}(K_{\vec{u},h}^i)$ denotes the number of i -simplices in the subcomplex $K_{\vec{u},h}^i$. By computing the Euler characteristic alongside a filtration, we obtain the *Euler Characteristic Curve*. This construction works for general filtrations and is not restricted to the height filtration.

3.4. Euler Characteristic Transform

The Euler Characteristic Transform (ECT) [19] of a d -dimensional simplicial complex K , denoted by $\text{ECT}_K : \mathbb{S}^{d-1} \rightarrow \mathbb{Z}^{\mathbb{R}}$, is defined by

$$\vec{v} \rightarrow \text{EC}_{\vec{v},K}, \quad (3)$$

where the direction \vec{v} is chosen from the $(d-1)$ -dimensional unit sphere \mathbb{S}^{d-1} . That is, the ECT is the set of all Euler Characteristic Curves obtained over the height filtrations along all possible directions. The ECT is the heart of our method. We use it as a topological descriptor to capture the important topological features of 3D images to define our topological loss functions.

Distance between ECTs. The distance between two ECTs corresponding to two complexes K_1 and K_2 is defined by

$$d(\text{ECT}_{K_1}, \text{ECT}_{K_2}) = \int_{\vec{u} \in \mathbb{S}^{d-1}} \|\text{EC}_{\vec{u},K_1} - \text{EC}_{\vec{u},K_2}\|^2 d\vec{u}, \quad (4)$$

where $\|\cdot\|$ is the l_2 -norm. We use this distance to compute the topology based loss function to train our neural network. In practice, the integration in equation (4) is computed using the Monte Carlo method, i.e., we compute the average of the l_2 -norms between the Euler curves along a finite number randomly sampled directions from \mathbb{S}^{d-1} .

4. Our Method: Euler Characteristic Transform-based Loss

In this section, first we describe the overall workflow of our 3D image reconstruction method and how our loss function fits into a neural network training procedure (subsec-

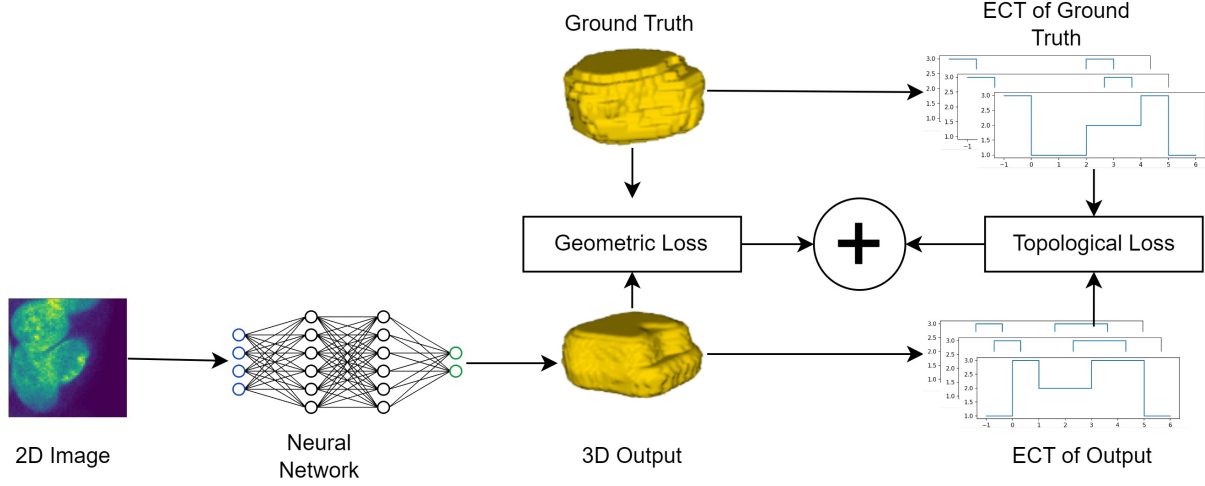


Figure 2. Workflow of our proposed method. Given a 2D image, a neural network produces a 3D output. The neural network is then trained on the sum of a geometric loss function like DICE loss or BCE and our proposed topological loss function, the distance between the ECTs of the images. Neural network image generated from [12].

tion 4.1). Subsequently, we give the detailed algorithm to compute the proposed loss functions (subsections 4.2, 4.3).

4.1. Overview

In our method, we develop a loss function based on the ECT to train a neural network for 3D image reconstruction from a single 2D slice. Figure 2 shows the workflow of our model, which is explained in the following steps.

1. Given a 2D slice, it is first passed through a neural network that gives an output 3D image I , where for each voxel x of I , the model assigns the likelihood of x being part of the true 3D image.
2. Given the 3D prediction I and the 3D ground truth Y , we use the DICE loss combined with a scaled ECT-based loss function, denoted L_{TL} , to optimize the neural network. Mathematically, this can be represented as:

$$L(I, Y) = L_{DICE}(I, Y) + \lambda L_{TL}(I, Y), \quad (5)$$

where λ is the weight parameter for the topological loss term.

This is a similar setup as described by Waibel *et al.* [20], however, our method differs in the details of the topology-based loss $L_{TL}(I, Y)$ and as a result the efficacy of it as well. Next, we discuss our algorithm to train a neural network by computing the topological loss terms based on ECT, in detail.

4.2. ECT Based Training Algorithm

Given a dataset of 2D slices and corresponding 3D images, we first train the SHAPR model based on the proposed ECT-based loss function.

Algorithm 1 TRAINSHAPRMODELBASEDONECT

Input: X - 2D image slice,

Y - Corresponding 3D ground-truth image,

t - Number of thresholds,

Θ_0 - Initial model parameters

Output: Trained SHAPR Model

- 1: $\Theta \leftarrow \Theta_0$ % Initialize model parameters
 - 2: **for** epoch = 1, 2, ..., N **do**
 - 3: $I \leftarrow \text{SHAPR}(X, \Theta)$
 - 4: $L_{Topo} \leftarrow 0$ % Initialize the Topology Loss
 - 5: % Unfold and Sort the Distinct Voxel Values of I and Y in Array R
 - 6: $R \leftarrow \text{SortDistinct}(I \cup Y)$
 - 7: $m \leftarrow \text{length}(R)$ % Number of Voxels in I
 - 8: % Compute ECT-based Topological Loss
 - 9: **for** thresh = $R(\lfloor \frac{m}{t} \rfloor), R(\lfloor \frac{2m}{t} \rfloor), \dots, R(\lfloor m \rfloor)$ **do**
 - 10: $A \leftarrow \text{BinaryImg}(I, \text{thresh})$
 - 11: $B \leftarrow \text{BinaryImg}(Y, \text{thresh})$
 - 12: Sample l directions $\{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_l\}$ from \mathbb{S}^2
 - 13: $\text{ECT}_A \leftarrow \text{ComputeECT}(A, \{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_l\})$
 - 14: $\text{ECT}_B \leftarrow \text{ComputeECT}(B, \{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_l\})$
 - 15: $L_{Topo} += (\text{ECT}_A - \text{ECT}_B)^2$
 - 16: **end for**
 - 17: % Compute Total Loss
 - 18: $L = L_{DICE} + \lambda L_{Topo}/n$
 - 19: % Perform Gradient Update Step to Update the Model Parameters Θ with Learning Rate α
 - 20: $\Theta \leftarrow \Theta - \alpha \nabla_{\Theta} L$
 - 21: **end for**
-

For ease of understanding, in Algorithm 1, we demonstrate the training of the SHAPR model on a single training sample, i.e., using a 2D slice image X and its ground-truth 3D image Y . In every epoch (or training step), the image X is first passed to the SHAPR model and an output 3D image I is produced by the model (Line 3). Then at each step, we compute the ECT-based loss function using Monte Carlo sampling. For this, we sort the distinct voxel values of I and Y in a one-dimensional array R and find t equally spaced thresholds $R(\lfloor \frac{m}{t} \rfloor), R(\lfloor \frac{2m}{t} \rfloor), \dots, R(\lfloor m \rfloor)$ where m is the total number of voxels in I (Lines 5-9). For each threshold τ , we compute the binary images $A = \mathbb{I}[x \geq \tau \mid x \in I]$ and $B = \mathbb{I}[x \geq \tau \mid x \in Y]$ corresponding to I and Y , respectively, where \mathbb{I} is the indicator function (Lines 10-11). Next, for each of these binary images A and B , we approximate the Euler Characteristic Transforms ECT_A and ECT_B as in Algorithm 2 using l sampled directions from \mathbb{S}^2 (Lines 12-14). Finally, the topological loss function is computed using the average of l_2 -norms between ECT_A and ECT_B , for all thresholds. A scaled version of this topological loss (here, λ is the scaling factor) is added with the standard DICE loss to compute the total loss (Line 18). The model parameters Θ are then updated by optimizing this loss using a gradient descent method (Line 20). Note that we choose equally spaced thresholds on the sorted array R to obtain a more varied set of images as compared to thresholding based on the range of voxel values.

4.3. ECT Computation

In this sub-section, we explain the details of approximating the ECT for a binary image A using l sampled directions $\{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_l\}$ from \mathbb{S}^2 whose pseudocode is given in Algorithm 2. Broadly, we first construct a cubical complex C from the binary image A using the method explained in Section 3.1 (Line 1, Algorithm 2). Then for each sampled direction \vec{u}_i we compute the Euler Curve of C along the direction \vec{u}_i (Lines 4-6). The obtained set of l Euler Curves is returned as our Euler Characteristic Transform (Line 7).

Algorithm 2 COMPUTEECT

Input: A - 3D Binary Image, l sampled directions $\{\vec{u}_1, \vec{u}_2, \dots, \vec{u}_l\}$ from the unit sphere \mathbb{S}^2

Output: ECT_A

```

1:  $C \leftarrow \text{CubicalComplex}(A)$ 
2:  $ECT_A \leftarrow []$  % Initialize as an empty array
3: % Compute Euler curves along the  $l$  directions
4: for  $i = 1, 2, \dots, l$  do
5:    $ECT_A.add(\text{EulerCurve}(C, \vec{u}_i))$ 
6: end for
7: return  $ECT_A$ 

```

The Euler Curve computation of a cubical complex C

along a sampled direction \vec{u} is described in Algorithm 3. We compute the minimum h_{min} and maximum h_{max} of all heights of the vertices $\mathbf{v}_0, \mathbf{v}_1, \dots, \mathbf{v}_n$ in the cubical complex C along the direction \vec{u} (Lines 1-2). For a chosen parameter M , we sample the height field at $M + 1$ equally spaced heights of step-size dh (Line 5). For each sampled height, we calculate the Euler Characteristic κ of the sub-complex $C_{\vec{u},h}$ (Lines 7-10). We return the list of obtained values as our discrete representation of the Euler curve. Note that the smaller the step size dh , the closer our representation is to the continuous Euler Curve.

Algorithm 3 EULERCURVE

Input: C - Cubical complex, \vec{u} - Direction vector

Output: $EC_{u,C}$

```

1:  $h_{min} \leftarrow \min(\vec{u} \cdot \mathbf{v}_0, \dots, \vec{u} \cdot \mathbf{v}_n)$ 
2:  $h_{max} \leftarrow \max(\vec{u} \cdot \mathbf{v}_0, \dots, \vec{u} \cdot \mathbf{v}_n)$ 
3:  $EC_{u,C} \leftarrow []$  % Initialize as an empty array
4:  $h \leftarrow h_{min}$ 
5:  $dh = (h_{max} - h_{min})/M$  % Step length with parameter  $M$ 
6: % Compute Euler curve of  $M + 1$  steps
7: while  $h \leq h_{max}$  do
8:    $EC_{u,C}.add(\kappa(C_{\vec{u},h}))$ 
9:    $h \leftarrow h + dh$ 
10: end while
11: return  $EC_{u,C}$ 

```

5. Theoretical Properties

In this section, we analyze and prove some important properties of the transform and the proposed loss function to evaluate our method.

5.1. Injectivity Property

Turner *et al.* [19] have shown that ECT over the space of simplicial complexes in \mathbb{R}^3 is injective. Betthausen [2] has generalized this result for ϕ -minimal sets. In our method, for computing the ECT-based loss function between two 3D images, we obtain a sequence of binary images corresponding to a sequence of threshold values for each 3D image and compute the ECT for each of the binary images (Algorithm 1, Line 9-13). In the following lemma, we provide a criterion to choose the number of thresholds so that such sequences of binary images are different for two different 3D images. Subsequently, sequences of ECTs for two different 3D images will also be different. Let, $\mathcal{F}(I, t)$ denote the sequence of t binary images from the 3D image I , for t equally spaced threshold values. Then we have the following result.

Lemma 5.1. *Given two 3D images I_1 and I_2 , $\mathcal{F}(I_1, t) = \mathcal{F}(I_2, t)$ iff $I_1 = I_2$, provided t is greater than or equal to the number of distinct voxel values in $I_1 \cup I_2$.*

Proof. We prove if $I_1 \neq I_2$ then $\mathcal{F}(I_1, t) \neq \mathcal{F}(I_2, t)$. Since $t \geq$ distinct number of voxel values in $I_1 \cup I_2$, we threshold on every distinct value in $I_1 \cup I_2$. As $I_1 \neq I_2$, at some coordinate x , $I_1(x) \neq I_2(x)$. Without loss of generality, let $I_1(x) < I_2(x)$. Now when we threshold at value $I_2(x)$, $I_1(x)$ becomes 0, while $I_2(x)$ is 1. This implies that $\mathcal{F}(I_1, t) \neq \mathcal{F}(I_2, t)$, as at the image thresholded at $I_2(x)$, we obtain different images. ■

We note, the distance function between two ECTs, in equation (4), satisfies the metric property [10, 19]. We use this in the subsequent proofs.

5.2. A Discussion on Stability of ECT-Based Loss

A commonly studied property in computational topology is the stability of a transform, that is the effect of perturbations on the input to the transformed output [5, 17]. We discuss a similar property for the case of ECT on binary images. We bound the possible change in the ECT of a binary image by a constant proportional to the size of the image and the size of the change in the input. We first prove a necessary lemma for our proof in Lemma 5.2. We then show that the distance between two EC's is bounded in Theorems 5.3 and 5.4. Subsequently we prove that the ECT is bounded in Corollary 1. We then discuss the effect of thresholding on stability.

Lemma 5.2. *A vertex in a d -dimensional grid is a part of at most 3^d cubes of any dimension.*

Proof. Consider a vertex $\mathbf{v}_0 = (x_1, x_2, \dots, x_d)$ in the interior of the grid. Every k -cube, that has \mathbf{v}_0 as a vertex, can be uniquely determined by k adjacent vertices of \mathbf{v}_0 along different dimensions in the grid. Along the i -th dimension \mathbf{v}_0 has two adjacent vertices $(x_1, x_2, \dots, x_i \pm 1, \dots, x_d)$, along positive and negative directions.

Now to count the number of k -cubes, that \mathbf{v}_0 is a part of, we simply count the number of ways we can choose k possible directions from the total d directions, which is $\binom{d}{k}$. Then for each of these chosen directions, we can either choose the adjacent vertices along the positive or negative direction, i.e. in 2^k ways. So the total number of k -cubes, that \mathbf{v}_0 is a part of, is $2^k \binom{d}{k}$. Summing up over all dimensions we get:

$$\sum_{k=0}^d \binom{d}{k} 2^k = 3^d.$$

Note that we performed the calculation for an interior vertex. For the vertices on the boundary of the grid, each will be a part of fewer cubes. So we can bound the number of cubes, that a single vertex is a part of, by 3^d . ■

Theorem 5.3. *Let I and I^* be two d -dimensional binary images with vertex set V s.t. they differ only at one voxel.*

Then along an arbitrary direction \vec{u} ,

$$D(\text{EC}_{\vec{u}, I}, \text{EC}_{\vec{u}, I^*}) \leq 3^d n / \sqrt{d}$$

where $n = |V|$ and $D(\text{EC}_{\vec{u}, I}, \text{EC}_{\vec{u}, I^})$ is the l_2 -norm between $\text{EC}_{\vec{u}, I}$ and $\text{EC}_{\vec{u}, I^*}$, i.e.*

$$D(\text{EC}_{\vec{u}, I}, \text{EC}_{\vec{u}, I^*}) = \sqrt{\int_{h_{\min}}^{h_{\max}} (\chi(C_{\vec{u}, h}) - \chi(C_{\vec{u}, h}^*))^2 dh}.$$

Proof. Let C be the cubical complex with vertices $\mathbf{v}_1, \mathbf{v}_2, \dots, \mathbf{v}_n$ associated with image I , using the construction described in Section 3.1. Let $h_1 \leq h_2 \leq \dots \leq h_n$ be the ordered list of heights of the vertices along \vec{u} . Then

$$D(\text{EC}_{\vec{u}, I}, \text{EC}_{\vec{u}, I^*}) = \sqrt{\sum_{i=1}^{n-1} \int_{h_i}^{h_{i+1}} (\chi(C_{\vec{u}, h}) - \chi(C_{\vec{u}, h}^*))^2 dh}.$$

Since EC is a piecewise constant function that changes only at the heights of vertices, we can rewrite it as,

$$D(\text{EC}_{\vec{u}, I}, \text{EC}_{\vec{u}, I^*}) = \sqrt{\sum_{i=1}^{n-1} (h_{i+1} - h_i) (\chi(C_{\vec{u}, h_i}) - \chi(C_{\vec{u}, h_i}^*))^2}.$$

Let, $e = \max\{h_2 - h_1, \dots, h_n - h_{n-1}\}$. Then

$$D(\text{EC}_{\vec{u}, I}, \text{EC}_{\vec{u}, I^*}) \leq \sqrt{e} \sqrt{\sum_{i=1}^{n-1} (\chi(C_{\vec{u}, h_i}) - \chi(C_{\vec{u}, h_i}^*))^2}.$$

Since for $\mathbf{x} \in \mathbb{R}^n$, $\|\mathbf{x}\|_2 \leq \|\mathbf{x}\|_1$, we have

$$\begin{aligned} D(\text{EC}_{\vec{u}, I}, \text{EC}_{\vec{u}, I^*}) &\leq \sqrt{e} \sum_{i=1}^{n-1} |\chi(C_{\vec{u}, h_i}) - \chi(C_{\vec{u}, h_i}^*)| \\ &= \sqrt{e} \sum_{i=1}^{n-1} \left| \sum_{j=0}^d (-1)^j (\text{Card}(C_{u, h_i}^j) - \text{Card}(C_{u, h_i}^{*j})) \right| \\ &\leq \sqrt{e} \sum_{i=1}^{n-1} \sum_{j=0}^d \left| (-1)^j (\text{Card}(C_{u, h_i}^j) - \text{Card}(C_{u, h_i}^{*j})) \right| \end{aligned}$$

Now for any sub-complex of C , the only cubes that can change are the ones that have \mathbf{v}_0 as a constituent vertex. So, using Lemma 5.2, we can bound the inner summation by 3^d . Thus we have

$$D(\text{EC}_{\vec{u}, I}, \text{EC}_{\vec{u}, I^*}) \leq \sqrt{e} 3^d n.$$

Next, we provide a bound for e to complete our proof. Every vertex $\mathbf{v}_0 = (x_1, \dots, x_d)$ has at least d adjacent vertices, say $\{\mathbf{v}_i : i = 1, \dots, d\}$ where $\mathbf{v}_i = (x_1, \dots, x_i \pm 1, \dots, x_d)$. We seek to find an upper bound of the minimum difference between the heights of the vertex \mathbf{v}_0 and any of its adjacent vertices over all possible directions $\vec{u} = (u_1, \dots, u_d) \in \mathbb{S}^{d-1}$.

This can be obtained by solving the following optimization problem:

$$\max_{\vec{u} \in \mathbb{S}^{d-1}} \min_{i \in \{1,2,\dots,d\}} (|\mathbf{v}_i \cdot \vec{u} - \mathbf{v}_0 \cdot \vec{u}|) = \max_{\vec{u} \in \mathbb{S}^{d-1}} \min_{i \in \{1,2,\dots,d\}} |u_i|$$

with $\|\vec{u}\| = 1$. The direction vector \vec{u} that maximises this function is the vector with all equal components, i.e., $(1/\sqrt{d}, \dots, 1/\sqrt{d})$. Thus, we obtain an upper bound of e as $1/\sqrt{d}$. ■

Theorem 5.4. *Let I and I^* be two d -dimensional binary images with vertex set V which differ at k voxels $\mathbf{v}_1, \dots, \mathbf{v}_k$. Then along an arbitrary direction \vec{u} ,*

$$D(\text{EC}_{\vec{u},I}, \text{EC}_{\vec{u},I^*}) \leq k3^d n / \sqrt{d}$$

where $n = |V|$.

Proof. From I , we construct a sequence of k images I_0, I_1, \dots, I_k , defined as follows:

$$I_i(\mathbf{v}) = \begin{cases} I^*(\mathbf{v}), & \mathbf{v} = \mathbf{v}_i \\ I_{i-1}(\mathbf{v}), & \text{otherwise} \end{cases}$$

for $i = 1, 2, \dots, k$ and $I_0 = I$. Observe that $I_k = I^*$ and that I_i and I_{i+1} differ by only one voxel for all i from 0 to $k-1$. Using the triangle inequality of a metric repeatedly and using Theorem 5.3,

$$\begin{aligned} D(\text{EC}_{\vec{u},I}, \text{EC}_{\vec{u},I^*}) &\leq \sum_{i=0}^{k-1} D(\text{EC}_{\vec{u},I_i}, \text{EC}_{\vec{u},I_{i+1}}) \\ &\leq \sum_{i=0}^{k-1} 3^d n / \sqrt{d} = k3^d n / \sqrt{d}. \end{aligned}$$

Corollary 1. *Let I and I^* be two d -dimensional binary images with vertex set V s.t. they differ at k voxels $\mathbf{v}_1, \dots, \mathbf{v}_k$. Then,*

$$D(\text{ECT}_I, \text{ECT}_{I^*}) \leq k3^d n / \sqrt{d} \times \text{Surface area of } \mathbb{S}^{d-1}$$

where $n = |V|$.

Proof. From theorem 5.4), the distance between two ECTs, in equation (4), can be bounded as

$$D(\text{ECT}_I, \text{ECT}_{I^*}) \leq k3^d n / \sqrt{d} \times \int_{\vec{u} \in \mathbb{S}^{d-1}} 1 \, du$$

We note, in our proposed algorithm, we perform a thresholding operation on the real-valued image to convert it into a set of binary images. On performing this operation, our transformation ceases to be continuous. As a result, obtaining a similar upper bound on our ECT-based loss function is not possible. ■

6. Experimental Results

We test the efficacy of our topological loss function by adding it to the SHAPR model and testing it on two biomedical datasets which have been used in the prior work [20,21].

1. **Red Blood Cells(RBC):** This is a dataset of 825 3D images obtained from a confocal microscope [16]. These cells are categorized into 9 designated categories: spherocytes, stomatocytes, discocytes, echinocytes, keratocytes, knizocytes, acanthocytes, cell clusters, and multilobates. The dimensionality of each image is $64 \times 64 \times 64$.
2. **Nuclei:** This is a dataset of 887 3D images of nuclei of human-induced pluripotent stem cells. The dimensionality of each image is $64 \times 64 \times 64$.

These datasets are publicly available.¹ Due to the limited dataset size, we follow the evaluation procedure of Waibel *et al.* [20]. That is we perform 5-fold cross-validation partitioning the dataset into five folds with a train/validation/test split of 60%/20%/20%. We ensure that each image of a dataset appears in the test split exactly once. We compare three different approaches to determine the improvements of our proposed loss. Namely, the baseline SHAPR [21], SHAPR with the Wasserstein-based loss [20], and finally SHAPR with our ECT-based loss. For the baseline SHAPR and the Wasserstein loss based implementation, we use the code made available by Waibel *et al.* [20].²

We follow the same training procedure as in Waibel *et al.* [20], that is, we train all the variants of SHAPR for a maximum of 100 epochs, using early stopping with a patience parameter of 15 epochs. We also perform data augmentation before training by performing random horizontal or vertical flipping as well as 90° rotations with a 33% probability for an augmentation to be applied on a sample. We track our experiments using WANDB [3]. In the testing phase, we apply Otsu's thresholding [14] to convert our image into a binary image. This binary image is then compared with the ground truth to calculate three metrics from the prior works, namely, IoU error, relative Volume error and relative surface error. We drop the roughness error from the prior works [20,21] as we believe it does not serve as a useful metric to measure the accuracy of the reconstruction. It is defined as the difference between the predicted image and a 3D smoothened Gaussian version of the image. As seen in Figure 3, even the ground truth is rough in nature and will have a large roughness error.

We train the baseline and Wasserstein loss based model using the hyperparameters reported in Waibel *et al.* [20]. For our ECT-based loss model, we use a scaling parameter $\lambda = 0.01$. The number of thresholds we consider per pair of

¹<https://hmgubox2.helmholtz-muenchen.de/index.php/s/YAds7dA2TcxSDtr>

²https://github.com/aidos-lab/SHAPR_torch

IoU Error (\downarrow)			Volume Error (\downarrow)			Surface Error (\downarrow)		
Dataset	RBC	Nuclei	Dataset	RBC	Nuclei	Dataset	RBC	Nuclei
Baseline	0.49 ± 0.09	0.64 ± 0.10	Baseline	0.50 ± 0.35	0.61 ± 0.57	Baseline	0.20 ± 0.14	0.38 ± 0.31
Wasserstein	0.49 ± 0.10	0.62 ± 0.10	Wasserstein	0.45 ± 0.35	0.66 ± 0.58	Wasserstein	0.22 ± 0.16	0.38 ± 0.31
ECT	0.48 ± 0.12	0.66 ± 0.11	ECT	0.40 ± 0.32	0.53 ± 0.51	ECT	0.19 ± 0.15	0.34 ± 0.29
(a)			(b)			(c)		

Table 1. Performance of different variants of the SHAPR model on the Red blood cells and Nuclei dataset. Represented as mean \pm standard deviation, with the best performing algorithm highlighted in **bold**.

images is 40 (n in Algorithm 1). The number of directions we consider in evaluating the integral of the distance function is 100 (l in Algorithm 2). Finally, the parameter M or number of steps (Algorithm 3) we take as 30.

We can see the results of our experiments in Table 1. We observe that on most metrics over both the datasets the ECT-based loss performs the best. We see the most significant improvements in the Volume Error where the ECT-based loss outperforms the previous best by 11.2% on the RBC dataset and 19.6% on the Nuclei dataset. We also see significant improvements in the Surface Error, the ECT-based loss outperforms the previous best by 14.5% on the RBC dataset and 10.8% on the Nuclei dataset.

We can visualize the outputs of the various methods on the Nuclei and RBC dataset in Figures 3 and 4. It is interesting to observe that adding a topology based loss clearly improves the quality of the reconstruction, as neither of the topology based methods produce artifacts while the baseline does (Figure 3). This is expected since the topology based losses optimize for topological invariants to obtain topologically correct reconstructions. We also observe in Figure 4 that the topology based methods are trying to capture the valley in the ground truth, while the baseline does not. Note that in the current reconstruction problem we cannot expect perfect reconstructions since the problem is ill-posed.

7. Conclusion

In this paper, we present a novel ECT-based topological loss function that can be used to aid the training of neural networks for the challenging task of 3D image reconstruction from a single image. We not only show empirical improvement but also discuss some important theoretical properties of our loss and ECT in general. Our ECT-based loss can be used to describe the topological distance between any two images. Our method could thus potentially be employed to aid neural networks in *any* vision task, including image segmentation or 3D image reconstruction from multiple images. Another natural extension of our work would be to consider the persistent homology transform (PHT) instead of the ECT. While both are injective, the persistence diagram is more informative than the Euler Curve at the cost of additional computation. It would be interesting to explore whether this provides any benefit.

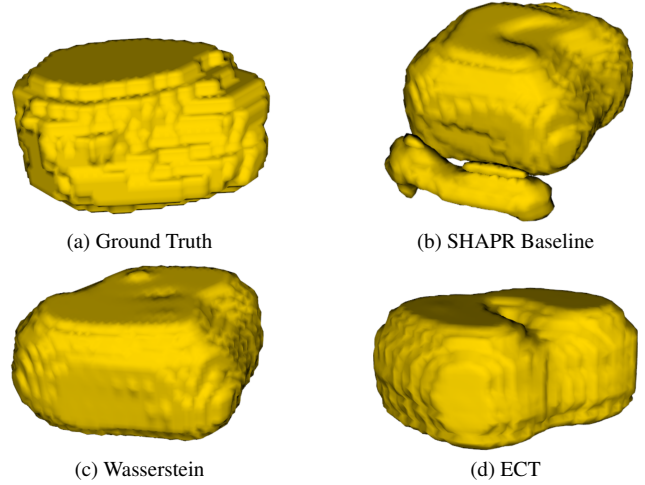


Figure 3. Qualitative results on the Nuclei dataset. The result using (b) SHAPR has topological artifacts, whereas, the results using topology-based loss functions (c) Wasserstein and (d) ECT do not have any topological artifacts.

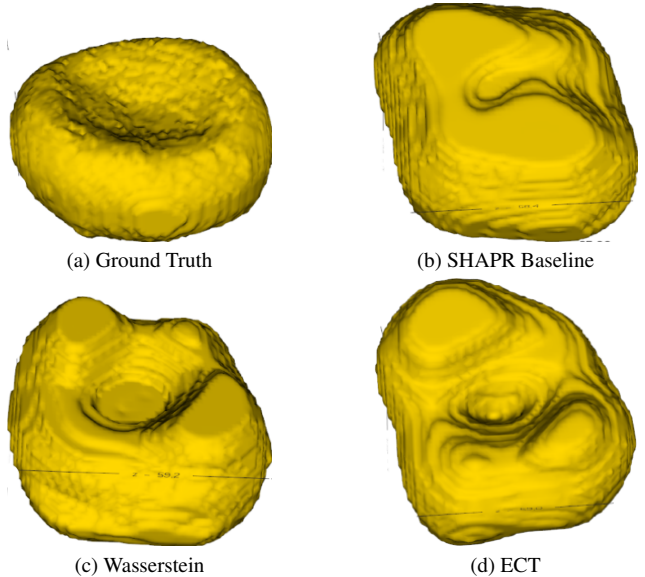


Figure 4. Qualitative results on the RBC dataset. The curvature of the valley in the ground truth is approximately captured by the topology-based loss functions (c) Wasserstein and (d) ECT, whereas, (b) the baseline is unable to capture it.

References

- [1] Madjid Allili, Konstantin Mischaikow, and Allen Tannenbaum. Cubical homology and the topological classification of 2d and 3d imagery. In *Proceedings 2001 international conference on image processing (Cat. No. 01CH37205)*, volume 2, pages 173–176. IEEE, 2001. 2
- [2] Leo M Betthausen. *Topological reconstruction of grayscale images*. PhD thesis, University of Florida, 2018. 5
- [3] Lukas Biewald. Experiment tracking with weights and biases, 2020. Software available from wandb.com. 7
- [4] Angel X Chang, Thomas Funkhouser, Leonidas Guibas, Pat Hanrahan, Qixing Huang, Zimo Li, Silvio Savarese, Manolis Savva, Shuran Song, Hao Su, et al. Shapenet: An information-rich 3d model repository. *arXiv preprint arXiv:1512.03012*, 2015. 1, 2
- [5] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. In *Proceedings of the twenty-first annual symposium on Computational geometry*, pages 263–271, 2005. 6
- [6] Herbert Edelsbrunner and John L Harer. *Computational topology: an introduction*. American Mathematical Society, 2022. 2
- [7] Xian-Feng Han, Hamid Laga, and Mohammed Bennamoun. Image-based 3d object reconstruction: State-of-the-art and trends in the deep learning era. *IEEE transactions on pattern analysis and machine intelligence*, 43(5):1578–1604, 2019. 1, 2
- [8] Felix Hensel, Michael Moor, and Bastian Rieck. A survey of topological machine learning methods. *Frontiers in Artificial Intelligence*, 4:681108, 2021. 2
- [9] Xiaoling Hu, Yusu Wang, Li Fuxin, Dimitris Samaras, and Chao Chen. Topology-aware segmentation using discrete morse theory. *arXiv preprint arXiv:2103.09992*, 2021. 2
- [10] Qitong Jiang, Sebastian Kurtek, and Tom Needham. The weighted euler curve transform for shape and image analysis. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, pages 844–845, 2020. 6
- [11] Nikos Kolotouros, Georgios Pavlakos, and Kostas Daniilidis. Convolutional mesh regression for single-image human shape reconstruction. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4501–4510, 2019. 1, 2
- [12] Alexander LeNail. Nn-svg: Publication-ready neural network architecture schematics. *J. Open Source Softw.*, 4(33):747, 2019. 4
- [13] Doruk Oner, Adélie Garin, Mateusz Kozinski, Kathryn Hess Bellwald, and Pascal Fua. Persistent homology with improved locality information for more effective delineation. Technical report, 2022. 2
- [14] Nobuyuki Otsu. A threshold selection method from gray-level histograms. *IEEE Transactions on Systems, Man, and Cybernetics*, 9(1):62–66, 1979. 7
- [15] Bastian Rieck, Tristan Yates, Christian Bock, Karsten Borgwardt, Guy Wolf, Nicholas Turk-Browne, and Smita Krishnaswamy. Uncovering the topology of time-varying fMRI data using cubical persistence. *Advances in Neural Information Processing Systems*, 33:6900–6912, 2020. 2
- [16] Greta Simionato, Konrad Hinkelmann, Revaz Chachanidze, Paola Bianchi, Elisa Fermo, Richard van Wijk, Marc Leonetti, Christian Wagner, Lars Kaestner, and Stephan Quint. Red blood cell phenotyping from 3d confocal images using artificial neural networks. *PLoS Computational Biology*, 17(5):e1008934, 2021. 7
- [17] Primož Skraba and Katharine Turner. Wasserstein stability for persistence diagrams. *arXiv preprint arXiv:2006.16824*, 2020. 6
- [18] Xingyuan Sun, Jiajun Wu, Xiuming Zhang, Zhoutong Zhang, Chengkai Zhang, Tianfan Xue, Joshua B Tenenbaum, and William T Freeman. Pix3d: Dataset and methods for single-image 3d shape modeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2974–2983, 2018. 1, 2
- [19] Katharine Turner, Sayan Mukherjee, and Doug M Boyer. Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA*, 3(4):310–344, 2014. 1, 2, 3, 5, 6
- [20] Dominik J. E. Waibel, Scott Atwell, Matthias Meier, Carsten Marr, and Bastian Rieck. Capturing shape information with multi-scale topological loss terms for 3d reconstruction. In Linwei Wang, Qi Dou, P. Thomas Fletcher, Stefanie Speidel, and Shuo Li, editors, *Medical Image Computing and Computer Assisted Intervention (MICCAI)*, pages 150–159, Cham, Switzerland, 2022. Springer. 1, 2, 4, 7
- [21] Dominik J. E. Waibel, Niklas Kiermeyer, Scott Atwell, Ario Sadafi, Matthias Meier, and Carsten Marr. SHAPR predicts 3d cell shapes from 2d microscopic images. *iScience*, 25(11):105298, 2022. 1, 2, 7
- [22] Dominik J. E. Waibel, Ernst Röell, Bastian Rieck, Raja Giryes, and Carsten Marr. A diffusion model predicts 3d shapes from 2d microscopy images. *arXiv preprint arXiv:2208.14125*, 2022. 2
- [23] Fan Wang, Huidong Liu, Dimitris Samaras, and Chao Chen. Topogan: A topology-aware generative adversarial network. In *Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part III 16*, pages 118–136. Springer, 2020. 2