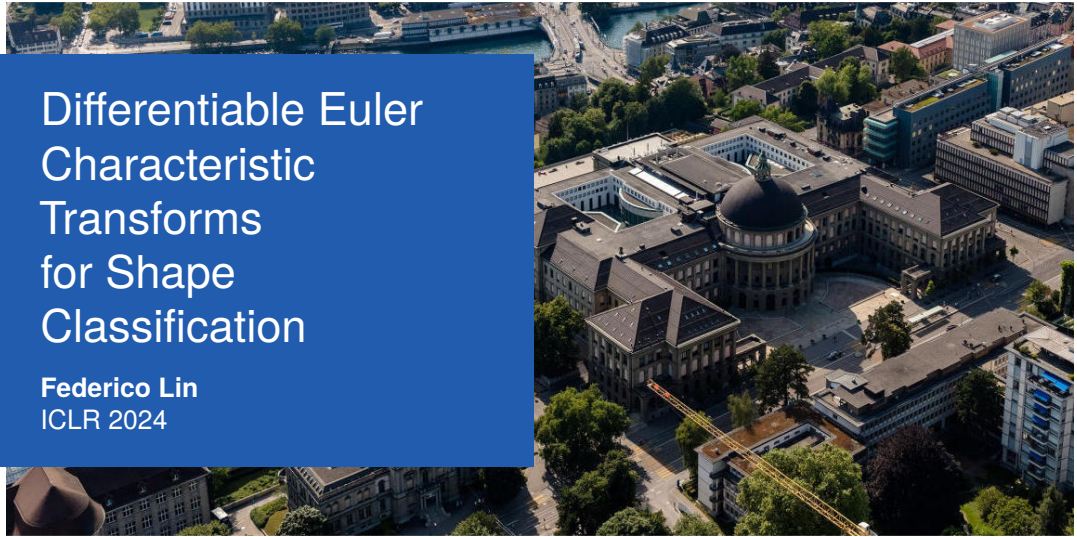


Differentiable Euler Characteristic Transforms for Shape Classification

Federico Lin

ICLR 2024

Today's Path

1. **Motivation:** why local GNN message passing can miss global structure.
2. **Related work:** one nearby idea using simplices.
3. **Euler characteristic:** a simple topological count.
4. **ECT:** turn one count into directional curves.
5. **DECT:** replace hard switches by smooth switches.
6. **Experiments:** what the method is used for and where it helps.

The main story is:

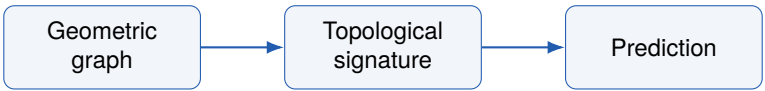
$$\text{graph} \longrightarrow \text{topological sweep} \longrightarrow \text{trainable neural layer.}$$

Why do we need topology for graph and shape classification?

We first look at why standard GNNs are powerful, but still mostly local.

The Big Question

Can topology become a trainable neural network layer?



We want global shape information, but still compatible with gradient descent.

Why Not Just Use a GNN?

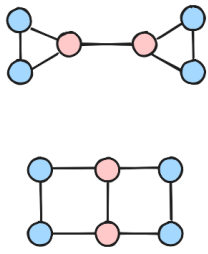
A graph neural network updates each node by aggregating information from its neighbors.

After:

- 1 layer: a node sees its 1-hop neighborhood,
- 2 layers: it sees 2-hop neighborhoods,
- k layers: it sees up to k-hop neighborhoods.

This is strong for **local patterns**, but global structures such as loops or long-range organization can be harder.

Motivation: add an explicit topological signal.



One nearby idea: make message passing more topological.

Instead of only passing information between nodes, simplicial message passing also uses higher-order structures such as edges and triangles.

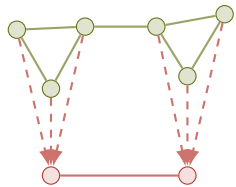
Related Work: Message Passing Simplicial Networks

Weisfeiler and Lehman Go Topological: Message Passing Simplicial Networks

Main idea:

- go beyond node-only message passing,
- lift a graph to a simplicial complex,
- let higher-order objects exchange information.

A triangle is not treated only as three independent edges. It can also act as a higher-order unit.



message passing through simplices

The basic topological count behind the whole method.

We start from a simple formula: vertices minus edges plus faces.

Euler Characteristic

For a simplicial complex K, the Euler characteristic is an alternating count:

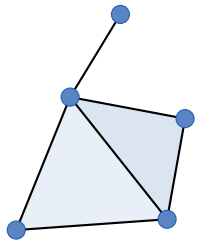
$$\chi(K) = \sum_p (-1)^p c_p$$

where c_p is the number of p-simplices.

In low dimensions:

$$\chi(K) = c_0 - c_1 + c_2 - c_3 + \cdots$$

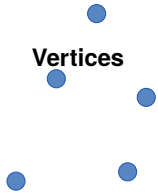
$$\chi(K) = |\text{vertices}| - |\text{edges}| + |\text{faces}| - \cdots$$



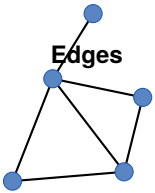
example graph

$|V| = 5,\ |E| = 6,\ |F| = 2,\ \chi = 1$

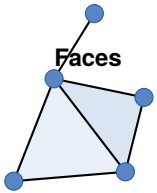
How the Counting Works



$|V| = 5$



$|E| = 6$

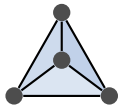


$|F| = 2$

$$\chi = |V| - |E| + |F| = 5 - 6 + 2 = 1$$

Examples with $\chi = 2$

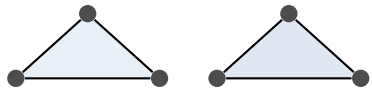
Tetrahedron surface graph



$$|V| = 4, \quad |E| = 6, \quad |F| = 4$$

$$\chi = 4 - 6 + 4 = 2$$

Two separate filled triangles



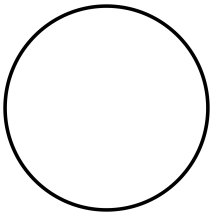
$$|V| = 6, \quad |E| = 6, \quad |F| = 2$$

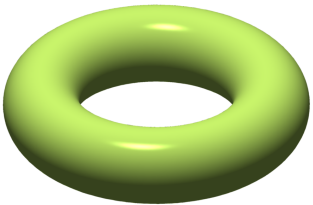
$$\chi = 6 - 6 + 2 = 2$$

One Number is Not Enough

The Euler characteristic compresses a whole shape into one number.

$$\chi(\text{circle}) = 0, \qquad \chi(\text{torus}) = 0.$$





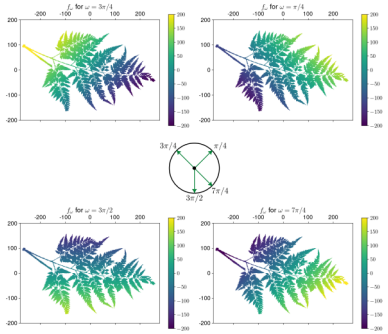
Same number, very different shapes.

How do we turn one Euler characteristic number into a shape descriptor?

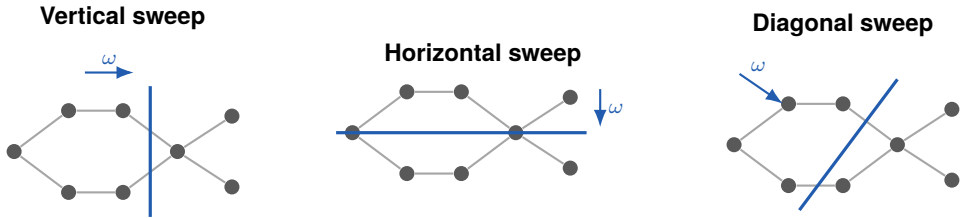
We sweep the graph from many directions and record a curve each time.

Directional Height Functions

Different directions give different views of the same geometric graph.



Same Graph, Different Directions

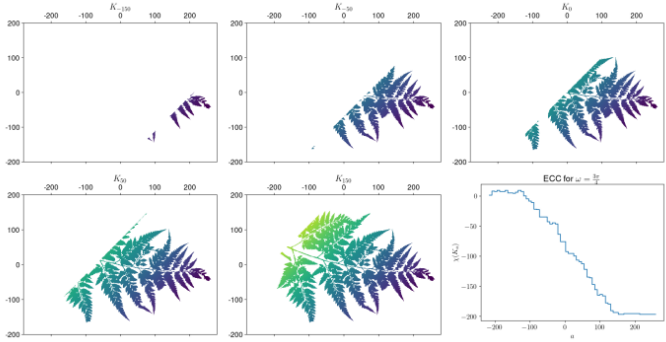


The same graph gives different Euler characteristic curves depending on the direction.

From Sweep to Curve

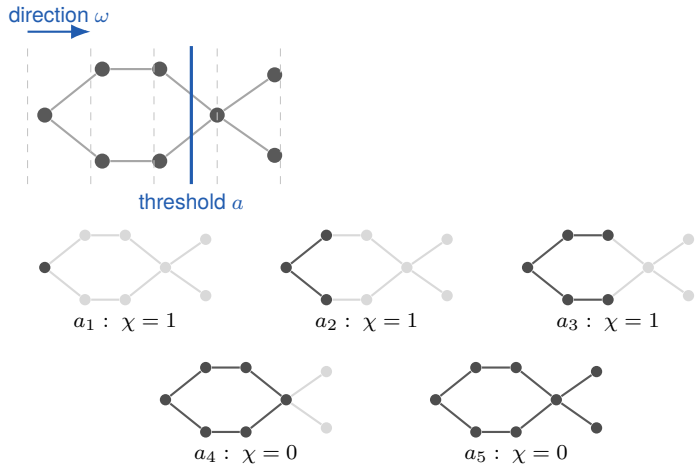
For one fixed direction, the shape is revealed step by step.

The Euler characteristic changes during the sweep.



Sweeping the Graph: One Direction

Instead of computing one number, we reveal the graph gradually from one direction and track how the revealed graph changes.



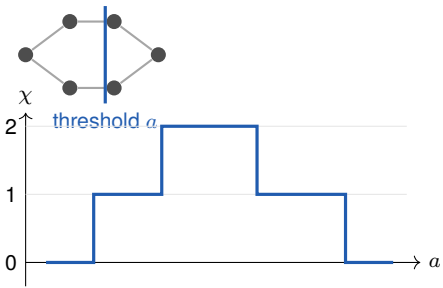
Euler Characteristic Curve

For one direction ω,

$$\mathrm{ECC}_\omega : \mathbb{R} \to \mathbb{Z}, \qquad a \mapsto \chi(K_a).$$

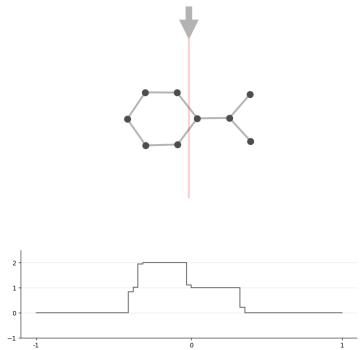
As a moves, we record how χ changes.

A vertex usually increases χ. An edge may decrease χ, especially when it connects two already visible vertices.

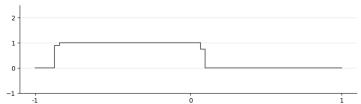


Euler characteristic curve

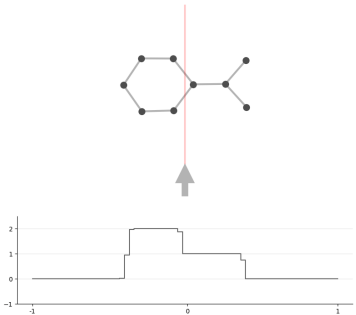
ECT Snapshot 1



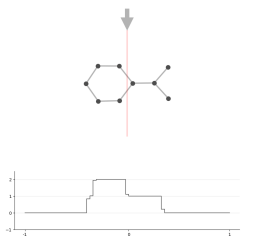
ECT Snapshot 2



ECT Snapshot 3



ECT as a Dynamic Topological Signature



The GIF shows how the topological signature changes as the direction varies.

Euler Characteristic Transform

One direction gives one Euler characteristic curve.

The Euler Characteristic Transform collects these curves over many directions:

$$\mathrm{ECT}(K) : S^{d-1} \to \mathbb{Z}^{\mathbb{R}}, \qquad \omega \mapsto \mathrm{ECC}_{\omega}.$$

So the ECT is a family of ECCs, one for each direction.

Where does the neural-network part enter?

The paper replaces discrete jumps by smooth approximations, so the transform can be trained by backpropagation.

Classical ECT: Where the Hard Switch Appears

Classical ECT is a signed sum of pieces that appear during the sweep:

$$\mathrm{ECC}_\omega(a) = \sum_i s_i \, \mathbf{1}[h_i(\omega) \le a].$$

Here:

$$s_i = \begin{cases} +1 & \text{vertices and faces,} \\ -1 & \text{edges.} \end{cases}$$

So each vertex, edge, or face contributes only after its height crosses the threshold.

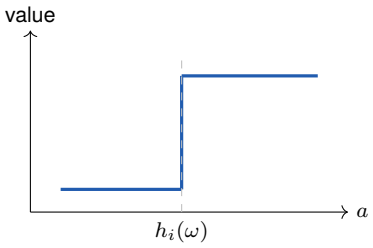
Why Classical ECT is Hard to Train

The hard switch is:

$$\mathbf{1}[h_i(\omega) \leq a].$$

It suddenly jumps from 0 to 1.

That is mathematically clean, but bad for gradient descent because the gradient is zero almost everywhere and undefined at the jump.



hard indicator

The Paper's Main Move

$$\cancel{\mathbf{1}[h_i(\omega) \leq a]} \qquad \Longrightarrow \qquad \operatorname{sigmoid}(\lambda(a - h_i(\omega)))$$

Instead of a hard switch, use a smooth switch.

$$\mathrm{DECT}(a, \omega) = \sum_i s_i \operatorname{sigmoid}(\lambda(a - h_i(\omega)))\,.$$

Now the topological signature is differentiable.

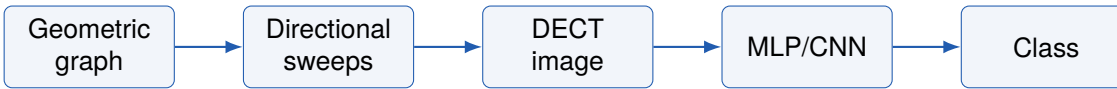
What Becomes Learnable?

Once the transform is differentiable, the model can learn:

- directions,
- thresholds,
- input coordinates,
- downstream neural network weights.

The ECT becomes a trainable layer, not only a fixed descriptor.

Pipeline



The novel part is making the topological representation differentiable.

What does DECT actually do in practice?

We look at inputs, learned directions, reconstruction-style losses, and classification results.

What is the Input? MNIST as a Graph

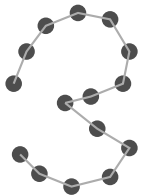
The experiments use graph-like versions of image data.

For MNIST:

- pixels or superpixels become nodes,
- spatial adjacency gives edges,
- node positions give geometry.

So DECT sees a geometric graph, not just a flat image.

sketch: digit as point cloud



Learning Directions

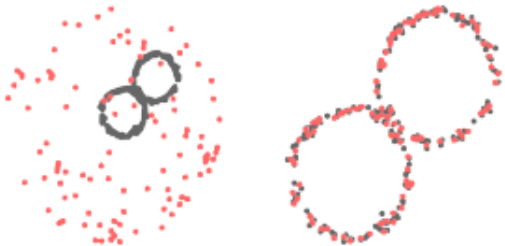
The viewing directions do not have to be fixed.

They can be learned from data.

This lets the model choose informative views.

(a) Learning directions

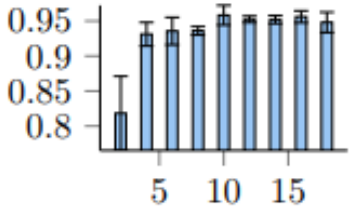
Learning Coordinates



(b) Learning coordinates

DECT can also be used as a loss to optimize coordinates toward a target topological signature.

Accuracy with Few Directions



Even a small number of directions gives useful classification performance.

MNIST-Superpixel

Method	Accuracy	Epoch runtime (s)
GAT (Veličković et al., 2018)	95.54 ± 0.21	42.26
GCN (Kipf & Welling, 2017)	90.71 ± 0.22	83.41
GIN (Xu et al., 2019)	96.49 ± 0.14	39.22
GraphSage (Hamilton et al., 2017)	97.31 ± 0.10	113.12
MLP	95.34 ± 0.14	22.74
ECT+CNN (ours)	93.00 ± 0.80	4.50
ECT+MLP (ours)	97.20 ± 0.10	10.80

DECT models are competitive with GNNs, while being much faster per epoch.

Graph Benchmarks

	Params.	BZR	COX2	DHFR	Letter-low	Letter-med	Letter-high
GAT	5K	80.3 ± 2.0	79.2 ± 2.6	72.8 ± 3.2	90.0 ± 2.2	63.7 ± 6.0	43.7 ± 4.1
GCN	5K	80.5 ± 2.4	**79.4 ± 1.8**	**76.7 ± 3.8**	81.4 ± 1.6	62.0 ± 2.1	43.1 ± 1.7
GIN	9K	81.7 ± 4.9	77.9 ± 2.4	64.7 ± 8.3	85.0 ± 0.6	67.1 ± 2.5	50.9 ± 3.5
ECT+CNN (ours)	4K	**81.8 ± 3.2**	70.4 ± 0.9	67.9 ± 5.0	**91.5 ± 2.1**	**76.2 ± 4.8**	**63.8 ± 6.0**
ECT+CNN (ours)	65K	84.3 ± 6.1	74.6 ± 4.5	72.9 ± 1.6	96.8 ± 1.2	86.3 ± 2.0	85.4 ± 1.3

The strongest gains appear on noisy Letter graph datasets.

What should we take away?

DECT is elegant and fast, but it also has practical trade-offs.

My Take: Pros and Cons

What I like

- elegant bridge between topology and deep learning;
- more interpretable than many geometric models;
- fast compared with several GNN baselines;
- usable as both a layer and a loss.

What I would question

- only finitely many directions in practice;
- alignment of shapes still matters;
- choices of directions and thresholds are task-dependent;
- smoothing may lose exact topological information.

Takeaway

The Euler Characteristic Transform summarizes how topology appears from many directions.

The paper makes it differentiable by replacing hard jumps with smooth transitions.

This allows topology to become part of an end-to-end trainable model.

$$\text{topology} \quad + \quad \text{gradient descent} \quad = \quad \mathrm{DECT}$$

Questions?

ETH zürich

Federico Lin
ETH Zurich
fedlin@student.ethz.ch

Code:
https://github.com/aidos-lab/DECT

Differentiable Euler Characteristic Transforms for Shape Classification