- https://github.com/vadimlebovici/eulearning: eulearning is a Python package to compute Euler characteristic profiles of multi-parameter filtrations.
- https://arxiv.org/pdf/2212.01666v2/ : This paper contains Euler Charactersitic details for shape smoothening which is very useful for big data
- https://github.com/lcrawlab/SINATRA: Written in R for sub-image selection problem. A statistical model is used to classify the shapes based on their topological summaries. Here, we make use of a Gaussian process classification model with a probit link function. Thus, it can be used for summarisation of topology of 3-D shapes.
- https://github.com/aidos-lab/dect-evaluation/tree/main/experiment: Link for DECT evaluation by the author
- https://github.com/aidos-lab/dect/tree/main: Link for DECT evaluation by the author.

Possible Utility of DECT

- Can be used for optimisation of point-clouds, mesh or grids.
- Can be used for benchmarking classification techniques against persistence diagrams methodology.
- Can be utilised as the loss function for classification tasks.

Detailed Summary of "Differentiable Euler Characteristic Transforms for Shape Classification"

Objective

- To enhance the traditional Euler Characteristic Transform (ECT), making it differentiable for integration into deep learning models.
- DECT bridges topological data analysis (TDA) and machine learning by addressing the limitations of ECT.

Background

- **ECT** is a topological descriptor that captures geometrical and topological features of data (e.g., graphs, point clouds, meshes).
- It uses the Euler Characteristic, a scalar value computed as the alternating sum of simplices (e.g., vertices, edges, faces).
- Traditional ECT suffers from:
  - Lack of task-specific learning.
  - Computational inefficiency in handling large datasets or complex data structures.

Maths

$$\begin{aligned} \mathrm{ECT}\colon & S^{n-1} \times \mathbb{R} \to \mathbb{Z} \\ & \xi, h \mapsto \sum_{k}^{\dim K} (-1)^k \sum_{\sigma_k} S\left(\lambda\left(h - f_\xi(x_{\sigma_k})\right)\right) \end{aligned},$$

- σ_k is a k-simplex, and $x_{\sigma k}$ represents its feature vector.
- S is the sigmoid function, providing a smooth approximation.
- λ controls the steepness of the sigmoid, balancing approximation tightness.

Contributions

1. **Differentiable ECT (DECT):**
   - Introduced a smooth approximation of ECT using sigmoid functions, enabling gradient-based optimization.
   - Allows ECT to serve as a trainable layer or loss function in neural networks.
2. **Scalability:**
   - Implements parallelized computations with GPU acceleration, ensuring high performance across datasets and tasks.
   - Handles mixed data modalities (point clouds, graphs, meshes) efficiently.
3. **Improved Integration:**
   - Traditional ECT concatenates features, losing directional information. DECT uses a learnable embedding for Euler Characteristic Curves (ECCs), preserving key details.
   - Outputs are permutation-invariant and suitable for classification tasks.
4. **Applications:**
   - Applied to diverse datasets (synthetic and real-world), showcasing DECT's flexibility in tasks such as shape classification and point cloud optimization.

Key Methods

1. **Mathematical Improvements:**
   - The ECT is redefined using differentiable summations of simplicial components with sigmoid functions.
   - Filters are applied to create multi-scale, nested subcomplexes, capturing topology across scales.
2. **Architecture:**
   - DECT is integrated into simple Multi-Layer Perceptrons (MLPs) and Convolutional Neural Networks (CNNs).
   - Combines ECCs into a high-dimensional representation using pooling layers and classifies with MLPs.
3. **Optimization:**

- DECT optimizes both directions and point cloud positions to match target topological structures.
- Supports both shape classification and geometry-aware point cloud adjustments.

Experiments

1. **Synthetic Data Classification:**
    - Tested on 2-manifolds (spheres, tori, Möbius strips) represented as point clouds, graphs, and meshes.
    - DECT achieved 100% classification accuracy across all modalities.
2. **Real-World Data:**
    - Applied on MNIST-Superpixel, a geometric graph dataset.
    - Results:
        - DECT (with CNNs) achieved competitive accuracy (97.2%) while being **10×** faster than GNN-based methods.
3. **Point Cloud Optimization:**
    - DECT adjusted noisy point clouds to fit target geometries (e.g., circles).
    - Demonstrated robustness to noise and flexibility in learning topological representations.
4. **Comparison with GNNs:**
    - Benchmarked against state-of-the-art methods on small graph datasets (e.g., BZR, COX2, Letter datasets).
    - Achieved similar or better performance with fewer parameters and faster runtime.

Results

- DECT provides:
    - **High accuracy:** Competitive with more complex models like Graph Attention Networks (GATs).
    - **Efficiency:** Orders of magnitude faster training due to vectorized operations and GPU support.
    - **Flexibility:** Works with varied data formats and sizes without requiring extensive preprocessing.

Table 3: Results of 5 runs on small graph benchmark data sets. Parameter numbers are approximate because the number of classes differs. The high consistency and performance of our method on the 'Letter' data sets is notable.

	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

Advantages

- **Task-Specific Learning:** Unlike traditional ECT, DECT adapts its parameters to optimize for specific tasks.
- **Scalability:** Parallelized computations make DECT applicable to large-scale datasets.
- **Robustness:** Handles noisy and complex data effectively, preserving essential topological features.