

Encoding the Euler Characteristic Transform

Nello Blaser
Odin Hoff Gardaa
Lars M. Salbu

Department of Informatics, University of Bergen, Norway

NELLO.BLASER@UIB.NO

ODIN.GARDA@UIB.NO

LARS.SALBU@UIB.NO

Elena Xinyi Wang
Bastian Rieck

Department of Informatics, University of Fribourg, Switzerland

XINYI.WANG@UNIFR.CH

BASTIAN.GROSSENBACHER@UNIFR.CH

Abstract

The Euler Characteristic Curve (ECC) records the Euler characteristic of a linearly embedded cell complex as a function of filtration height in a given direction, and the Euler Characteristic Transform (ECT) is the injective shape descriptor obtained by collecting ECCs over many directions. How the ECT is encoded for a neural network is itself an inductive bias, conventionally fixed by discretizing each ECC. We introduce a continuous encoding: for each direction and each vertex it records the net Euler-characteristic change attributed to that vertex, producing a per-direction token sequence that a small transformer maps to a feature vector. We separate the resulting pipeline into two stages on orthogonal axes: an *ECC encoder* that acts within each direction, mapping its curve to a fixed-length vector, and an *ECT representation* that acts across directions, aggregating the per-direction vectors into one. We study six ECT representation architectures spanning a range of inductive biases, from a structure-agnostic feedforward baseline to convolutional and complex-valued models that preserve equivariance under planar rotations. Across six classification benchmarks covering point clouds, graphs, cubical complexes, and meshes, the continuous encoding improves accuracy on all six datasets, and control experiments attribute the gain to the tokenization itself rather than to the added transformer capacity. The representation architecture matters less than the encoding, and the payoff from its inductive biases depends on the encoding: a feedforward network performs best under continuous encoding but is less robust under discretization than convolutional architectures.

1. Introduction

The Euler Characteristic Transform (ECT) has emerged as a powerful descriptor for shape analysis, combining topological connectivity information and geometric structure into a computationally efficient representation. Originally introduced by [Turner et al. \(2014\)](#), the ECT operates by filtering a shape along multiple directions and computing the Euler characteristic—a topological invariant that captures information about connected components, holes, and voids—at each filtration step. The resulting transform has been proven injective for embedded simplicial complexes in $\mathbb{R}^d$ ([Curry et al., 2022](#); [Ghrist et al., 2018](#); [Schapira, 1995](#); [Turner et al., 2014](#)), making it a sufficient descriptor for shape comparison and classification tasks. The Differentiable Euler Characteristic Transform (DECT) of [Röell and Rieck \(2024\)](#) made the computation differentiable with respect to both sampling directions and input coordinates, enabling end-to-end learning within neural networks. The

(D)ECT has subsequently been used in several distinct roles in neural network pipelines: (i) as a global shape descriptor fed to a classifier (McGuire, 2024; Röell and Rieck, 2024; Toscano-Duran et al., 2026), (ii) as a local feature for graph representation learning (Amboage et al., 2026; von Rohrscheidt and Rieck, 2025), (iii) as a topology-aware loss term (Nadimpalli et al., 2023; Röell and Rieck, 2024), and (iv) as an invertible descriptor for generative modeling (Röell and Rieck, 2025). Our work is squarely in the first category and asks how the global ECT should be encoded once it has been produced.

The integration of ECT into neural network architectures has taken several distinct approaches, with different strategies for encoding the directional and multi-scale information inherent in the transform. In existing work the ECT is typically discretized first, i.e., the Euler Characteristic Curve (ECC) along each direction is evaluated at a fixed grid of height thresholds, producing a fixed-dimensional matrix that the downstream network operates on. Prior architectures process this matrix either as an unordered set of curves, via a DeepSets (Zaheer et al., 2017) model (Röell and Rieck, 2024), or as a 2D image fed to a standard CNN (McGuire, 2024; Röell and Rieck, 2024).

The encoding strategies described above all act on a discretized ECT: each ECC is evaluated at a fixed grid of H height thresholds, producing a $D \times H$ matrix, where D is the number of directions. The grid is a modeling choice with real costs. The resolution H becomes a hyperparameter that has to be tuned per dataset, trading expressivity against memory and compute—and the natural scale of that trade-off varies between datasets, since the relevant features in a 28×28 silhouette and in a high-resolution building mesh “live” at very different scales. The grid also forces the same resolution in regions where the ECC is changing rapidly and in regions where it is flat, even though all of the information can be attributed to the vertices. This suggests an alternative: encode the ECT by the steps *themselves* rather than by samples of the function between them. A representation that records what changes, where, and by how much immediately sidesteps the resolution hyperparameter entirely and is exact by construction.

Contributions. This paper makes the following contributions:

- We introduce a *continuous* representation of the ECT that records, for each sampling direction and each vertex of the complex, the net Euler characteristic change attributed to that vertex. The representation is grid-free, exact, parsimonious, and inherently differentiable, and a small transformer encoder maps it to a per-direction feature vector that replaces the discretized ECT.
- We systematically study six neural network architectures for representing the ECT: a feedforward network, Deep Set, 1D convolution, 2D convolution, 1D complex convolution, and a hybrid model. The latter four are designed to preserve equivariance under planar rotations of the input; the first two serve as baselines without this structure.
- We provide an empirical evaluation across six classification benchmarks for different data types—point clouds, graphs, cubical complexes, and meshes—comparing all architectures with both continuous and discrete ECT representations, with attention to both accuracy and training stability.

2. Continuous ECC Encoding

We propose a novel *continuous* representation that stores the ECCs exactly, without using a grid. For the sake of notational simplicity, we reserve our exposition to the 2D case. For a cell complex K linearly embedded in $\mathbb{R}^2$ with N vertices, and a given direction $\mathbf{w}_i = (\cos \theta_i, \sin \theta_i)$ corresponding to angle θ_i , the filtration assigns to each cell $\tau \in K$ the height

$$h(\tau, \mathbf{w}_i) = \max_{v \in \tau} \langle v, \mathbf{w}_i \rangle,$$

which is attained at the single vertex of τ with maximum projection onto $\mathbf{w}_i$. Define the *responsible vertex* of a cell τ in direction $\mathbf{w}_i$ as

$$r(\tau, \mathbf{w}_i) = \arg \max_{v \in \tau} \langle v, \mathbf{w}_i \rangle.$$

Each cell can then be attributed entirely to its responsible vertex. For each vertex v and direction $\mathbf{w}_i$, the net change in Euler characteristic as the filtration sweeps through $\langle v, \mathbf{w}_i \rangle$ is

$$\Delta\chi(v, \mathbf{w}_i) = \sum_{k \geq 0} (-1)^k |\{ \tau \in k\text{-cells} : r(\tau, \mathbf{w}_i) = v \}|,$$

combining the vertex’s own contribution with the signed contributions of every higher-dimensional cell that enters the filtration at the same threshold. Summing $\Delta\chi(v, \mathbf{w}_i) \mathbf{1}[h \geq \langle v, \mathbf{w}_i \rangle]$ over all vertices reproduces the ECC exactly, and the discretized ECT can be recovered at any resolution. Vertices with $\Delta\chi(v, \mathbf{w}_i) = 0$ for all directions ($\mathbf{w}_i$) carry no information and are discarded, leaving $N_{\text{active}} \leq N$ tokens. For point clouds, where the complex has no higher-dimensional cells, $\Delta\chi \equiv 1$ at every vertex and $N_{\text{active}} = N$. Each vertex contributes its height $\langle v, \mathbf{w}_i \rangle$ as a real-valued token coordinate that is linear in the vertex position and smooth in the sampling angle, so the representation is differentiable by construction. The multiplicities $\Delta\chi(v, \mathbf{w}_i)$ are integer-valued and piecewise constant in the vertex positions, so they carry no gradient and are non-smooth only on a measure-zero set of tied projections.

To learn directly from the continuous representation, we process a sequence of tokens for each direction with a shared transformer encoder. For a given (cell complex, direction) pair, the input is a sequence of 2-dimensional tokens $\{(\langle v, \mathbf{w}_i \rangle, \Delta\chi(v, \mathbf{w}_i))\}_v$. This tokenization is also shown in Figure 1. A linear input projection maps each token to a d_{model} -dimensional embedding, and a transformer encoder exchanges information across tokens within the same direction. Max-pooling over the token axis aggregates the variable-length sequence into a fixed-size vector per (cell complex, direction) pair, and a final linear projection produces a d_{out} -dimensional output. The result is a $D \times d_{\text{out}}$ -dimensional matrix that takes the place of the discretized ECT matrix as input to any of the neural representation architectures described in Section 3. Within a single cell complex, all D directions are batched together for the transformer; cell complexes are processed sequentially to avoid padding to a common token count across the batch.

The continuous encoder removes a tuned resolution hyperparameter and the sigmoid approximation it requires, replacing the discretized ECT with a $D \times d_{\text{out}}$ feature matrix derived directly from the $O(N)$ vertices of the complex. It substitutes for the discretized ECT, including in every representation architecture studied next.

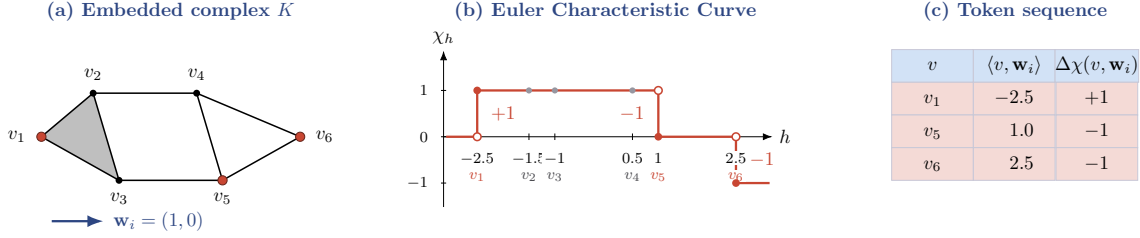


Figure 1: (a) Embedded simplicial complex K , consisting of 6 vertices $v_1, \dots, v_6$, 8 edges and 1 face. (b) Euler Characteristic Curve of K in the direction $\mathbf{w}_i$. (c) Token sequence $\{(\langle v, \mathbf{w}_i \rangle, \Delta\chi(v, \mathbf{w}_i))\}_v$ after masking vertices without change.

3. ECT Representations

The ECT has several notable equivariance properties. It is inherently equivariant with respect to permutation of the input nodes or points. It can be made translation-equivariant through centering the shape during preprocessing (Crawford et al., 2020; Munch, 2025). And it is equivariant under rotations of the embedded shape: rotating the input by an angle θ corresponds to a circular shift of the sampling directions by the same angle. In practice when computing the ECC at D regularly spaced directions, rotational equivariance holds for rotations by multiples of $2\pi/D$. The preservation of these equivariances depends on the neural network architecture.

We propose and evaluate six distinct neural network architectures for processing ECC encodings to generate ECT representations. The encoded ECT of a shape embedded in $\mathbb{R}^2$ can be represented as a matrix $\mathbf{E} \in \mathbb{R}^{D \times H}$, where D is the number of sampling directions and H is the number of height thresholds or the transformer output dimension. Each row corresponds to an ECC computed along a specific direction $\mathbf{w}_i$. All proposed architectures start with a continuous ECT layer followed by either discretization or an ECC transformer, producing an encoded ECT matrix. Their representation architecture then transforms the ECT to a (possibly equivariant) vector representation $\mathbf{z}$. And finally, a fully connected classification head is used to predict class probabilities. See also Figure 2.

The ECT is computed via a dedicated layer that processes embedded simplicial or cubical complexes. For fixed angles, the directions are regularly distributed according to $\theta_i = 2\pi i/D$ for $i \in \{0, 1, \dots, D-1\}$. This regular distribution is essential for maintaining rotational equivariance, as discussed below. Alternatively, the direction corresponding to the angles can be optimized jointly with other network parameters to allow the network to learn task-specific directional sampling patterns, at the cost of breaking rotational equivariance.

Feedforward network. The feedforward architecture takes the simplest approach by flattening the entire ECT matrix into a single vector and processing it through fully-connected layers. This baseline architecture makes no assumptions about the structure of the ECT and treats it as an arbitrary collection of features. Formally, the neural representation $\mathbf{z}$, from which we compute the output y is just $\mathbf{z} = \phi(\text{ECT})$ and $y = \psi(\mathbf{z})$, where the encoder ϕ extracts representation $\mathbf{z}$ from the ECT and the decoder ψ produces the final output y .

This architecture thus serves as a baseline that must learn rotation invariance purely from data rather than benefiting from architectural inductive biases.

Deep Set. Here we treat the ECT as an unordered set of ECCs, exploiting the permutation invariance property of set functions. Following the framework of [Zaheer et al. \(2017\)](#), the architecture consists of three components: a *feature extraction network* ϕ applied independently to each ECC, a permutation-invariant *aggregation function* $\oplus$, and a *decoder network* ψ . For data in $\mathbb{R}^2$ with fixed, regularly distributed angles this architecture is only rotationally equivariant if the angle information is excluded as input features to ϕ . However, excluding the angles as feature information is lossy, because there is no way to keep track of which angles are close to one another in the Deep Set architecture. In early experiments, this architecture did not learn well without angle information and we have therefore included angles in all Deep Sets experiments reported.

2D convolution. The ECT matrix can be directly viewed as a 2D image, where one dimension corresponds to sampling directions and the other to height thresholds. This natural interpretation allows the application of standard 2D convolutions, with the network learning hierarchical spatial features through successive convolutional layers. The key to achieving rotational equivariance in this architecture lies in the careful implementation of padding. Specifically, rotational equivariance requires *circular padding* along the directional dimension. When the input shape rotates by angle α , the ECT matrix undergoes a corresponding circular shift along the direction axis. With circular padding, convolutions commute with these circular shifts, thereby preserving rotational equivariance. Note that the 2D convolution also has approximate equivariance to translations of the height thresholds, which is not a reasonable inductive bias, but we include the architecture because it has been suggested before ([McGuire, 2024](#); [Röell and Rieck, 2024](#)).

1D convolution. Instead, we apply 1D convolutions along the directional dimension. This approach can be understood as treating each row of the ECT matrix as a separate signal channel, with the convolutions capturing patterns in how the Euler characteristic varies across different sampling directions. As with the 2D convolutional case, rotational equivariance in the 1D architecture requires *circular padding* along the directional dimension.

1D complex convolution. This architecture takes a fundamentally different approach by representing ECCs in the complex domain, where angular information is naturally encoded in the phase. This is inspired by rotationally equivariant learning from contours ([Gardaa and Blaser, 2025](#)). Each encoded ECC at height h is first converted to a complex representation by multiplying it with a phase factor determined by the sampling direction and then applying 1D complex convolution where all convolutions are in the complex domain and the activation function preserves the phase structure while applying nonlinearity to the magnitude. Complex convolutions in this form are inherently equivariant to rotations ([Gardaa and Blaser, 2025](#)).

Hybrid. Combining the benefits of 1D convolutions with the flexibility of fully-connected layers, this architecture begins with 1D convolutional layers with circular padding. After these convolutional layers, the output is flattened and processed by a fully-connected layer with ReLU activation. This design allows the network to first extract rotationally equivariant

features through convolutions before using the greater flexibility of dense connections for final feature aggregation.

The six architectures differ in their inductive biases about which structure of the ECT is informative: the feedforward network assumes none, the Deep Set assumes permutation symmetry of directions, the 2D/1D convolutions assume circular ordering of directions, and the complex convolution and hybrid assume rotational equivariance specifically. The next section measures which inductive biases pay off across point clouds, graphs, cubical complexes, and meshes.

4. Experiments

The aim of our subsequent experiments is to analyze the ramifications of different ECC encoding and ECT representation strategies on downstream performance.

4.1. Experimental Setup

We implemented three ECC encoding strategies (discrete, discrete with a transformer applied to the discretized ECC and continuous) and all six ECT representation architectures described above and evaluated them on a range of classification tasks (Figure 2). In the discrete case we used $H = 32$ height thresholds, and in the continuous case the output dimension $d_{\text{out}} = 32$ matches the resolution parameter used by the discretized ECT, ensuring identical downstream architectures for both variants. The ECT was computed using $D = 64$ fixed directions, resulting in a 64×32 matrix representation for each shape. All models were trained using the Adam optimizer with standard PyTorch default hyperparameters ($\beta_1 = 0.9, \beta_2 = 0.999$), and cross-entropy loss for classification. For all experiments, we used a learning rate of 0.0001 and whenever possible batch size of 128. Due to memory constraints, we reduced the batch size to 16 for the 3D mesh data. The main experiments are reported with fixed angles, with a separate experiment about learned angles for the point cloud dataset. Specific architectures are reported in the appendix. Our goal is to compare different neural representations of the ECT and do not include any architectures that bypass the ECT in our experiments.

All architectures produce a 64-dimensional representation $\mathbf{z}$ that feeds into a common lightweight classification head. This head consists of two fully-connected layers: the first reduces from 64 to 16 dimensions with ReLU activation, and the second maps from 16 dimensions to the number of output classes with softmax activation.

We evaluate on four data types. Three are embedded in $\mathbb{R}^2$: point clouds and cubical complexes derived from Fashion-MNIST (Xiao et al., 2017), and the letter-low, letter-med, and letter-high graph datasets from TUDataset (Morris et al., 2020), which carry 2-dimensional vertex coordinates. The fourth is a triangle-mesh dataset in $\mathbb{R}^3$, SwissBuildings (Federal Office of Topography swisstopo, 2025), a building-type classification task. For the 2D datasets, we sample $D = 64$ directions equally spaced on the circle; for the 3D meshes, we sample them on a cylinder grid that preserves the circular structure along the azimuthal axis, so the convolutional architectures still apply circular padding there. See Section B for more details on the SwissBuildings dataset. All datasets were split into training and

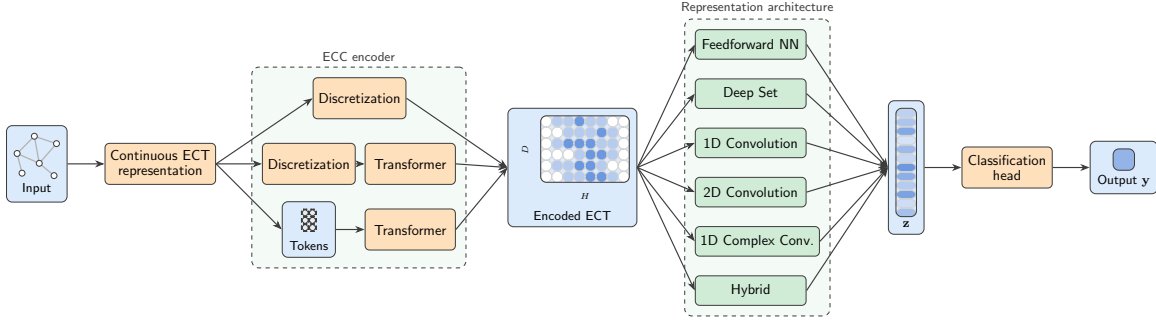


Figure 2: Overview of the pipeline. The continuous ECT representation feeds three interchangeable paths inside the ECC encoder: discretization onto a fixed grid (top), discretization followed by a transformer (middle), or tokenization followed by a transformer (bottom). All three produce a $D \times H$ matrix that drives the same representation architecture, classification head, and output. Blue rectangles denote data; orange rectangles denote fixed layers; green rectangles denote choices.

test data and for each of the 5 training runs, the training data was split into training and validation data. Because of their respective sizes, we trained the point cloud and mesh datasets for 20 epochs, whereas we trained the graph datasets for 100 epochs. We report the test accuracy for the epoch that achieved the best validation accuracy. All code used to produce our results is available on <https://github.com/blasern/ect-representation>.

4.2. Experimental Results

Table 1 presents the *improvements* in accuracy of continuous ECC encoding compared to the discrete encoding. To distinguish the contributions of the tokenization and the transformer architecture, we also show the discrete version with additional transformer encoding. For classification accuracies for all encoders and representations across all benchmark datasets, see Table S.3 and Figure S.1. The datasets span a wide difficulty range. Letter-low is nearly saturated, with all architectures exceeding 0.9 accuracy. Note also that the Fashion-MNIST-derived datasets (FashionMNISTPointCloud and FashionMNISTCubical) and Letters datasets are not rotationally symmetric: the source images have a canonical upright orientation, so rotational equivariance is not a useful inductive bias for these tasks, which partly explains the strong performance of the feedforward network there.

The continuous encoding improves accuracy on all six datasets. The gains are largest on SwissBuildings (+0.095) and FashionMNISTPointCloud (+0.077), with smaller improvements on Letter-high (+0.059) and FashionMNISTCubical (+0.059). Adding a transformer on top of the discrete encoding produces no consistent gains, but for some of the representations, the transformer can have an effect (SwissBuildings accuracy in the feedforward architecture moves from 0.505 to 0.606). However, for the best architectures for each dataset, the benefit of the continuous representation comes from the tokenization itself rather than from the added transformer capacity.

Dataset	Discrete	Continuous	Discrete Transformer
FashionMNISTPointCloud	0.702	0.778 (+0.077)	0.717 (+0.016)
Letter-low	0.936	0.959 (+0.023)	0.968 (+0.032)
Letter-med	0.828	0.843 (+0.015)	0.820 (-0.008)
Letter-high	0.751	0.810 (+0.059)	0.675 (-0.076)
SwissBuildings	0.717	0.812 (+0.095)	0.680 (-0.037)
FashionMNISTCubical	0.778	0.837 (+0.059)	0.768 (-0.010)

Table 1: For each dataset, the accuracy and in parenthesis the accuracy difference between the best discrete model and (i) the best continuous model and (ii) the best discrete-transformer model. Positive signs indicate that the presented versions improved accuracy, while negative signs indicate that the discrete version performed better.

Overall, the influence of the ECT representation is smaller and less clear than the influence of the ECC encoder, but the two interact. With continuous encoding, the feedforward network is the strongest performer on four of the six datasets, edged out only on Letter-low by the hybrid 1D convolution and on SwissBuildings by the 1D complex convolution. With discrete encoding the feedforward network is unstable—standard errors above 0.2 on FashionMNISTPointCloud and a collapse to an accuracy of 0.505 on SwissBuildings—and the hybrid 1D convolution becomes the most reliable performer. Continuous tokenization narrows this gap by giving the feedforward network a more learnable input; the convolutional architectures, with their built-in directional structure, deliver more stable training across all three encoders. Figure S.2 compares sample efficiency on FashionMNISTPointCloud across architectures and encoders. The dominant effect is the positive interaction between the continuous encoder and the feedforward architecture. Only the hybrid architecture with any encoder, comes close to the feedforward architecture with the continuous encoder. The Deep Set and convolution architectures lag behind the top two performers throughout, regardless of encoder.

5. Conclusion and Discussion

We introduced a continuous tokenization of the ECT that records the net Euler-characteristic change attributed to each vertex along each direction, eliminating the height-resolution hyperparameter of the discretized ECT. Across point clouds, graphs, cubical complexes, and meshes, this representation improves accuracy on all of the six primary benchmarks, with gains up to +0.095 on SwissBuildings. The improvement is attributable to the tokenization itself rather than to added model capacity: applying a transformer on top of the discretized ECT does not give consistent gains. Among the representation architectures, the feedforward network is the strongest performer when paired with the continuous encoder for most datasets.

A natural direction for future work is extending the ECT representation framework to 3D shapes that are equivariant to general 3D rotations. The architectures presented here exploit the circular structure of the 2D rotations via circular padding and complex phase encoding.

In 3D, the ECT instead integrates over directions on the sphere S^2 rather than S^1 , requiring $SO(3)$ -equivariance rather than $SO(2)$ -equivariance. Several equivariant architectures have been developed for signals defined on the sphere (Ballerin et al., 2026; Esteves et al., 2020; Shen et al., 2020). Adapting the ECT representation architectures to the 3D setting will likely require borrowing from these approaches, for instance by treating the ECT as a scalar field on S^2 and applying spherical convolutions that commute with $SO(3)$ rotations. This is a non-trivial extension, as the efficient circular padding tricks used here have no direct spherical analogue and $SO(3)$ -equivariant architectures are computationally demanding. But such extensions to the 3-dimensional case would substantially broaden the applicability of learned ECT representations to problems in 3D shape analysis.

Acknowledgments

This work has received funding from the Swiss National Science Foundation (Grant 235713), the European Union (Grant 101126560) and the Swiss State Secretariat for Education, Research, and Innovation (SERI).

References

- Juan P Garcia Amboage, Ernst Röell, Patrick Schnider, and Bastian Rieck. LEAP: Local ECT-based learnable positional encodings for graphs. In *International Conference on Learning Representations*, 2026. URL <https://openreview.net/forum?id=8XFPhByERc>.
- Francesco Ballerin, Nello Blaser, and Erlend Grong. $SO(3)$ -equivariant neural networks for learning from scalar and vector fields on spheres, 2026. URL <https://arxiv.org/abs/2503.09456>.
- Lorin Crawford, Anthea Monod, Andrew X Chen, Sayan Mukherjee, and Raúl Rabadán. Predicting clinical outcomes in glioblastoma: an application of topological and functional data analysis. *Journal of the American Statistical Association*, 115(531):1139–1150, 2020.
- Justin Curry, Sayan Mukherjee, and Katharine Turner. How many directions determine a shape and other sufficiency results for two topological transforms. *Transactions of the American Mathematical Society, Series B*, 9(32):1006–1043, 2022.
- Carlos Esteves, Ameesh Makadia, and Kostas Daniilidis. Spin-weighted spherical CNNs. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 8614–8625. Curran Associates, Inc., 2020.
- Federal Office of Topography swisstopo. swissBUILDINGS3D 3.0 Beta. <https://www.swisstopo.admin.ch/en/landscape-model-swissbuildings3d-3-0-beta>, 2025. Accessed 29 April 2026.
- Odin Hoff Gardaa and Nello Blaser. Rotatouille: Rotation equivariant deep learning for contours. In *The Fourth Learning on Graphs Conference*, 2025. URL <https://openreview.net/forum?id=VbCLyz3uW7>.

- Robert Ghrist, Rachel Levanger, and Huy Mai. Persistent homology and euler integral transforms. *Journal of Applied and Computational Topology*, 2(1):55–60, 2018. ISSN 2367-1734. doi: 10.1007/s41468-018-0017-1. URL <https://doi.org/10.1007/s41468-018-0017-1>.
- Sarah McGuire. Classification of Euler Characteristic Transforms using Convolutional Neural Networks. Applied Algebraic Topology Network Talk, May 2024. URL <https://www.youtube.com/watch?v=hajjmIm6K8o>.
- Christopher Morris, Nils Morten Kriege, Franka Bause, Kristian Kersting, Petra Mutzel, and Marion Neumann. Tudataset: A collection of benchmark datasets for learning with graphs. In *ICML 2020 Workshop on Graph Representation Learning and Beyond*, 2020.
- Elizabeth Munch. An invitation to the Euler characteristic transform. *The American Mathematical Monthly*, 132(1):15–25, 2025.
- Kalyan Varma Nadimpalli, Amit Chattopadhyay, and Bastian Rieck. Euler characteristic transform based topological loss for reconstructing 3d images from single 2d slices. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 571–579, 2023.
- Ernst Röell and Bastian Rieck. Differentiable Euler characteristic transforms for shape classification. In *International Conference on Learning Representations*, 2024. URL <https://openreview.net/forum?id=M0632iPq3I>.
- Ernst Röell and Bastian Rieck. Point cloud synthesis using inner product transforms. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen, editors, *Advances in Neural Information Processing Systems*, volume 38, pages 14551–14583. Curran Associates, Inc., 2025.
- P. Schapira. Tomography of constructible functions. In *Applied algebra, algebraic algorithms and error-correcting codes (Paris, 1995)*, volume 948 of *Lecture Notes in Comput. Sci.*, pages 427–435. Springer, Berlin, 1995. ISBN 3-540-60114-7. URL https://doi.org/10.1007/3-540-60114-7_33.
- Wen Shen, Binbin Zhang, Shikun Huang, Zhihua Wei, and Quanshi Zhang. 3d-rotation-equivariant quaternion neural networks. In Andrea Vedaldi, Horst Bischof, Thomas Brox, and Jan-Michael Frahm, editors, *Computer Vision – ECCV 2020*, pages 531–547, Cham, 2020. Springer International Publishing.
- Victor Toscano-Duran, Florian Rottach, and Bastian Rieck. Molecular machine learning using Euler characteristic transforms. In Aurelio López Fernández, Alejandro Rodríguez-González, Raquel Leirós-Rodríguez, Christian Mata Miquel, and Víctor Manuel González Suárez, editors, *Artificial Intelligence in Biomedicine*, pages 391–405, Cham, Switzerland, 2026. Springer.
- 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.

- Julius von Rohrscheidt and Bastian Rieck. Diss-l-ECT: Dissecting graph data with local Euler characteristic transforms. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 61790–61809. PMLR, 2025.
- Han Xiao, Kashif Rasul, and Roland Vollgraf. Fashion-MNIST: a novel image dataset for benchmarking machine learning algorithms, 2017.
- Manzil Zaheer, Satwik Kottur, Siamak Ravanbakhsh, Barnabas Poczos, Russ R Salakhutdinov, and Alexander J Smola. Deep sets. *Advances in neural information processing systems*, 30, 2017.

Appendix A. Architecture Specifications

Below we list detailed descriptions of each architecture as used in the experiments. Table S.1 shows the number of parameters for the different representation models. The classification heads are excluded because their number of parameters depends on the number of classes in the different datasets. Note that the transformer architecture for the continuous ECT has 69216 parameters and roughly doubles the total number of parameters. Similarly, the transformer architecture for the discrete ECT has 69152 parameters.

Model	Parameters
Hybrid 1D convolution	85264
1D convolution	78272
2D convolution	81840
1D complex convolution	96256
Feedforward network	131136
Deep Set	78528

Table S.1: Number of parameters for the different models. Complex parameters are counted as 2 real parameters.

The encoded ECT is a $D \times H$ matrix ($D = 64$ directions, $H = 32$ heights or transformer output dimension). The feedforward and 2D convolutional architectures consume it in this directions-first layout. The Conv1D-based architectures (Deep Set, 1D conv, 1D complex conv, hybrid) transpose it to channel-first form so the convolution slides along the directional axis D , which becomes the length, while the height/feature axis becomes the channels; when angle information is added as an extra channel the channel count is $H + 1$.

A.1. Feedforward Network Architecture

The Feedforward network provides a simple baseline by flattening the entire $D \times H = 64 \times 32$ ECT matrix into a 2048-dimensional vector and processing it through a single fully-connected layer that reduces to 64 dimensions, followed by ReLU activation. This architecture makes no structural assumptions about the ECT and does not incorporate angle information, treating the representation purely as an unordered collection of features. As discussed, this flattening operation breaks rotational equivariance by imposing a fixed ordering on the directional dimension.

A.2. Deep Set Architecture

Our Deep Set implementation uses four pointwise Conv1D layers with kernel size 1, effectively implementing the per-curve feature extraction network ϕ through 1D convolutions applied independently to each direction. The architecture progressively transforms the $(H + 1)$ -dimensional input (which includes one angle feature) through hidden dimensions of 128, 256, 128, and finally to 64 features, with ReLU activations between each layer. The aggregation function $\oplus$ is implemented as max pooling over the directional dimension, which combines

the 64-dimensional representations from all $D = 64$ directions into a single 64-dimensional feature vector. This vector then feeds into the shared classification head.

A.3. 2D Convolutional Architecture

The 2D CNN processes the ECT as a two-channel image with dimensions $64 \times 32 \times 2$ (directions $\times$ heights $\times$ channels). The first channel contains the ECT values themselves, while the second channel contains angle differences broadcast across all height thresholds. The architecture consists of four convolutional layers with progressively increasing channel counts: the first layer transforms the 2-channel input to 16 features using a 3×3 kernel, followed by layers that increase the channel count to 32, maintain it at 32, and finally expand to 64 features. The kernel sizes also increase for the deeper layers, with the third and fourth layers using 5×5 kernels to capture larger spatial patterns. Each convolution is followed by ReLU activation, and after the final layer, global max pooling aggregates information across both spatial dimensions to produce a 64-dimensional feature vector for classification. An important implementation detail for maintaining rotational equivariance is the custom padding scheme, using circular padding on the directional dimension and standard zero padding on the height dimension.

A.4. 1D Convolutional Architecture

The 1D CNN applies convolutions along the directional dimension, processing a transposed $(H + 1) \times D = 33 \times 64$ -dimensional input. The architecture employs four 1D convolutional layers with progressively increasing kernel sizes to capture patterns at multiple scales. Starting from the $(H + 1)$ -dimensional input (which includes angle information), the first layer with kernel size 1 expands to 128 channels, followed by a second layer with kernel size 3 that reduces to 64 channels. The third and fourth layers maintain 64 channels while using larger kernels of size 5 and 7 respectively, allowing the network to integrate information across increasingly large directional neighborhoods. All convolutions use circular padding to enable rotational equivariance, and ReLU activations follow each layer. After the final convolution, max pooling over the directional dimension produces a 64-dimensional feature representation.

A.5. 1D Complex Convolutional Architecture

The complex CNN operates entirely in the complex domain. The architecture begins by converting the real-valued ECT to a complex polar representation where each value is multiplied by $e^{i\theta_i}$ determining the phase. This creates a complex-valued $H \times D = 32 \times 64$ -dimensional input that naturally encodes both the ECT magnitudes and the directional angles. Four complex convolutional layers process this representation with circular padding. The first layer uses kernel size 1 to expand from $H = 32$ input channels to 32 complex feature channels, while the second layer with kernel size 3 increases this to 64 channels. The third and fourth layers maintain 64 channels using kernel size 5, allowing the network to capture increasingly complex patterns in the complex-valued features. Between layers, we apply the ModTanh activation function, which extends the standard tanh nonlinearity to complex numbers by computing $\text{ModTanh}(z) = \tanh(|z|) \cdot z/(|z| + \epsilon)$ with $\epsilon = 10^{-6}$. This activation preserves the phase structure of the complex representations while applying nonlinearity

to the magnitudes. After the final complex convolution, we convert back to real values by computing the magnitude $|z|$ of each complex feature. Max pooling over the directional dimension then aggregates these magnitude features to produce the final 64-dimensional representation.

A.6. Hybrid Architecture

The hybrid begins with three 1D convolutional layers with circular padding and progressively increasing kernel sizes (1, 3, and 5), transforming the $(H + 1) \times D$ input through channel dimensions of 64, 64, and finally 16. After these convolutional layers, the output is flattened to a $D \times 16 = 1024$ dimensional vector, which is then processed by a fully-connected layer that reduces to 64 dimensions with ReLU activation.

A.7. Equivariance

All the stated equivariance properties have been experimentally confirmed. Note that when angles are learnable parameters rather than fixed and regularly distributed, the rotational equivariance of all architectures is fundamentally broken. This is of course desirable, because it only makes sense to learn angles if some angles contain more relevant signal than others, in which case the task is not rotationally equivariant. In these cases, we can incorporate angles into the architectures dependent on the model architecture. The feedforward network does not incorporate angle information at all because that would be the same for all datapoints and the Deep Set architecture always concatenates the angle as an additional feature to each ECC. The 1D, hybrid, and 2D convolutional architectures add the difference between adjacent angles as an additional channel. The 1D complex convolution architecture always includes angle information.

A.8. Directional Transformer Encoder for Continuous and Discrete ECC

The directional transformer was configured with the hyperparameters listed in Table S.2.

Hyperparameter	Value
Token dimension (input)	2
Model width d_{model}	64
Attention heads	4
Encoder layers	2
Feedforward width	128
Output dimension d_{out}	32

Table S.2: Hyperparameters of the directional transformer encoder.

Appendix B. Dataset Construction

Fashion-MNIST point cloud. To create the point cloud data, each image was converted by treating pixel intensities as a probability distribution and sampling 200 planar coordinates

Encoder	Representation	FashionMNIST- PointCloud	Letter- low	Letter- med	Letter- high	SwissBuildings	FashionMNIST- Cubical
Continuous	1D complex conv.	0.683 (0.015)	0.884 (0.011)	0.616 (0.065)	0.460 (0.074)	0.812 (0.009)	0.756 (0.012)
	1D conv.	0.669 (0.007)	0.898 (0.022)	0.692 (0.019)	0.478 (0.037)	0.803 (0.016)	0.778 (0.006)
	2D conv.	0.633 (0.018)	0.916 (0.019)	0.668 (0.046)	0.488 (0.053)	0.799 (0.013)	0.768 (0.011)
	Deep set	0.710 (0.011)	0.902 (0.022)	0.734 (0.039)	0.517 (0.062)	0.792 (0.009)	0.771 (0.009)
	Feedforward	0.778 (0.004)	0.958 (0.009)	0.843 (0.030)	0.810 (0.012)	0.805 (0.005)	0.837 (0.004)
	Hybrid 1D conv.	0.746 (0.005)	0.959 (0.013)	0.823 (0.026)	0.687 (0.038)	0.777 (0.016)	0.808 (0.007)
Discrete	1D complex conv.	0.634 (0.013)	0.915 (0.018)	0.650 (0.024)	0.474 (0.051)	0.717 (0.019)	0.702 (0.005)
	1D conv.	0.611 (0.022)	0.871 (0.039)	0.638 (0.021)	0.460 (0.030)	0.704 (0.007)	0.678 (0.017)
	2D conv.	0.518 (0.234)	0.828 (0.069)	0.476 (0.027)	0.343 (0.029)	0.686 (0.016)	0.692 (0.013)
	Deep set	0.672 (0.005)	0.867 (0.024)	0.716 (0.014)	0.524 (0.034)	0.684 (0.009)	0.697 (0.010)
	Feedforward	0.311 (0.211)	0.936 (0.009)	0.788 (0.023)	0.724 (0.013)	0.505 (0.116)	0.775 (0.002)
	Hybrid 1D conv.	0.702 (0.009)	0.926 (0.011)	0.828 (0.022)	0.751 (0.040)	0.654 (0.007)	0.778 (0.006)
Discrete	1D complex conv.	0.524 (0.020)	0.821 (0.034)	0.474 (0.067)	0.330 (0.029)	0.665 (0.016)	0.612 (0.015)
Transformer	1D conv.	0.401 (0.275)	0.837 (0.021)	0.558 (0.049)	0.386 (0.032)	0.677 (0.012)	0.641 (0.010)
	2D conv.	0.396 (0.272)	0.831 (0.006)	0.521 (0.030)	0.373 (0.026)	0.668 (0.017)	0.616 (0.009)
	Deep set	0.411 (0.284)	0.860 (0.016)	0.612 (0.031)	0.387 (0.023)	0.680 (0.010)	0.658 (0.009)
	Feedforward	0.343 (0.332)	0.968 (0.009)	0.820 (0.018)	0.675 (0.022)	0.606 (0.040)	0.768 (0.006)
	Hybrid 1D conv.	0.717 (0.005)	0.938 (0.019)	0.800 (0.025)	0.599 (0.046)	0.616 (0.017)	0.746 (0.008)

Table S.3: Test accuracies for the continuous, discrete, and discrete transformed ECT. Standard errors are computed from 5 repeated runs. Best results per encoder are colored, best overall results are bold. Multiple results are highlighted if within one standard error of the maximal value.

via multinomial sampling with replacement. The integer coordinates were then perturbed with uniform noise in the range $[-0.5, 0.5]$ to produce continuous-valued point clouds.

Fashion-MNIST cubical complex. Each grayscale image was binarized at a fixed intensity threshold 0.3, and the resulting foreground pixels were used to construct a 2D cubical complex with vertices, edges between horizontally or vertically adjacent pixels, and square 2-cells for each filled 2×2 block.

SwissBuildings. Derived from the `swissBUILDINGS3D 3.0 Beta` release, this building-classification dataset has labels indicating building type. We restricted the dataset to include the 10 classes with more than 1000 buildings. To avoid geographic leakage between training and test data, we split by geographic tile rather than by individual building. The class distribution is heavily imbalanced, so we apply per-epoch class-balanced subsampling: at the start of every epoch, each class is capped at the size of the smallest class by drawing a random subsample of indices. For this dataset we sampled directions on a cylinder grid: n_ϕ azimuthal angles equally spaced on $[0, 2\pi]$ paired with n_z elevations equally spaced on $[-1, 1]$, and then project the resulting $(n_\phi \cdot n_z)$ points onto the sphere by normalization. This preserves the circular structure along the ϕ axis, so the convolutional architectures still apply circular padding there. We use $D = n_\phi \cdot n_z = 64$ with $n_z = 4$.

Appendix C. Complete results

Table S.3 and Figure S.1 show the complete results of test set accuracies for all tested models and across all datasets. Figure S.2 shows how test accuracy on the FashionMNIST-PointCloud data increases with training data size.



Figure S.1: Test accuracies for the continuous, discrete, and discrete transformed ECT for all datasets. Standard errors (in parenthesis) are computed from 5 repeated runs.

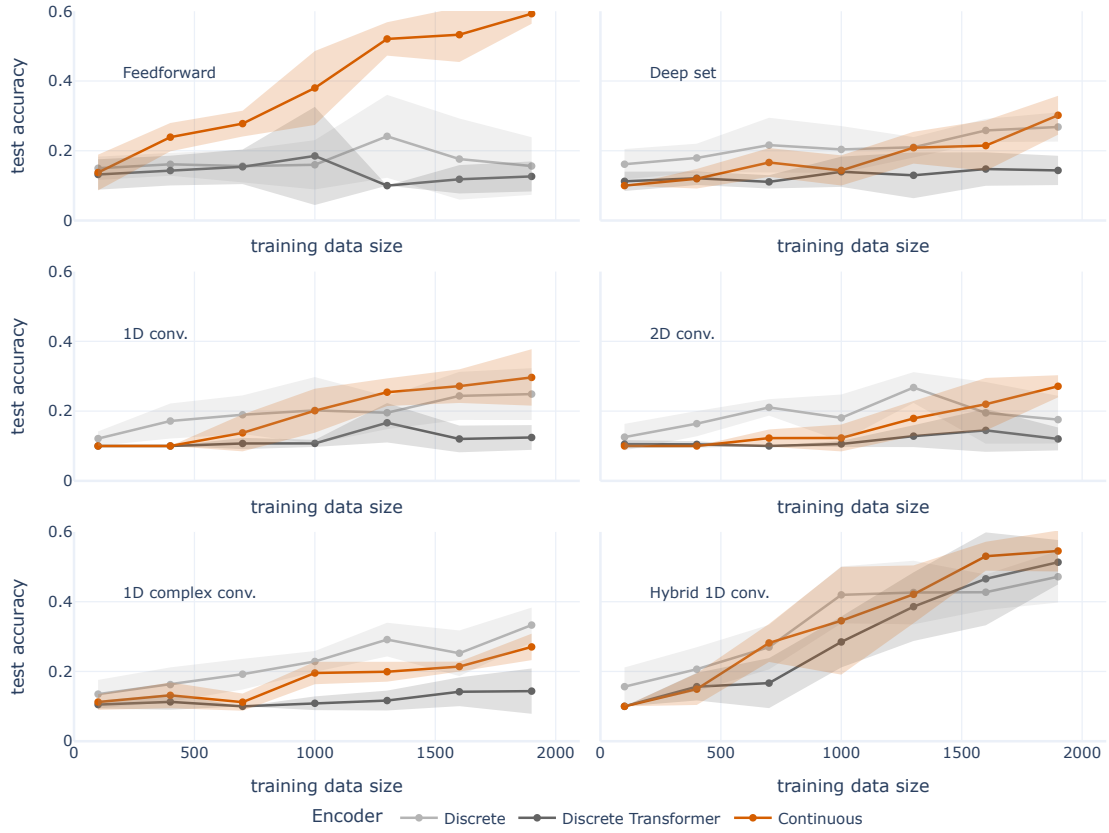


Figure S.2: Test accuracy on FashionMNISTPointCloud for different models for increasing training data sizes. Error bands are standard errors computed from 5 repeated runs.

