

Non-Euclidean Self-Organizing Maps

Dorota Celińska-Kopczyńska, Eryk Kopczyński

Institute of Informatics, University of Warsaw

{dot,erykk}@mimuw.edu.pl

Abstract

Self-Organizing Maps (SOMs, Kohonen networks) belong to neural network models of the unsupervised class. In this paper, we present the generalized setup for non-Euclidean SOMs. Most data analysts take it for granted to use some subregions of a flat space as their data model; however, by the assumption that the underlying geometry is non-Euclidean we obtain a new degree of freedom for the techniques that translate the similarities into spatial neighborhood relationships. We improve the traditional SOM algorithm by introducing topology-related extensions. Our proposition can be successfully applied to dimension reduction, clustering or finding similarities in big data (both hierarchical and non-hierarchical).

1 Introduction

Self-Organizing Maps (SOMs, also known as Kohonen networks) belong to neural network models of the unsupervised class allowing for dimension reduction in data without a significant loss of information. SOMs preserve the underlying topology of high-dimensional input and transform the information into one or two-dimensional layer of neurons. The projection is nonlinear, and in the display, the clustering of the data space and the metric-topological relations of the data items are visible [Kohonen, 1997]. In comparison to other techniques of reducing dimensionality, SOMs have many advantages. They do not impose any assumptions regarding the distributions of the variables and do not require independence among variables. They allow for solving nonlinear problems; their applications are numerous, e.g., in pattern recognition (see, e.g., [Grossberg and Carpenter, 1991]), brain studies [Bezdek *et al.*, 1993; Reddick *et al.*, 1997; Pérez-Aguila, 2013] or biological modeling [Mazzatorta *et al.*, 2003; Boniecki *et al.*, 2012]. At the same time, they are relatively easy to implement and modify [Kohonen, 1997; Asan and Ercan, 2012].

A typical setup for SOM assumes usage of a region of Euclidean plane. On the other hand, non-Euclidean geometries are steadily gaining attention of the data scientists [Wasserman, 2018; Chazal and Michel, 2017]. In particular, hyperbolic geometry has been proven useful in data visualization

[Munzner, 1998] and the modeling of scale-free networks [Krioukov *et al.*, 2010; Papadopoulos *et al.*, 2012]. Such a usefulness comes from the exponential growth property of hyperbolic geometry, which makes it much more appropriate than Euclidean for modeling and visualizing hierarchical data. Since the idea of SOM is rooted in geometry, we can expect to gain new insights from non-Euclidean SOM setups. Surprisingly, there are nearly no attempts to do so. Even if there have been propositions to use hyperbolic geometry in SOMs [Ritter, 1999; Ontrup and Ritter, 2001], other possibilities of inclusion of non-Euclidean geometries and different topologies (e.g., spherical geometry, quotient spaces) have been neglected. There is also no research on characteristics of data that affect the quality of Self-Organizing Maps.

Against this background, our contributions in this paper can be summarized as follows:

- We are the first to present the generalized setup for non-Euclidean SOMs. Our proposition allows for usage of (so far neglected or absent) quotient spaces. In consequence, we get a more regular and visually appealing results that the previous setups.
- By using Goldberg-Coxeter construction, our proposition allows for easy scalability of the templates. It also makes spheres a worthy counterpart for analysis – we are no longer restricted to usage of platonic solids.
- To our best knowledge, we are the first to extend SOM setup by non-Euclidean aspects other than the shape of the template. We introduce geometry-related adjustments in the dispersion function. Moreover, we show that our proposition improves the results in comparison to traditional Gaussian dispersion.
- Our quantitative analysis proves that the shape of data matters for Self-Organizing Maps. We use measures of topology preservation from the literature.
- The results of non-Euclidean SOMs have interpretation. Usage of different geometries allows us to find and highlights various aspects in the data sets. E.g., spherical geometry allows for an easy examination of polarization, and hyperbolic geometry due to the exponential growth fosters finding similarities. This makes non-Euclidean SOMs suitable both as stand-alone technique, but also as an auxiliary one to include in other models.

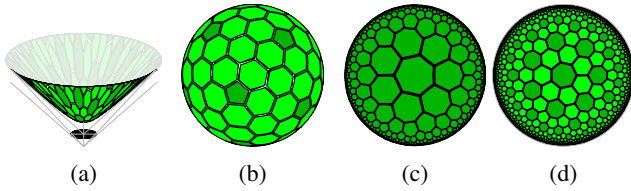


Figure 1: Representations of non-Euclidean geometries: (a) Minkowski hyperboloid; (b) Sphere. Hyperbolic tessellations in Poincaré disk model: (c) order-3 heptagonal tiling, (d) bitruncated order-3 heptagonal tiling.

2 Prerequisites

2.1 Non-Euclidean Geometries

Most data analysts take it for granted to use some subregions of a flat space as their data model, which means utilizing constructs which follow the principles of the Euclidean geometry. However, by the assumption that the underlying geometry is non-Euclidean, we obtain a new degree of freedom for the techniques of analysis which translate the similarities into spatial neighborhood relationships [Ritter, 1999]. Recall that formally, the Euclidean plane is the set of points $\{(x, y); x, y \in \mathbb{R}^2\}$, with the metric $d(a, b) = \|a - b\|$, where $\|(x, y)\| = \sqrt{x^2 + y^2}$. The sphere \mathbb{S}^2 is the set of points $\{(x, y, z); x, y, z \in \mathbb{R}^3, x^2 + y^2 + z^2 = 1\}$, with the metric $d(a, b) = 2\text{asin}(\|a - b\|/2)$, where $\|(x, y, z)\| = \sqrt{x^2 + y^2 + z^2}$. The Minkowski hyperboloid \mathbb{H}^2 is the set of points $\{(x, y, z); x, y, z \in \mathbb{R}^3, x^2 + y^2 + 1 = z^2, z > 0\}$, with the metric $d(a, b) = 2\text{asinh}(\|a - b\|/2)$, where we use the Minkowski metric $\|(x, y, z)\| = \sqrt{x^2 + y^2 - z^2}$.

All the geometries mentioned are characterized by constant curvature K : $K = 0$ in the case of Euclidean plane, while in \mathbb{H}^2 we have $K < 0$, and in \mathbb{S}^2 we have $K > 0$.

Figure 1 shows two tilings of \mathbb{H}^2 , the order-3 heptagonal tiling and its bitruncated variant, in the Poincaré disk model. In the Poincaré disk model, points of \mathbb{H}^2 are represented by points inside a disk. In the hyperbolic metric, all the triangles, heptagons and hexagons in each of the tessellations in Figure 1 are actually of the same size, and the points on the boundary of the disk are infinitely far from the center.

2.2 Tessellations of Non-Euclidean Spaces

Tessellations from Figure 1 can be interpreted as metric spaces, where the points are the tiles, and the distance $\delta(v, w)$ is the number of edges crossed to reach w from v . Such metric spaces have properties similar to the underlying surface.

Schläfli symbol. In a regular tessellation every face is a regular p -gon, and every degree has degree q (we assume $p, q \geq 3$). We say that such a tessellation has a Schläfli symbol $\{p, q\}$. Such a tessellation exists on the sphere if and only if $(p-2)(q-2) < 4$, plane if and only if $(p-2)(q-2) = 4$, and hyperbolic plane if and only if $(p-2)(q-2) > 4$.

Contrary to the Euclidean tessellations, we cannot scale hyperbolic or spherical tessellations. On a hyperbolic plane of curvature -1 , every face in a $\{q, p\}$ tessellation will have area $\pi(q\frac{p-2}{p} - 2)$. Thus, among hyperbolic tessellations of form $\{q, 3\}$, $\{7, 3\}$ is the finest, and they get coarser and coarser as

q increases. Regular spherical tessellations correspond to the platonic solids.

2.3 Self-Organizing Maps: General Idea

SOM network consists of two layers: the input layer containing the variables in the input data, and the output layer of the resulting clustering.

We describe every element in the input data D using k variables: $D \subseteq \mathbb{R}^k$. The elements of $x \in D$ are the values of the k variables which serve as the basis for clustering.

Neurons are traditionally arranged in a lattice. For each neuron i in the set of neurons we initialize the weight vector $w_i \in \mathbb{R}^k$. Weights are links that connect the input layer to the output layer. The final results may depend on the distribution of the initial weights [Asan and Ercan, 2012]. We assign the initial weights randomly. The neurons need to be exposed to a sufficient number of different inputs to ensure the quality of learning processes. In a usual setup, the formation of the SOM is controlled by three parameters: the learning rate η , the number of iterations t_{max} , and the initial neighborhood radius $\sigma(t_{max})$. Every iteration involves two stages.

Competition stage. We pick $x_t \in D$. The neurons compete to become activated. Only the node that is the most similar to the input data x_t will be activated and later adjust the values of weights in their neighborhood. For each neuron i in the set of neurons we compute the value of the scoring function $d(w_i, x_t) = \|w_i - x_t\|$. The neuron for which the value of the scoring function is the lowest becomes the winning neuron.

Adaptation. For a given input, the winning neuron and its neighbors adapt their weights. The adjustments enhance the responses to the same or to a similar input that occurs subsequently. This way the group of neurons specializes in attracting given pattern in input data. The input data x_t affects every other neuron j with the factor of $d_{\sigma(t)}(r) = \eta \exp(-r^2/2\sigma(t)^2)$, where r is the distance between the neuron j and the winning neuron i , and $\sigma(t)$ is the neighborhood radius in the iteration t ; we take $\sigma(t) = \sigma(t_{max})(1-t/t_{max})$ [Kohonen, 1997; Asan and Ercan, 2012; Ritter, 1999; Ontrup and Ritter, 2001].

This dispersion has a natural interpretation in the Euclidean geometry. Imagine the information as particles spreading between neurons according to the random walk model: each particle starts in neuron i , and in each of time steps, the information can randomly spread (with probability p) to one of the adjacent neurons. From the Central Limit Theorem we know that the distribution of particles after tC time steps approximates the normal distribution with variance proportional to t , which motivates using the function $d_{\sigma(t)}(r)$. Heat conduction is a well-known physical process which works according to similar rules, but where time and space are continuous.

3 Our Contribution

The core idea of the SOM algorithm is using a deformable template to translate data similarities into spatial relationships. The overwhelming majority of SOM applications use subregions of Euclidean space. Instead, we use non-Euclidean geometries to take advantage of their properties, such as the exponential growth of hyperbolic space. While

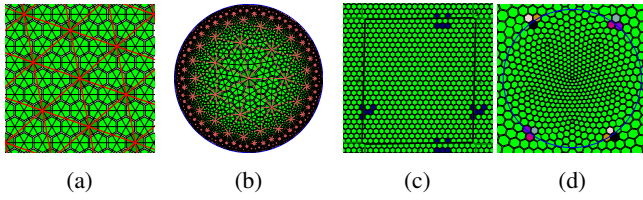


Figure 2: Goldberg-Coxeter construction: (a) Euclidean plane; (b) $GC_{2,1}\{7, 3\}$. Fundamental domains: (c) torus, (d) elliptic plane.

the basic idea has appeared in [Ritter, 1999; Ontrup and Ritter, 2001], we improve on it in the following ways.

3.1 Choice of the Tessellation

Continuous computations can be costly and prone to precision errors. Continuity is also not always essential. Usually, SOMs are based on the regular grids. Ritter [Ritter, 1999] argues that spherical tessellations are not useful in data analysis, because there are only five regular tessellations, namely platonic solids. Those solids are rather coarse and provide limited possibilities for manipulations of neighborhoods, even in comparison with the Euclidean surfaces. Similarly, regular hyperbolic tessellations such as $\{7, 3\}$ suffer because the exponential growth is too fast.

We combat these issues while losing only a bit of regularity by using the Goldberg-Coxeter construction. This construction adds additional hexagonal tiles. Consider the hexagonal grid $\{6, 3\}$ on the plane, and take an equilateral triangle X with one vertex in the point $(0, 0)$ and another vertex in the point obtained by moving a steps in a straight line, turning 60 degrees right, and moving b steps more. The tessellation $GC_{a,b}\{p, 3\}$ is obtained from the triangulation $\{3, p\}$ by replacing each of regular triangles with a copy of X . In Figure 2, brown lines depict the underlying regular triangulation. Regular tessellations are a special case where $a = 1, b = 0$. Figure 1b shows the result of applying the Goldberg-Coxeter construction to the sphere.

3.2 Using Closed Manifolds

The effects caused by the neurons on the boundary having less neighbors may make the maps less regular and less visually attractive. This problem does not appear on the sphere, which is a closed manifold. On the other hand, it is magnified in hyperbolic geometry, where the perimeter of a region is proportional to its area, causing a large fraction of the neurons to be affected by the boundary effects.

We combat these issues by using quotient spaces. A quotient space is obtained by identifying points in the manifold. For example, a square torus, a quotient space of the Euclidean plane, is obtained by identifying points (x, y) and (x', y') such that $x - x'$ and $y - y'$ are both integers. We call the original Euclidean plane the **covering space** of the torus. Intuitively, a quotient space is created by cutting a fragment of a surface and gluing its edges.

While the torus is usually presented in introductory topology books in its wrapped, donut-like shape, we present our quotient spaces in the covering space presentation, such as

in Fig. 2 (c)). We show the covering space of our manifold; our quotient space corresponds to the periodically repeating part of the picture. Such a presentation lets us present the whole manifold on a single picture, and is much more clean, especially for hyperbolic or non-orientable quotient spaces. Intuitively, the covering space presentation simulates how the manifold is perceived by the native beings (or neurons).

In the spherical geometry, we can identify each point with its antipodal point, obtaining the elliptic plane (Figure 2 (d)). The elliptic plane is non-orientable: a right-handed neuron would see a left-handed version of themselves on the other pole. Figure 2 (d) depicts the stereographic projection of the elliptic plane; the blue circle is the equator.

The sphere is a surface of genus 0, while the torus is a surface of genus 1; the genus of an orientable surface is, intuitively, the number of “holes” in it. Orientable quotient spaces of the hyperbolic plane have genus greater than 1, or equivalently, Euler characteristics $\chi = 2 - 2g < 0$. If we tile a surface with Euler characteristics χ with a pentagons, b hexagons and c heptagons in such a way that three polygons meet in every vertex, the following relationship will hold: $6\chi = a - c$. Thus, a sphere can be tiled with 12 pentagons (dodecahedron), a torus can be tiled with only hexagons, and hyperbolic quotient spaces can be tiled with only hexagons and -6χ heptagons. See the full version [Celińska-Kopczyńska and Kopczyński, 2021] for the details on our manifolds.

3.3 Dispersion

The natural interpretation of the dispersion function mentioned in the Prerequisites section no longer works in non-Euclidean geometry. In particular, the exponential nature of the hyperbolic plane makes the random walk process behave very differently in larger time frames (see, e.g., [Grigor’yan and Noguchi, 1998] for a study of heat conduction in the hyperbolic plane). For example, it is well known that the random walk on a two-dimensional Euclidean grid returns to the starting point (and any other point) with probability 1. In a two-dimensional hyperbolic grid, this probability decreases exponentially with distance. Interestingly, Ontrup and Ritter [Ritter, 1999; Ontrup and Ritter, 2001] who originally introduced non-Euclidean SOMs did not discuss this issue. In applications we may also use quotient spaces, which changes the situation even further – the information particle could reach the same neuron j in many different ways (e.g., by going left or by going right).

For that reason, we use a different dispersion function, based on numerically simulating the random walk process on our manifold. We compute the probability $P_{i,j,t}$ that the information particle starting in neuron i will end up in neuron j after t time steps. This probability can be computed with a dynamic programming algorithm: for $t = 0$ we have $P_{i,j,t} = 1$ if and only if $i = j$ and 0 otherwise; for $t + 1$ we have $P_{i,j,t+1} = P_{i,j,t} + p \sum_k P_{i,k,t} - P_{i,j,t}$, where we sum over all the neighbors k of the neuron j . See Algorithm 1 for the pseudocode which computes $P_{i,j,t}$. In this pseudocode, $N(i)$ denotes the neighborhood of the neuron i . This algorithm has time complexity $O(n^2T)$. Our application is based on highly symmetric surfaces, which lets us to reduce the time complexity by taking advantage of the symmetries.

Algorithm 1 Dispersion algorithm .

Parameter: the set of all nodes of a network V ; neighborhoods $N(i)$; the number of time steps T and precision p

Output: the dispersion array $P_{i,j,t}$ for $t = 0, \dots, T$

```

1: for  $i, j \in V$  do
2:    $P_{i,j,t} := 0$ 
3: end for
4: for  $i \in V$  do
5:    $P_{i,i,t} := 1$ 
6: end for
7: for  $t = 0, \dots, T - 1$  do
8:   for  $i, j \in V$  do
9:      $P_{i,j,t+1} := P_{i,j,t}$ 
10:  end for
11:  for  $k \in N(i)$  do
12:     $P_{i,j,t+1} := P_{i,j,t+1} + p \cdot (P_{i,k,t} - P_{i,j,t})$ 
13:  end for
14: end for
    
```

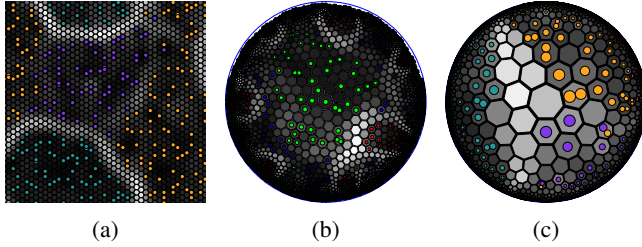


Figure 3: Example results of SOM on the iris flower (b) and palmer-penguins (ac) dataset. (a) a torus, (b) Klein quartic, (c) a disk in \mathbb{H}^2 .

In iteration t , the weights are updated for every neuron j according to the formula: $w_j := w_j + \frac{\eta P_{i,j,t}}{\max_t P_{i,j,t}} (x_t - w_j)$, where i is the winning neuron, and $f(t) = T(1 - \frac{t}{t_{max}})^2$.

4 Example Visualizations of our Results

To visualize the result of the proposed algorithm we will use the classic iris flower dataset by Fisher [Fisher, 1936] and the palmerpenguins dataset [Horst *et al.*, 2020]. Figure 3 depicts the visual example result of the SOM clustering. Coloring of the tiles allows for the examination of the clusters. We utilize a standard tool: inverted U-matrix (unified distance matrix) where the Euclidean distance between the representative vectors of neurons in the neighborhood is depicted in grayscale. The darker the shade, the less dissimilarity there is within the neighborhood. We may see the smooth and gradual changes.

While static visualizations work for Euclidean geometry, hyperbolic geometry is useful for visualization of hierarchical data, where we can focus by centering the visualization on any node in the hierarchy [Lamping *et al.*, 1995; Munzner, 1998]. Our visualization engine lets the viewer to smoothly change the central point of the projection to any point in the manifold, and thus clearly see the spatial relationships of every cluster.

The locations and the neighborhoods returned by SOMs have interpretation. Given that competition and adaptation

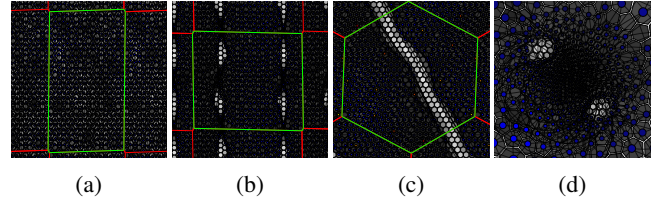


Figure 4: (a) square torus to rectangular torus; (b) rectangular torus to square torus; (c) sphere to hex torus; (d) hex torus to sphere

stages force the neighborhood to attract similar objects, the distance between the neurons becomes a measure for similarity: the further, the less similar objects are. We may use the resulting classification and mapping in further analyses.

5 Experiments

Our general experimental setup is as follows.

- We construct the original manifold O . Let T_O be the set of tiles and E_O be the set of edges between the tiles.
- We map all the tiles into the Euclidean space $m : T_O \rightarrow \mathbb{R}^d$, where d is the number of dimensions.
- We construct the target embedding manifold E . Let T_E be the set of tiles and E_E be the set of edges between the tiles.
- We apply our algorithm to the data given by m , This effectively yields an embedding $e : T_O \rightarrow E_O$.
- We measure the quality of the embedding.

To limit the effects of randomness (random initial weight of neurons, random ordering of data) we apply this process independently 100 times for every pair of manifolds.

In Figure 4 the effects of four runs are shown. Small gray polygons (hexagons) represent the tiles of E . The green and red polygons depict the fundamental domain. Every circle represents a tile from T_O that has been mapped to the manifold E , to the tile shown in the visualization. Edges between circles correspond to the original edges E_O between them.

In Figures 4a and 4b, both E and O are tori of different shapes. In the run shown in Figure 4a we obtain what we consider a successful map: the topology of the data is recovered correctly (despite the different shape of the two tori). Figure 4b shows an unsuccessful recovery of topology. In this case, the original torus O has been cut into two cylinders O_1 and O_2 , which are respectively mapped to cylinders E_1 and E_2 in E ; however, the two maps $O_1 \rightarrow E_1$ and $O_2 \rightarrow E_2$ are mirrored. This issue is clearly visible in our visualization: parts of the boundary areas between E_1 and E_2 contain no tiles, and the edges show singularities.

Figures 4c and 4d show a pair of mappings where E and O have different topologies (a sphere and a torus). Since the topologies are different, there is no way to map T_E to T_O without singularities. In Figure 4c our algorithm has stretched the sphere O on the poles, obtaining a cylinder; that cylinder is then mapped to a cylinder obtained by cutting the torus. The torus has been cut along the white area. Most edges are mapped nicely, to pairs of close cells, but some edges close

to the poles will have to go around the cylinder. Figure 4d is the inverse case. The torus is obtained by removing two disks at the poles, and gluing the boundaries of the removed disks. Edges which connect the boundaries of the two disks go across the whole sphere, while the remaining edges have the correct topological structure.

5.1 Embedding into \mathbb{R}^d

We need to embed the manifold O into \mathbb{R}^d in such a way that both its topology and geometry are preserved, that is, distances in \mathbb{R}^d are a good representation of actual geodesic distances in O . We use the following methods.

Natural. Well-known embeddings are known for the following cases:

- Sphere \mathbb{S}^2 has a well-known embedding to \mathbb{R}^3 .
- The square torus has an embedding to $\mathbb{R}^4 = \mathbb{R}^2 \times \mathbb{R}^2$, obtained by representing the torus as $\mathbb{T}^2 = \mathbb{S}^1 \times \mathbb{S}^1$ and mapping every circle \mathbb{S}^1 to \mathbb{R}^2 . For the rectangular torus, we use two circles of sizes corresponding to the length of edges. For the hexagonal torus, we use three circles, corresponding to the three axes.

Signpost. For closed hyperbolic manifolds we use the following method. We choose the subset of tiles t_1, \dots, t_d as *signposts*. Then, for every tile t , we set $m(t) = (\delta(t, t_1), \dots, \delta(t, t_d))$. In most cases we choose the signposts to be the tiles of degree other than 6.

Landscape. For hyperbolic and Euclidean disks, we use the following method. We find all the zig-zag lines in the tessellation. These zig-zag lines go along the edges and split the manifold into two parts. They are obtained by turning alternately to left and right at every vertex. Let L be the set of all lines. We assign a random Gaussian vector v_l to each straight line $l \in L$. The central tile t_0 has all coordinates equal to 0. For any other tiles t , we find the set L_t of all the straight lines separating t_0 and t , and set $m(t) = \sum_{l \in L_t} v_l$.

We call this *landscape method* because it is inspired by the method used to generate natural-looking landscapes in HyperRogue [Kopczyński *et al.*, 2017]. It represents the reason why hyperbolic geometry may appear in real-life data such as social network analysis: every line $l \in L$ represents a part of the social network becoming polarized or specialized.

5.2 Measuring the Quality of Final Embedding

Our final embeddings should correctly preserve the topology. Our primary method of measuring topology preservation is based on the ideas of Villmann *et al* [Villmann *et al.*, 1994]. This measure is based on the neuron weights w_t for every tile $t \in T_E$. For every tile $t \in T_E$, let p_t be the point in the manifold O which is the closest to w_t . Define the Voronoi cell V_t as the set of points in the manifold O which are closer to p_t than to any other $p_{t'}$, i.e., $V_t = \{x \in O : \forall t' \in T_E | x - p_t| \leq |x - p_{t'}|\}$. Two Voronoi cells V_t and $V_{t'}$ are adjacent if $V_t \cap V_{t'} \neq \emptyset$. This way, we obtain two graphs on the set of tiles T_E : the graph $G_E = (T_E, E_E)$ where two tiles are adjacent iff there is an edge in E between them, and the graph $G_V = (T_E, E_V)$ where two tiles t, t' are adjacent iff their Voronoi cells are adjacent.

For an embedding that ideally preserves the topology and also preserves the geometry well enough, we have $E_V = E_E$. In general, let $d_E(t_1, t_2)$ and $d_V(t_1, t_2)$ be the length of the shortest paths between tiles t_1 and t_2 in the graphs G_V and G_E . We define the *Villmann measure* of the embedding as $v = \max_{(t_1, t_2) \in E_E} d_V(t_1, t_2) + \max_{(t_1, t_2) \in E_V} d_E(t_1, t_2) - 2$. Ideally, we get $v = 0$; embeddings which stretch the manifold or have local folds yield small values of v ($2 \leq v \leq 4$), and embeddings which do not preserve the topology yield larger values. An embedding does not preserve the topology if one of two cases hold: the induced map from E to O is not continuous (making the first component of v large) or the induced map from O to E is not continuous (making the second component of v large) [Villmann *et al.*, 1994]. See the full version for other methods of measuring embedding quality.

5.3 Quantitative Results

We use the following parameters: $t_{max} = 30000$ iterations, learning coefficient $\eta = 0.1$, dispersion precision $p = 10^{-4}$, T is the number of dispersion steps until the max value/min value ≤ 1.6 , 60 landscape dimensions, manifolds with about 520 tiles. Computing 57600 embeddings takes 4 hours on 8-core Intel(R) Core(TM) i7-9700K CPU @ 3.60GHz. Our implementation is included in the RogueViz non-Euclidean geometry engine. The results of our experiments, code and visualizations are at https://figshare.com/articles/software/Non-Euclidean_Self-Organizing_Maps_code_and_data/_16624393.

Comparison of simulated and Gaussian dispersion. We use the aforementioned measures of quality to check if simulated dispersion improves the quality of the embedding in comparison to Gaussian. In the Gaussian dispersion function, we take the discrete distance between tiles as the distance between two neurons. We take advantage of the paired nature of the data and compute the differences between the values of the quality measure obtained with Gaussian and the simulated dispersion. We use Wilcoxon test to check if the difference is statistically insignificant, against the alternative hypothesis that the results from simulated dispersion are better. We use 1% significance level.

Our results show that the embeddings obtained with simulated dispersion have usually lower Villmann’s measure than the embeddings obtained with Gaussian dispersion. This remains true whether we use single or double density of the original manifold, or when we restrict only to closed/disk O , or closed/disk E . The p-value is 0.00 in all cases. The results are slightly different for other measures of embedding quality; see the full version for more details.

Quality of shape recovery. We have already shown that using simulated dispersion improves the quality of the embedding. However, SOM on even correctly chosen manifold may be prone to errors. Here, we will analyze the factors that affect the errors in embeddings.

OLS regression fails to account for the qualitative difference between low values of Villmann measure (only local geometric errors) and high values (discontinuities). Therefore for topological errors where correct embeddings are possible, we will use censored regression model (tobit model) [Tobin,

O	closed			disk			closed	disk
E	closed			disk			disk	closed
Effect type	$E(y x)$	$E(y x, y > 0)$	$P(y > 0 x)$	$E(y x)$	$E(y x, y > 0)$	$P(y > 0 x)$	$\Delta E(y)/\Delta x_k$	
(Intercept)	—	—	—	—	—	—	29.70 [†]	32.42 [†]
O hyperbolic	1.79 [†]	1.79 [†]	3.5e-07 [†]	-0.22	-0.18	-0.024	0.47 [†]	-2.57 [†]
O spherical	1.94 [†]	1.94 [†]	1.5e-07 [†]	—	—	—	0.81 [†]	—
same_manifold	-15.15 [†]	-14.70 [†]	-0.08 [†]	-5.90 [†]	-7.23 [†]	-0.65 [†]	—	—
same_geometry	-3.09 [†]	-3.09 [†]	-1.1e-06 [†]	-6.07 [†]	-4.52 [†]	-0.49 [†]	-3.16 [†]	-2.70 [†]
both_orientable	0.32 [†]	0.32 [†]	4.8e-08 [†]	—	—	—	1.20 [†]	-0.043
both_symmetric	-1.34 [†]	-1.34 [†]	-1.7e-07 [†]	—	—	—	-1.83 [†]	0.35 [†]
diff_curv_pos	-9.53 [†]	-9.53 [†]	0	4.84 [†]	3.93 [†]	0.53 [†]	3.34 [†]	-5.79 [†]
diff_curv_neg	-10.30 [†]	-10.30 [†]	0	6.73 [†]	5.47 [†]	0.74 [†]	-10.20 [†]	-12.81 [†]
diff_samples	0.017 [†]	0.017 [†]	0	—	—	—	0.025 [†]	0.013 [†]
same_topology	-9.02 [†]	-9.01 [†]	-0.00082 [†]	—	—	—	—	—
N	25600			4900			11200	11200
(pseudo) R^2	0.2099			0.1355			0.5882	0.5488
R^2_{adj}	—			—			0.5879	0.5485

Table 1: Factors affecting quality of SOM embedding – Villmann measure. Partial effects for OLS and marginal effects for tobit. [†], **, *denote significance at 1%, 5%, 10% level, accordingly. In all regressions, p-values for joint significance tests equaled 0.000.

1958; Greene, 2020]. To this end, we left censor values of Villmann measures lower than 8 to 0.

In the case of closed manifolds embedded on close manifolds, out of 25600 embeddings we got 2082 embeddings without topological errors (8.1%). If disks were embedded to disks, out of 4900 embeddings 2499 were free of topological errors (51%). For correctly chosen manifolds this percentage was significantly higher (94.3% for closed ones and 100% for disks). The probability of SOMs making topological errors ($P(y > 0|x)$) decreases if we correctly choose manifold, topology or at least geometry. Differences in curvature vastly increase both the probability of SOM yielding tears and the number of those errors (for whole sample $E(y|x)$ and conditionally if SOM erred ($E(y|x, y > 0)$) in the case of disks. Again, original hyperbolic closed manifolds are harder to recover in comparison to Euclidean manifolds; also, orientability increases the difficulty of the task. The results are stable for the double density of the original manifold (full version).

6 Discussion

Choosing the manifold. One of the major concerns regarding using non-Euclidean SOMs is the choice of the underlying manifold. Depending on what is the core interest of the researcher, the choice of the underlying manifold may vary. It is typical for multidimensional analysis techniques that the eventual choice of the setup can be subjective. To make sure the results are robust, one may conduct the simulations on distinct spaces. The Goldberg-Coxeter construction lets us use similar number of neurons for different manifolds, controlling for the number of possible groups. Later diagnostics may include comparison of information criteria.

Other parameters. The full version contains results of changing other parameters, such as the landscape dimension and the density of original manifolds.

7 Conclusions

In this paper, we provide the general setup for non-Euclidean SOMs. We utilize symmetric quotient spaces to make our maps uniform, the Goldberg-Coxeter construction to remove the limitations related to the number and size of available grids, and suggest using a dispersion function different than Gaussian to match the underlying geometry.

It is surprising to us that the idea of using non-Euclidean templates seems to have been neglected after the initial papers [Ritter, 1999; Ontrup and Ritter, 2001]. There is research on extending the SOM algorithm to the cases where the data D is no longer considered a subset of \mathbb{R}^k with Euclidean distances, but rather based the distances are based on dissimilarity matrices or Mercer kernels [Rossi, 2014; Martín-Merino and Muñoz, 2004]. While such data representations are sometimes referred to as non-Euclidean, they are not directly related to non-Euclidean geometry. Such approaches can be seen as orthogonal to ours: they run SOM on Euclidean lattices but change the representation of D , while in our approach, the data manifold is still embedded into \mathbb{R}^k , but we change the template. One possible direction of further research is to combine both approaches. However, contrary to our approach, the non-geometrical nature of these settings makes them less usable for visualization.

In this paper, we restricted ourselves to two-dimensional geometries. An exciting future direction is using three-dimensional visualizations, applying the recent advances in Virtual Reality and the visualization of Thurston geometries [Coulon *et al.*, 2020; Kopczyński and Celińska-Kopczyńska, 2022]. We believe that our approach can be adapted to such geometries, yielding insightful visualizations.

We are grateful to the referees for their insightful and constructive comments. This work has been supported by the National Science Centre, Poland, grant UMO-2019/35/B/ST6/04456.

References

- [Asan and Ercan, 2012] Umut Asan and Secil Ercan. An introduction to self-organizing maps. In C. Kahraman, editor, *Computational Intelligence Systems in Industrial Engineering: with Recent Theory and Applications*, pages 299–319. Atlantis Press, 2012.
- [Bezdek *et al.*, 1993] James C. Bezdek, Lawrence O. Hall, and Laurence P. Clarke. Review of mr image segmentation techniques using pattern recognition. *MEDICAL PHYSICS-LANCASTER PA-*, 20:1033–1033, 1993.
- [Boniecki *et al.*, 2012] Piotr Boniecki, Krzysztof Nowakowski, Robert Tomczak, Sebastian Kujawa, and Hanna Piekarska-Boniecka. The application of the Kohonen neural network in the nonparametric-quality-based classification of tomatoes. In *ICDIP 2012*, volume 8334, pages 440 – 444. SPIE, 2012.
- [Celińska-Kopczyńska and Kopczyński, 2021] Dorota Celińska-Kopczyńska and Eryk Kopczyński. Non-euclidean self-organizing maps, 2021. <https://arxiv.org/abs/2109.11769>.
- [Chazal and Michel, 2017] Frédéric Chazal and Bertrand Michel. An introduction to topological data analysis: fundamental and practical aspects for data scientists. Technical report, 2017.
- [Coulon *et al.*, 2020] Rémi Coulon, Elisabetta A. Matsuoto, Henry Segerman, and Steve J. Trettel. Ray-marching thurston geometries, 2020. arXiv 2010.15801. To appear in *Experimental Mathematics*.
- [Fisher, 1936] Ronald A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7(2):179–188, 1936.
- [Greene, 2020] William H. Greene. *Econometric Analysis*. Pearson Education Limited, 8 edition, 2020.
- [Grigor’yan and Noguchi, 1998] Alexander Grigor’yan and Masakazu Noguchi. The heat kernel on hyperbolic space. *Bulletin of the London Mathematical Society*, 30(6):643–650, 1998.
- [Grossberg and Carpenter, 1991] Stephen Grossberg and Gail A. Carpenter. *Pattern recognition by self-organizing neural networks / edited by Gail A. Carpenter and Stephen Grossberg*. MIT Press Cambridge, Mass, 1991.
- [Horst *et al.*, 2020] Allison Marie Horst, Alison Presmanes Hill, and Kristen B Gorman. *palmerpenguins: Palmer Archipelago (Antarctica) penguin data*, 2020.
- [Kohonen, 1997] Teuvo Kohonen, editor. *Self-organizing Maps*. Springer-Verlag, Berlin, Heidelberg, 1997.
- [Kopczyński *et al.*, 2017] Eryk Kopczyński, Dorota Celińska, and Marek Čtrnáct. HyperRogue: Playing with hyperbolic geometry. In *Procs of Bridges*, pages 9–16, Phoenix, Arizona, 2017. Tessellations Publishing.
- [Kopczyński and Celińska-Kopczyńska, 2022] Eryk Kopczyński and Dorota Celińska-Kopczyńska. Real-time visualization in anisotropic geometries. *Experimental Mathematics*, 0(0):1–20, 2022.
- [Krioukov *et al.*, 2010] Dmitri Krioukov, Fragkiskos Papadopoulos, Maksim Kitsak, Amin Vahdat, and Marián Boguñá. Hyperbolic geometry of complex networks. *Phys. Rev. E*, 82:036106, Sep 2010.
- [Lamping *et al.*, 1995] John Lamping, Ramana Rao, and Peter Pirolli. A focus+context technique based on hyperbolic geometry for visualizing large hierarchies. In *Procs of CHI ’95*, pages 401–408, New York, NY, USA, 1995. ACM Press/Addison-Wesley Publishing Co.
- [Martín-Merino and Muñoz, 2004] Manuel Martín-Merino and Alberto Muñoz. Extending the som algorithm to non-euclidean distances via the kernel trick. In *ICONIP 2004*, volume 3316, pages 150–157, 11 2004.
- [Mazzatorta *et al.*, 2003] Paolo Mazzatorta, Marjan Vracko, Aneta Jezierska, and Emilio Benfenati. Modeling toxicity by using supervised kohonen neural networks. *Journal of chemical information and computer sciences*, 43(2):485–492, 2003.
- [Munzner, 1998] Tamara Munzner. Exploring large graphs in 3d hyperbolic space. *IEEE Computer Graphics and Applications*, 18(4):18–23, 1998.
- [Ontrup and Ritter, 2001] Jörg Ontrup and Helge Ritter. Hyperbolic self-organizing maps for semantic navigation. In *Procs of NIPS’01*, pages 1417–1424, Cambridge, MA, USA, 2001. MIT Press.
- [Papadopoulos *et al.*, 2012] Fragkiskos Papadopoulos, Maksim Kitsak, M. Angeles Serrano, Marian Boguñá, and Dmitri Krioukov. Popularity versus Similarity in Growing Networks. *Nature*, 489:537–540, Sep 2012.
- [Pérez-Aguila, 2013] Ricardo Pérez-Aguila. Enhancing brain tissue segmentation and image classification via 1d kohonen networks and discrete compactness: An experimental study. *Engineering Letters*, 21(4), 2013.
- [Reddick *et al.*, 1997] Wilburn E. Reddick, John O. Glass, Edwin N. Cook, T. David Elkin, and Russell J. Deaton. Automated segmentation and classification of multispectral magnetic resonance images of brain using artificial neural networks. *IEEE Transactions on Medical Imaging*, 16(6):911–918, 1997.
- [Ritter, 1999] Helge Ritter. Self-organizing maps on non-euclidean spaces. In E. Oja and S. Kaski, editors, *Kohonen Maps*, pages 97–108. Elsevier, 1999.
- [Rossi, 2014] Fabrice Rossi. How many dissimilarity/kernel self organizing map variants do we need? In *WSOM*, 2014.
- [Tobin, 1958] James Tobin. Estimation of relationships for limited dependent variables. *Econometrica*, 26(1):24–36, 1958.
- [Villmann *et al.*, 1994] Thomas Villmann, Ralf Der, J. Herrmann, and Thomas Martinetz. Topology preservation in self-organizing feature maps: General definition and efficient measurement. pages 159–166, 01 1994.
- [Wasserman, 2018] Larry Wasserman. Topological data analysis. *Annual Review of Statistics and Its Application*, 5(1):501–532, 2018.