Permutation Entropy for Graph Signals

John Stewart Fabila-Carrasco, Chao Tan, *Senior Member, IEEE*, and Javier Escudero, *Senior Member, IEEE*

***Abstract*—Entropy metrics (for example, permutation entropy) are nonlinear measures of irregularity in time series (one-dimensional data). Some of these entropy metrics can be generalised to data on periodic structures such as a grid or lattice pattern (two-dimensional data) using its symmetry, thus enabling their application to images. However, these metrics have not been developed for signals sampled on irregular domains, defined by a graph. Here, we define for the first time an entropy metric to analyse signals measured over irregular graphs by generalising permutation entropy, a well-established nonlinear metric based on the comparison of neighbouring values within patterns in a time series. Our algorithm is based on comparing signal values on neighbouring nodes, using the adjacency matrix. We show that this generalisation preserves the properties of classical permutation for time series and the recent permutation entropy for images, and it can be applied to any graph structure with synthetic and real signals. We expect the present work to enable the extension of other nonlinear dynamic approaches to graph signals.**

***Index Terms*—Graph signal processing, Graph Laplacian, Permutation entropy, Adjacency matrix, Irregularity, Nonlinearity Dynamics, Topology, Entropy metric.**

I. Introduction

In THE analysis of time series, entropy is a common tool used to describe the probability distribution of the states of a system. Based on this concept, the seminal paper [1] introduced the so-called permutation entropy (PE) as a measure to quantify irregularity (or complexity) in time series, a fundamental challenge in data analysis. This entropy involves calculating permutation patterns, i.e., permutations defined by comparing neighbouring values of the time series. In the last years, PE has been applied in different fields as biomedicine [2], [3], physical systems [4] and economics [5]. Some variants, modifications and extensions of PE have been introduced, including: a multiscale step [6]; changes targeting signals with noise [7]; a variation for detecting heartbeat dynamic [8]; the inclusion of a nonlinear mapping to consider the differences between the amplitude values [9], [10]; considering time reversibility conditions [11], [12]; and extensions to higher dimensions [13].

Manuscript received October 1, 2021; revised March 7, 2022; accepted April 3, 2022. Date of publication April 13, 2022; date of current version April 27, 2022. The work of John Stewart Fabila-Carrasco and Javier Escudero was supported by Leverhulme Trust through a Research Project under Grant RPG-2020-158. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. David I Shuman. *(Corresponding author: John Stewart Fabila-Carrasco.)*

John Stewart Fabila-Carrasco and Javier Escudero are with the School of Engineering, Institute for Digital Communications, University of Edinburgh, West Mains Rd, Edinburgh EH9 3FB, U.K. (e-mail: math.john.stewart@gmail.com; javier.escudero@ed.ac.uk).

Chao Tan is with the School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China (e-mail: tanchao@tju.edu.cn).

Digital Object Identifier 10.1109/TSIPN.2022.3167333

A time series can be considered as a one-dimensional data vector (1D), while an image can be regarded as a two-dimensional regular data set (2D). In the field of image processing, several entropy algorithms have been proposed to quantify the irregularity of images as generalisations of their one-dimensional analogous. Examples include: 2D permutation entropy [13], 2D sample entropy [14], 2D dispersion entropy [15], and 2D distribution entropy [16]. Most of the methods are straightforwardly generalised to higher-dimensional periodic structures, for example, 3D dispersion entropy [17], 3D fuzzy entropy [18]. The generalisation comes from the fact that the underlying structure (the lattice graph or grid graph, for example for an image) is a periodic structure. Then, the algorithms [13]–[16] use the symmetry from the structure to compare the values of the signal. However, thus far, it is unclear how to generalise the two-dimensional methods to a general irregular domain (or graph).

The study of data defined on irregular graphs domains is the main interest of graph signal processing (GSP), an active research area in recent years [19], [20]. This is motivated by the fact that, new technological advances have enabled the recording of data from complex systems [21]. GSP is immediately useful in applications where measures are distributed on irregular domains. Examples include a network of weather stations, vehicular networks or power grids, among others [19]–[21]. In some cases, the signal domain is not a set of equidistant time points (time series) or a regular grid (image), and in some cases, the data is not related to space or time. Graphs can model such data and complex interactions, and these new relations may be included in the data processing techniques. Then, some conventional signal-processing operations can be extended to graphs, such as filtering in the spectral and vertex domain, interpolation, subsampling the data with regarding to the graph [20]–[22] and generating surrogate graph signals [23].

For a time series, the classical PE is computed based on the successive values of the time series or neighbouring values. These concepts are equivalents in 1D. However, for a signal on a graph, the concept of successive values is unclear, but we have the notion of neighbouring vertices. This concept is fundamental to generalise the permutation entropy for graphs signals ($\mathrm{PE_G}$). In particular, we will consider time series as a signal function on a 1D-graph (an undirected path) and an image as a signal function on a 2D-graph (a grid).

Of note, the concept of graph entropy has been defined in previous literature [24], [25]. However, this definition involves the computation of the spectrum of the Laplacian [26], its probability distribution and the Shannon entropy. Therefore, it measures the complexity/irregularity of the geometric structure

and topology of the graph, but not of the signals on the graph itself.

Thus, here we introduce a measure of the regularity of a signal over a graph, combining the signal values with the topology of the graph, thus extending entropy algorithms for time series and images to graphs.

A. Related Work

The computation of the classical PE requires four elements: 1) The input, most often a time series; 2) The choice of the parameters: the embedding dimension m and time delay L; 3) The computation of the permutation patterns; and 4) The computation of the entropy value. Most of the variations and improvements for the original PE method focus on one of the previous elements.

1) Input Permutation entropy was originally designed to analyse a univariate time series [1]. Beyond univariate time series, a multivariate permutation entropy was defined in [27]. Some variations of PE consider signals defined on regular domains, notably a 2D permutation entropy [13] and a 3D permutation entropy [17]. However, there are no PE extensions that consider signals defined on irregular input. This is where we make our main contribution. Our algorithm can be used for any signal defined on a regular (time series, image, etc.) or irregular domain (a graph).

2) Parameters Given a time series, calculating PE involves selecting parameters such as the embedding dimension m and time delays L. On this front, [28] proposed a way to choose the parameters automatically, [29] extended the scalar time delay to vector time delays, [30] presented non-uniform embeddings time delays, and [31] described a parameter optimisation strategy for multiscale PE.

3) Permutation patterns PE considers only the order of the values but not its amplitudes. In order to include information contained in the amplitude values, [32], [33] defined a weighted-permutation entropy, and [34] defined an amplitude-aware permutation entropy. These modifications affect how the patterns are defined.

4) Entropy computation Once the relative frequency for the distinct permutation patterns has been estimated, the original PE algorithm uses Shannon's definition to compute the entropy. Other entropy definitions have been used in the literature, for example, [35] presented a Rènyi-based permutation entropy.

All the improvements to classical PE presented above in 2), 3), and 4) are designed for univariate time series. Very recently, extensions have been introduced for regular 2D and 3D data [13], [17]. However, there is no previous work on how to extend entropy metrics (like PE or other related methods) to irregular domains. Our algorithm uses the same principles behind the classical PE, but we extend the input to any general graph. What is more, any variations of the original algorithm related to the choice of parameters, estimation of patterns, and calculation of entropy can be straightforwardly applied to our algorithm. However, this is beyond the scope of this paper, where we focus on presenting an PE_G and exploring the relationships of this metric with the signal and, crucially, the *graph*. Likewise, exploration of applications of PE (or PE_G) as features for classification tasks (such as in [36]) is outside of the scope of this paper.

A. Contributions

The main contributions of this article are:

- For the first time, the concept of a nonlinear entropy metric -permutation entropy- is extended, from unidimensional time series and two-dimensional images to data residing on the vertices of (irregular) undirected graphs.
- We explore how the permutation entropy of graph signals depends on both the signal and the graph. We also give conditions to change the graph while maintaining the entropy of a signal.
- We show that our algorithm can also be applied to signals on directed graphs and/or weighted graphs.
- We illustrate the application of the permutation entropy on graphs algorithm on well-established benchmark synthetic datasets and on real-world data, showing that it generalises well the behaviour of the unidimensional PE and the recently introduced two-dimensional permutation entropy.

A. Structure of the article

The outline of the paper is as follows: Section II introduces the classical permutation entropy and the notation on graph theory used in the article (including the basic definition of the normalised Laplacian). Section III presents the main contribution: the permutation entropy for graphs signals, including a version for weighted and directed graphs. In addition, the section presents some examples and study how geometric modification on the graph preserves the entropy values of the signal. Section IV shows how $\mathrm{PE_G}$ applies to real and synthetic signals residing on 1D, 2D and irregular graphs. The conclusions and future lines of research are presented in Section V and it concludes the paper.

II. Background and Notation

In this section, we introduce general background information, including the original permutation entropy (Section II-A), the definition of a graph and the notion of the normalised Laplacian (Section II-B). These definitions will be fundamental to generalise the permutation entropy from a time series to a general graph signal case.

A. Original Permutation Entropy

Permutation entropy (PE) measures the irregularity of a time series. The algorithm is based on the comparison of neighbouring values within patterns in the time series [1]. It is a simple, robust method and computationally very fast (as it depends linearly on the number of samples of the signal: $O(N)$). For a time series $\mathbf{X} = \{x_i\}_{i=1}^{N}$, the algorithm to compute PE is the following [2]:

1) For $2 \leq m \in \mathbb{N}$ the *embedding dimension* and $L \in \mathbb{N}$ the *delay time*, the *embedding vector* $\mathbf{x}_i^m(L) \in \mathbb{R}^m$ is

given by

$$\mathbf{x}_i^m(L) = (x_{i+jL})_{j=0}^{m-1} = \big(x_i, x_{i+L}, \ldots, x_{i+(m-1)L}\big) \tag{1}$$

for all $1 \leq i \leq N-(m-1)L$. For practical purposes, the authors [1] suggest working with $3 \leq m \leq 7$.

2) The m real numbers of the embedding vector $\mathbf{x}_i^m(L)$ are associated with natural numbers from 1 to m, and then arranged in increasing order. Then, each embedding vector $\mathbf{x}_i^m(L)$ is assigned to one of the $m!$ permutation (also called possible patterns) denoted by π.
Formally, the embedding vector

$$\mathbf{x}_i^m(L) = \big(x_i, x_{i+L}, \ldots, x_{i+(m-1)L}\big)$$

is arranged in the increasing order vector:

$$\big(x_{i+(k_1-1)L} \leq x_{i+(k_2-1)L} \leq \cdots \leq x_{i+(k_m-1)L}\big)\,.$$

Following the convention in [2], if some values are equal, the order is given by the corresponding $k's$. For example, if $x_{i+(k_{l1}-1)L} = x_{i+(k_{l2}-1)L}$ and $k_{l1} < k_{l1}$, we write $x_{i+(k_{l1}-1)L} \leq x_{i+(k_{l2}-1)L}$. This convention does not affect the results [37]. In particular, the constant vector $(1, 1, \ldots, 1)$ is mapped onto $(1, 2, \ldots, m)$. Therefore, any embedding vector $\mathbf{x}_i^m(L)$ is uniquely mapped onto the vector $(k_1, k_2, \ldots, k_m) \in \mathbb{N}^m$.

3) The relative frequency for the distinct permutation $\pi_1, \pi_2, \ldots, \pi_k$ where $k \leq m!$ is denoted by $p(\pi_1), p(\pi_2), \ldots, p(\pi_k)$. The permutation entropy PE for the time series $\mathbf{X}$ is computed as the Shannon entropy for the k distinct permutations as follows

$$\mathrm{PE}(m, L) = -\sum_{i=1}^{k} p(\pi_i) \ln p(\pi_i)\,.$$

It is clear that $0 \leq \mathrm{PE}(m, L) \leq \ln(m!)$, then, for convenience, it is normalised by $\ln(m!)$, then

$$0 \leq \frac{\mathrm{PE}(m, L)}{\ln(m!)} \leq 1\,.$$

The simple case is for $m = 2$ and $L = 1$. Given a time series $\mathbf{X}$, the idea of $\mathrm{PE}(m, L)$ is to organise the $N-1$ pair of neighbours according to their relative values. Let p_1 be the number of pair of neighbours such that $x_t < x_{t+1}$, represented by the permutation (1,2); and p_2 be the number of pair of neighbours such that $x_t > x_{t+1}$, represented by the permutation (2,1). Then, using Shannon's entropy:

$$\mathrm{PE}(m, L) = -\frac{p_1}{N-1} \log \frac{p_1}{N-1} - \frac{p_2}{N-1} \log \frac{p_2}{N-1}\,.$$

In permutation entropy, the ordering of the values is taken into account, but not the magnitude of changes.

An extension of Permutation Entropy to two-dimensional patterns (images) has very recently been published [13]. This two-dimensional algorithm takes rectangular windows across the image and, for each window, vectorises its contents. Then, steps 2 and 3 are applied.

B. Graphs, Graph Signals and the Normalised Laplacian

An *undirected graph* G is defined as the triple $G = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ which consists of a finite set of vertices or nodes $\mathcal{V} = \{1, 2, 3, \ldots, N\}$, an edge set $\mathcal{E} \subset \{(i, j) : i, j \in \mathcal{V}\}$ and $\mathbf{A}$ is the corresponding $N \times N$ symmetric adjacency matrix on edges with entries $1 = \mathbf{A}_{ij} = \mathbf{A}_{ji}$ if $(i, j) \in \mathcal{E}$ and 0 otherwise.

Along this article, we consider graphs containing no multiple edges, loops or isolated vertices, i.e. *simple* graphs.

A *graph signal* is a real function defined on the vertices, i.e., $\mathbf{X} : \mathcal{V} \longrightarrow \mathbb{R}$. The graph signal $\mathbf{X}$ can be represented as an N-dimensional column vector, $\mathbf{X} = [x_1; x_2; \ldots; x_N] \in \mathbb{R}^N$ (with the same indexing of the vertices).

It is well-known that the power of the adjacency matrix counts the number of k-walks between two vertices, i.e., the entry $(\mathbf{A}^k)_{i,j}$ is equal to the number of walks of length equal to k having the vertex i as start and vertex j as end.

Given a graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{A})$, we define a function on the vertices $\deg^k : \mathcal{V} \longrightarrow \mathbb{R}$ given by

$$\deg^k(i) := \sum_{j \in \mathcal{V}} (\mathbf{A}^k)_{ij} = \sum_{j \in \mathcal{V}} (\mathbf{A}^k)_{ji}\,; \tag{2}$$

for $k = 1$, we write $\deg^1(i) = \deg(i)$, i.e., the degree of a vertex i is the number of edges that are incident to it.

Given a vertex i, we define $\mathcal{N}_k(i)$ as the set of all vertices connected to the vertex i with a walk on k edges, i.e.,

$$\mathcal{N}_k(i) := \{\, j \in \mathcal{V} \,|\, \text{it exists a walk on } k \text{ edges joining } i \text{ and } j \,\}\,, \tag{3}$$

with the convention $\mathcal{N}_0(i) = \{i\}$ and $\mathcal{N}_1(i) = \mathcal{N}(i)$.

The normalised Laplacian is defined using the adjacency matrix as follows:

$$\Delta := I - \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}}\,,$$

where $\mathbf{D}$ is the *degree matrix*, i.e., a diagonal matrix given by $\mathbf{D}_{ii} = \deg(i)$.

III. Permutation Entropy for Graph Signals

This section introduces the permutation entropy for graph signals (denoted as $\mathrm{PE_G}$). In the original PE for time series, the construction of the embedding vectors given by (1) is made between values on consecutive steps (t and $t + L$) for $m = 2$. Consecutive values cannot be defined straightforwardly in irregular graphs.

As a motivation for the general definition, we show (with an example) how to compare between values on a fixed vertex and its neighbourhoods (Section III-A). For the general formulation, we will consider the topology of the graph encoded in the adjacency matrix to define the algorithm and construct the embedding vectors (Section III-B). Finally, we extend the algorithm for directed (Section III-C) and weighted graphs (Section III-D).

A. Motivation and Example

Consider the graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ and $\mathbf{X}$ be any signal on the graph. Similarly to PE for time series of order $m = 2$ and $L = 1$, we compare the signal value at the vertex i with respect

to the average of its neighbours, i.e., we will compare:

$$x_i \quad \text{and} \quad \frac{1}{\deg(i)} \sum_{j \in \mathcal{N}(i)} x_j \ .$$

Observe the relation with the normalised Laplacian, i.e.,

$$\frac{1}{\deg(i)} \sum_{j \in \mathcal{N}(i)} x_j = x_i - \Delta x_i = (I - \Delta) x_i = \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} x_i \ .$$

For each $i \in \mathcal{V}$, we define the pair where its first component is the value of the signal $\mathbf{X}$ on the node i and the second component is the average of the signal $\mathbf{X}$ on the neighbours of i, i.e.,

$$\mathbf{y}_i := (x_i, (I - \Delta)\, x_i) = \left(x_i, \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} x_i\right) \ . \qquad (4)$$

The pair is analogous for the embedding vector defined by (1) in PE.

We organize the N pairs according to their relative values. Let p_1 be the number of pairs for which $x_i < \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} x_i$, or equivalently $\Delta x_i < 0$ (represented by the permutation 12) and let p_2 be the number of the pairs for which $x_i > \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} x_i$ or equivalently $\Delta x_i > 0$ (represented by the permutation 21).

We define the permutation entropy of the graph signal $\mathbf{X}$ for embedding $m = 2$ and $L = 1$ as a measure of the probability of the permutation (1, 2) and (2, 1), so:

$$\mathrm{PE_G} = -\frac{p_1}{N} \log \frac{p_1}{N} - \frac{p_2}{N} \log \frac{p_2}{N} \ .$$

Intuitively, we are dividing the vertices of G according to the signal $\mathbf{X}$ into two subsets. One set corresponds to the vertices such that $\Delta x_i > 0$, i.e., it contains the *local maximums* of the signal on the graph domain. Similarly, the other set contains vertices such $\Delta x_i < 0$, i.e., the *local minimums*. The interpretation is analogous to the permutation entropy for time series (for the case $m = 2$ and $L = 1$), where the patterns are defined by the points where the function is increasing or decreasing.

Example 1: Consider the graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ and signal $\mathbf{X}$ shown in Fig. 1.

We construct the embedding vectors given by (4). We obtain one pair for each vertex, i.e., $\mathbf{y}_1 = (-1, -1.15)$, $\mathbf{y}_2 = (-2.3, -0.5)$, $\mathbf{y}_3 = (0.-1.325)$, $\mathbf{y}_4 = (-3, 2.5)$, $\mathbf{y}_5 = (1, 2.5)$, $\mathbf{y}_6 = (5, -0.333)$, $\mathbf{y}_7 = (1, 1.95)$ and $\mathbf{y}_8 = (-1.1, 1)$.

We have two patterns for the case $m = 2$. The pairs $\mathbf{y}_2, \mathbf{y}_4, \mathbf{y}_5, \mathbf{y}_7$ and $\mathbf{y}_8$ belong to the same pattern (where $x_i < \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} x_i$) and $\mathbf{y}_1, \mathbf{y}_3$ and $\mathbf{y}_6$ belong to the second pattern (where $x_i > \mathbf{D}^{-\frac{1}{2}} \mathbf{A} \mathbf{D}^{-\frac{1}{2}} x_i$).

The relative frequency of each permutation pattern is $\frac{5}{8}$ and $\frac{3}{8}$ respectively. Finally using Shannon's entropy, the $\mathrm{PE_G}$ value of the signal $\mathbf{X}$ is equal to $-\frac{5}{8}\ln(-\frac{5}{8}) - \frac{3}{8}\ln(-\frac{3}{8}) = 0.6616$. The normalised $\mathrm{PE_G}$ is $\frac{0.6616}{\ln(2)} = 0.9544$.

B. Permutation Entropy for Graphs Signals

Let $G = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ be a graph and $\mathbf{X} = \{x_i\}_{i=1}^N$ be a signal on the graph, the permutation entropy for the graph signals $\mathrm{PE_G}$ is defined as follow:

1) For $2 \leq m \in \mathbb{N}$ the *embedding dimension* and $L \in \mathbb{N}$ the *delay time*, we construct the embedding vector $\mathbf{y}_i^{m,L} \in \mathbb{R}^m$ given by

$$\mathbf{y}_i^{m,L} = \left(y_i^{kL}\right)_{k=0}^{m-1} = \left(y_i^0, y_i^L, \ldots y_i^{(m-1)L}\right) \ ,$$

for all $i = 1, 2, \ldots, N$ and where

$$y_i^{kL} = \frac{1}{|\mathcal{N}_{kL}(i)|} \sum_{j \in \mathcal{N}_{kL}(i)} x_j \qquad (5)$$

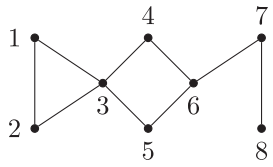
$$= \frac{1}{|\mathcal{N}_{kL}(i)|} (\mathbf{A}^{kL} \mathbf{X})_i \ . \qquad (6)$$

Recalling $\mathcal{N}_{kL}$ is defined by (3), it follows that $y_i^0 = x_i$ and $y_i^1 = (I - \Delta)x_i$.

2) The m real numbers of the embedding vector $\mathbf{y}_i^{m,L}$ are associated with integer numbers from 1 to m and then arranged in increasing order.
There are $m!$ permutation (also called possible patterns) π for an m-embedding vector.

3) The relative frequency for the distinct permutation $\pi_1, \pi_2, \ldots, \pi_k$ where $k \leq m!$ is denoted by $p(\pi_1), p(\pi_2), \ldots, p(\pi_k)$. The permutation entropy $\mathrm{PE_G}$ for signal $\mathbf{X}$ is computed as the Shannon entropy for the k distinct permutations

$$\mathrm{PE_G}(m, L) = -\sum_{i=1}^{k} p(\pi_i) \ln p(\pi_i) \ .$$



(a)

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix} \qquad \mathbf{X} = \begin{pmatrix} -1 \\ -2.3 \\ 0 \\ -3 \\ 1 \\ 5 \\ 1 \\ -1.1 \end{pmatrix}$$

(b) (c)

Fig. 1. 1(a) An example of a graph G, 1(b) its adjacency matrix $\mathbf{A}$, and 1(c) a graph signal $\mathbf{X}$.

In the next sections, without specification $L = 1$ is chosen. If all possible patterns have equal probability value, the $\mathrm{PE_G}$ reaches its highest value which is equal to $\ln(m!)$. Note that we use the normalised $\mathrm{PE_G}$ as $\dfrac{\mathrm{PE_G}}{\ln(m!)}$.

We use (5) to prove some properties of $\mathrm{PE_G}$, while (6) is more useful for a numerical implementation. Recall that the Laplacian of a isolated vertex x_i is defined in the literature as zero, hence $y_i^1 = (I - \Delta)x_i = x_i$ and $\mathbf{y}_i^{m,L}$ is the constant vector.

Following the convention in [2] is mapped onto $(1, 2, \ldots, m)$ and according to [37], the convention does not affect the result. For simplicity in the notation, along with this article, we will also assume that G has no isolated vertices to avoid $|\mathcal{N}_{kL}(i)| = 0$ in both equations.

Some of the similarities and differences between the original PE and the permutation entropy for graph signals $\mathrm{PE_G}$ are the following:

1) The main difference is the construction of the embedding vectors in step 1. With (1), PE constructs $N - (m - 1)L$ embedding vectors using m consecutive values of the signal. $\mathrm{PE_G}$ uses the adjacency matrix and (4) to obtain N embedding vectors (independent of m and L), each embedding vector corresponds to one vertex.
2) Step 2 (arrange the embedding vectors in increasing order) and 3 (computing Shannon's entropy) for both algorithms PE and $\mathrm{PE_G}$ are the same. The computational cost of $\mathrm{PE_G}$ depends on the the number of vertices and edges (see Section IV-E).
3) A time series $\mathbf{X}$ can be considered as a graph signal over the graph G given by the undirected path. The value given by PE for the time series $\mathbf{X}$ and the value given by the $\mathrm{PE_G}$ when $\mathbf{X}$ is considered as a graph signal are (in general) different. However, the dynamics detected by the PE is preserved with $\mathrm{PE_G}$, see Section IV. The *arrow of time* information is lost when we consider the undirected path. One way to preserve the information is to consider $\mathbf{X}$ as a graph signal on the directed path and define $\mathrm{PE_G}$ for directed graphs (see Section III-C).
4) The adjacency matrix $\mathbf{A}$ is a particular case of the weight matrix $\mathbf{W}$, where all non-zero weight values are equal to one. The general algorithm for weighted graphs is presented in Section III-D.

The entropy of the graph signal $\mathbf{X}$ depends on its numerical values and the graph topology. It is interesting to study these quantities (that depend on the geometric structure of the graph) under some geometric perturbation (delete edges, vertices and contractions [38], [39]). We will show that under some conditions, adding or deleting edges on the graph will preserve the permutation entropy of the signal $\mathbf{X}$.

Proposition 1: Let $\mathbf{X}$ be a graph signal over the graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ with entropy value $\mathrm{PE_G}$ for $m = 2$ and $L = 1$. Let $i, j \in \mathcal{V}$ be two vertices such that $\Delta x_i < 0 < \Delta x_j$.

1) If $\{i, j\} \notin \mathcal{E}$ and $x_i < x_j$, then $\mathrm{PE}_G = \mathrm{PE}_{G'}$ where $G' = G + \{i, j\}$.
2) If $\{i, j\} \in \mathcal{E}$ and $x_j < x_i$, then $\mathrm{PE}_G = \mathrm{PE}_{G'}$ where $G' = G - \{i, j\}$.

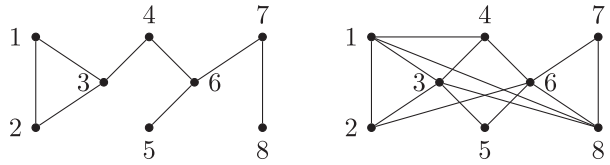
Proof: We will prove 1 and the proof for 2 is similar.

1) First, we will prove that $\Delta x_i < 0$ together with $x_i < x_j$ implies $\Delta_{G'} x_i < 0$ where $G' = G + \{i, j\}$. It follows by:

$$\Delta x_i < 0$$

$$deg_G(i) x_i < \sum_{k \in \mathcal{N}_G(i)} x_k$$

$$x_i \left(deg_G(i) + 1\right) < \sum_{k \in \mathcal{N}_G(i)} x_k + x_j$$



(a) Graph G_1 (b) Graph G_2

Fig. 2. Examples of two graphs. G_1 is a subgraph of G and G is a subgraph of G_2. Both graphs G_1 and G_2, preserve the entropy of the original signal defined on G.

$$x_i < \frac{1}{deg_{G'}(i)} \sum_{k \in \mathcal{N}_{G'}(i)} x_k$$

$$\Delta_{G'} x_i < 0 \, .$$

In an analogous way, it can be shown that $0 < \Delta x_j$ and $x_i < x_j$ imply that $0 < \Delta_{G'} x_j$. Therefore, for each vertex $k \in \mathcal{V}$ the pairs $(x_k, (I - \Delta)x_k)$ preserve the same order on both graphs (G and $G + \{i, j\}$). The relatives frequencies are equals, hence, their entropy are equals. ■

In the previous result, we prove a condition that preserves not only the entropy value but their relative frequencies. Hence, we can apply in an iterative way to generalise the result with the following corollary.

Corollary 1: Let $\mathbf{X}$ be a graph signal over the graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ with entropy value $\mathrm{PE_G}$ for $m = 2$ and $L = 1$. Define the sets with the following property

$$E_0 = \{(i, j) \,|\, i \in A, j \in B \text{ and } x_i < x_j\} \tag{7}$$

$$E_1 = \{(i, j) \,|\, i \in A, j \in B \text{ and } x_j < x_i\} \tag{8}$$

where $A = \{\, i \in \mathcal{V} \,|\, \Delta x_i < 0 \,\}$ and $B = \{\, i \in \mathcal{V} \,|\, 0 < \Delta x_i \,\}$.

- If $E' \subset E_0$ and $E' \cap \mathcal{E} = \emptyset$, then $\mathrm{PE}_G = \mathrm{PE}_{G'}$ where $G' = G + E'$.
- If $E' \subset E_1$ and $E' \subset \mathcal{E}$, then $\mathrm{PE}_G = \mathrm{PE}_{G'}$ where $G' = G - E'$.

Example 2: Consider the graph G and signal $\mathbf{X}$ given in Example 1. Define the sets $A = \{\, i \in \mathcal{V} \,|\, \Delta x_i < 0 \,\} = \{2, 4, 5, 7, 8\}$ and $B = \{\, i \in \mathcal{V} \,|\, 0 < \Delta x_i \,\} = \{1, 3, 6\}$.

The edge set given by $E_0 = \{(1, 4), (1, 8), (3, 8), (6, 2), (6, 8)\}$ fulfil the condition in (7). Define $G' = G + E'$ for any $E' \subset E_0$, then by Corollary 1 follows the entropy is the same, i.e., $\mathrm{PE}_G = \mathrm{PE}_{G'}$. The case $E' = E_0$ is shown in Fig. 2(a).

Similarly, $E_1 = \{(3, 5)\}$ (fulfil the condition in (8)), then G and $G - E_1$ have the same entropy. Graph $G - E_1$ is shown in Fig. 2(b).

In this sense, given a signal over a graph, with Proposition 1 we can find structures that preserve not only the same numeric entropy value but the *maximal* and *minimal* values of the signal on the same vertices.

The *invariance property* with respect to monotonic transformation of the time signal is an important property of the PE, i.e., if $\mathbf{X}$ is a time series, and f is an arbitrary strictly decreasing (or increasing) real function, then the classical PE of the time

series $\mathbf{X}$ and $f(\mathbf{X})$ are equal [1]. This function occurs, for example, when the data is measured with different equipment. In a similar scenario, the following proposition shows that some modification on the signal does not change the permutation entropy.

Proposition 2: Let $\mathbf{X}$ be a graph signal over the graph G and c a real (non zero) constant function defined on the vertices. The entropy of the signals: $\mathbf{X}$, $c\mathbf{X}$ and $c+\mathbf{X}$ are equal.

Proof: For any $2 \leq m \in \mathbb{N}$ and $L \in \mathbb{N}$, the embedding vector for the graph signal $\mathbf{X}$ are defined as $\mathbf{y}_i^{m,L} = (y_i^{kL})$ (see (5)). It is easy to show that the embedding vectors for $c\mathbf{X}$ are $c\mathbf{y}_i^{m,L}$ and the embedding vectors for $c+\mathbf{X}$ are $c+\mathbf{y}_i^{m,L}$. Therefore, the proportion of the patterns in the original signal $\mathbf{X}$ is preserved in the signals $c\mathbf{X}$ and $c+\mathbf{X}$. Therefore, its entropy values are equal.

■

The previous proposition shows a difference with respect to the definition of *smoothness* on graph signals. Formally, for a graph signal $\mathbf{X}$ on the G, the smoothness is measured in terms of the quadratic form of the normalised Laplacian

$$\mathbf{X}^T \Delta \mathbf{X} := \frac{1}{2}\sum_{i\sim j}(x_i - x_j)^2 \ . \tag{9}$$

Therefore, the smoothness of the signal $c\mathbf{X}$ is different (in general) from the smoothness of the signal $\mathbf{X}$. The algorithm $\mathrm{PE_G}$ is interested in the change of the patterns rather than the changes of values of the signal as in the smoothness definition.

C. Permutation Entropy for Signals on Directed Graph

In Section III-B we introduced the permutation entropy for undirected graphs. As a particular case, in this section, we introduce a permutation entropy algorithm for directed graphs, denoted as $\mathrm{PE}_{\overrightarrow{G}}$.

A *directed graph* or *digraph* is a graph where each edge has an orientation or direction. The directed edge (called also an *arc*) is an ordered pair (i,j) and it is drawn as an arrow from the vertex i to the vertex j. A *directed path* on k vertices is a directed graph that joins a sequence of different vertices with all the edges in the same direction and is denoted as $\overrightarrow{P}$, i.e. its vertices are $\{1,2,\ldots,k\}$ and its arcs $(i,i+1)$ for all $1 \leq i \leq k-1$.

The permutation entropy for signals on directed graphs will be almost identical to the presented in Section III-B except for a small change in the construction of the embedding vector.

1) For $2 \leq m \in \mathbb{N}$ the *embedding dimension* and $L \in \mathbb{N}$ the *delay time*, define the set

$$V^* = \{\, i \in \mathcal{V} \,|\, \overrightarrow{\mathcal{N}}_{(m-1)L}(i) \neq \emptyset \,\}\,, \tag{10}$$

where

$$\overrightarrow{\mathcal{N}}_k(i) = \{\, j \in \mathcal{V} \,|\, \text{it exists a directed path on } k \text{ arcs from } i \text{ to } j \,\}. \tag{11}$$

We construct the embedding vector $\mathbf{y}_i^{m,L} \in \mathbb{R}^m$ given by

$$\mathbf{y}_i^{m,L} = \left(y_i^{kL}\right)_{k=0}^{m-1} = \left(y_i^0, y_i^L, \ldots y_i^{(m-1)L}\right)\ , \tag{12}$$

for all $i \in V^*$ where

$$y_i^{kL} = \frac{1}{|\overrightarrow{\mathcal{N}}_{kL}(i)|}\sum_{j\in\overrightarrow{\mathcal{N}}_{kL}(i)} x_j\ . \tag{13}$$

Step 2 and 3 are the same as in Section III-B. The next proposition shows that the classical permutation entropy is the same if we consider the time series as a signal over a directed path. Therefore, we generalise the PE for all directed graphs with the same values as the original one.

Proposition 3: Let $\mathbf{X} = \{x_i\}_{i=1}^N$ be a time series and consider $G = \overrightarrow{P}$ the directed path on N vertices, then for all m and L the equality holds:

$$\mathrm{PE}(m,L) = \mathrm{PE}_G(m,L)\ .$$

Proof: For the embedding dimension m, delay time L and G the directed path with N vertices, then $\overrightarrow{\mathcal{N}}_k(i) = \{i+k\}$ for all $1 \leq k \leq N-i$ and $\emptyset$ otherwise (see (11)).

The set defined in (10) is $V^* = \{1,2,3,\ldots,N-(m-1)L\}$. Then, for all $i \in V^*$, by (13):

$$y_i^{kL} = \frac{1}{|\overrightarrow{\mathcal{N}}_{kL}(i)|}\sum_{j\in\overrightarrow{\mathcal{N}}_{kL}(i)} x_j = x_{i+kL}\ .$$

Therefore, the embedding vector $\mathbf{y}_i^{m,L} \in \mathbb{R}^m$ defined by (12) is

$$\mathbf{y}_i^{m,L} = \left(y_i^{kL}\right)_{k=0}^{m-1} = \left(x_i, x_{i+L}, \ldots x_{i+(m-1)L}\right)\ ,$$

for all $i \in V^* = \{1,2,3,\ldots,N-(m-1)L\}$, hence the are exactly the same embedding vectors defined in the original PE in (1), i.e. $\mathbf{y}_i^{m,L} = \mathbf{x}_i^m(L)$. Because steps 2) and 3) in both algorithms are the same, we conclude $\mathrm{PE} = \mathrm{PE_G}$.

D. Permutation Entropy for Signals on Weighted Graphs.

In most of the examples, the adjacency matrix usually it is enough. Nevertheless, the previous results and algorithms can be generalised for weighted graphs.

A *weighted undirected graph* G is defined as the triple $G = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ which consist of a finite set of vertices or nodes $\mathcal{V} = \{1,2,3,\ldots,N\}$, an edge set $\mathcal{E} = \{(i,j) : i,j \in \mathcal{V}\}$ and $\mathbf{W}$ is the corresponding $n \times n$ symmetric adjacency matrix weighted on edges with entries $0 \leq w_{ij} = w_{ji}$ the weight of edge (i,j).

Observe that $(\mathbf{W}^k)_{ij}$ is the sum of the product of all the weights of all the walks from the vertex i to the vertex j of length exactly k. We define a function on the vertices $\deg^k : \mathcal{V} \longrightarrow \mathbb{R}$ given by

$$\deg^k(i) := \sum_{j\in\mathcal{V}}(\mathbf{W}^k)_{ij} = \sum_{j\in\mathcal{V}}(\mathbf{W}^k)_{ji}\ . \tag{14}$$

Let $\mathbf{X} = \{x_i\}_{i=1}^N$ be a signal on the graph $G = (\mathcal{V}, \mathcal{E}, \mathbf{W})$, the permutation entropy of the signal $\mathbf{X}$ on the weighted graph G is the same that the presented in Section III-B, but replace the (5) by the following:

$$y_i^{kL} = \frac{1}{\deg^k(i)}(\mathbf{W}^{kL}\mathbf{X})_i\ .$$

Similarly to Section III-C, we can extend the algorithm for weighted directed graphs.

E. Permutation Entropy for Signals With Vector Embedding Delays

In [29], [30], PE is generalised for a vector of embedding delays (rather than a scalar). Analogously, we can extend PE_G with vector embedding delays. For $2 \leq m$ the *embedding dimension*, the *embedding delay* is represented as a $m-1$ element vector $L = (l_1, l_2, \ldots, l_{m-1})$. The embedding vector $\mathbf{y}_i^{m,L} \in \mathbb{R}^m$ is given by

$$\mathbf{y}_i^{m,L} = \left(y_i^0, y_i^{l_1}, y_i^{l_1+l_2}, \ldots, y_i^{\sum_{j=1}^{m-1} l_j} \right) ,$$

for all $i = 1, 2, \ldots, N$ (see (5)).

In particular, if $\mathbf{X}$ is a time series, we can prove (similar to Proposition 3) that PE with vector embedding delays [29] is equal to $\mathrm{PE_G}$, where $G = \overrightarrow{P}$ is the directed path on N vertices and the embedding vectors are defined above.

IV. Experiments and Discussion

In this section, we apply our algorithm to synthetic and real data, including signals on 1D (time series), 2D (image) and irregular domains (graph). We show that $\mathrm{PE_G}$ is a suitable generalisation of the original PE for time series, but with the advantage that the input could be any graph signal.

In 1D, the equality $\mathrm{PE_G} = \mathrm{PE}$ holds if the underlying graph G is the directed path (Proposition 3). The values differ slightly when G is an undirected graph. However, $\mathrm{PE_G}$ can detect different dynamics for synthetic signals (logistic map, autoregressive models) and for real signals (fantasy data set). In case the input is 2D, our algorithm gives similar results to the recently introduced two-dimensional permutation entropy [13].

Finally, we apply the algorithm for signals defined in general graphs. Fixing the underlying graph G and considering a signal $\mathbf{X}$, we show the $\mathrm{PE_G}$ value depends on the irregularity of the signal. We also consider the reverse case, fixing the signal $\mathbf{X}$ (for example, a set of n random values). We consider several underlying graphs G with $\mathbf{X}$ as a graph signal, and we study the impact on the graph structure has on the entropy measure.

A. Examples on 1D and the Classical Permutation Entropy

In this section, we consider a time series $\mathbf{X} = \{x_i\}_{i=1}^N$ and three underlying graphs: G_1 a directed path, G_2 an undirected path, and G_3 a directed path with the reverse orientation respect G_1. For any m and L, recall that the classical PE gives equal results as PE_{G_1} (Proposition 3), hence the permutation for graph signals has the same properties as the classic one in 1D. Moreover, we verified computationally that the values of PE_{G_1} are almost the same as PE_{G_3}, and for $N > 30$, its difference is always in the order of 10^{-16} (computational accuracy of MATLAB).

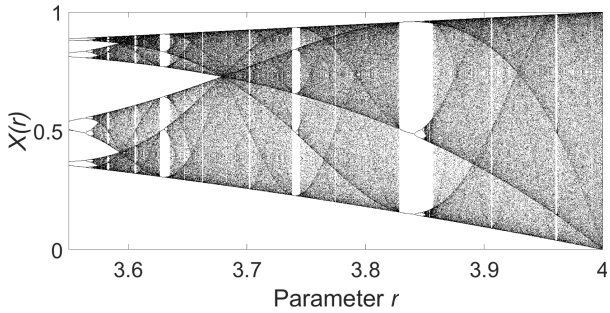
1) The Logistic Map: PE has been used to detect dynamical changes in time series [1], [2]. A non linear dynamic system used to show the performance of PE is the logistic map, given by

$$x_{n+1} = r x_n (1 - x_n) .$$

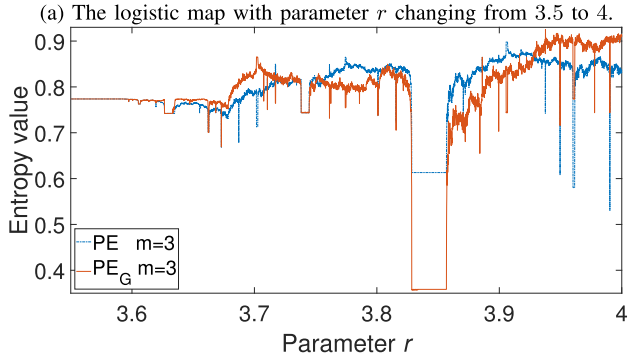
The analysis is relevant for the parameter r. Thus, we vary the parameter $3.55 < r < 4.0$ with increments r in steps of 10^{-5} at each iteration. We define the sequence given by $\mathbf{X}(r) = \{x_i\}_{i=1}^N$ where the sample points are $N = 2^{12}$. To discard transients, we delete the first 2^{10} sample points. The initial value is $x_0 = 0.65$. Fig. 3(a) shows the time series, where each point of the discrete time is plotted for each value of r, i.e., the bifurcation diagram for the logistic map for $r \in (3.55, 4.0)$.

We created 4501 time series. For each sequence, we consider two underlying graphs: G_1 a directed path and G_2 an undirected path, both on N vertices. Finally, we compute its permutation entropy for $m = 3$ and $L = 1$ (see Fig. 3(b)). It is known that chaotic behaviour starts for $3.5699 \leq r \leq 4$. The entropy algorithm is able to detect *island of stability*, i.e., values of r such the data sequence shows non-chaotic behaviour. The largest window is $1 + \sqrt{8} \approx 3.8284 < r < 3.8415$, this range of r shows oscillation among three values [40]. The algorithm (with both underlying graphs G_1 and G_2) detects the window (for any embedding dimension m). However, the wider gap between the values for G_2 indicates a large sensitivity of the algorithm to detect the changes of the dynamic on the data. A similar effect in other islands of stability occurs. This fact is in agreement with other previous studies [1], [2].

For computing $\mathrm{PE_G}$, we used different combination of embedding dimension ($m = 2, 3, 4$), delay time $L = 1, 2, 3$, lengths of time series and initial condition. In all the cases, $\mathrm{PE_G}$



(a) The logistic map with parameter r changing from 3.5 to 4.



(b) Entropy values computed with underlying graph G_1 (directed path) and G_2 (undirected path). Both algorithms use $m = 3$ and $L = 1$.

Fig. 3. Diagram for the logistic map and the values of the permutation entropy for graph signals.

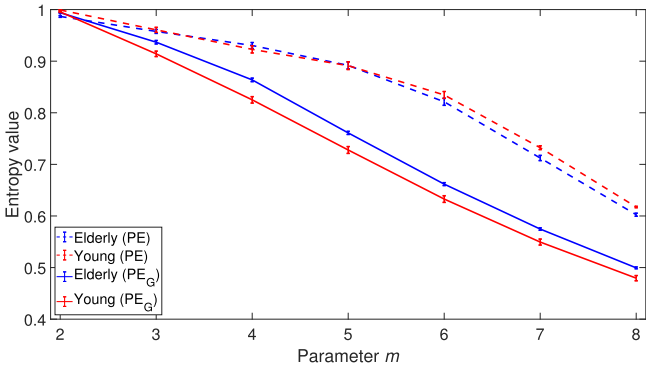


Fig. 4. Averaged entropy values and standard error bars for embedding dimensions $2 \leq m \leq 8$.

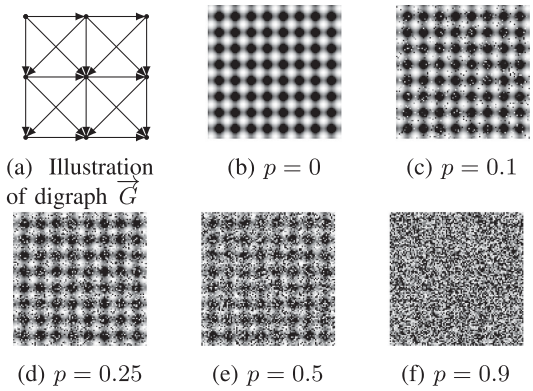


Fig. 5. Examples of images generated by the MIX process in (15). The size of each image is 100×100 pixels.

had good performance in detecting the dynamic changes in the time series.

2) Heartbeat Time Series: The Fantasia database has been analysed widely to validate the performance of some entropies algorithms [7], [8]. We use 10 heartbeat time series: 5 correspond to young subjects (aged between 21 and 34 years) and 5 recordings from elderly subjects (aged between 68 and 85 years). Each time series is divided into samples of 800 points with an overlap of 400 points. The classical PE is computed (or equivalently, its permutation graph entropy for the directed path) for each sample. We also consider each time series as a graph signal on the undirected path, and $\mathrm{PE_G}$ is computed for each case. We consider the embedding dimensions $2 \leq m \leq 8$ for the computation. The averaged entropy values with their standard error bars are shown in Fig. 4.

The analysis shows that the elderly and young subjects are not indistinguishable by the classical PE for $3 \leq m \leq 6$. Considering $\mathbf{X}$ as a graph signal on the undirected path, the algorithm $\mathrm{PE_G}$ can difference the subjects for all embedding dimensions (except $m = 2$). Changing the size of the samples and/or intersection does not change this behaviour. In addition, we observe that the entropy values of the elderly subjects are consistently higher than the young subjects for all embedding dimensions with $\mathrm{PE_G}$. The results are consistent for a larger time delay. In contrast, the order in PE values depends on the parameter m, that is, the ranking of elderly and young people is not consistent.

B. Permutation Entropy for Images (2D)

One of the main advantages of our algorithm is the fact that it can be applied on any graph, including the directed graph shown in Fig. 5(a) (or its undirected version), where any signal can be regarded as an image. Therefore the permutation entropy (described in Section III-C) gives us a metric of the regularity/complexity of images.

An image can be considered a graph signal defined on the 2D grid. One can apply the classical Shannon entropy to the probability distribution of the image values (without permutation patterns), but this would ignore its geometrical structure. In contrast, some 2D algorithms inherently use such geometrical structure. [13] introduces a 2D permutation entropy (PE_{2D}), which implicitly uses the spatial information in the images when building patterns from neighbouring pixels before computing the entropy of this distribution of patterns. PE_{2D} was used to discriminate periodic from synthesized images. The distribution entropy for images (DistE_{2D}) presented in [16] also uses the geometrical information to compute a distance matrix which eventually feeds into Shannon entropy. DistE_{2D} is used to analyse the irregularity of small-sized textures. In [41], the Wada index used the geometrical information based on the standard box-counting algorithm. The distribution of different colours in two-dimension was obtained, and the truncated Shannon entropy was applied. The Wada index was used to detect the existence of Wada basin boundaries. The algorithms PE_{2D}, DistE_{2D} and Wada index use the geometrical structure of the regular 2D grid. However, their definition depends strongly on the regularity of the structure (of the image). It is unclear how the methods can be applied to irregular domains or graph domains. PE_G also allows the consideration of the geometrical structure of the 2D grid but, as a major novelty, the definition does not depend on the regularity of the image. Hence, PE_G can deal with irregular domains, but the behaviour is similar to PE_{2D} or DistE_{2D} in the cases where the domains are regular.

For the analysis, we choose the directed graph $\overrightarrow{G}$ because: 1) the directed adjacency matrix has more entries equal to zero than the undirected version and hence the algorithm is faster, 2) the algorithm PE_{2D} presented in [13] (and almost every 2D algorithm) implicitly uses this orientation in the vectorisation, 3) the orientation preserves more information of the *geometry* of the graphs and hence gives us better results, 4) if $\overrightarrow{G}$ is a 2D graph with size $N \times 1$ or $1 \times N$, then $\mathrm{PE}_{\overrightarrow{G}}$ is equal to the classical PE, hence, our algorithm is a natural generalisation and 5) choosing vertex 1 or any other vertex as an origin of the orientation gives almost identical results, because of its symmetry.

To assess the ability of $\mathrm{PE_G}$ and similarly to [14], [16], we use the two-dimensional processing MIX_{2D}.

1) MIX_{2D} *Processing:* Let $X_{i,j} = \sin\left(\frac{2\pi i}{12}\right) + \sin\left(\frac{2\pi j}{12}\right)$ and let $Z_{i,j}$ be a random variable where $Z_{i,j} = 0$ with probability $1 - p$ and $Z_{i,j} = 1$ with probability p. In addition, consider $Y_{i,j}$ a matrix of random values ranged in $[-\sqrt{3}, \sqrt{3}]$. The MIX_{2D}

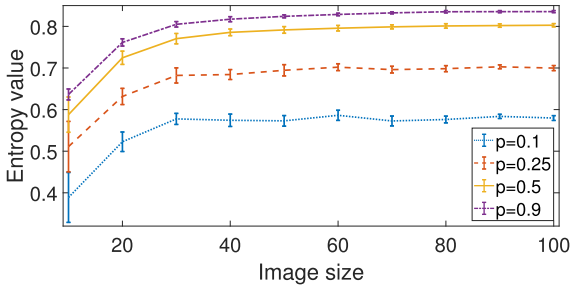


Fig. 6. Mean value and standard deviation of results obtained by $\mathrm{PE_G}$ computed from 20 realisations from MIX processing.

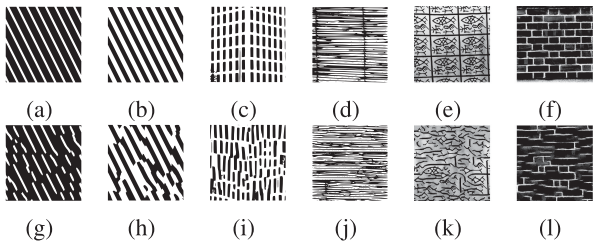


Fig. 7. Examples of periodic textures (a) to (f) and their corresponding synthetic texture (g) to (l).

process is defined by the equation:

$$\mathrm{MIX}_{2D}(p)_{i,j} = (1 - Z_{i,j})\, X_{i,j} + Z_{i,j} Y_{i,j}\ . \qquad (15)$$

Fig. 5 shows samples for different values of p, and the underlying graph consider. To understand the effect of the size of image, we create 10 different realization of $\mathrm{MIX}_{2D}(p)$ for each $p = 0.1, 0.25, 0.5, 0.9$ whose size changed from 10×10 to 100×100 (for larger size, the results are similar). For each realisation, we compute its $\mathrm{PE}_{\overrightarrow{G}}$ (Fig. 5a) with embedding dimension $m = 6$. In Fig. 6 is shown the mean and standard deviation values of $\mathrm{PE}_{\overrightarrow{G}}$. We also compute the PE_{2D} with embedding dimension $d_x = 3$ and $d_y = 2$ (see [13]). In both methods, 6! permutation patterns are possible and the results are similar.

As p goes to 1, the image gets closer to random noise (more irregular) and then the $\mathrm{PE}_{\overrightarrow{G}}$ gets close to 1. In particular, if $r < s$, then the entropy value of $\mathrm{MIX}_{2D}(r)$ is smaller than $\mathrm{MIX}_{2D}(s)$. These results are in agreement with the bidimensional entropy algorithms as the distribution [16] and sample entropy [14].

2) Artificial Periodic and Synthesized Textures: We use the same six periodic textures and their corresponding synthesized textures (as in [13]) to show how $\mathrm{PE_G}$ changes when a periodic turns into a synthesized texture. The images dataset are downloaded from [42]. The original textures and their corresponding texture (sized 256×256) are depicted in the same order in Fig. 7. Considering the directed graph depicted in Fig. 5(a) and for $m = 4$, we compute the $\mathrm{PE_G}$. Results in Table 1 show that the permutation entropy of a synthesized texture is higher than of its corresponding periodic texture. Hence, the algorithm discriminates synthetic periodic from periodic textures in agreement with [13], [16].Similarly to the discussion in 1D examples, the main novelty of this paper is that we can apply the same algorithm (without modification) for any irregular graph,

TABLE I
NUMERICAL VALUES OF $\mathrm{PE_G}$ FOR PERIODIC TEXTURES AND SYNTHETIC TEXTURES IN FIG. 7

Periodic texture	(a)	(b)	(c)	(d)	(e)	(f)
Entropy value	0.568	0.328	0.823	0.792	0.817	0.865
Synthetic texture	(g)	(h)	(i)	(j)	(k)	(l)
Entropy value	0.623	0.484	0.842	0.829	0.852	0.875

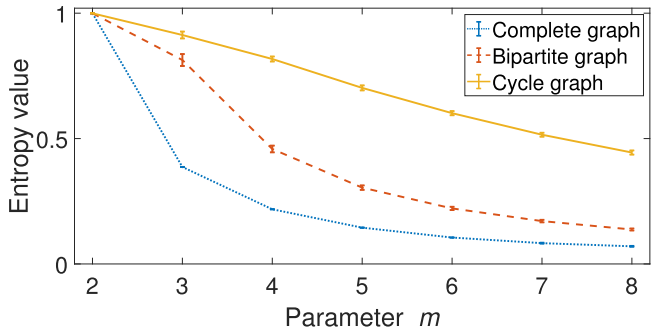


Fig. 8. Permutation entropy measures of the random signal $\mathbf{X}$ on several underlying graphs G on 500 vertices.

included synthetic signals and real-world 2D data. $\mathrm{PE_G}$ offers similar performance to standard 2D algorithms in the literature. Our algorithm uses the spatial structure but does not depend on the regularity of the graph. Almost all the methods in 2D depend on the regularity of the grid, and are not straightforward apply to irregular domains. We can apply $\mathrm{PE_G}$ to any graph signal as the next section shows.

C. Permutation Entropy for Signals on Graphs

Here, the signal $\mathbf{X} = \{x_i\}_{i=1}^{N}$ will be white Gaussian noise. The entropy has different values depending on the graph underlying, even if $\mathbf{X}$ remains the same.

1) Random Signals on Different Graph Structures: We show that the algorithm is able to distinguish different values of regularity degree of the graph.

We considerer several graph structures G on N vertices with $\mathbf{X}$ as a graph signal: G_1 the cycle graph, G_2 the complete bipartite graph with bipartition V_1 and V_2 where $|V_1| = |V_2| = N/2$, and G_3 the complete graph. Considering $N = 500$, we generate 20 realisations of signal $\mathbf{X}$ and we compute $\mathrm{PE_G}$ for $G = G_1, G_2$ and G_3 for different embedding values m. In Fig. 8, we show the mean and standard deviation for dimensions $2 \leq m \leq 8$.

Observe that G_1, G_2 and G_3 are 2, $N/2$ and $N-1$ regular graphs, respectively. The analysis shows the random signal $\mathbf{X}$ on the cycle graph cycle G_1 has larger entropy values than the same signal on the complete graph G_3. In general, the entropy value for the signal decreases as the degree of regularity of the graph increases. Formally, denote by $G(k, N)$ a k-regular graph on N vertices, then: $\lim_{m \to N} \mathrm{PE}_{G(k,N)}(m) = 0$ and a larger value of k increases the convergence ratio. Hence $\mathrm{PE}_{G(k_1,N)}(m) < \mathrm{PE}_{G(k_2,N)}(m)$ for all m and $k_2 \ll k_1$. Then, for the same signal, the algorithm is able to detect different degrees regularity on the graph structure.

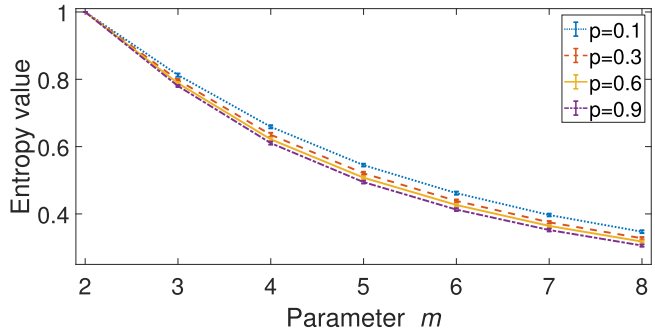


Fig. 9. Erdő-Rényi model for values p equal to 0.1, 0.3, 0.6 and 0.9 and $N = 2000$. Mean value of its PE_{G} and standard deviation for 20 simulations.

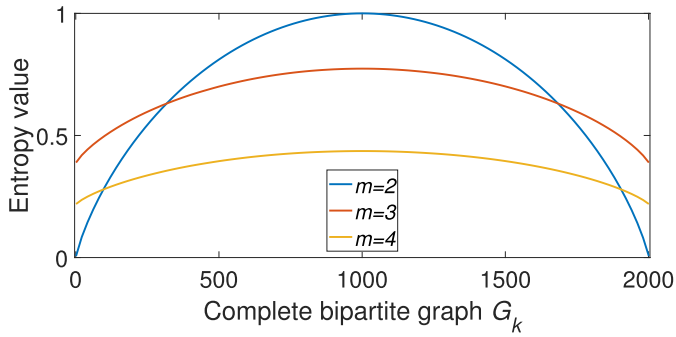


Fig. 10. Permutation entropy values of the random signal $\mathbf{X}$ on the complete bipartite graph G on k and $2000 - k$ vertices.

2) Erdő-Rényi Graphs: The random graph model introduced by Erdő-Rényi (ER graphs) is parametrised by the number of vertices N, the probability p, and it is denoted by $G_{N,p}$. Therefore, ER graphs are models with random connections and they are used to represent common real-world data.

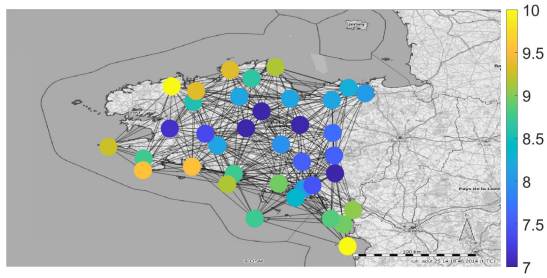
For a fixed $N = 2000$, we consider the ER graph $G_{N,p}$ for several values of $p = 0.1, 0.3, 0.6, 0.9$. For 20 realisations of the signal $\mathbf{X}$ we compute its $\mathrm{PE}_{G_{N,p}}$ for $2 \leq m \leq 7$. In Fig. 9, we show the mean and standard deviations. In all the cases (except for $m = 2$) there is no overlapping on the intervals. Then, for a fixed $m > 2$ a smaller value of p implies a smaller number of edges (less connectivity) and therefore a larger entropy value. Then $\mathrm{PE}_{G_{N,p1}} < \mathrm{PE}_{G_{N,p2}}(m)$ for all $m > 2$ and $p2 \ll p1$, and the algorithm is able to detect different connectivity degree on the graph structure.

3) Controlling the Entropy Value by Changing the Graph Topology: Any graph signal can be more regular/irregular depending on the topology graph. Let $\mathbf{X}$ be any signal with N points and consider the embedding dimension $m = 2$, for any $\alpha \in [0, 1]$ we will be able to construct a graph G such its entropy of the signal $\mathbf{X}$ is equal (or close enough depending on N) to α.

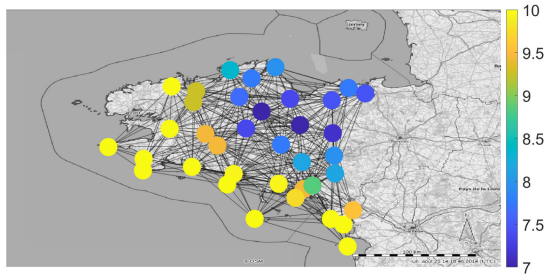
Let $1 \leq k \leq N - 1$, and without loss of generality, suppose that $x_1, x_2, \ldots, x_k$ are the k largest values from the signal $\mathbf{X}$. Consider G_k the complete bipartite graph with partition $A = \{1, 2, \ldots, k\}$ and $B = \{k + 1, k + 2, \ldots, N\}$. In this case, $PE_{G_k} = -\frac{N-k}{N}\ln(\frac{N-k}{N}) - \frac{k}{N}\ln(\frac{k}{N})$.

In particular, if $k = 1$ then G_1 is the star graph with centre on the vertex 1. The entropy $\mathrm{PE}_{G_1} = -\frac{N-1}{N}\ln(\frac{N-1}{N}) - \frac{1}{N}\ln(\frac{1}{N}) \to 0$ as $N \to \infty$. Then, we have constructed a graph structure with small entropy for the signal $\mathbf{X}$. Similarly, for N even, we consider $k = N/2$, the entropy of the signal $\mathbf{X}$ on the graph $G_{\frac{N}{2}}$ is $-\ln(\frac{1}{2})$ and its normalised entropy is equal to 1.

In Fig. 10, we show for $N = 2000$ the entropy of a signal random signal $\mathbf{X}$ with underlying graph the bipartite graph G_k for $1 \leq k \leq 1000$. Observe that permutation entropy for the graph G_k and G_{N-k} are equal because for the graph symmetry. Then, for $m = 2$ and any value $\alpha \in [0, 1]$ we construct a graph G_k with entropy equal to α (or close enough). Observe that this construction is optimised for $m = 2$, and the range of the entropy for larger dimensions is narrower. Similar constructions can be done to maximise or minimise the entropy for a fixed embedding dimension m.



(a) A temperature observation at 14:00 (January 27, 2014), denoted as signal $\mathbf{X}_1$.



(b) A temperature observation at 04:00 (January 23, 2014), denoted as signal $\mathbf{X}_2$.

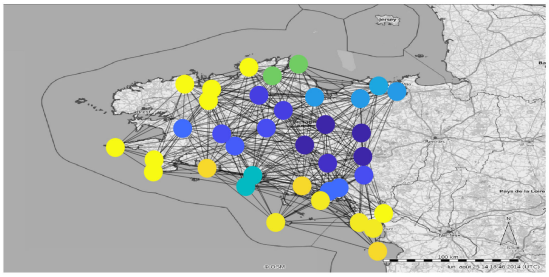
Fig. 11. Two different readings of temperatures in Brittany during January 2014.

D. A Real-World Data Example: Temperature Data

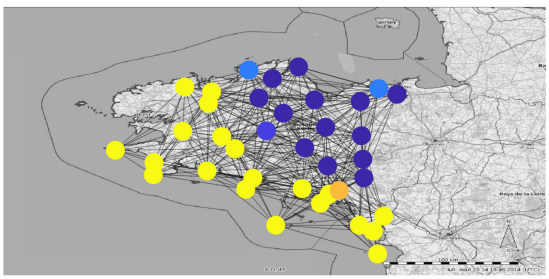
We use the temperature readings of ground stations observed in Brittany for January 2014 [43]. The graph is defined as follows: each vertex represents the ground station, and the weighted edges between vertices are given by a Gaussian kernel of the Euclidean distance between vertices [22]:

$$W_{ij} = \begin{cases} \exp\left(\frac{-d(i,j)^2}{2\sigma_1^2}\right) & \text{if } d(i,j) \leq \sigma_2 \\ 0 & \text{otherwise.} \end{cases}$$

Similarly to [43], we use $\sigma_1^2 = 5.1^8$ and $\sigma_2 = 10^5$. Let $\mathbf{X}_1$ be the signal corresponding to the temperature observation at 14:00 (January 27, 2014) shown in Fig. 11(a) and $\mathbf{X}_2$ the signal at 04:00 (January 23, 2014) shown in Fig. 11(b). The signal $\mathbf{X}_1$



(a) Patterns of the signal $\mathbf{X}_1$ for $m = 4$.



(b) Patterns of the signal $\mathbf{X}_2$ for $m = 4$.

Fig. 12. Each vertex corresponds to one pattern of the signal. Equal colours are equal patterns.

is more *irregular* than $\mathbf{X}_2$ in the following sense: In $\mathbf{X}_1$, the maximum/minimum values are more dispersed along with the map. In contrast, the extreme values are more grouped in $\mathbf{X}_2$. In Fig. 11(b), we see that the lowest temperatures are localised in the North-East, the highest in the South-West and the transition between ground stations are smother. While in Fig. 11(a), the distribution of the temperatures is more random. This fact is captured by the patterns computed for the entropy.

Formally, the irregularity is measured using $\mathrm{PE_G}$ in both signals. For example, consider $m = 4$, $L = 1$ for computing the entropy, which requires up to 24 permutation patterns. For each vertex, one pattern is generated ((5)). In the signal $\mathbf{X}_1$, 13 patterns appear with distribution $\{1, 1, 1, 2, 2, 2, 3, 3, 3, 3, 4, 4, 8\}$ (see Fig. 12(a)) and for the signal $\mathbf{X}_2$, 5 patterns are formed with distribution $\{1, 1, 2, 13, 20\}$ (see Fig. 12(b)). The $\mathrm{PE_G}$ of the signal $\mathbf{X}_1$ is 0.7529, while for $\mathbf{X}_2$ is 0.3313.

In general, the entropy values for the signal $\mathbf{X}_1$ are $0.9995, 0.8729, 0.7529, 0.5438$ for $m = 2, 3, 4, 5$ and for $\mathbf{X}_2$ are $0.9740, 0.5667, 0.3313, 0.2739$ for $m = 2, 3, 4, 5$. In all the cases, the entropy for $\mathbf{X}_2$ is smaller than $\mathbf{X}_1$ (this fact is true for $1 \leq L$).

In fact, the difference (and relations) of entropy values for 04:00 and 14:00 are preserved for all embedding dimensions m. We compute the entropy for all temperature measures (one per hour for 31 days). Table 2 shows the average of the entropy values for the temperature signals at 04:00 and at 14:00. The entropy value of the temperature measurement is higher at 14:00 than at 04:00. We speculate that this reflects a more irregular distribution of temperatures over the ground at 14:00 (during the early afternoon when temperatures could reach higher values

TABLE II
AVERAGE OF $\mathrm{PE_G}$ VALUES FOR THE TEMPERATURE MEASUREMENTS AT 04:00 AND 14:00

	$m = 2$	$m = 3$	$m = 4$	$m = 5$
Temperature at 04:00	0.969	0.830	0.572	0.421
Temperature at 14:00	0.973	0.849	0.618	0.443

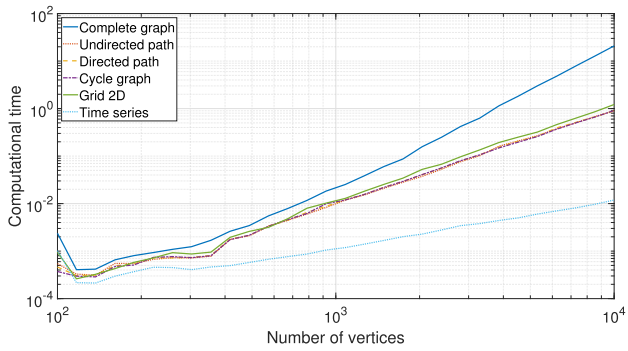


Fig. 13. Computational time of $\mathrm{PE_G}$ with different underlying graphs and signal lengths.

depending on the geographical situation of each station) than at 4:00 in the middle of the night.

In this example, we see how the performance of $\mathrm{PE_G}$ is different from the smoothness signal definition in (9). While the smoothness of the $\mathbf{X}_1$ is 221.9 and for $\mathbf{X}_2$ is 138.2, i.e. is smoother $\mathbf{X}_1$ than $\mathbf{X}_2$, visually it seems more *regular* $\mathbf{X}_2$ than $\mathbf{X}_1$. Recall that we are interested in the change of the pattern and the smoothness in the change of values (Proposition 2 and (9)).

E. Computational Cost

In order to assess the computational time of $\mathrm{PE_G}$ compared with the classical PE, we use random values with different lengths, changing from 100 to 10,000 sample points. The optimised implementation of PE [44] was used in this article. We treat a time series as a signal on a path graph, and we use the adjacency matrix to compute $\mathrm{PE_G}$ with embedding dimension $m = 2$, time delay $L = 1$. Our algorithm can deal with diverse graph topologies. Thus, we consider several underlying graphs with random signals $\mathbf{X}$: a complete graph, a cycle, a path, and the 2D grid (see Fig. 5(a)). The results are depicted in Fig. 13. The simulations were carried out using a PC with Intel(R) Core(TM) i7 with MATLAB R2021a.

Our algorithm is more computationally expensive than the classical PE. However, this is expected as $\mathrm{PE_G}$ does not assume any a priori information about the structure of the domain where the signal was sampled. We note that the computational time increases with the number of vertices and edges in the graph. The results show that the computational times of the cycle, the 2D grid, and the path (undirected and directed) are very close because the number of edges increases linearly with the number of vertices. For the complete graph, the number of edges increases quadratically with the number of vertices. Thus, the number of non-zero entries on the adjacency matrix (used for $\mathrm{PE_G}$) also increases quadratically, leading to a computational

cost of $O(N^2)$. We postulate, though, that more efficient implementations for $\mathrm{PE_G}$ can be done for special kinds of graphs using, for example, the graph's symmetries (as 1D and 2D graphs) or when its adjacency matrix is symmetric (undirected graphs). However, these optimisations are beyond the scope of this paper. Here we report the cost of the general algorithm for either directed or undirected unweighted graphs.

V. Conclusion and Future Work

In this paper, we generalised the permutation entropy for graph signals. Some modifications and extensions of the classical PE have been developed in the literature [7], [13], [15]. However, we introduce for the first time an entropy measure for signals on general irregular domains defined by graphs. In particular, we observe that by considering the underlying graph G as a path (1D), the results of $\mathrm{PE_G}$ coincide with the results for standard algorithms on time series (as the original PE [1]). Moreover, our graph algorithm also enables applying PE-related analysis to images (2D).

We also observe that the results depend on how much information we have about the underlying graph. Weights or directions on the edges give different kinds of relations between the signals at neighbouring vertices captured by the algorithm.

We explore how the same signal changes its entropy depending on the topology of the graph and how the same underlying graph with signals with different dynamics has different $\mathrm{PE_G}$. It demonstrates the importance of the signal and graph for computing the entropy values.

Some future lines of research are the following:

- Extend other entropy metrics (e.g., dispersion entropy [9]) to irregular domains, including multivariate signals [45].
- Generate surrogate graph signals and test the nonlinearlity of the signals defined on the graph.
- Study the relationship between properties of the graph (for example, the spectrum of the graph Laplacian), and the regularity of the signal. This would also be useful to help determine how to define the graph for a given graph signal that would be subject to entropy analysis.

We expect the algorithm presented in this paper to enable the extension of similar techniques that inspect nonlinear dynamics from data acquired over irregular graphs.

The MATLAB code used in this paper is freely available at https://github.com/JohnFabila/PEG.

Acknowledgment

We thank the anonymous reviewers whose comments helped improve and clarify this manuscript.

References

[1] C. Bandt and B. Pompe, "Permutation entropy: A natural complexity measure for time series," *Phys. Rev. Lett.*, vol. 88, no. 17, 2002, Art. no. 174102.

[2] Y. Cao, W. W. Tung, J. B. Gao, V. A. Protopopescu, and L. M. Hively, "Detecting dynamical changes in time series using the permutation entropy," *Phys. Rev. E*, vol. 70, no. 4, 2004, Art. no. 046217.

[3] E. Olofsen, J. W. Sleigh, and A. Dahan, "Permutation entropy of the electroencephalogram: A measure of anaesthetic drug effect," *BJA, Brit. J. Anaesth.*, vol. 101, no. 6, pp. 810–821, 2008.

[4] R. Yan, Y. Liu, and R. X. Gao, "Permutation entropy: A nonlinear statistical measure for status characterization of rotary machines," *Mech. Syst. Signal Process.*, vol. 29, pp. 474–484, 2012.

[5] L. Zunino, M. Zanin, B. M. Tabak, D. G. Pérez, and O. A. Rosso, "Forbidden patterns, permutation entropy and stock market inefficiency," *Physica A, Stat. Mechanics Appl.*, vol. 388, no. 14, pp. 2854–2864, 2009.

[6] H. Azami and J. Escudero, "Improved multiscale permutation entropy for biomedical signal analysis: Interpretation and application to electroencephalogram recordings," *Biomed. Signal Process. Control*, vol. 23, pp. 28–41, 2016.

[7] Z. Chen, Y. Li, H. Liang, and J. Yu, "Improved permutation entropy for measuring complexity of time series under noisy condition," *Complexity*, 2019, pp. 1–12.

[8] C. Bian, C. Qin, Q. D. Ma, and Q. Shen, "Modified permutation-entropy analysis of heartbeat dynamics," *Phys. Rev. E*, vol. 85, no. 2, 2012, Art. no. 021906.

[9] H. Azami and J. Escudero, "Amplitude- and fluctuation-based dispersion entropy," *Entropy*, vol. 20, no. 3, pp. 1–21, 2018.

[10] M. Rostaghi and H. Azami, "Dispersion entropy: A measure for time-series analysis," *IEEE Signal Process. Lett.*, vol. 23, no. 5, pp. 610–614, May 2016.

[11] J. H. Martínez, J. L. Herrera-Diestra, and M. Chavez, "Detection of time reversibility in time series by ordinal patterns analysis," *Chaos, Interdiscipl. J. Nonlinear Sci.*, vol. 28, no. 12, 2018, Art. no. 123111.

[12] M. Zanin, A. Rodríguez-González, E. Menasalvas Ruiz, and D. Papo, "Assessing time series reversibility through permutation patterns," *Entropy*, vol. 20, no. 9, 2018, Art. no. 665.

[13] C. Morel and A. Humeau-Heurtier, "Multiscale permutation entropy for two-dimensional patterns," *Pattern Recognit. Lett.*, vol. 150, pp. 139–146, 2021.

[14] L. E. V. Silva, A. C. S. Senra Filho, V. P. S. Fazan, J. C. Felipe, and L. M. Junior, "Two-dimensional sample entropy: Assessing image texture through irregularity," *Biomed. Phys. Eng. Exp.*, vol. 2, no. 4, 2016, Art. no. 045002.

[15] H. Azami, L. E. V. da Silva, A. C. M. Omoto, and A. Humeau-Heurtier, "Two-dimensional dispersion entropy: An information-theoretic method for irregularity analysis of images," *Signal Processing, Image Commun.*, vol. 75, pp. 178–187, 2019.

[16] H. Azami, J. Escudero, and A. Humeau-Heurtier, "Bidimensional distribution entropy to analyze the irregularity of small-sized textures," *IEEE Signal Process. Lett.*, vol. 24, no. 9, pp. 1338–1342, Sep. 2017.

[17] D. Lebret *et al.*, "Three-dimensional dispersion entropy for uterine fibroid texture quantification and post-embolization evaluation," *Comput. Methods Programs Biomed.*, vol. 215, 2022, Art. no. 106605.

[18] A. S. F. Gaudencio *et al.*, "Three-dimensional multiscale fuzzy entropy: Validation and application to idiopathic pulmonary fibrosis," *IEEE J. Biomed. Health Informat.*, vol. 25, no. 1, pp. 100–107, Jan. 2021.

[19] A. Ortega, P. Frossard, J. Kovačević, J. M. Moura, and P. Vandergheynst, "Graph signal processing: Overview, challenges, and applications," *Proc. IEEE*, vol. 106, no. 5, pp. 808–828, 2018.

[20] L. Stanković, D. Mandic, M. Dakovic, M. Brajovic, B. Scalzo, and T. Constantinides, "Data Analytics on Graphs Part I: Graphs and Spectra on Graphs," *FNT Mach. Learn.*, vol. 13, no. 1, pp. 1–157, 2020.

[21] D. I. Shuman, S. K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Process. Mag.*, vol. 30, no. 3, pp. 83–98, May 2013.

[22] D. I. Shuman, B. Ricaud, and P. Vandergheynst, "Vertex-frequency analysis on graphs," *Appl. Comput. Harmon. Anal.*, vol. 40, no. 2, pp. 260–291, 2014.

[23] E. Pirondini, A. Vybornova, M. Coscia, and D. Van De Ville, "A spectral method for generating surrogate graph signals," *IEEE signal Process. Lett.*, vol. 23, no. 9, pp. 1275–1278, Sep. 2016.

[24] L. Han, F. Escolano, E. R. Hancock, and R. C. Wilson, "Graph characterizations from von neumann entropy," *Pattern Recognit. Lett.*, vol. 33, no. 15, pp. 1958–1967, 2012.

[25] F. Passerini and S. Severini, "The Von Neumann entropy of networks," 2008. [Online]. Available: https://arxiv.org/abs/0812.2597

[26] J. S. Fabila-Carrasco, F. Lledó, and O. Post, "Spectral gaps and discrete magnetic laplacians," *Linear Algebra Appl.*, vol. 547, pp. 183–216, 2018.

[27] S. He, K. Sun, and H. Wang, "Multivariate permutation entropy and its application for complexity analysis of chaotic systems," *Physica A, Stat. Mechanics Appl.*, vol. 461, pp. 812–823, 2016.

[28] A. Myers and F. A. Khasawneh, "On the automatic parameter selection for permutation entropy," *Chaos*, vol. 30, no. 3, 2020, Art. no. 033130.

[29] D. J. Little and D. M. Kane, "Permutation entropy with vector embedding delays," *Phys. Rev. E*, vol. 96, no. 6, pp. 1–8, 2017.
[30] M. Tao, K. Poskuviene, N. F. Alkayem, M. Cao, and M. Ragulskis, "Permutation entropy based on non-uniform embedding," *Entropy*, vol. 20, no. 8, pp. 1–13, 2018.
[31] X. Wang, S. Si, Y. Wei, and Y. Li, "The optimized multi-scale permutation entropy and its application in compound fault diagnosis of rotating machinery," *Entropy*, vol. 21, no. 2, 2019, Art. no. 170.
[32] J. Xia, Jianan, P. Shang, J. Wang, and W. Shi, "Permutation and weighted-permutation entropy analysis for the complexity of nonlinear time serie," *Commun. Nonlinear Sci. Numer. Simul.*, vol. 31, no. 1-3, pp. 60–68, 2016.
[33] B. Fadlallah, B. Chen, A. Keil, and J. Principe, "Weighted-permutation entropy: A complexity measure for time series incorporating amplitude information," *Phys. Rev. E*, vol. 87, no. 2, 2013, Art. no. 022911.
[34] H. Azami and J. Escudero, "Amplitude-aware permutation entropy: Illustration in spike detection and signal segmentation," *Comput. Methods Programs Biomed*, vol. 128, pp. 40–51, 2016.
[35] S. Chen, P. Shang, and Y. Wu, "Weighted multiscale rènyi permutation entropy of nonlinear time series," *Physica A, Stat. Mechanics Appl.*, vol. 496, pp. 548–570, 2018.
[36] M. Landauskas, M. Cao, and M. Ragulskis, "Permutation entropy-based 2D feature extraction for bearing fault diagnosis," *Nonlinear Dyn.*, vol. 102, no. 3, pp. 1717–1731, 2020.
[37] D. Cuesta–Frau, M. Varela–Entrecanales, A. Molina–Picó, and B. Vargas, "Patterns with equal values in permutation entropy: Do they really matter for biosignal classification," Complexity, vol. 2018, 2018, Art. no. 1324696.
[38] J. S. Fabila-Carrasco, "The discrete magnetic Laplacian: Geometric and spectral preorders with applications," PhD dissertation, Universidad Carlos III de Madrid, 2020. [Online]. Available: https://e-archivo.uc3m.es/handle/10016/31372
[39] J. S. Fabila-Carrasco, F. Lledó, and O. Post, "Spectral preorder and perturbations of discrete weighted graphs," *Mathematische Annalen*, 2020, pp. 1–49.
[40] C. Zhang, "Period three begins," *Math. Mag.*, vol. 83, no. 4, pp. 295–297, 2010.
[41] L. Saunoriene, M. Ragulskis, J. Cao, and M. A. Sanjuan, "Wada index based on the weighted and truncated Shannon entropy," *Nonlinear Dyn.*, vol. 104, pp. 739–751, 2021.
[42] Graphics.stanford.edu, "Texture Analysis and Synthesis," 2013. [Online]. Available: https://graphics.stanford.edu/projects/texture/demo/synthesis_eero.html
[43] B. Girault, "Stationary graph signals using an isometric graph translation," in *Proc. 23rd Eur. Signal Process. Conf.*, 2015, pp. 1516–1520.
[44] X. Li, G. Ouyang, and D. A. Richards, "Predictability analysis of absence seizures with permutation entropy," *Epilepsy Res.*, vol. 77, no. 1, pp. 70–74, 2007.
[45] J. S. Fabila-Carrasco, C. Tan, and J. Escudero, "Multivariate permutation entropy, a cartesian graph product approach," *arXiv:2203.00550*. [Online]. Available: https://arxiv.org/abs/2203.00550