Dispersion Entropy for Graph Signals

Chaos, Solitons and Fractals 175 (2023) 113977

Contents lists available at ScienceDirect

Chaos, Solitons and Fractals

journal homepage: www.elsevier.com/locate/chaos

Dispersion entropy for graph signals

John Stewart Fabila-Carrasco [a,*], Chao Tan [b], Javier Escudero [a]

[a] School of Engineering, Institute for Digital Communications, University of Edinburgh, West Mains Rd, Edinburgh, EH9 3FB, Scotland, UK
[b] School of Electrical and Information Engineering, Tianjin University, Tianjin 300072, China

ARTICLE INFO

Keywords:
Entropy metrics
Graph signals
Dispersion entropy
Graph topology
Nonlinear analysis

ABSTRACT

We present a novel method, called Dispersion Entropy for Graph Signals, DE_G, as a powerful tool for analysing the irregularity of signals defined on graphs. DE_G generalizes the classical dispersion entropy concept for univariate time series, enabling its application in diverse domains such as image processing, time series analysis, and network analysis. Furthermore, DE_G establishes a theoretical framework that provides insights into the irregularities observed in graph centrality measures and in the spectra of operators acting on graphs. We demonstrate the effectiveness of DE_G in detecting changes in the dynamics of signals defined on both synthetic and real-world graphs, by defining a mix process on random geometric graphs or those exhibiting small-world properties. Our results indicate that DE_G effectively captures the irregularity of graph signals across various network configurations, successfully differentiating between distinct levels of randomness and connectivity. Consequently, DE_G provides a comprehensive framework for entropy analysis of various data types, enabling new applications of dispersion entropy not previously feasible, and uncovering nonlinear relationships between graph signals and their graph topology.

1. Introduction

The field of data analysis has significantly benefited from the evolution of entropy-based measures [1], vital tools for assessing irregularities and nonlinear behaviours in data [2,3]. These measures have a broad spectrum of applications, spanning across various fields such as finance [4,5], biology [6,7], industrial process [8,9] and even conflicts and international events [10] to name a few. Among these entropy-based measures, Dispersion Entropy (DE) has carved a unique space for itself due to its distinct properties and effectiveness [11] .

Serving as a robust algorithm, DE is specifically designed to capture intricate dynamics within one-dimensional time series data. Its standout characteristic is the dual consideration of both the order and amplitude of data points, offering a comprehensive perspective on the system's dynamics [11,12].

DE boasts several distinct advantages. It is capable of identifying changes in noise bandwidth as well as concurrent shifts in frequency and amplitude, making it a highly adaptable tool for signal analysis [11]. Its superior performance has been demonstrated in applications on real-valued signals, where it outperforms other entropy measures in distinguishing different groups within datasets [13]. Furthermore, it exhibits high computational efficiency, necessitating significantly less computation time compared to several other entropic measures [14]. This property makes it especially suited for processing large or complex datasets.

Owing to its versatility and efficiency, DE has been employed in various domains. These include EEG analysis [13], diagnosis and monitoring of rotary machines [9], and fault detection in rolling bearings [15]. With the increasing complexity and volume of data, the utility of DE continues to grow, making it a tool of choice in diverse applications.

The ongoing expansion of complex data recorded in a distributed way over networks, including transportation systems [16], social and web networks [17], has led to growing interest in broadening the scope of entropy metrics beyond univariate time series to encompass more general domains. There have been recent advancements in extending entropy measures to analyse images (2D data) [18] and irregular domains such as graphs [19]. Despite the improvements made in DE for 1D and 2D data [12,20], a void remains in its application for data defined on graphs. Bridging this gap would allow us to analyse the nonlinear behaviour of real-world systems with graph-based structures, where the conventional DE was not previously applicable. This would provide a powerful framework for data analysis across a broad range of applications in the field of Graph Signal Processing (GSP) [16].

Smoothness is a fundamental property extensively studied in GSP [16,21,22], typically through the use of the combinatorial Laplacian's quadratic form. Intuitively, a graph signal is considered smooth if connected vertices exhibit similar values [21]. Nonetheless, this definition may not fully capture the complex dynamics of graph signals due to its

* Corresponding author.
E-mail address: John.Fabila@ed.ac.uk (J.S. Fabila-Carrasco).

https://doi.org/10.1016/j.chaos.2023.113977
Received 12 May 2023; Received in revised form 31 July 2023; Accepted 24 August 2023
Available online 15 September 2023

relationship with the spectrum [23]. To address this limitation, here we propose a novel method, enabled by our new method DE_G to assess irregularity in graph signals, which effectively captures the irregularity of graph signals, providing critical insights into the underlying graph structure and data.

To evaluate our method's performance, we employ synthetic and real-world graphs, including random geometric graphs (used to model wireless sensor networks [24]) and small-world networks (observed widely in biological systems [25], social networks [26], and complex systems [27]). In our analysis, we generalize the mix process – a stochastic process combining a sinusoidal signal with random dynamics controlled by the parameter $p \in [0, 1]$ – to the setting of graph signals. This process has been employed to assess the performance of various entropy metrics in time series [14,28] and images [29]. Moreover, we analyse centrality measures, which assign ranking values to the graph's vertices based on their position or importance within the graph. Centrality measures play a crucial role in social network analysis for evaluating the importance of vertices in communication [30,31].

Contribution

In this paper, we propose a method for defining Dispersion Entropy for Graph Signals, denoted as DE_G. Our approach generalizes the classical univariate definition of DE by incorporating topological information through the adjacency matrix. We demonstrate the effectiveness of DE_G on synthetic and real-world datasets, and characterize the relationship between graph topology and signal dynamics. Our results indicate that DE_G is a promising technique for analysing graph data, holding potential for numerous applications in fields such as biomedicine and social sciences.

2. Background and notation

This section aims to provide the foundational background and set the notations that are used throughout the remainder of this study. We also present a brief overview of the classical Dispersion Entropy method for univariate time series.

2.1. Notation

Let $G = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ denote a *simple undirected graph*, where $\mathcal{V}$ denotes a finite set of vertices $\mathcal{V} = \{v_1, v_2, v_3, \dots, v_N\}$, such that no isolated vertices exist. The edge set $\mathcal{E} \subset \{(v_i, v_j) \mid v_i, v_j \in \mathcal{V}\}$ contains no multiple edges or self-loops. The adjacency matrix $\mathbf{A}$ is a symmetric $N \times N$ matrix that reflects the topology of the graph, such that $\mathbf{A}_{ij} = \mathbf{A}_{ji} = 1$ if an edge exists between (v_i, v_j), i.e., if $(v_i, v_j) \in \mathcal{E}$ and $\mathbf{A}_{ij} = \mathbf{A}_{ji} = 0$ otherwise.

The degree matrix $\mathbf{D}$ is a diagonal matrix where each diagonal entry $\mathbf{D}_{ii}$ is the degree of the vertex v_i. The combinatorial Laplacian matrix of the graph, Δ, is defined by $\Delta = \mathbf{D} - \mathbf{A}$ and the normalized Laplacian, $\mathbf{L}$, is defined as $\mathbf{L} = \mathbf{D}^{-1/2}\Delta\mathbf{D}^{-1/2}$. These matrices serve as key tools in the spectral analysis of graphs [32,33].

In the context of this study, a *graph signal* is a real mapping $\mathbf{X} : \mathcal{V} \to \mathbb{R}$. For computational purposes, it is convenient to represent the graph signal $\mathbf{X}$ as an N-dimensional column vector in $\mathbb{R}^{N\times 1}$, written as $\mathbf{X} = \left[x_1, x_2, \dots, x_N\right]^T$, where the indexing corresponds to the vertices.

2.2. Example

Consider the graph G shown in the left panel of Fig. 1, composed of vertices

$$\mathcal{V} = \{v_1, v_2, v_3, v_4, v_5, v_6, v_7, v_8\} ;$$

and edges $\mathcal{E}$ as indicated by the connecting lines.

The graph signal $\mathbf{X} : \mathcal{V} \to \mathbb{R}$ maps each vertex $v_i \in \mathcal{V}$ to a real number $x_i \in \mathbb{R}$, resulting in $\mathbf{X} = \left[x_1, x_2, \dots, x_8\right]^T$, an 8×1 column vector. The signal values are visualized using red lines for positive values and blue lines for negative values as shown in Fig. 1.

In the middle panel of Fig. 1, we present the adjacency matrix $\mathbf{A}$ of the graph G. This symmetric matrix, with dimensions $N \times N$ (in this case, 8×8), has entries $\mathbf{A}_{ij} = \mathbf{A}_{ji}$ that are set to 1 if an edge exists between v_i and v_j, and 0 otherwise

2.3. The classical dispersion entropy for time series

Dispersion Entropy (DE) is an important tool used for analysing time series data [13]. The wide array of applications and the robustness of the DE method make it an essential technique for time series data analysis. More importantly, it has proven to be an effective tool for handling nonlinear and non-stationary data, which are common in real-world applications. We provide a concise step-wise explanation of the DE method here, with a detailed mathematical derivation available in the Appendix A:

1. The samples of the signal are discretized into different classes according to the signal values, resulting in a classified signal.
2. The classified signal is scanned looking for patterns, composed of a series of samples given an embedding dimension and delay time. Each pattern corresponds to a unique dispersion pattern.
3. The occurrence frequency of each potential dispersion pattern is calculated, indicating the dominance of certain patterns in the time series.
4. The DE is computed based on these frequencies using the concept of Shannon entropy, hence offering a measure of the complexity or irregularity in the time series.

3. Dispersion Entropy for Graph Signals (DE_G)

The Dispersion Entropy for Graph Signals (DE_G) extends the classical method to graph signals, adapting the entropy calculation to the underlying graph structure, providing unique insights into the complexity and interconnectedness of graph-based data.

3.1. The algorithm

Let $\mathbf{X}$ be a graph signal defined on G, $2 \le m \in \mathbb{N}$ be the *embedding dimension*, $l \in \mathbb{N}$ be the *delay time* and $c \in \mathbb{N}$ be the *class number*. The DE_G is defined as follows:

1. The *embedding matrix* $\mathbf{Y} \in \mathbb{R}^{N\times m}$ is given by $\mathbf{Y} = [\mathbf{y}_0, \mathbf{y}_1, \dots, \mathbf{y}_{m-1}]$, defined by
$$\mathbf{y}_k = D^{kl}\mathbf{A}^{kl}\mathbf{X} \in \mathbb{R}^{N\times 1} , \quad k = 0, 1, \dots, m-1 , \tag{1}$$
where D^{kl} is the diagonal matrix $D^{kl}_{ii} = 1/\sum_{j=1}^{N}(\mathbf{A}^{kl})_{ij}$.
2. *Map function.* Each entry of the embedding matrix $\mathbf{Y}$ is mapped to an integer number from 1 to c, called a class. The function $F : \mathbb{R} \to \mathbb{N}_c$ where $\mathbb{N}_c = \{1, 2, \dots, c\}$ is applied element-wise on the matrix $\mathbf{Y}$, i.e. $F(\mathbf{Y}) \in \mathbb{N}_c^{N\times m}$ where $F(\mathbf{Y})_{ij} = F(y_{ij})$.
3. *Dispersion patterns.* Each vertex of the graph is assigned an *embedding vector* and mapped to a unique dispersion pattern. Formally, the *embedding vectors* consist of m integer numbers (ranged from 1 to c) corresponding to each row of the matrix $F(\mathbf{Y})$, i.e., $\mathrm{row}_i(F(\mathbf{Y})) = \left(F(y_{ij})\right)_{j=1}^{m}$ for $i = 1, 2, \dots, N$. The set of dispersion patterns is $\Pi = \{\, \pi_{u_1u_2\dots u_m} \mid u_i \in \mathbb{N}_c \,\}$. Each embedding vector (one for each vertex of the graph) is uniquely mapped to a dispersion pattern, i.e., $v_i \to \mathrm{row}_i(F(\mathbf{Y})) \to \pi_{u_1u_2\dots u_m}$ where $u_1 = F(y_{i1}), u_2 = F(y_{i2}), \dots, u_m = F(y_{im})$.
4. *Relative frequencies.* For each dispersion pattern $\pi \in \Pi$, its relative frequency is obtained as:
$$p(\pi) = \frac{|\{v_i \mid v_i \in \mathcal{V} \text{ and } v_i \text{ has type } \pi\}|}{N} , \tag{2}$$
where $||$ represents cardinality.

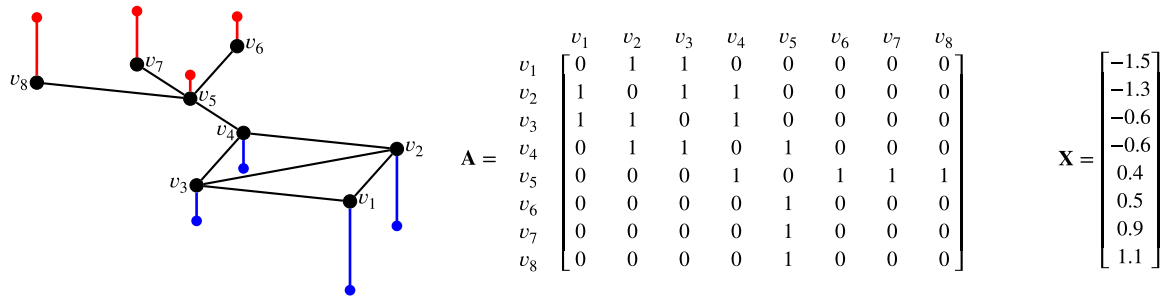


Fig. 1. Representation of a simple undirected graph, its adjacency matrix, and associated graph signal. Left: A simple undirected graph G, with vertices v_i linked by edges. The graph signal $\mathbf{X}$ is shown by vertical lines, each representing the value of the function $\mathbf{X}$ at the corresponding vertex. Centre: Adjacency matrix $\mathbf{A}$, an $N \times N$ symmetric matrix showcasing the connectivity between vertices. Right: Vectorized form of the graph signal $\mathbf{X}$, translating the function $\mathbf{X} : \mathcal{V} \to \mathbb{R}$ into an N-dimensional vector.

5. The *Dispersion Entropy for Graph Signals* $\mathrm{DE_G}$ is computed as the normalized Shannon's entropy for the distinct dispersion patterns as follows:

$$\mathrm{DE_G}(\mathbf{X}, m, l, c) = -\frac{1}{\ln(c^m)} \sum_{\pi \in \Pi} p(\pi) \ln p(\pi) \,. \tag{3}$$

3.2. Properties

The $\mathrm{DE_G}$ algorithm offers several unique features and properties.

The *embedding matrix* $\mathbf{Y} \in \mathbb{R}^{N \times m}$ is a critical component encapsulating the intricate topological relationships between the graph and signal. Typically, an embedding dimension is chosen within the range $3 \le m \le 7$, with the delay time often set to $l = 1$, as suggested in [2].

Eq. (1) plays a pivotal role in the algorithm, not only because it intertwines the graph's topology and the signal's values, but also due to the fact that its simple and efficient implementation belies its profound geometric implications.

Each column vector $\mathbf{y}_k$ is calculated by averaging the signal values of the neighbouring vertices. The first column of the matrix $\mathbf{Y}$ corresponds to the original graph signal, i.e., $\mathbf{y}_0 = \mathbf{X}$. For $k = 1$, the Eq. (1) simplifies to $\mathbf{y}_1 = D\mathbf{A}\mathbf{X}$, where $D_{ii} = 1/\sum_{j=1}^{N}(\mathbf{A})_{ij} = deg(v_i)$, and $deg(v_i)$ denotes the number of edges connected to the vertex v_i. This results in:

$$(D\mathbf{A}\mathbf{X})_i = \frac{1}{deg(v_i)} \sum_{(v_i, v_j) \in \mathbf{E}} \mathbf{X}_j \,, \quad i = 1, 2, \ldots, N \,.$$

This demonstrates that the second column associated with the vertex v_i is the average of the graph signal's values of the vertices connected to v_i. Consequently, the second column is related to the normalized Laplacian L, more specifically, $\mathbf{y}_1 = \mathbf{X} - L\mathbf{X}$.

In a more general context, for $\mathbf{y}_k = D^{kl}\mathbf{A}^{kl}\mathbf{X}$, a similar interpretation applies. The exponent of the adjacency matrix $\mathbf{A}^{kl}$ indicates the number of kl-walks between two vertices. That is, the element $(\mathbf{A}^{kl})_{ij}$ equals $|W_{kl}(v_i, v_j)|$, where

$$W_{kl}(v_i, v_j) = \{\text{Walks of length } kl \text{ from vertex } v_i \text{ to vertex } v_j\} \,.$$

Hence, we have:

$$(D^{kl}\mathbf{A}^{kl}\mathbf{X})_i = \frac{\sum_{j=1}^{N} |W_{kl}(v_i, v_j)| \mathbf{X}_j}{\sum_{j=1}^{N} |W_{kl}(v_i, v_j)|} \,, \quad i = 1, 2, \ldots, N \,.$$

Here, the numerator is the weighted signal with respect to the number of walks between the vertices, and the denominator serves as a normalization factor. This factor ensures that computations are properly balanced and scaled, which is crucial for accurate graph-signal analysis.

Map functions. To address limitations in assigning the signal $\mathbf{X}$ to only a limited number of classes, various map functions $F : \mathbb{R} \to \mathbb{N}_c$ have been proposed [11]. The Normal Cumulative Distribution Function (NCDF) is commonly utilized [34]. The map $G : (0, 1) \to \mathbb{N}_c$ is defined as $G(x) = round(cx + 0.5)$, where rounding increases or decreases a number to the nearest integer. The map $\mathrm{NCDF} : \mathbb{R} \to (0, 1)$ is defined as:

$$\mathrm{NCDF}(x) = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{x} e^{\frac{-(t-\mu)^2}{2\sigma^2}} \mathrm{d}t \,, \tag{4}$$

where μ and σ represent the mean and standard deviation of $\mathbf{X}$, respectively. Thus, $F = G \circ \mathrm{NCDF} : \mathbb{R} \to \mathbb{N}_c$ is the map function used in our implementation of the $\mathrm{DE_G}$ algorithm.

Dispersion patterns. The number of possible dispersion patterns that can be assigned to each embedding vector is c^m. Moreover, the number of embedding vectors constructed in the $\mathrm{DE_G}$ algorithm is N, one for each vertex. In contrast, classical DE has a number of embedding vectors dependent on the parameters m and l, specifically, $n - (m-1)l$.

The *normalized Shannon's entropy* provides a measure of irregularity that can be used to compare signals defined on different graphs. The value of this normalized entropy ranges from 0 (regular behaviour) to 1 (irregular behaviour). It is noteworthy to clarify that the usual Shannon's entropy given by $-\sum_{\pi \in \Pi} p(\pi) \ln p(\pi)$ takes values between 0 to $\ln(c^m)$, where c^m represents the number of potential dispersion patterns. Therefore, by normalizing the Shannon's entropy with $\ln(c^m)$, we ensure that the entropy values are scaled to fall in the interval $[0, 1]$.

Table 1 summarizes the main parameters of the $\mathrm{DE_G}$ algorithm, providing a clear overview of their role in the computation process and the typical values used.

3.3. Example

We exemplify the utilization of $\mathrm{DE_G}$ with the graph signal $\mathbf{X}$ and graph G introduced in Section 2.2 (also illustrated in Fig. 1). We establish the class number $c = 3$, the embedding dimension $m = 2$, and the delay time $l = 1$, and undertake the following sequence of steps:

Step 1: We initially compute the embedding matrix $\mathbf{Y} \in \mathbb{R}^{8 \times 2}$, as defined in Eq. (1). The resultant matrix is shown in Fig. 2(a).

Step 2: We apply the map function F to each entry of the normalized matrix, using the Normal Cumulative Distribution Function (NCDF) as formulated in Eq. (4), where $\mu(\mathbf{X}) = 0.11$ and $\sigma(\mathbf{X}) = 0.44$. The map function F transforms the entries into an integer range from 1 to $c = 3$, resulting in the matrix illustrated in Fig. 2(b).

Step 3: Subsequently, we map each row of $F(\mathbf{Y})$ to a distinctive dispersion pattern. Given the parameters c and m, there are $c^m = 3^2 = 9$ possible dispersion patterns. These patterns are presented in Fig. 2(c).

Step 4: We compute the relative frequencies of each dispersion pattern, utilizing Eq. (2), and display the results in Fig. 2(d).

Step 5: In the final step, we compute $\mathrm{DE_G}$ using Eq. (3), which yields the following outcome:

$$\mathrm{DE_G}(\mathbf{X}, m, l, c) = -\frac{1}{\ln(9)} \sum_{\pi \in \Pi} p(\pi) \ln p(\pi) = 0.4434 \,.$$

The $\mathrm{DE_G}$ value encapsulates the regularity of the signal propagated across the graph. As is apparent, the graph is primarily dichotomous:

Table 1
Summary of parameters used in DE_G algorithm.

Category	Symbol	Description	Typical values or expression	Reference
Notation	G	Simple and finite graph	$G = (\mathcal{V}, \mathcal{E}, \mathbf{A})$	[16,22,23]
	$\mathcal{V}$	Set of vertices	N	[16,22,23]
	$\mathcal{E}$	Set of edges	Undirected	[16,22,23]
Input of DE_G	$\mathbf{A}$	Adjacency matrix	$N \times N$ matrix	[16,22,23]
	$\mathbf{X}$	Graph signal defined on G	$N \times 1$ column vector	[16,22,23]
Parameters of DE_G	m	Embedding dimension	$3 \leq m \leq 7$	[2]
	l	Delay time	$l = 1$	[2]
	c	Class number	$c^m \ll N$	[11]
Algorithm steps	Y	Embedding matrix	$N \times m$ matrix	Section 3.1
	$F : \mathbb{R} \to \mathbb{N}_c$	Map function	NCDF	[11]
Algorithm Output	$DE_G(\mathbf{X}, m, l, c)$	Computed DE_G value	$0 \leq DE_G \leq 1$	Section 3.1
MIX Process	d	Dimension	$d = 1, 2$	[7,29,35]
	a	Noise amplitude	$a = 3$	[7,29,35]
	f	Frequency	$\frac{1}{12} \leq f \leq 5$ Hz	[7,29,35]

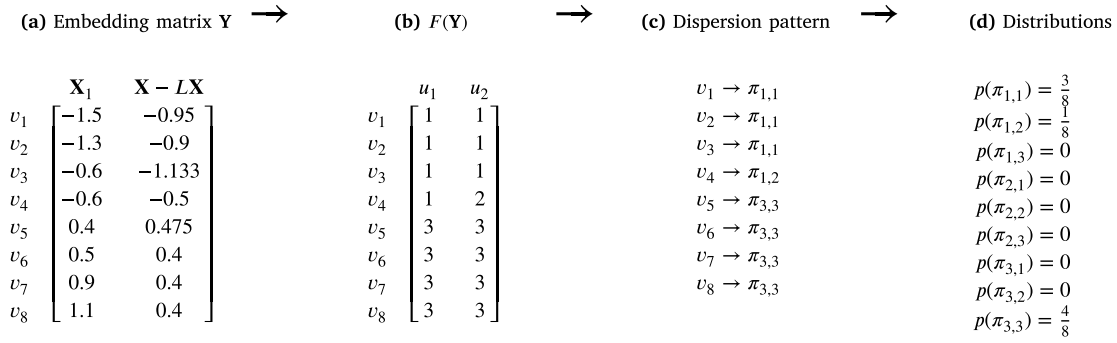


Fig. 2. Illustration of the step involved in the DE_G algorithm.

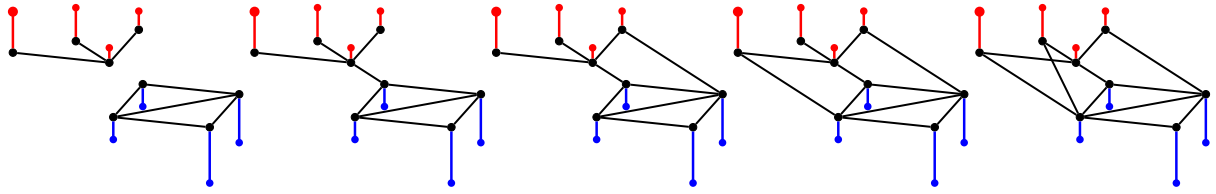


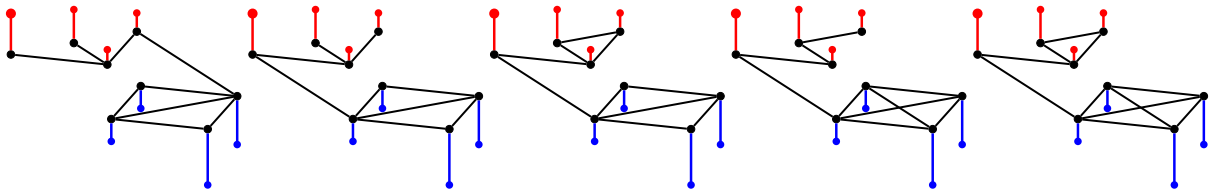
Fig. 3. All graphs have the same set of vertices, then we can consider the same graph signal on each $\mathbf{X}$, but the underlying graph is different.

the first group consists of vertices with positive values (v_1, v_2, v_3, v_4), all of which are interconnected; the second group contains vertices with negative values (v_5, v_6, v_7, v_8), which also exhibit interconnections. The only cross-group connection exists between v_4 and v_5, thereby introducing irregularity into the signal.

Changes on the underlying graph that increases the irregularity of the signal. Significantly, the removal of this intergroup edge (between v_4 and v_5) leads to a more regular signal, as reflected by a decreased entropy value (see Fig. 3a). Conversely, introducing more edges between the two groups intensifies the signal's irregularity and correspondingly elevates the entropy values, as illustrated in Fig. 3c-e. This pattern elucidates that introducing edges between vertices from differing classes (thus different values) engenders more irregular signals. Even if the signal remains unchanged, the underlying topology varies, a shift accurately captured by the DE_G algorithm.

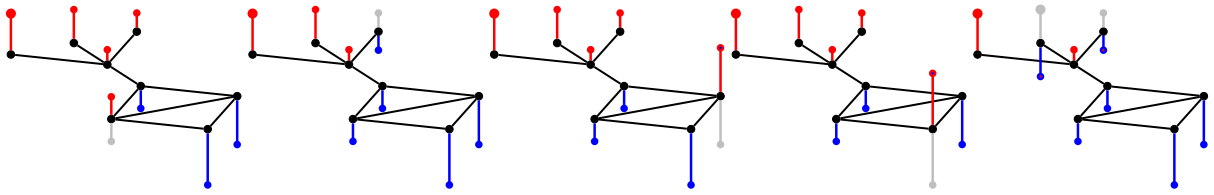
Changes on the underlying that preserves the irregularity of the signal. On the other hand, if we augment the graph with additional edges that join vertices within the same class, the signal irregularity remains similar, as detected by the DE_G algorithm, producing similar or identical entropy values. In Fig. 4a-e, we display several underlying graphs with identical graph signals, all yielding the same entropy values, further exemplifying this point.

Same graph but different signals. It is observed that the algorithm DE_G is capable of discerning the dynamics of diverse graph signals that are embedded within the same graph structure. Let us consider the graph signal defined in Fig. 1 as our first example. The entropy computed by DE_G for this configuration is 0.4434. When we alter the signal on vertex v_3 from $\mathbf{X}_3 = -0.6$ to $\mathbf{X}_3 = 0.6$ (as shown in Fig. 5a), the entropy rises due to increased irregularity in the signal. Furthermore, the level of irregularity is even more pronounced if we modify the value on vertex v_2 from $\mathbf{X}_2 = -1.3$ to $\mathbf{X}_2 = 1.3$ (refer to Fig. 5c). The reason being, the magnitude of irregularity has grown substantially. This effect is further magnified if we elevate the vertex with the lowest signal value, v_1, to a higher value (changing from $\mathbf{X}_1 = -1.5$ to $\mathbf{X}_1 = 1.5$, as depicted in Fig. 5d). A sharp increase in entropy is also observed when we change the signal values on two vertices as demonstrated in Fig. 5e. This underscores the sensitivity of the entropy measure to variations in graph signals. Lastly, Fig. 5b shows the resultant entropy value when we alter a single signal value belonging to the red group (positive values). In this case, v_6 is modified from $\mathbf{X}_6 = 0.5$ to $\mathbf{X}_6 = -0.5$. The ability of DE_G to distinguish these modifications and appropriately quantify the increase in entropy highlights its effectiveness in tracking changes in the dynamics of graph signals, even within the same underlying graph structure.



(a) $DE_G = 0.4434$ (b) $DE_G = 0.4434$ (c) $DE_G = 0.4434$ (d) $DE_G = 0.4434$ (e) $DE_G = 0.4434$

Fig. 4. All graphs have the same set of vertices, then we can consider the same graph signal on each $\mathbf{X}$, but the underlying graph is different, however the dynamical is similar in all of them (two separated groups, red and blue, joined by one and only one edge.).



(a) $DE_G = 0.5521$ (b) $DE_G = 0.5714$ (c) $DE_G = 0.6800$ (d) $DE_G = 0.7886$ (e) $DE_G = 0.8675$

Fig. 5. To assess the dynamics of similar graph signals with respect to the underlying graph topology, we maintained the same underlying graph as depicted in Fig. 1 for all scenarios. We also considered the same signal as that in Fig. 1, which has an entropy $DE_G = 0.4434$, with the only difference being that we altered the sign of some vertices. This exercise allows us to examine the behaviour of similar signals on the same graph. The original graph signal is illustrated in grey.

3.4. Dispersion entropy for directed graphs

The algorithm DE_G provides a tool for analysing undirected graph signals, and can be extended to *directed graphs* with minor modifications (see Appendix B). Additionally, the algorithm can be applied to any graph signal, but for time series, it produces the same values as the classical DE [11]. This is established in Proposition 1

Proposition 1 (*Equivalence of* DE *and* DE_G *for Time Series*). *Let* $\mathbf{X} = \{x_i\}_{i=1}^N$ *be a time series and* $\overrightarrow{G} = \overrightarrow{P}$ *is the directed path on* N *vertices. Then, for all* m, c *and* l *the following equality holds:*

$$DE(m, l, c) = DE_{\overrightarrow{P}}(m, l, c) .$$

Proof. Please refer to the Appendix C. □

4. MIX Process and DE_G

In this section, our objective is to demonstrate the detection of irregularities, not only in simple graphs as illustrated in Section 3.3, but also in more complex structures such as Random Geometric Graphs, and when dealing with more complex signals as is the case of the MIX process.

4.1. Random geometric graph

A *d-dimensional Random Geometric Graph* (RGG) is a graph in which each vertex $v_i \in \mathcal{V}$ is assigned a random d-dimensional coordinate $v_i \to \mathbf{z}_i = (z_i^1, \dots, z_i^d) \in [0,1]^d$. The structure of a RGG is significantly influenced by the proximity parameter $0 \leq r$. Two vertices $v_i, v_j \in \mathcal{V}$ are connected by an edge if and only if the Euclidean distance between their coordinates is less than or equal to r, i.e., $|\mathbf{z}_i - \mathbf{z}_j| \leq r$.

For larger values of r, more edges will be formed as the condition $|\mathbf{z}_i - \mathbf{z}_j| \leq r$ is more likely to be satisfied, leading to a denser and more connected graph. On the other hand, smaller values of r impose a more stringent condition for edge creation, resulting in sparser and more disconnected graphs. Hence, r acts as a tunable parameter controlling the sparsity and connectivity of the resulting RGG (see [36]).

4.2. MIX process

We introduce a graph-based stochastic process MIX_G defined on RGGs to assess the performance of DE_G in capturing complex signal dynamics.

First we define $MIX_{\mathbb{R}^d} : \mathbb{R}^d \to \mathbb{R}$ given by

$$MIX_{\mathbb{R}^d} = (1 - R)S + RW \quad , \tag{5}$$

where $R : \mathbb{R}^d \to [0,1]$ is a random variable with a probability p of taking the value 1 and a probability $1 - p$ of taking the value 0, $W : \mathbb{R}^d \to [-\sqrt{a}, \sqrt{a}]$ is uniformly distributed white noise, and $S(\mathbf{z}) = S(z_i^1, \dots, z_i^d) = \sum_{j=1}^d \sin\left(2\pi f z_i^j\right)$. Observe that the function S is determined by the function $\sin(2\pi f x)$ with period $\frac{1}{f}$ and frequency f. Hence, the $MIX_{\mathbb{R}^d}$ is uniquely determined by the dimension d, the amplitude of the noise a and the frequency f.[1]

The selection of parameters $d = 1$, $f = 1/12$ Hz, $a = 3$ and equidistant sampling was proposed by [7] and $f = 5$ Hz in [35]. The choice was made such that the MIX process in time series could not be differentiated based on their sample means and standard deviations. The same parameter selection, except that $d = 2$, i.e. for images is studied in [29].

4.3. MIX process on RGGs

Let G be a d-dimensional RGG with N vertices $\mathcal{V} = \{v_1, v_2, v_3, \dots, v_N\}$, and $v_i \to \mathbf{z}_i = (z_i^1, \dots, z_i^d) \in [0,1]^d$. The graph signal MIX_G is

[1] In this paper, we utilize the term "frequency" to describe the parameter f within the MIX process. However, it is important to note that f is essentially the frequency of the sine function that constitutes the main component of the MIX process. Although the MIX process possesses periodic properties with a period of $\left(\frac{1}{f}, \dots, \frac{1}{f}\right)$, it would be technically incorrect to label f as the frequency of the MIX process itself. This convention can be traced back to the case when $d = 1$, wherein f serves as the frequency of both the sine function and the MIX process. Moreover, we have adopted the units in the domain as seconds, thus defining f in Hertz, in alignment with the frequency of the sine function rather than the MIX process. Despite these detailed clarifications, it is critical to recognize that such convention does not impact the interpretation of our results. The primary objective remains to employ the MIX process as a benchmark to validate the effectiveness of DE_G in diverse applications.

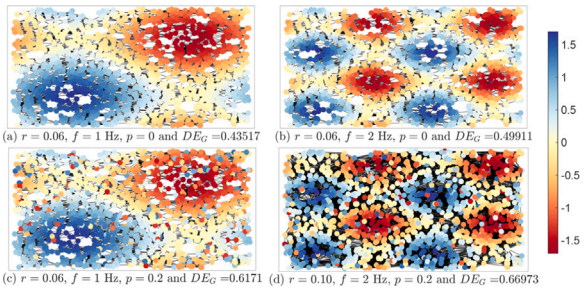


Fig. 6. Examples of RGGs with $N = 1,500$ and values $r = 0.06$ and $r = 0.10$. The graph signals are generated by the MIX_G process with different parameter values.

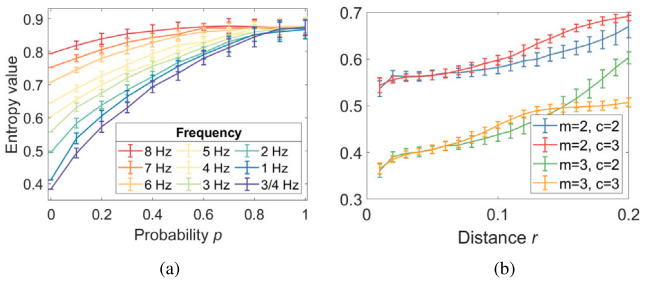


Fig. 7. Entropy values (a) for a fixed graph, increasing the noise and for several frequencies and (b) the underlying graph is more connected.

defined as the restriction of the $\text{MIX}_{\mathbb{R}^d}$ (Eq. (5)) to the graph domain, i.e.,:

$$\text{MIX}_G(v_i) = ((1-R)S + RW)(\mathbf{z}_i) \quad \text{for} \quad 1 \le i \le N\ . \tag{6}$$

Similarly to the general process $\text{MIX}_{\mathbb{R}^d}$ (Section 4.2), the MIX_G is defined for the graph G and determined by probability p, the noise amplitude a, the dimension d and the frequency f.[2]

The construction of a d-dimensional RGG requires selecting two parameters, r and d. The graph signal generated by the MIX_G process incorporates random noise (determined by p) with different amplitudes (determined by a) into some values of the sinusoidal signal (determined by f). Finally, for the algorithm DE_G, and according to Table 1, we employ a fixed embedding dimension of $m = 3$, the number of classes set at $c = 3$, time delay $l = 1$, and NCDF as the nonlinear map (similar results are obtained for others nonlinear mappings and values of m, c, and l).

Our algorithm, DE_G, detects changes in the frequency of the signal (increasing f), the presence of white noise (increasing p), and the graph connectivity (increasing r) by increasing the entropy values of DE_G, but it is robust to the amplitude of the noise a. Fig. 6 illustrates the effectiveness of DE_G in detecting the dynamics of the MIX_G process.

Analysis of the dimension d. For clarity and simplicity in our discussion, we represent the RGG in a 2-dimensional space, i.e., $d = 2$. Hence, each v_i corresponds to $\mathbf{z}_i = (z_i^1, z_i^2)$, and $S(v_i) = \sin\left(2\pi f z_i^1\right) + \sin\left(2\pi f z_i^2\right)$ for all $1 \le i \le N$. It is crucial to note that our discussion's mathematical principles and results do not depend on this choice of the dimension d and the choice was made for better visualization. However, the analysis is valid for any higher dimension but will require more complex figures.

Impact of the frequency f and probability p in the construction of the graph signal MIX_G. We analyse the impact of different parameter values on the irregularity of the graph signal MIX_G by fixing the underlying RGG with constant $N = 1500$ and $r = 0.06$.

Increasing the frequency f of the MIX_G process results in a more irregular graph signal. The frequency $f = 1$ Hz and $f = 2$ Hz of the sine function in Eq. (6) are depicted in Fig. 6(a)-(b). This increase in the frequency produces more variation in the graph signal values between neighbouring vertices. Our algorithm DE_G detects these dynamics by increasing the entropy values. Similarly, an increase in the randomness parameter p results in a more random signal. The parameters $p = 0$ and $p = 0.2$ in Eq. (6) are depicted in Fig. 6(a), (c). The DE_G algorithm detects the change in randomness, by increasing the entropy values.

More generally, we compute the entropy values for a range of frequencies from 3/4 Hz to 8 Hz, as well as for different levels of noise, with probabilities ranging from 0 to 1. The results of 30 realizations are depicted in Fig. 7(a), showing the mean and standard deviation. The DE_G algorithm effectively detects the increasing irregularity of the signal by increasing the entropy values. Moreover, the algorithm can distinguish between different levels of irregularity in the MIX_G signal based on the chosen value of p.

Analysis of the distance r. By fixing the graph signal, we investigate the behaviour of the DE_G algorithm as the underlying graph changes. Specifically, we examine the impact of increasing the distance parameter r from 0.01 to 0.3 used for construct the RGG with $N = 1,500$ vertices. Entropy values are computed for 20 realizations, and the mean and standard deviation are depicted in Fig. 7(b) for several values of m and c. As r increases, the number of edges increases, connecting more distant vertices with different values. The resulting patterns are more irregular, with more changes and a wider distribution, leading to an increase in the entropy value.

Robustness of the amplitude a. The performance of the DE_G algorithm is robust to variations in the amplitude a of the white noise, as defined in Eq. (6). This robustness was observed across a variety of experiments ($0 \le a \le 5$) and persisted despite changes to the characteristics of the graph signal dynamics. This robustness can be attributed to the utilization of a mapping function in step 2 of the DE_G algorithm. Specifically, although the introduction of noise leads to changes in the values of the embedding matrix, a significant proportion of these altered values are mapped to the same class number. Consequently, the overall distributions of permutation patterns and, by extension, the entropy values remain largely similar. This robustness to changes in noise amplitude is a key feature of the DE_G algorithm's performance, and indicates that the graph version, DE_G, maintains this property of the original DE for univariate time series [11] and images [12].

5. The spectrum of the Laplacian and DE_G

Let $\mathbf{X}$ be a graph signal; *the smoothness of* $\mathbf{X}$ is given by $\mathbf{X}^T \Delta \mathbf{X}$ [16]. We examine the relationship between DE_G and the spectrum of Δ acting on an RGGs (similar results are obtained for other random graphs).

Let G be a RGG with $N = 1,500$ vertices. The eigenvalues of Δ and its corresponding eigenvectors are denoted by $\sigma = \{\lambda_1 \le \lambda_2 \le \cdots \le \lambda_N\}$ and $\{f_i\}_{i=1}^N$, respectively. The smoothness of each eigenvector is evaluated and normalized based on the classical definition, i.e., $\lambda_N^{-1} f_i^T \Delta f_i$, and the results are shown in Fig. 8. Each eigenvector f_i is considered as a graph signal and DE_G is computed for $c = 2, 3, 4$ and $m = 2$. The results are depicted in Fig. 8. The smoothness definition is an increasing function, i.e., smaller eigenvalues correspond to smoother eigenvectors (also known as graph Fourier modes [37]). Such information is limited especially when eigenfunctions associated with equal eigenvalues (and equal smoothness) exhibit different levels of irregularity. By applying the DE_G algorithm, we can better understand and analyse the dynamics of these eigenfunctions.

The dispersion entropy computed for different values of m and c enables us to capture abrupt changes in entropy values as the dynamics of the eigenfunctions change. Fig. 9 depicts six eigenvectors

[2] As with the description in the previous section, the term "frequency" f primarily represents the frequency of the $\sin\left(2\pi f z_i^j\right)$, that define the MIX_G process and not the frequency of graph signal MIX_G. It is worth mentioning that common terms such as frequency, sampling, filtering, usually associated with Fourier Transform, lack a universally agreed-upon definition in the Graph Signal Processing domain.

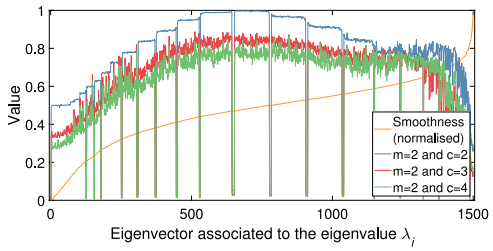


Fig. 8. Entropy values of DE_G and smoothness based on the Laplacian Δ for the eigenvalues as graph signals.

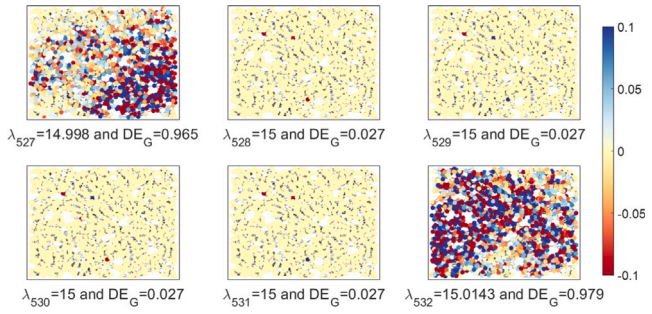


Fig. 9. Several eigenfunctions and their entropy values.

$\{f_j\}_{j=527}^{532}$ corresponding to the eigenvalues $\{\lambda_j\}_{j=527}^{532}$. The definition of smoothness of f_j coincides with the value λ, and the eigenvalue $\lambda_{528} = 15$ has a multiplicity equal to four, and its eigenfunctions $\{\lambda_j\}_{j=528}^{531}$ exhibit a regular behaviour, while f_{527} and f_{532} are more irregular. Hence, classical definitions are not able to fully capture the difference in dynamics within the graph signals. In contrast, the DE_G algorithm is capable of detecting them. In particular, the entropy value of the eigenfunctions is nearly close to 0 if the signal exhibits a more regular dynamics and close to 1 for the most irregular eigenfunctions. Thus, DE_G detects eigenvalues with high multiplicity, useful for the construction of isospectral graphs [38].

6. Small-world networks and DE_G

We evaluate the performance of DE_G in detecting dynamics on signals defined on small-world networks, generated by the Watts–Strogatz model [26], and changing the mean degree k and rewiring probability p. Let G be a small-world network with $N = 1,500$ and various graph signals, including a random signal, a recurrence relation (logistic map [2]), a stochastic process (Wiener process [39]), and a periodic signal (sine).

Fixing k, changing p. By fixing $k = 1$, we analyse the effect of the parameter p (ranging from 0 to 1) in the construction of the network G_p and the entropy values. We compute DE_G for each graph signal for 20 realizations, and the mean and standard deviation are depicted in Fig. 10(a). For $p = 0$, the underlying graph G_p is a cycle of N vertices. A path graph is a geometric perturbation of a cycle [40] and due to Proposition 1, we can consider the values of $p = 0$ to be the classical DE. The classical DE is able to detect the dynamics of various signals, but its computation does not involve the topological structure, thus it only works for the path graph. In contrast, DE_G takes into account not only the signal information but also the graph structure. In this setting, the dynamics of the random signal is almost constant, because it is not affected by G_p. The Wiener process and sine signals exhibit lower entropy values for $p = 0$ (e.g., the cycle), as their dynamics stem from either periodicity (sine) or stochastic processes (Wiener). However, as p increases, the underlying graph becomes more random, and hence the entropy value also increases. In any case, DE_G is still able to distinguish

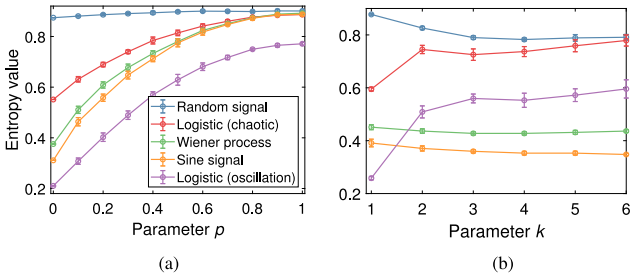


Fig. 10. Entropy values for different signals defined on a small-world network generated by the Watts–Strogatz model.

Table 2
Overview of the graph structures used, including the number of vertices and edges.

Underlying graph	$\lvert\mathcal{V}\rvert$	$\lvert\mathcal{E}\rvert$	Reference
Minnesota road network	2,642	3,303	[41]
Social circles: Facebook	3,959	84,243	[42]
Arxiv GR-QC collaboration	5,241	14,484	[43]
The US power grid	4,941	6,594	[25]
Pointcloud (Stanford Bunny)	2,503	13,726	[44]
Sphere	4,000	22,630	[45]

the random signal from the periodic signal and the Wiener process (for all $p < 0.8$). Two logistic map signals are generated, one with oscillatory behaviour ($r = 3.3$) and one with chaotic behaviour ($r = 3.7$). These characteristics are well detected by DE_G for all values of p.

Fixing p, changing k. By fixing $p = 0.05$, the underlying graph G_k where k ranges from 1 to 6, thereby increasing the connectivity. In Fig. 10(b), we present the entropy values for each graph signal. The entropy values for the sine and Wiener signals almost remain constant, independent of G_k, due to their periodicity and stochastic dynamics. However, the logistic map exhibits a higher degree of variability in its entropy values as k increases. This is because the logistic map is defined by a recurrence formula, where each value depends only on the previous sample, and if k increases, the underlying G_k has more connections between neighbourhoods. This may disrupt the recurrence relation, generating more irregular signals and resulting in higher entropy values. Conversely, the random signal shows a reduction in entropy values as k increases, as the creation of more connections leads to a more robust average value due to the law of large numbers.

7. Graph Centrality Measures and DE_G

Each centrality measure can be considered as a graph signal, allowing the application of the DE_G algorithm to assess the irregularity of centrality measures on real and synthetic graphs (refer to Table 2).

We used six centrality measures as graph signals, namely [30,31]: *Eigenvector centrality*, *Betweenness*, *Closeness*, *Harmonic centrality*, *Degree* and *Pagerank*. The DE_G algorithm leverages the graph topology to effectively detect irregularities generated by each centrality measure, as demonstrated in Fig. 11. In particular, the *Eigenvector Centrality* produces smooth signals [21] in most graphs, and this is reflected in low entropy values. Well-connected vertices tend to appear on the shortest paths between other vertices. When the graph has only a few such vertices, the entropy of the *Betweenness* measure is lower. In cases where the graph has a more irregular distribution of vertices with this characteristic (e.g., in the sphere due to its symmetry), the entropy values are higher. A similar effect occurs when considering the average length of the shortest path between the vertex and all other vertices, as detected by the *Closeness* measure. Finally, the *Degree* and *PageRank* measures produce more irregular graph signals because each signal's value defined on the graph depends only on local properties (the degree or the number and importance of the other vertices connected to it) rather than global properties (such as average paths between vertices in the previous measures).

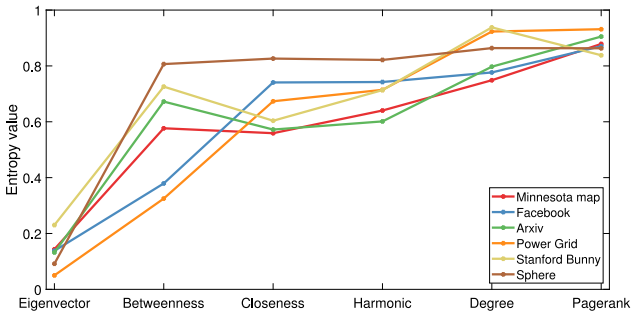


Fig. 11. The dispersion entropy for various centrality measures.

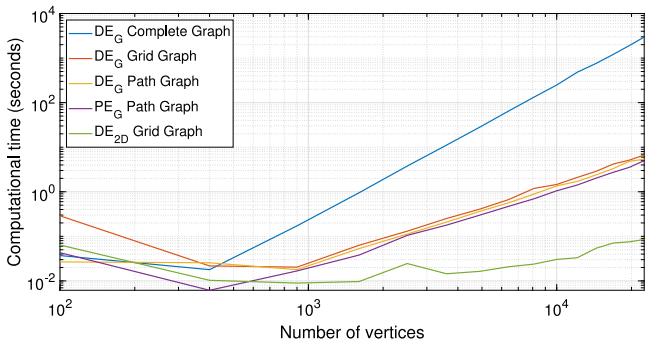


Fig. 12. Computational time of DE_G, PE_G, and DE_{2D} with varying node number for grid, path, and complete graphs.

8. Comparing DE_G and other entropy metrics performance

In Section 3.4, we establish an equivalence between DE and DE_G for directed paths. In this context, we extend our discussion to include a comparison between the performance and computational time of DE_G and other entropy metrics. The Permutation Entropy for Graph Signals, denoted by PE_G [19], marked the first entropy metric specifically designed for graph-based data analysis. Both methods rely on the adjacency matrix, but PE_G primarily focuses on the order of amplitude values (local properties), which might result in the loss of valuable information regarding the amplitudes (global properties). DE_G addresses these limitations by providing a more comprehensive way to characterize the dynamics of graph signals. We conducted the same previous analysis with PE_G (see the Appendix D), and found that DE_G consistently outperforms PE_G in all cases, highlighting the potential of our novel method for effectively analysing graph signal irregularities.

8.1. Computational cost

The computational cost of DE_G was compared to PE_G using G as a 2-dimensional grid graph of size $n \times n$ with a MIX process defined on G. The entropy values were computed for G contains n^2 vertices, with values varying from 10^2 to 150^2. The results are displayed in Fig. 12. The computational times for both metrics PE_G and DE_G (for equals underlying graphs) were almost identical, with a minor increase in DE_G due to the mapping computation in Step 3 of the DE_G algorithm (Section 3.1).

The MIX process signal, defined on a grid, was treated as an image to apply the Dispersion Entropy for Images (DE_{2D}) presented in [12]. The computational time for this setting was also evaluated (see Fig. 12). DE_G was more computationally demanding than DE_{2D}, which was anticipated due to DE_G not making any prior assumptions about the signal sample domain structure.

Notably, the computational time of DE_G escalates not only with the image size (for an image of size $n \times n$, we must calculate the power of the adjacency matrix of size $n^2 \times n^2$) but also with the graph's connectivity. When a path graph was considered instead of the grid, the computational time was almost unchanged due to the edge count increasing linearly with the number of vertices (n^2). However, for the $n \times n$ vertices in a complete graph, the edge count increases quadratically with the number of vertices. Consequently, the number of non-zero entries in the adjacency matrix also escalates quadratically, leading to increased computational costs (see Fig. 12).

The DE_{2D} algorithm does not factor in the underlying graph topology, therefore yielding identical results for path, grid, or complete graphs. However, this algorithm is unsuitable for irregular domains. Thus, while DE_{2D} is faster for regular images or time series, our algorithm is applicable to any domain, though similar results are obtained for images. We hypothesize that efficient implementations for DE_G could be developed for special types of graphs using the graph's symmetries or when its adjacency matrix is periodic. However, these optimizations are beyond this study's scope, which reports the cost of the general algorithm for images under various graph settings.

Simulations were conducted on a PC with an Intel(R) Core(TM) i7 using MATLAB R2023b.

9. Conclusions

We have introduced Dispersion Entropy for Graph Signals (DE_G), a method that enhances the analysis of irregularities in graph signals. Our approach generalizes classical dispersion entropy, enabling its application to a wide array of domains, including real-world graphs, directed, undirected and weighted graphs, and unveiling novel relationships between graph signals and graph-theoretic concepts (e.g., eigenvalues and centrality measures). Notably, DE_G allows the application of the Dispersion Entropy concept not only to univariate time series, but also to multivariate time series and images. By overcoming the limitations of the classical smoothness definition, DE_G offers a more comprehensive approach to analysing graph signals and holds significant potential for further research and practical applications, as it effectively captures the complex dynamics of signals across diverse topology configurations.

CRediT authorship contribution statement

John Stewart Fabila-Carrasco: Conceptualization, Methodology, Formal analysis, Writing – original draft. **Chao Tan:** Supervision. **Javier Escudero:** Conceptualization, Supervision, Writing – review & editing.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: John Stewart Fabila-Carrasco reports financial support was provided by the Leverhulme Trust via a Research Project Grant (RPG-2020-158). Javier Escudero reports financial support was provided by the Leverhulme Trust via a Research Project Grant (RPG-2020-158). John Stewart Fabila-Carrasco reports a relatonship with The University of Edinburgh that includes: employment. Javier Escudero reports a relatonship with The University of Edinburgh that includes: employment. Chao Tan reports a relatonship with Tianjin University that includes: employment. For the purpose of open access, the author has applied a Creatve Commons Attributon (CC BY) licence to any Author Accepted Manuscript version arising from this submission. Elsevier and Jisc have established an agreement to support authors in the United Kingdom who wish to publish open access. The University of Edinburgh is cover by this agreement.

Data availability

No data was used for the research described in the article.

Acknowledgements

This work was supported by the Leverhulme Trust via a Research Project Grant (RPG-2020-158). For the purpose of open access, the author has applied a Creative Commons Attribution (CC BY) licence to any Author Accepted Manuscript version arising from this submission We are grateful to the anonymous reviewers whose insightful comments significantly improved the manuscript.

Appendix A. Classical dispersion entropy for univariate time series

This appendix provides a concise description of the classical Dispersion Entropy (DE) for time series [13]. Consider a univariate signal $x = \{x_1, x_2, \dots, x_N\}$ of length N. The DE calculation proceeds as follows:

1. The signal values $x_j, j = 1, 2, \dots, N$ are classified into c classes, resulting in a classified signal $u_j, j = 1, 2, \dots, N$. Various mapping techniques can be applied for this classification.
2. Time series $\mathbf{u}_i^{m,c}$ of embedding dimension m and delay l are formed as $\mathbf{u}_i^{m,c} = \left\{u_i^c, u_{i+l^c}^c, \dots, u_{i+(m-1)l}^c\right\}$. Each of these time series is mapped to a dispersion pattern $\pi_{v_0 v_1 \dots v_{m-1}}$. The total number of possible dispersion patterns is c^m.
3. The relative frequency of each potential dispersion pattern $\pi_{v_0 \dots v_{m-1}}$ is calculated by
$$p\left(\pi_{v_0 \dots v_{m-1}}\right) = \frac{|\left\{i \mid i \le N-(m-1)l, \mathbf{u}_i^{m,c} \text{ has type } \pi_{v_0 \dots v_{m-1}}\right\}|}{N-(m-1)l},$$
where $||$ represents cardinality. This frequency represents the proportion of the time series assigned to each dispersion pattern.
4. Finally, DE is computed following the Shannon entropy definition as
$$\mathrm{DE}(\mathbf{x}, m, c, l) = -\frac{1}{\ln(c^m)} \sum_{\pi=1}^{c^m} p\left(\pi_{v_0 \dots v_{m-1}}\right) \cdot \ln\left(p\left(\pi_{v_0 \dots v_{m-1}}\right)\right).$$

Appendix B. Dispersion entropy for directed graphs

In the paper, we have introduced the Dispersion Entropy for graph signals, denoted as DE_{G}, in the context of *undirected* graphs. To extend this concept to *directed graphs* or *digraphs*, the approach remains analogous, with the primary distinction being the need to incorporate specific constraints on the rows of the embedding matrix. These constraints are imposed by the well-defined vectors $\mathbf{y}_k$.

Let $\overrightarrow{G} = (\mathcal{V}, \mathcal{E}, \mathbf{A})$ be a digraph with N vertices, where $\mathbf{A}$ denotes the adjacency matrix of the directed graph, and $\mathbf{X} = \{x_i\}_{i=1}^n$ is a signal defined on $\overrightarrow{G}$. Given an *embedding dimension* m with $2 \le m \in \mathbb{N}$, a *delay time* $L \in \mathbb{N}$, and a *class number* $c \in \mathbb{N}$, the Dispersion Entropy for Directed Graphs ($DE_{\overrightarrow{G}}$) is defined as follows:

1. *Embedding matrix.* Let $\mathcal{V}^* \subset \mathcal{V}$ be the set given by:
$$V^* = \{\, i \in \mathcal{V} \mid \sum_{j=1}^{n} (\mathbf{A}^{kl})_{ij} \neq 0 \text{ for all } k = 0, 1, \dots, m-1 \,\}.$$
The *embedding matrix* $\mathbf{Y}^* \in \mathbb{R}^{|V^*| \times m}$ is given by:
$$\mathbf{Y}^* = [\mathbf{y}_0^*, \mathbf{y}_1^*, \dots, \mathbf{y}_{m-1}^*] \tag{B.1}$$
where $\mathbf{y}_k^* \in \mathbb{R}^{|V^*| \times 1}$, given by the restriction of $\mathbf{y}_k$ to the vertices in V^*, i.e., $\mathbf{y}_k^* = \mathbf{y}^k|_{V^*}$.
2. *Map function.* Each element of the embedding matrix $\mathbf{Y}^*$ is mapped to an integer number from 1 to c, called a class, i.e., we define a function $F : \mathbb{R} \to \mathbb{N}_c$ where $\mathbb{N}_c = \{1, 2, \dots, c\}$ that applies element-wise on the matrix $\mathbf{Y}^*$, i.e. $F(\mathbf{Y}^*) \in \mathbb{N}_c^{N \times m}$ where $F(\mathbf{Y}^*)_{ij} = F(y_{ij}^*)$.
3. *Dispersion patterns.* Each row of the matrix $F(\mathbf{Y}^*)$, called an *embedding vector*, is mapped to a unique dispersion pattern. Formally, the *embedding vectors* consist of m integer numbers (from 1 to c) corresponding to each row of the matrix $F(\mathbf{Y}^*)$, i.e., $\mathrm{row}_i(F(\mathbf{Y}^*)) = \left(F(y_{ij}^*)\right)_{j=1}^m$ for $i = 1, 2, \dots, N$. The set of dispersion patterns is defined as $\Pi = \{\, \pi_{v_1^* v_2^* \dots v_m^*} \mid v_i^* \in \mathbb{N}_c \,\}$. Each embedding vector is uniquely mapped to a dispersion pattern, i.e., $\mathrm{row}_i(F(\mathbf{Y}^*)) \to \pi_{v_1^* v_2^* \dots v_m^*}$ where $v_1 = F(y_{i1}^*), v_2 = F(y_{i2}^*), \dots, v_m = F(y_{im}^*)$.
4. *Relative frequencies.* For each dispersion pattern $\pi \in \Pi$, its relative frequency is obtained as:
$$p(\pi) = \frac{|\{v_i \mid v_i \in \mathcal{V}^* \text{ and } v_i \text{ has type } \pi\}|}{|\mathcal{V}^*|},$$
where $||$ represents cardinality.
5. *Shannon's entropy.* The *dispersion entropy for graph signals* $DE_{\overrightarrow{G}}$ is computed as the normalized Shannon's entropy for the distinct dispersion patterns as follows:
$$DE_{\overrightarrow{G}}(\mathbf{X}, m, l, c) = -\frac{1}{\ln(c^m)} \sum_{\pi \in \Pi} p(\pi) \ln p(\pi).$$

Properties

The $DE_{\overrightarrow{G}}$ algorithm for directed graphs exhibits the following properties:

The directed graph version of $DE_{\overrightarrow{G}}$ serves as a generalization of its undirected counterpart. If G is an undirected connected (non-trivial) graph, then $\mathcal{V}^* = \mathcal{V}$, and all the steps remain the same in both the directed and undirected versions of the algorithm.

The restriction process $\mathbf{y}_k^* = \mathbf{y}^k|_{V^*}$ is equivalent to the vertex virtualisation process presented in [40].

Similarly, the $DE_{\overrightarrow{G}}$ algorithm can be extended to weighted (directed or undirected) graphs by restricting the subset to
$$V^* = \{\, i \in \mathcal{V} \mid \sum_{j=1}^{n} (\mathbf{W}^{kl})_{ij} \neq 0 \text{ for all } k = 0, 1, \dots, m-1 \,\}.$$
where $\mathbf{W}$ represents the weighted adjacency matrix. This generalization allows for a more comprehensive analysis of graph signals in various contexts.

Appendix C. Proof of Proposition 1

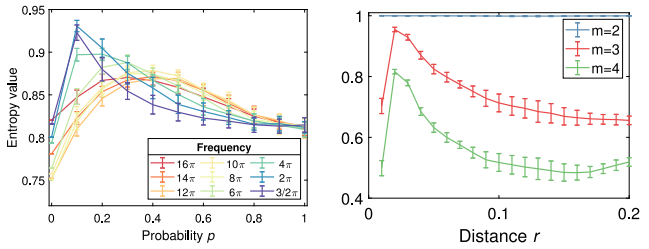
The classical dispersion entropy for time series was established in the literature by [11]. In the following proposition, we demonstrate that when the DE_{G} is restricted to time series (considering the directed path as the underlying graph), the DE_{G} is equivalent to the classical DE.

A *directed path* on k vertices is a directed graph that connects a sequence of distinct vertices with all edges oriented in the same direction, denoted as $\overrightarrow{P}$. Its vertices are given by $\mathcal{V} = \{v_1, v_2, v_3, \dots, v_k\}$ and its arc set is $\mathcal{E} = \{(v_i, v_{i+1}) \mid 1 \le i \le N-1\}$.

Proposition 1 (*Equivalence of* DE *and* DE_{G} *for Time Series*)**.** Let $\mathbf{X} = \{x_i\}_{i=1}^N$ be a time series and consider $\overrightarrow{G} = \overrightarrow{P}$ the directed path on n vertices, then for all m, c and l, the following equality holds:
$$\mathrm{DE}(m, l, c) = \mathrm{DE}_{\overrightarrow{P}}(m, l, c).$$

Proof. The adjacency matrix for the directed path $\mathbf{A}$ is given by
$$\mathbf{A}_{ij} = \begin{cases} 1 & \text{if} \quad i = 1, 2, \dots, N-1 \quad \text{and} \quad j = i+1, \\ 0 & \text{otherwise}. \end{cases}$$

For any $k \in \mathbb{N}$, the matrix $\mathbf{A}^k$ is given by
$$\left(\mathbf{A}^k\right)_{ij} = \begin{cases} 1 & \text{if} \quad i = 1, 2, \dots, N-k \quad \text{and} \quad j = i+k, \\ 0 & \text{otherwise}, \end{cases}$$



(a) (b)

Fig. D.13. Entropy values using $\mathrm{PE_G}$ (a) for a fixed graph, increasing the noise and for several frequencies and (b) the underlying graph is more connected.

in particular, for all $k = 0, 1, \ldots, m-1$

$$\sum_{j=1}^{N} (\mathbf{A}^{kl})_{ij} = \begin{cases} 1 & \text{if} \quad i = 1, \ldots, N-(m-1)L, \\ 0 & \text{otherwise} \end{cases}.$$

Thus, we have

$$\mathbf{y}_k^* = \mathbf{y}_k|_{V^*} = D\mathbf{A}^{kl}\mathbf{X}\Big|_{V^*} = [x_{1+kl}, x_{2+kl}, \ldots, x_{i+kl}, \ldots, x_{N-(m-1)l}]^T .$$

The *embedding matrix* is given by:

$$\mathbf{Y}^* = \begin{pmatrix} x_1 & x_{1+L} & \cdots & x_{1+(m-1)L} \\ x_2 & x_{2+L} & \cdots & x_{2+(m-1)L} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N-(m-1)L} & x_{N-(m-2)L} & \cdots & x_N \end{pmatrix},$$

and, given a map function $F : \mathbb{R} \to \mathbb{N}_c$ defined by $F = G \circ \mathrm{NCDF} : \mathbb{R} \to \mathbb{N}_c$, the matrix $F(\mathbf{Y}^*)$ is given by:

$$F(\mathbf{Y}^*) = \begin{pmatrix} z_1 & z_{1+L} & \cdots & z_{1+(m-1)L} \\ z_2 & z_{2+L} & \cdots & z_{2+(m-1)L} \\ \vdots & \vdots & \ddots & \vdots \\ z_{N-(m-1)L} & z_{N-(m-2)L} & \cdots & z_N \end{pmatrix}.$$

Subsequently, the embedding vectors are represented as $\mathrm{row}_i(F(\mathbf{Y})) = \left(z_i, z_{i+L}, \ldots z_{i+(m-1)L}\right)$. Due to the fact that $|\mathcal{V}| = N-(m-1)L$, the *relative frequencies* and *Shannon's entropy* associated with the graph-based dispersion entropy ($DE_{\vec{G}}$) and the classical dispersion entropy (DE) are identical. □

Appendix D. Comparing $\mathrm{DE_G}$ and $\mathrm{PE_G}$ Performance

In this section, we demonstrate the superior performance of the Dispersion Entropy for Graph Signals ($\mathrm{DE_G}$) over the Permutation Entropy for Graph Signals, denoted by $\mathrm{PE_G}$ [19]. By applying both algorithms to all the examples in the manuscript, we consistently observe that $\mathrm{DE_G}$ outperforms $\mathrm{PE_G}$, highlighting the potential and efficacy of $\mathrm{DE_G}$ for analysing graph signal irregularities.

Following the same setting used to produced Figs. 2, 3, 5 and 6 in the manuscript, we substitute $\mathrm{PE_G}$ for $\mathrm{DE_G}$. The results are depicted in Fig. D.13, D.14, D.15 and D.16, respectively.

Random graphs and $\mathrm{PE_G}$

The $\mathrm{PE_G}$ algorithm is not able to detect increasing of the signal irregularity (due to frequency increments) and is unable to differentiate between distinct levels of irregularity in the MIX_G signal based on the parameter p (Fig. D.13(a)). Similarly, in Fig. D.13(b), as graph connectivity increases (by raising r) the algorithm saturates for an embedding dimension of $m = 2$. To achieve accurate characterizations, it is necessary to increase $m > 2$ and even that, the behaviour is not monotonous, whereas $\mathrm{DE_G}$ performs well with smaller embedding dimensions.

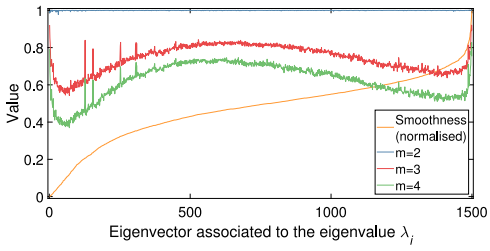
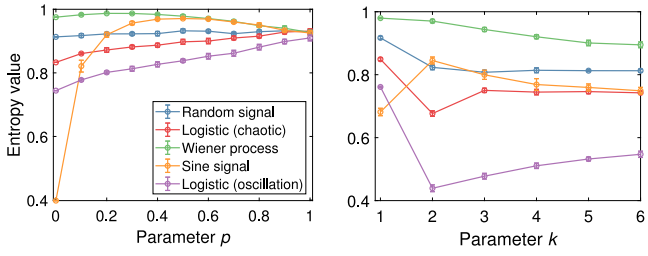


Fig. D.14. Entropy values of $\mathrm{PE_G}$ and smoothness based on the Laplacian Δ for the eigenvalues as graph signals.



(a) (b)

Fig. D.15. Entropy values of $\mathrm{PE_G}$ for different signals defined on a small-world network generated by the Watts–Strogatz model.

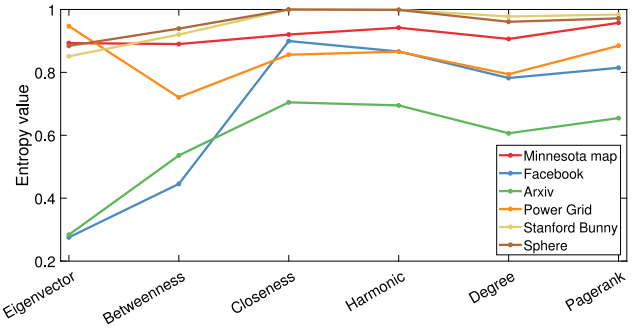


Fig. D.16. The permutation entropy for various centrality measures.

The spectrum of the Laplacian and $\mathrm{PE_G}$

The entropy values of $\mathrm{PE_G}$ exhibit a highly consistent and regular behaviour, with minimal variations (Fig. D.14). Despite the varying degrees of irregularity in the eigenvalues (as shown in Fig. 4 of the manuscript), the $\mathrm{PE_G}$ algorithm fails to detect these differences.

Small-world networks and $\mathrm{PE_G}$

The stochastic dynamics of the Wiener process are not adequately characterized by $\mathrm{PE_G}$ (Fig. D.15(a)), as its entropy values are higher than those of random behaviour (random signal). Periodic dynamics are detected only with lower parameter values of p, and the chaotic and oscillation behaviour (Logistic map) are identified by $\mathrm{PE_G}$, which is consistent with the results presented in [19]. However, as the parameter k is increased (Fig. D.15(b)), the performance of $\mathrm{PE_G}$ remains similar when the parameter p is changed. This is due to $\mathrm{PE_G}$ considering the order of the values but not their amplitude.

Graph centrality measures and $\mathrm{PE_G}$

Smooth signals produced by the *Eigenvector Centrality* are not effectively detected by $\mathrm{PE_G}$ (with the exception of the Arxiv and Facebook graphs). The remaining centrality measures yield similar entropy

values, making it challenging to establish a relationship with PE_G (Fig. D.16). This limitation highlights the greater value of DE_G for such analyses.

References

[1] Azami H, Faes L, Escudero J, Humeau-Heurtier A, Silva LE. Entropy analysis of univariate biomedical signals: Review and comparison of methods. Front Entropy Across Discip Panor Entropy Theory Comput Appl 2023;233–86.

[2] Bandt C, Pompe B. Permutation entropy: A natural complexity measure for time series. Phys Rev Lett 2002;88(17):174102.

[3] Richman JS, Lake DE, Moorman JR. Sample entropy. In: Methods in enzymology, vol. 384. Elsevier; 2004, p. 172–84.

[4] Benedetto F, Mastroeni L, Vellucci P. Modeling the flow of information between financial time-series by an entropy-based approach. Ann Oper Res 2021;299:1235–52.

[5] Yin Y, Shang P. Weighted multiscale permutation entropy of financial time series. Nonlinear Dynam 2014;78:2921–39.

[6] Cao Y, Tung Ww, Gao J, Protopopescu VA, Hively LM. Detecting dynamical changes in time series using the permutation entropy. Phys Rev E 2004;70(4):046217.

[7] Pincus SM. Approximate entropy as a measure of system complexity. Proc Natl Acad Sci 1991;88(6):2297–301.

[8] Li Y, Wang S, Li N, Deng Z. Multiscale symbolic diversity entropy: A novel measurement approach for time-series analysis and its application in fault diagnosis of planetary gearboxes. IEEE Trans Ind Inf 2021;18(2):1121–31.

[9] Rostaghi M, Ashory MR, Azami H. Application of dispersion entropy to status characterization of rotary machines. J Sound Vib 2019;438:291–308.

[10] Zanin M, Martínez JH. Analyzing international events through the lens of statistical physics: The case of Ukraine. Chaos 2022;32(5).

[11] Rostaghi M, Azami H. Dispersion entropy: A measure for time-series analysis. IEEE Signal Process Lett 2016;23(5):610–4.

[12] Azami H, da Silva LEV, Omoto ACM, Humeau-Heurtier A. Two-dimensional dispersion entropy: An information-theoretic method for irregularity analysis of images. Signal Process, Image Commun 2019;75:178–87.

[13] Azami H, Rostaghi M, Abásolo D, Escudero J. Refined composite multiscale dispersion entropy and its application to biomedical signals. IEEE Trans Biomed Eng 2017;64(12):2872–9.

[14] Azami H, Escudero J. Amplitude-and fluctuation-based dispersion entropy. Entropy 2018;20(3):210.

[15] Li Y, Geng B, Jiao S. Dispersion entropy-based Lempel-Ziv complexity: A new metric for signal analysis. Chaos Solitons Fractals 2022;161.

[16] Ortega A, Frossard P, Kovačević J, Moura JMF, Vandergheynst P. Graph signal processing: Overview, challenges, and applications. Proc IEEE 2018;106(5):808–28.

[17] Huang W, Bolton TA, Medaglia JD, Bassett DS, Ribeiro A, Van De Ville D. A graph signal processing perspective on functional brain imaging. Proc IEEE 2018;106(5):868–85.

[18] Morel C, Humeau-Heurtier A. Multiscale permutation entropy for two-dimensional patterns. Pattern Recognit Lett 2021;150:139–46.

[19] Fabila-Carrasco JS, Tan C, Escudero J. Permutation entropy for graph signals. IEEE Trans Signal Inf Process Netw 2022;8:288–300.

[20] Wang Y, Xu Y, Liu M, Guo Y, Wu Y, Chen C, et al. Cumulative residual symbolic dispersion entropy and its multiscale version: Methodology, verification, and application. Chaos Solitons Fractals 2022;160:112266.

[21] Dong X, Thanou D, Frossard P, Vandergheynst P. Learning Laplacian matrix in smooth graph signal representations. IEEE Trans Signal Process 2016;64(23):6160–73.

[22] Shuman DI, Narang SK, Frossard P, Ortega A, Vandergheynst P. The emerging field of signal processing on graphs: Extending high-dimensional data analysis to networks and other irregular domains. IEEE Signal Process Mag 2013;30(3):83–98.

[23] Stanković L, Daković M, Sejdić E. Introduction to graph signal processing. Vertex-Freq Anal Graph Signals 2019;3–108.

[24] Kenniche H, Ravelomananana V. Random geometric graphs as model of wireless sensor networks. In: 2010 the 2nd international conference on computer and automation engineering, vol. 4. IEEE; 2010, p. 103–7.

[25] Watts DJ, Strogatz SH. Collective dynamics of 'small-world' networks. Nature 1998;393(6684):440–2.

[26] Newman ME, Strogatz SH, Watts DJ. Random graphs with arbitrary degree distributions and their applications. Phys Rev E 2001;64(2):026118.

[27] Newman ME. Models of the small world. J Stat Phys 2000;101:819–41.

[28] Pincus SM, Goldberger AL. Physiological time-series analysis: What does regularity quantify? Am J Physiol - Heart Circ Physiol 1994;266(4 35-4).

[29] Silva LEV, Senra Filho A, Fazan VPS, Felipe JC, Junior LM. Two-dimensional sample entropy: Assessing image texture through irregularity. Biomed Phys Eng Express 2016;2(4):045002.

[30] Borgatti SP, Everett MG. A graph-theoretic perspective on centrality. Social Networks 2006;28(4):466–84.

[31] Das K, Samanta S, Pal M. Study on centrality measures in social networks: a survey. Soc Netw Anal Min 2018;8:1–11.

[32] Chung FRK. Spectral graph theory, vol. 92. Providence, RI, USA: American Mathematical Society; 1997.

[33] Mohar B, Alavi Y, Chartrand G, Oellermann O. The Laplacian spectrum of graphs. Graph Theory Comb Appl 1991;2(871–898):12.

[34] Montgomery DC, Runger GC. Applied statistics and probability for engineers. John wiley & sons; 2010.

[35] Żurek S, Grabowski W, Wojtiuk K, Szewczak D, Guzik P, Piskorski J. Relative consistency of sample entropy is not preserved in MIX processes. Entropy 2020;22(6):694.

[36] Dall J, Christensen M. Random geometric graphs. Phys Rev E 2002;66(1):016121.

[37] Girault B, Ortega A, Narayanan SS. Irregularity-aware graph fourier transforms. IEEE Trans Signal Process 2018;66(21):5746–61.

[38] Fabila-Carrasco J, Lledó F, Post O. A geometric construction of isospectral magnetic graphs. Anal Math Phys 2023;13(4):64.

[39] Wiener N. The homogeneous chaos. Am J Math 1938;60(4):897–936.

[40] Fabila-Carrasco JS, Lledó F, Post O. Spectral preorder and perturbations of discrete weighted graphs. Math Ann 2022;382(3–4):1775–823.

[41] Gleich D. The matlabbgl matlab library. 2015.

[42] Leskovec J, Mcauley J. Learning to discover social circles in ego networks. Adv Neural Inf Process Syst 2012;25.

[43] Leskovec J, Kleinberg J, Faloutsos C. Graph evolution: Densification and shrinking diameters. ACM Trans Knowl Discov Data 2007;1(1):2–es.

[44] Turk G, Levoy M. Zippered polygon meshes from range images. In: Proc. 21st annu. conf. comput. graph. interact. tech. 1994, p. 311–8.

[45] Perraudin N, Paratte J, Shuman D, Martin L, Kalofolias V, Vandergheynst P, et al. GSPBOX: A toolbox for signal processing on graphs. 2014, arXiv preprint arXiv:1408.5781.