Maastricht University

Master Research Project Data Science & AI

Semester 2 2022/2023

Exploring Audio Data with Recurrence Plots: β-Divergence, Visualisation Techniques, and Computational Methods

Group 14

Sree Showrya Kotala	i6206796
Amirul Karim Tanim	i6287477
Muhammad Danial Khan	i6339391
Jean William Petronella Marie Janssen	i6211969

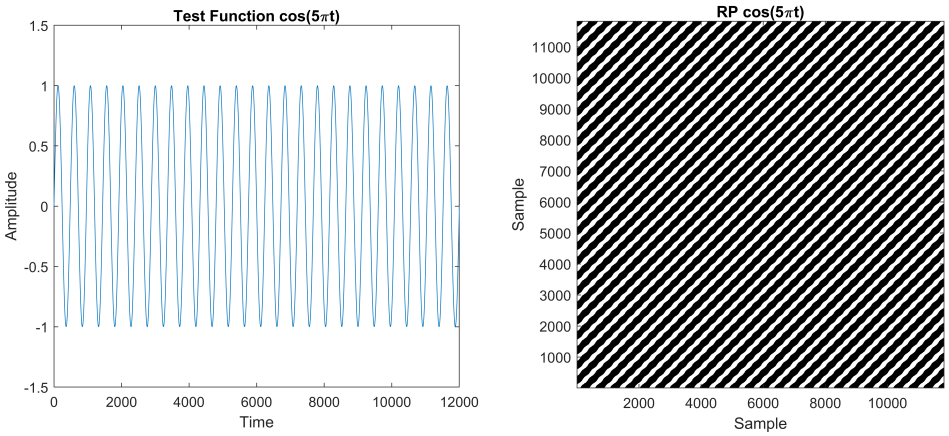
Department of Advanced Computing Studies

Supervisors: Martijn Boussé, Philippe Dreesen

April 1, 2024

1 Introduction

A recurrence plot (RP) is a visual tool used in nonlinear data analysis. It serves as a visual representation, taking the form of a graph represented by a square matrix. In this matrix, the elements correspond to specific instances when a particular state of a dynamic system repeats itself. The columns and rows of the matrix represent pairs of specific times. In technical terms, the RP enables us to identify and visualize the instances when the trajectory of a dynamic system in phase space revisits approximately the same region. By examining the RP, we can observe patterns that indicate when the system's trajectory tends to occupy similar areas in its phase space. This information is valuable in understanding the repetitive behavior and characteristics of the dynamic system under consideration.



(a) Time domain plot (b) Euclidean distance RP

Figure 1: (a) Time domain plot of $cos(5\pi t)$. (b) Recurrence Plot (RP) of the signal after thresholding. The black line segments indicate the presence of recurrent patterns or repetitions in the underlying time series data.

RP-based methods have found widespread applications in numerous fields of research Marwan et al. (2007). While they are particularly popular in physiology, where they have been used to analyze heartbeat intervals, monitor diseases, and detect cardiac arrhythmia precursors, their utility extends to neuroscience, genomics, ecology, physics, chemistry, earth science, astrophysics, engineering, and economics. In neuroscience and life sciences, RPs and Recurrence Quantification Analysis (RQA) have been employed to study various physiological processes. They have been applied to diverse data sources such as electromyography, eye movements, postural fluctuations, EEG, neuronal signals, and voice streams. The analysis of DNA sequences using RQA has revealed intriguing long-range correlations, while other studies confirm such patterns in different DNA sequences. In earth science, RPs have been used to compare palaeo-climate and modern rainfall data, align time scales in geophysical profiles, and estimate climatic variables. Economists have employed RPs and RQA to investigate economic time series, such as exchange rates and stock indices, for chaos, correlations, and prediction purposes. Overall, RP-based methods offer versatile tools for visualizing and analyzing complex systems in various scientific disciplines, enabling deeper insights and facilitating advancements in our understanding of the underlying phenomena.

RP analysis can be used as a tool to detect stuttering in audio data. Stuttering refers to a speech disorder when the flow of speech is interrupted by involuntary pauses and repetition of sounds. Stuttering diagnosis involves a speech-language pathologist (SLP) who examines the individual's medical, developmental, and family history. The SLP assesses speech and language skills, including fluency, voice quality, articulation, and grammar, using standardised tests, observations, and audio/video recordings. SLPs use speech analysis software to analyse speech influences, voice quality, and articulation, aiding in diagnosis and severity determination. However, these tools cannot detect pauses in speech, requiring SLPs to review video data alongside software analysis. Supplementary data is often used to ensure a comprehensive diagnosis. The software-generated plots and trends provide valuable insights for SLPs in treating speech and language disorders. By examining the recurrence patterns of a speech signal, it may be possible to identify repetitions, hesitations, and other temporal disruptions that are indicative of stuttering. The recurrence plot can also reveal patterns of speech disfluency that are not necessarily perceptible to the human ear. After an RP is constructed for each window, it can then be analysed using various RQA measures

to extract features that are indicative of stuttering. These features can then be used to train machine learning models to detect and classify stuttering in audio data automatically. Overall, recurrence plot analysis may offer a promising approach to detecting and characterising stuttering in speech signals.

In summary, this project aims to find the usefulness of RPs in audio data analysis specifically focusing on potential for stuttering diagnosis. However, due to the complex nature of audio data, the focus had to be diverted toward finding efficient visualisation and computational methods to generate RPs. Advantages of using β-divergences as a distance metric are explored as well.

2 Context and Research Questions

2.1 Problem statement

Visualizations are often more comprehensible and user-friendly than raw data, making them a preferred method of interpretation for many individuals. Repetition of a specific event is a very common scenario in practical systems. By applying RPs, we can easily generate an intuitive visualization that helps to identify the recurrences or repetition of dynamical systems. These recurrences show up as lines or artifacts (single block not forming a line) in visualization as seen in Figure 1. In the context of stuttering analysis, we aim to employ visual tools to identify instances of recurrence in speech and to distinguish between stuttered and fluent speech based on visual characteristics. Due to the intricate nature of speech signals, we face three major issues as described below:

1. Computational tractability: Speech signals have a high sampling rate which results in a large sample length. Calculation of RPs for large sample lengths becomes intractable in terms of time required and memory overload.

2. Due to large sample lengths, generating a visual representation of RPs becomes unmanageable.

3. 'Classical' RPs which employ Euclidean distance as a distance metric do not reveal noticeable information for audio data. This calls for an alternative method of using β-divergences as a means to reveal more information from RPs.

The emphasis on visualisation and computational approaches is driven by the need to simplify the understanding and analysis of RPs for complex speech signals in stuttering research. Another focus of the project was to explore the characteristics of Recurrence Plots via β-divergences (RP-BD) when applied to audio signals. Our definition of a 'classical' RP uses a Euclidean distance when computing the (dis)similarity between ordered segments of a signal to itself. This yields a symmetric visualisation - one that is mirrored along the main diagonal. We wanted to investigate an alternative methodology in which an asymmetric distance such as β-divergence is used. An asymmetric distance would not only allow for information to be more densely represented but potentially induces more interesting relationships and noticeable characteristics to reveal themselves. Empirical evaluations would help in identifying patterns and characteristics that emerge from the visualisation. Meanwhile, qualitative metrics were computed to help provide more concrete evidence of the identified characteristics and the degree to which they occur.

2.2 Research questions

The focus of the project is to investigate the contribution (if any) of the RPs in audio data analysis in stuttering research. Exploration of β-divergence as a metric and finding efficient visualisation and computational methods was a major part of the project. Based on this, we propose the following research questions:

1. Do RP-BD reveal more information on stutter data when compared to classical RPs?
   (a) What characteristics vary in the visualisations from RP-BDs compared to classical RPs?
   (b) How well are the RP-BDs able to tackle noise in audio recordings to an equivalent degree as classical RPs?

2. How useful are available audio datasets for stuttering in helping to reveal any additional information RP-BDs provide over classical RPs?

3. How can the qualitative analysis of RPs be used for the diagnosis of stuttering?
   (a) What are the qualitative characteristics to stutter data and how can we identify them?
4. Is it possible to develop efficient visualisation techniques and computational methods for RPs given the complex nature of audio signal?

3 Literature review

Looking at the state of the art there are three main approaches to tackling the problem of detecting and classifying stuttering. Table 1 summarises the findings of Sheikh et al. (2022) that are most relevant to our project.

Most of the existing works detect and identify stuttering either by language models or by Automatic Speech Recognition (ASR) systems. Usually, these systems first convert the audio signals into their corresponding textual form and then apply language models to detect or identify stuttering. Pálfy & Pospíchal (2011) reported the best recognition rates using Radial Basis Function (RBF) (96.4% accuracy) and linear Kernel (98% accuracy) as Kernel functions in Support Vector Machine (SVM) respectively. With the recent developments in convolutional neural networks, the feature representation of stuttered speech is moving towards spectrogram representations from conventional MFCCs (Sheikh et al., 2022). Kourkounakis et al. (2020) exploited the use of spectrograms (as a greyscale image) as a sole feature extractor for stutter recognition. They achieved state-of-the-art performance in identifying stuttering events. Among the Deep Learning-based Automated Stutter Identification Systems (ASIS) the FluentNet classifier proposed by Kourkounakis et al. (2021) and the spectrogram feature representations of stuttered speech above are the most effective. However, for a large set of stuttered speakers, StutterNet proposed by Sheikh et al. (2021) is the most effective one.

We want to highlight that although the performances for a machine learning (ML) / neural network (NN) technique are high, the methods are not only complex to the layman, but the black box nature of the approach makes their decisions hard to explain. Although a statistical approach could also be considered for our project focus to speed up the workflow of diagnosis of stuttering, we instead chose to explore the first track of the feature extraction approach.

To allow for features to be extracted that are intuitive and explainable to a medical practitioner, we propose the use of RPs. Visualisations from RPs help provide a qualitative analysis that highlights the recurrences of dynamic systems. There are currently few papers on the qualitative interpretation of the visualisations from RPs. The outputs can also then be further processed with recurrence quantification analysis (RQA) to extract quantified measures of the properties and characteristics of the plots and used in machine learning approaches. Marwan & Kraemer (2023) bring to light how the integration of RP with ML techniques has revolutionised the analysis of dynamical systems: enabling more accurate and efficient identification of patterns and features in complex data. The project aims to conduct a similar analysis using RPs and RQMs to more intuitively convey the behaviour of the dynamical system.

As highlighted in Section 2.1, a classical RP uses a Euclidean distance measure, this is a symmetric distance that results in redundancy in the lower left portion of the visualisations. To more densely encode information, we propose using an asymmetric divergence. Namely, the β-divergence to measure the (dis)similarity when computing the visualisations from the RPs. As Euclidean distances and β-divergences are a subset of Bregman-Divergences as described by Olaya & Otman (2021) and Hennequin et al. (2010), by implementing a Bregman-divergence we are able to fairly compare the two divergences by adjusting the functions used in the Bregman-Divergence equation. This allows for us to use a parametric approach to the divergence measure in the computation of the RPs.

Furthermore, when generating RPs the visualisations are sensitive to the parameters used. Marwan et al. (2007) indicated methods to derive the appropriate values for the parameters m, τ, and ε. In addition, they provide insight into the qualitative interpretation of RPs visualisation and will be used to justify the qualitative analysis we conduct on the stutter data.

The focus of the project is to provide qualitative and quantitative analysis for RPs generated on stutter data, doing so while comparing the added benefits (if any) of an asymmetric divergence i.e. the β-divergence. From these RPs, we can compute RQMs and investigate their contribution to the detection and classification of stuttering in individuals. We want to be able to propose a solution that is intuitive for a practitioner to interpret and the RQM metrics provide a qualitative measure to support and speed up the diagnosis of an individual's degree of stuttering by the practitioner.

Approach	Research Works
Feature-based approach	• Acoustic-based feature extraction methods are one of the common techniques used in stuttering detection. Mahesha et al. compared LPC, LPCC, and MFCC for syllable repetition, word repetition, and prolongation and showed that LPCC-based multi-class SVM (92% accuracy) outperforms LPCC (75% acc.) and MFCC(88% acc.) based SVM stutter recognition models. • On the other hand, Hariharan et al. elaborated on the effect of LPC, LPCC, and WLPCC features for stuttering (repetition and prolongation only) recognition events. They concluded that the WLPCC feature-based stuttering recognition models outperform LPC and LPCC. • Arjun et al. used LPC and MFCCs as input features and concluded that MFCCs perform better than LPCs.
Statistical approach	• Hidden Markov Models (HMM) has been the center of contemporary speech recognition systems and hence have been extended successfully to disfluency classification. Tan et al. used 12 MFCC features with HMMs and achieved an average recognition rate of 93%. This tool recognizes only normal and stutter utterances and is not classifying different types of disfluencies. • Wisniewski et al. used Euclidean distance as a codebook based on 20 MFCCs with HMMs. They reported an average recognition rate of 70% for two stuttering classes including blocks and prolongation.
Machine Learning and Neural Network Approach	• Chee et al. presented an MFCC feature-based KNN and LDA classification model for repetition and prolongation types of disfluencies. The models report the best average accuracies of 90.91% for KNN (with k=1) and 90.91% for LDA on the UCLASS dataset. • Santoso et al. proposed modulation spectrum feature-based Bidirectional LSTM (BiLSTM) to detect the causes of errors in speech recognition systems. They further improved the system by introducing an attention-based BiLSTM classifier for stuttering event detection. • Kourkounakis et al. proposed a FluentNet which reported an average accuracy of 91.75% and 86.7% on UCLASS and LibriStutter datasets respectively. Additionally, Sheikh et al. recently proposed a StutterNet which performs well with a large set of stuttered speakers. • Lea et al. curated the SEP-28K dataset and showed that increasing the amount of training data improved relative stutter detection performance by 28%.

Table 1: Summary of contemporary approaches in stuttering identification

4 Background Concepts

Recurrence plots (RP) are a visualisation technique used in nonlinear time series analysis to study the structure and dynamics of a system. Recurrence plots are constructed by representing the pairwise distances between segments in a time series using binary values, where a value of 1 indicates that two segments are close in space and time, and a value of 0 indicates that they are far apart. In essence, an RP is a square matrix $RP_{i,j}, i, j = 1, ...N$. Using the pre-determined threshold T, we can calculate the element $RP(i, j)$, and when this element is equal to 1, the distance between $x(t_i)$ and $x(t_j)$ exceed T. If this is not the case the value of $RP(i, j)$ is 0. The following parameters are used in the construction of recurrence plots:

- Time delay (τ): This parameter represents the time lag between two segments in the time series that are considered for comparison. In the construction of RP, τ determines the diagonal structure of the plot.

- Embedding dimension (m): This parameter represents the number of dimensions used to embed the time series in phase space. In the construction of RP, m determines the size of the square matrix used to represent the RP. Higher embedding dimensions allow for a more detailed representation of the system's structure, but may also introduce noise and computational complexity.

- Threshold (T): A threshold value T is chosen to determine which pairs of segments are considered recurrent. Pairs of points whose distance falls below the threshold value are

considered to be recurrent and are represented as black dots in the RP. On the other hand, pairs of segments whose distance is greater than the threshold value are not recurrent and are represented as white dots in the RP.

- (Auto) correlation: This parameter represents the degree of similarity or dependence between segments in the time series. In the construction of recurrence plots, the (auto)correlation of a segment and its surrounding segments can produce patterns indicative of the behaviour of the underlying structures or dynamics of the system.

As previously mentioned in Section 2.1, an asymmetrical RP could provide a different representation of information. In order to investigate this, β-divergence is introduced. "Févotte & Idier (2011) described β-divergence as a family of cost functions parameterised by a single shape parameter β that takes the Euclidean distance, the Kullback-Leibler divergence, and the Itakura-Saito divergence as special cases ($\beta = 2$, 1, 0 respectively)." The β-divergence can be defined as the following:

$$d_\beta(x|y) \stackrel{def}{=} \begin{cases} \frac{1}{\beta(\beta-1)}(x^\beta + (\beta-1)y^\beta - \beta x y^{\beta-1}) & \beta \in \mathbb{R}\backslash\{0,1\} \\ x\log\frac{x}{y} - x + y & \beta = 1 \\ \frac{x}{y} - \log\frac{x}{y} - 1 & \beta = 0 \end{cases} \tag{1}$$

In these formulae, x and y represent the point on matrices X and Y that represent a sound signal. β-divergence will be used to generate RPs within the research, but its added value has to be studied.

Once the suitable parameters have been chosen, the next stage is to construct the RP. The construction process of the RP begins by embedding the time series data into a higher-dimensional space utilising the chosen embedding dimension and time delay. To compute the distances between points, two distance metrics will be used, Euclidean distance and β -divergence. The resulting distances are then plotted, after the threshold is applied, as black dots on a two-dimensional grid, with each point on the grid corresponding to a pair of segments in the time series. Segments that are close to each other in the time series will appear as clusters of black dots on the RP, while segments that are far apart in the time series will be scattered throughout the RP. The resulting RP provides a visual representation of the recurrence structure of the time series, allowing for the identification of patterns and features that may not be apparent from the original time series. After this, a comparative analysis between the two metrics will be performed to identify any significant differences that can be utilised for analysis purposes.

5 Methodology

5.1 Dataset and Pre-processing

The end goal of this project is to explore the potential of a Recurrence Plot (RP) in qualitative and quantitative analysis of speech stuttering. Hence, a major part of the initial research incorporated finding a suitable stuttering dataset. Multiple datasets were found and the LibriStutter dataset was proposed as one of the potential datasets to start with (Kourkounakis et al., 2021). It is a synthesised stuttering dataset curated from the LibriSpeech dataset (Panayotov et al., 2015). The LibriSpeech is an Automated Speech Recognition (ASR) dataset based on public-domain audio books, while the LibriStutter dataset is synthesised with five different types of stuttering incorporated into a small part of the LibriSpeech dataset. The details of the pros and cons of different datasets are described in Appendix A.1.

Recurrence Plot in the context of speech data and stuttering is an unexplored research territory. The attempt of constructing RPs from the speech data of the LibriStutter dataset was hindered by various challenges including computational intractability and lack of resources for interpretation. The computational challenges diverted the focus of the project to finding efficient methods for visualisation and computation. On the other hand, the challenges of utilising the dataset were handled by curating a small dataset by incorporating both the LibriSpeech and the LibriStutter datasets. Twenty sets of corresponding audio signals from the two datasets were curated. A set includes two signals: one from the LibriSpeech dataset (the non-stutter signal) and the other corresponding signal from the LibriStutter dataset (the stutter signal). This small dataset includes four types of stuttering: sound repetition, word repetition, phrase repetition, and prolongation. It should be noted that to handle the computational intractability of calculating RPs, the complete audio signals from the datasets were not used. Instead, different stuttering and corresponding

non-stuttering segments were extracted by manually listening to the signals. This was done using the free open-source Audacity software.

Both datasets are fairly noise free. Hence, noise reduction was not necessary. However, the sampling rate for LibriSpeech and LibriStutter datasets are 16 kHz and 22 kHz respectively. This mismatch is resolved by downsampling the LibriStutter signals. The motivation behind curating a dataset using the fluent vs stutter datasets stem from the fact that it makes it easier to interpret the RPs for corresponding utterances. However, it is not expected that such corresponding data will be available in a practical setup when diagnosing stuttering. This dataset is solely curated to better interpret and compare the RPs of stutter vs non-stutter signals for the same utterance.

5.2 Blocking Method

Traditionally, RPs are calculated using loops to compare the similarity between each pair of points in the time series data. The process involves iterating through the data points and computing the distance or dissimilarity between each pair. This is done by comparing the values of the data points at different time steps or dimensions. The computed distances are then used to construct the RP matrix. The loop-based approach involves nested iterations over the time series data, making it a computationally intensive process, especially for large datasets. For each pair of points, the loop calculates the distance using a distance metric such as Euclidean distance or other dissimilarity measures. This process is repeated for every pair, resulting in a dense matrix that represents the recurrence plot.

Converting the traditional method of calculating RPs into a matrix operation using Kron products can significantly improve computational efficiency. The Kron product allows for the simultaneous computation of distances between pairs of points, eliminating the need for explicit loops. By exploiting the properties of matrix operations, the computation can be paralleled and optimised, leading to faster execution times and reduced memory usage. Instead of comparing each pair of points individually, the Kron product enables the calculation of distances for multiple pairs simultaneously. This is achieved by generating expanded matrices that contain repeated copies of the data points. By performing element-wise operations on these expanded matrices, the distances between pairs can be computed in a matrix format.

In the realm of optimising computation and parallelisation techniques, the primary focus has been on efficiently generating RQAs. Due to the computational complexity associated with long time series, generating complete RPs have often been omitted. Instead, RQAs are constructed using the symmetry property of RPs, utilising either the upper or lower diagonal to compute the necessary RQA measures (Martinovič & Zitzlsberger, 2018). Several techniques worked on approximation, techniques described by Rawald et al. (2014), Fukino et al. (2016) and another technique employed was using randomly selected microstates, that are very small subsets of RPs and statistically calculating the RQAs from them (Froguel et al., 2022). To further enhance the efficiency of RQA computation, blocked-based approaches have been adopted, leveraging parallelization techniques and the power of Graphics Processing Units to accelerate the process(Rawald et al., 2014),(Rawald et al., 2017).

The audio data under consideration posed a significant challenge due to its extremely large size, resulting in impractical run times and excessive memory usage when employing traditional distance metric calculation techniques or even matrix techniques. Attempting to process such voluminous data led to issues such as memory overflows or run times that extended up to half an hour (specification in Section 2). This became particularly problematic when dealing with longer signals or higher embedding dimensions, impeding a comprehensive analysis of the raw audio. One potential approach that was explored to tackle this challenge involved downsampling the data, aiming to reduce the number of data points. However, this strategy often resulted in a notable degradation of the signal quality, causing the key characteristics of the reconstructed phase space RPs to diminish. Consequently, this compromised the accuracy and reliability of the subsequent analysis. The emphasis on asymmetric divergences in the current scope has limited the utilisation of the symmetry property inherent in RPs. As a result, the potential benefits of exploiting the symmetry property have not been fully explored. Therefore, it became crucial to find an alternative solution that could overcome the limitations imposed by the large dataset, ensuring both efficient computation and preservation of the essential signal features for reliable analysis.

To address these challenges, a systematic approach was devised as Algorithm 1, aiming to divide the complex problem into smaller, more manageable segments. To begin, a suggested block size is determined, which serves as an initial estimate for dividing the embedding matrix. The length of the embedding matrix is then calculated to determine its size. Based on the suggested block size, the number of blocks is determined by dividing the size of the embedding matrix. Next, the

block size is recalculated to ensure an even division of the embedding matrix. This adjustment guarantees that each block has a consistent and balanced representation of the data. Once the block size is finalised, the embedding matrix is generated by dividing it into blocks according to the updated block size.

For each block, the Algorithm performs the Kronecker(Kron) product operation. This operation involves element-wise multiplication between the current block and a matrix of ones, as well as between a matrix of ones and the current block. By applying the Kron product in both directions, element-wise differences can be taken in a more systematic and optimised manner, further reducing computational complexity.

Now, the Algorithm considers combinations of blocks. For each combination, it calculates the distance between the blocks using a specified distance metric, such as Squared EU distance, IS-divergence, β-divergence, or KL-divergence. In the case of squared EU distance, a simple element-wise difference is performed. This step provides insights into the relationships and similarities among different segments of the embedding matrix.

The distance matrix computed in the previous step is summed along the embedding dimension, resulting in a single block that represents the combined information from the individual blocks in the combination. This summation aggregates the data and allows for a holistic view of the embedding structure. The data is then reshaped to the size of the block to create a plot between the two blocks. The resulting plot is then added to its designated location to complete plot, which represents the final distance plot of the complete signal. This sub-region plot visually represents the composition and patterns present in the original data, providing a comprehensive overview of the embedding matrix and the signal.

To ensure positive values in the graph, the Algorithm takes the absolute value of the complete plot. Finally, the values in the complete plot are normalised to bring them within a desired range or scale. This normalisation step facilitates better interpretation and visualisation of the data.

Algorithm 1 Blocking Method

procedure BLOCKING METHOD(Signal X, Embedding m,Time delay τ,Threshold T, $SuggestedBlockSize$)

Calculate the length of the Input Signal X

$$N = length(X)$$

Calculate the length K of the embedding matrix.

$$K = N - \tau(m-1)$$

Determine the $NumberOfBlocks$

$$NumberOfBlocks = ceil(K\ /SuggestedBlockSize)$$

Recalculate the $BlockSize$ based on the number of blocks.

$$BlockSize = floor(K\ /NumberOfBlocks)$$

Generate the Embedding Matrix Blocks H.

Iterate over each $EmbeddingMatrixBlocks$:

Generate the $ForwardKroneckerProduct$ of the current block with ones.

$$KronF = Kron(ones(BlockSize, 1), H)$$

Generate the $ReverseKroneckerProduct$ of ones with the current block.

$$KronR = Kron(H, ones(BlockSize, 1))$$

For Each combination of $KronBlocks$

Calculate the $Distance$ between the $Kronblocks$.

$$D_{ij} = \| \vec{}KronF_i \ - \vec{}KronR_j \ \|$$

Sum the $Distance$ matrices along the embedding dimension.
Reshape the result into a $SquarePlot$.
Add the resulting $SquarePlot$ to its designated location in the $CompletePlot$.

Take the absolute value of the $CompletePlot$.
Normalise the values in the $CompletePlot$.
Apply thresholding T

Return : $ReccurencePlot$

end procedure

5.3 Compression

After successfully addressing the challenge of efficiently computing RPs a new obstacle emerged when attempting to perform further analysis, such as computing RQAs. This hurdle stemmed from the large sizes of the RPs, making it challenging to conduct meaningful comparisons between audio data. Comparing audio data typically required a significant number of samples, often with

high sampling frequencies. As a result, the resulting RPs were extremely large. Although downsampling initially seemed like a potential solution, it was soon discovered that this approach could significantly alter the signal, leading to a loss of key characteristics in the resulting RPs.

To overcome this challenge, Algorithm 1 was enhanced by introducing a compression method to compute summaries of regions instead of individual sample points. This approach involved dividing the data into smaller windows or blocks, allowing for the compression of RPs within each block into a single sample. This compression greatly reduced the overall size of the data while still preserving many essential features from the larger neighbourhood of the original block.

The primary distinction in this Algorithm 2 lies in the post-processing step following the calculation of the distance matrix, similar to Algorithm 1. However, instead of reshaping the computed distance matrix to form a distance plot for the sub-region, a novel approach is employed. This approach involves aggregating the information of the entire region into a single data point, effectively compressing the plot. The degree of compression applied to the plot is determined by the size of the compression window or Block size. By utilising larger windows, larger regions can be compressed, leading to smaller overall plots. This compression technique allows for the representation of the entire region's characteristics in a more concise manner, enabling a reduction in the overall size of the plot.

By implementing this modified approach, it became possible to compute summaries of regions and effectively reduce the size of the RPs. This method proved advantageous in preserving crucial information while mitigating the computational challenges posed by the large size of the data.

Algorithm 2 Compression Method

procedure Compression Method(Signal X, Embedding m,Time delay τ,Threshold T, $SuggestedBlockSize$)

Calculate the length of the Input Signal X

$$N = length(X)$$

Calculate the length K of the embedding matrix.

$$K = N - \tau(m-1)$$

Determine the $NumberOfBlocks$

$$NumberOfBlocks = ceil(K\ /SuggestedBlockSize)$$

Recalculate the $BlockSize$ based on the number of blocks.

$$BlockSize = floor(K\ /NumberOfBlocks)$$

Generate the Embedding Matrix Blocks H.

Iterate over each $EmbeddingMatrixBlocks$:

Generate the $ForwardKroneckerProduct$ of the current block with ones.

$$KronF = Kron(ones(BlockSize, 1), H)$$

Generate the $ReverseKroneckerProduct$ of ones with the current block.

$$KronR = Kron(H, ones(BlockSize, 1))$$

For Each combination of $KronBlocks$

Calculate the $Distance$ between the $Kronblocks$.

$$D_{ij} = \| \bar{}KronF_i - \bar{}KronR_j \|$$

Sum the $Distance$ matrix.

Add the resulting $Point$ to its designated location in the $CompletePlot$.

Take the absolute value of the $CompletePlot$.

Normalise the values in the $CompletePlot$.

Apply thresholding T

Return : $ReccurencePlot$

end procedure

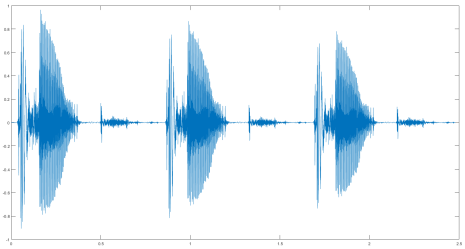
5.4 Recurrence Plots vs Spectrograms

KL-divergences are commonly used in audio applications for being a scale invariant metric. Given that audio data is inherently prone to large variations in scales, a scale-invariant metric is vital to comparing differences between two samples in their content rather than their amplitude. A β-divergence also has this scale-invariant property. Classically, when visualising an audio signal an easy approach is to use a spectrogram. Spectrograms are robust, efficient and easily implemented, however, spectrograms are not scale invariant. Resulting in the amplitude of the signal playing a crucial role on the final output visualisation.

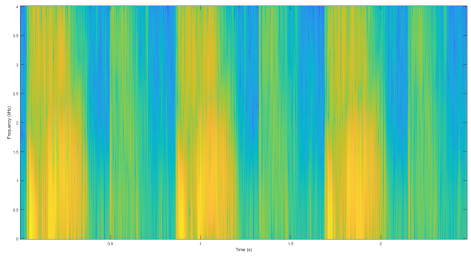
This is usually not an issue, however, the assumption was made that when an individual stutters, as they go through the process of stuttering that they begin to get slightly quieter. This reduction is amplitude would show as a varying spectrogram image, making it harder to distinguish if it's two

similar sounding words or the same word being stuttered. In a RP with a scale invariant metric such as the β or KL-divergence, we expect the structures of the image to remain consistent. A RP is also more intuitive and easy to interpret than a spectrogram. Individuals rarely need to think in terms of the frequency domain and how signals are a composition of various frequencies and thus the idea of 'recurrence' is easier to interpret.

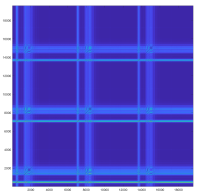
So to explore and show the benefits of using the β-divergence over a classical audio visualisation technique such as a spectrogram we have devised the following proof of concept. An audio recording is made of a word such as 'Pop'. This exact recording is then copied and attached to the original recording twice. Each repetition has it's amplitude reduced by a given decay %. E.g., with a 10% decay the first wave has an amplitude of 100% of the recording, the second has 90%, and the third has 81%. Similarly, in a 'perfect repetition decay' signal with a 20% decay, the first, second and third waves would have relative amplitudes of 100%, 80% and 64% respectively. An example of a 'perfect repetition decay' signal with a decay of 10% is shown in Figure **??**.



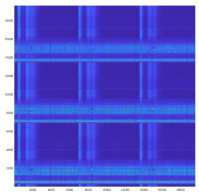
(a) Audio signal with a decay ratio of 10% - each repetition of the word has the amplitude decayed by 10% relative to the previous utterance of the same word.



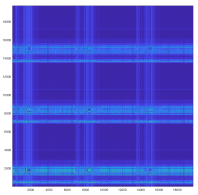
(b) Spectrogram of the perfect repetition 10% decay signal. Notice that the frequency content remains largely the same, but the brilliance of the frequencies is decaying as expected.



(c) Recurrence plot of the above signal computed with a squared Euclidean distance metric



(d) Recurrence plot of the above signal computed with a β-divergence metric



(e) Recurrence plot of the above signal computed with a Kullback-Leibler-divergence metric

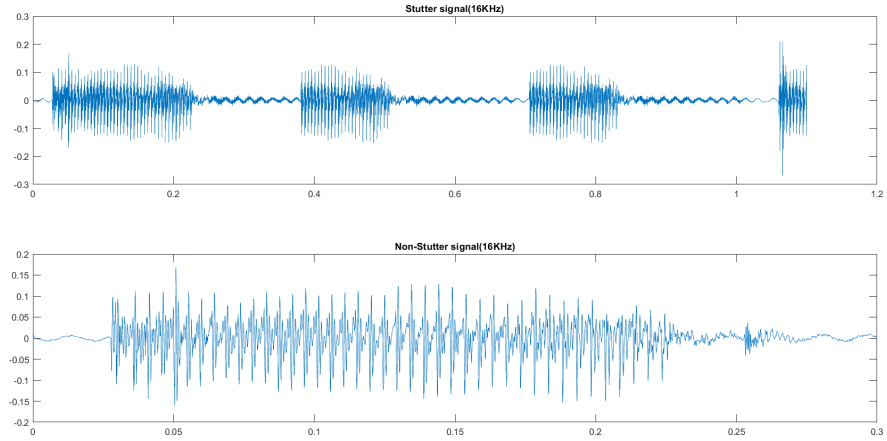
Figure 2: 'Perfect repetition decayed' example with decay of 10% as proof of concept to motivate the use of RPs over Spectrograms as RPs using β or KL-divergences are scale invariant. Note the asymmetric RPs produced by the β or KL-divergences in contrast to the symmetric RP produced by the squared Euclidean distance RP that showcases more information about the signal in the same plot window. At this decay level one is still able to tell that the word being repeated is the same one, hence is a person stutters but doesn't quiet down more than 20% in volume while doing so, a spectrogram could theoretically produce a very similar plot that could help with detecting stuttering. Meanwhile the RPs having the same structure of thin band followed by thicker band, where the band thicknesses stays consistent throughout the repetitions - showing that it is indeed the same word being uttered.

5.5 Interpreting Recurrence Plots

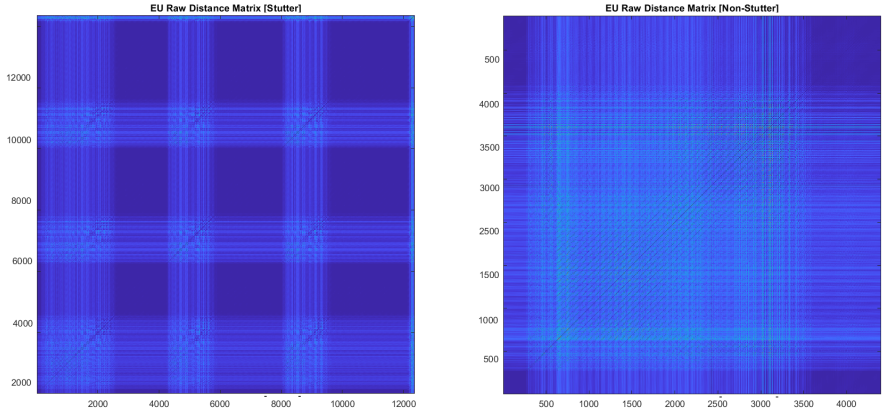
5.5.1 Qualitative Analysis

As mentioned earlier, one of the challenges was interpreting RPs generated from speech signals. Since this is an unexplored territory, no resources were found to interpret RPs for stutter signals. Our curated dataset of stutter vs non-stutter for the same utterance helped in this case. It was useful to do a qualitative analysis by comparing the corresponding stutter vs non-stutter signals. A comparative analysis is shown in Figure 3.

A visual inspection of the RPs in Figure 3b reveals a pattern of activity and inactivity. The regions where vertical and horizontal lines intersect represent the regions of activity. The empty regions with no lines represent the inactivity of speech. If we look closely, it is noticeable that a squeezed version of the non-stutter RP is present in the lower left corner of the stutter RP. The



(a) Time domain plot



(b) Euclidean distance RP

Figure 3: (a) Time domain plot of stutter vs non-stutter signal. It is evident that the stutter signal is basically a repetition of the squeezed non-stutter signal. (b) Euclidean distance RP of stutter vs non-stutter signal. A repeating pattern of crossed lines and empty regions is visible on the RP for the stutter signal.

rest of the part in stutter RP represents the recurring state of stutter. Further analysis will be discussed in section 6.5.1.

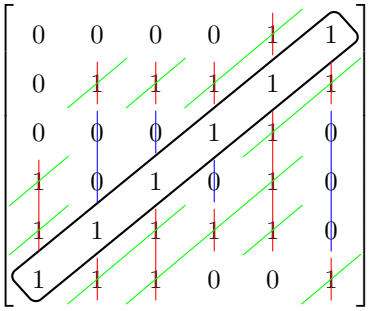
5.5.2 Quantitative Analysis

Recurrence Quantification Analysis(RQA) is a powerful tool used in the analysis of RP. RQA involves the use of various measures to quantify the different characteristics of the recurrence structure present in the RP. These measures can be used to extract meaningful information from the RP and provide insights into the underlying dynamics of the time series being analysed. To conduct a quantitative analysis the following subset of metrics provided in an open source RQA implementation based on the implementation by Marwan et al. (2007) were chosen. When calculating the metrics a parameter l_{min} represents the minimum length of contiguous 'recurrent' / 'non-recurrent' states for the portion to be considered a line, the default value for this length is 2. A Theiler window of size 1 is also applied, omitting the points along the Line of Identity (LOI) from being included in the metric calculation.

- Recurrence Rate (RR): The percentage of recurrence points in the RP, representing the proportion of time that the system spends in a recurrent state.
- Determinism (DET): The proportion of recurrent points that form diagonal lines of size at least l_{min}, representing the predictability or determinism of the system.

- Average diagonal length (L): The average length of the diagonal lines in the RP, representing the typical duration of recurrent states.
- L_{max} : The maximum length of diagonal lines in the RP
- Laminarity (LAM) : The proportion of recurrent points that form vertical lines at least of size at least l_{min}.
- Trapping Time (TT): The average length of vertical lines; represents time that the system spends within a recurrent state before transitioning to a non-recurrent state.
- V_{max} : Maximal length of the vertical lines in the RP
- Maximum White Vertical Line (RT_{max}) : Maximal length of the non-recurrent state: white in a monotone RP (blue in our figures) that does not run off the top or bottom of the RP.
- Recurrence time of second type ($T2$) : Average length of the white(blue) vertical lines

From Figure 3 shows an RP of a stutter signal where the individual stutters - particularly, stutter type 3: word repetition. The vertical and horizontal structures in the plot are synonymous throughout all stutter types and prompt us to use the vertical line metrics we've chosen. The regions of activity appear as square shaped blocks that form diagonal lines, hence we also chose to store these diagonal line metrics to help us quantise the patterns we are seeing. The white lines (blue lines in our visualisations) are only counted if they do not run off the top or bottom of the RP. We hypothesise that the white line metrics will help us determine the gaps between the region of activity and potentially let us dissociate stutter signals from non stutter signals. An example of the calculation of the RQAs we've selected is provided for the example matrix below.



First the LOI (black outline) is omitted from the calculations, next we compute the histograms for the diagonal, vertical, and white vertical lines highlighted by the green, red and blue lines respectively. Note that the white vertical lines are only consider 'inside the RP' if the endpoints of the white line are not at the top or bottom of the RP. Also note that $l_{min} = 2$ for these calculations.

Count	**1**	**2**	**3**	**4**	**5**	**6**
Diagonal Histogram	6	3	1	0	0	0
Vertical Histogram	8	2	1	0	0	0
White Vertical Histogram	2	1	1	0	0	0

We then calculate the metrics as follows.

$$RR = \frac{6 \times \mathbf{1} + 3 \times \mathbf{2} + 1 \times \mathbf{3}}{36 - 6} = \frac{15}{30} \tag{2a}$$

$$DET = \frac{3 \times \mathbf{2} + 1 \times \mathbf{3}}{15} = \frac{9}{15} \tag{2b}$$

$$L = \frac{3 \times \mathbf{2} + 1 \times \mathbf{3}}{3 + 1} = \frac{9}{4} \tag{2c}$$

$$L_{max} = \mathbf{3} \tag{2d}$$

$$LAM = \frac{2 \times \mathbf{2} + 1 \times \mathbf{3}}{15} = \frac{7}{15} \tag{2e}$$

$$V = \frac{2 \times \mathbf{2} + 1 \times \mathbf{3}}{2 + 1} = \frac{7}{3} \tag{2f}$$

$$V_{max} = \mathbf{3} \tag{2g}$$

$$RT_{max} = \mathbf{3} \tag{2h}$$

$$T2 = \frac{1 \times \mathbf{2} + 1 \times \mathbf{3}}{1 + 1} = \frac{5}{2} \tag{2i}$$

RQA metrics were computed on 5 different perfect repetition decayed signals with a decay of 10%, 20%, 30%, 40%, and 50%. RQA metrics were also computed on the dataset of stutter vs non stutter signals that was curated as described in Section 5.1.

6 Result and Analysis

6.1 Specifications

All experiments, from which the results were retrieved, were run on the following hardware:

1. CPU: AMD Ryzen 5 3600 6-core processor
2. RAM: 16 GB DDR4

6.2 Block Method

The implementation of the Blocking method was necessitated by the need for optimisation when constructing Recurrence Plots (RPs) from large audio samples. The conventional methods proved to be impractical, leading to either memory errors or prohibitively long execution times. These limitations posed challenges in visualising the data and conducting subsequent analyses on the RPs. The introduction of the Blocking method provided a breakthrough, enabling the visualisation of previously unattainable audio signals. Notably, the method exhibited significantly improved runtime compared to the traditional approaches that attempt to construct RPs in their entirety. Its flexibility, achieved through configurable block sizes, ensured efficient memory utilisation and enhanced computational efficiency. This proved particularly advantageous when dealing with larger datasets or higher embedding dimensions, which significantly increase the number of computational points involved.

To evaluate the effectiveness and optimisation of the Blocking method in terms of execution time, a series of experiments were conducted. Initially, individual parameters were examined to determine their impact on runtime. Through a systematic variation of the tau parameter from one to hundred in twenty-one steps, it was determined that tau had no discernible effect on the runtime. This experiment was repeated five times to account for any inconsistencies and smooth out the results.

Subsequently, the remaining parameters, namely the length of the signal, embedding dimension, β value, and block size, were subjected to more comprehensive experiments. These experiments aimed to explore the relationships between these parameters and the corresponding execution times.

Table 2: Parameters values used in the testing

Signal length	2000	3000	4000	5000	6000	
Embedding	2	4	6	8	10	12
Block Size	250	500	750	1000	Full	
β Value	0	.5	1	2		

This resulted in six hundred runs that were repeated five times to account for any inconsistencies and smooth out the results. It was observed that increasing the signal length or embedding dimension had a direct impact on the run times, as they significantly increased the number of computation points. Among the distance metrics used, the generalised β-divergence was found to be the most computationally expensive, while the Squared Euclidean distance was the least demanding. The run times for KL-divergence and IS-divergence were comparable.

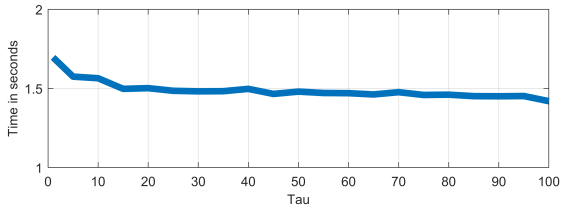


Figure 4: Run-time trend of τ indicating minimal effect

When comparing the run-times between different block sizes and the traditional matrix-based techniques, a consistent trend was observed for smaller signal lengths as seen in Figures 5a,5b,5c. The block sizes performed worse in general when compared to the traditional technique. The smallest block size of 250 performed the worst and the largest size block of 1000 performed the best but when embedding was sufficiently high, it performed only marginally better than the traditional technique. However, a deviation from this trend was observed for a signal length of 5,000 samples as seen in Figure 5d. In this case, all block sizes other than the smallest performed similarly, with comparable performance for lower embedding dimensions and notably improved performance for higher embedding dimensions. In the final experiment conducted on a signal length of 6,000 samples as shown in Figure 5e, for all the blocks other than the smallest, comparable results were obtained for lower embedding dimensions. However, as the embedding dimension increased, the difference in run times between the block method and the traditional matrix method became more pronounced. Even the overall worst-performing block performed remarkably well. Interestingly, a minimal variance was observed in the run times across different block sizes, even with increased embedding dimensions, when compared to the traditional matrix method.

6.3 Validity of Block Method

In order to assess the effectiveness of the proposed Blocking method, a comprehensive set of experiments was conducted, involving various types of signals. For each signal, Rps were constructed using four different distance metrics: squared Euclidean distance, IS-divergence, KL divergence, and generalised β-divergence. The objective was to compare the results obtained through the Blocking method with the outcomes of the conventional matrix based approach, where RPs are computed in their entirety. During the experimentation, different block sizes were tested within the Blocking method. No discrepancies were observed when comparing the computed RPs using the Blocking method against those obtained from the traditional approach. To validate this, a point-by-point comparison was performed between the RPs derived from both methods, resulting in a plot with solely zero values. This outcome indicated that the Blocking method yielded same results to the conventional approaches. Figure 6 shows the results of both methods side by side for a single signal, other results are visualised in Appendix A.2 .

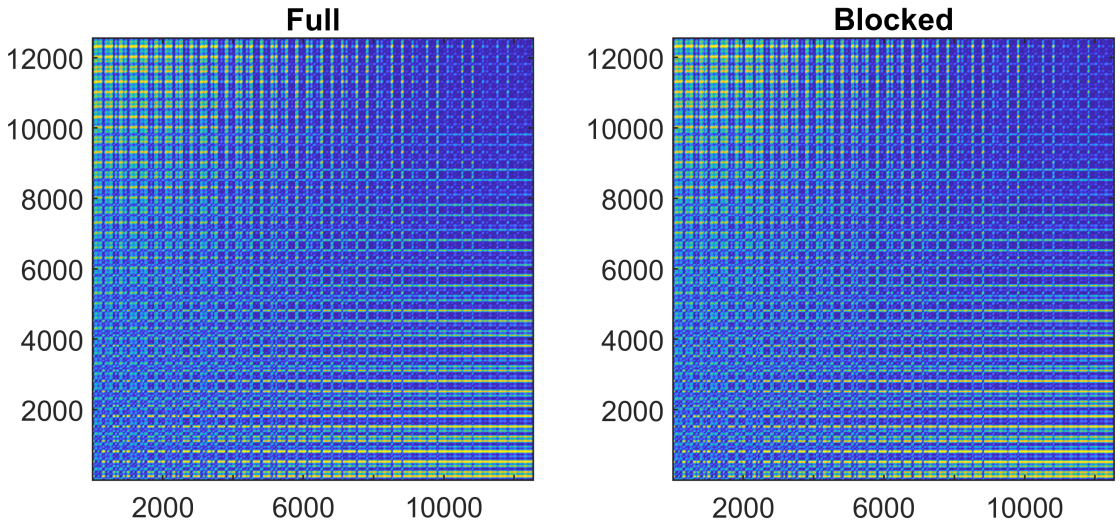
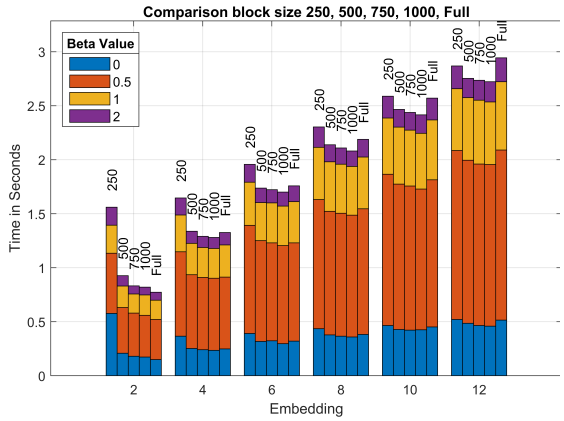
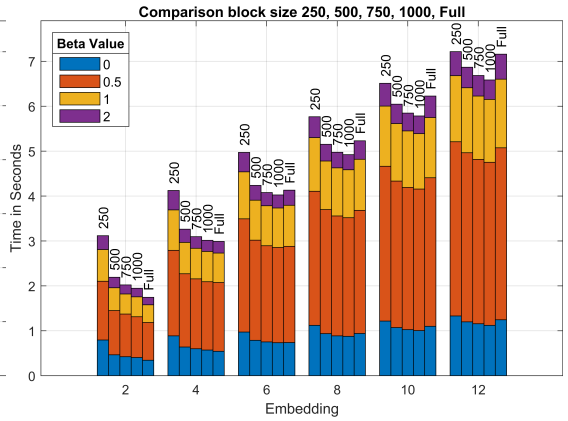


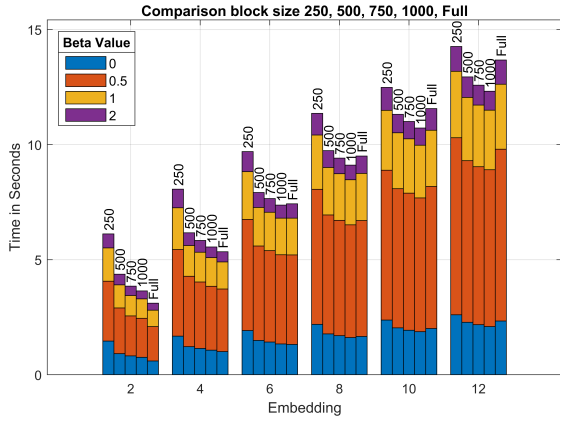
Figure 6: Distance plot comparison between Blocking method and traditional method, using block size of 500 and KL-divergence as the metric ($\beta = 1$)



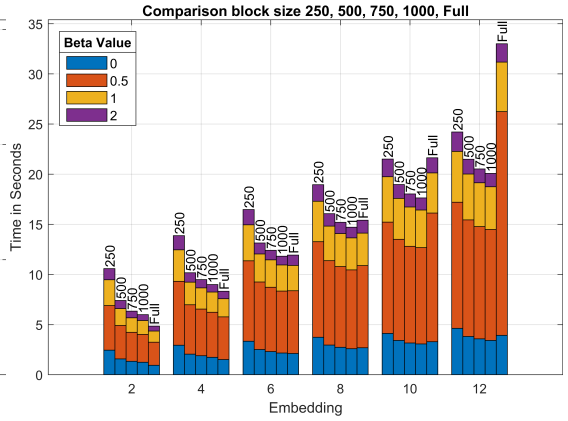
(a) Runtimes on signal size of 2000 samples



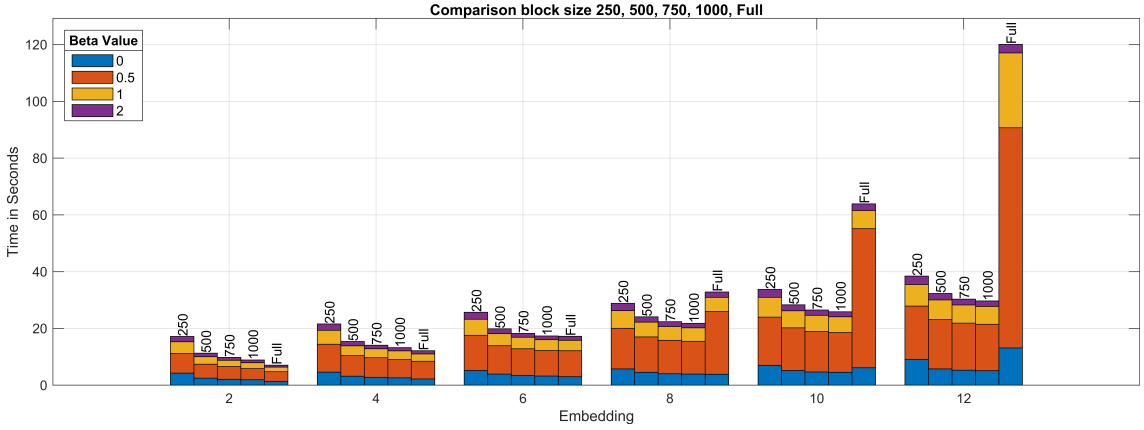
(b) Runtimes on signal size of 3000 samples



(c) Runtimes on signal size of 4000 samples



(d) Runtimes on signal size of 5000 samples

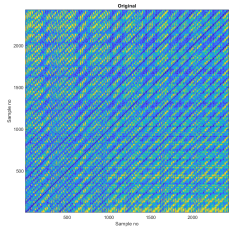


(e) Runtimes on signal size of 6000 samples

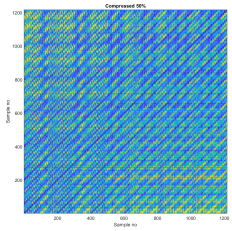
Figure 5: The cumulative runtime trend of four different values of β across multiple signal sizes, embedding dimension values, and various block sizes and its comparison to the traditional technique.

6.4 Effect of Compression vs Downsampling

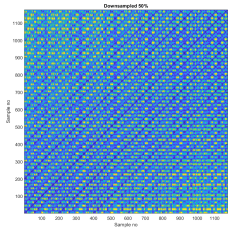
The blocking method takes care of the computational intractability of computing the RP matrix. However, some audio signals are still long enough that MATLAB faces memory overload when creating the RP image. To overcome this obstacle we are left with two potential solutions: Compression, or Downsampling the original signal. Downsampling significantly reduces the quality of the audio and the RPs and causes information loss. On the other hand, the compression method aggregates the information of the entire region into a single data point, compressing the plot while retaining most of the information. A comparative analysis for a complex toy signal $(cos(3\pi t) + sin(7\pi t) + 0.1t)$ is shown in Figure 7. It is evident from the figures that compressed RPs are more similar to the original RPs compared to the downsampled RPs. Analysis for another toy signal is available in Appendix A.3.



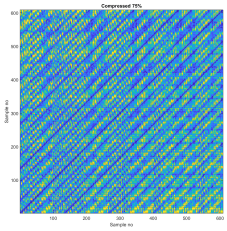
(a) Original RP



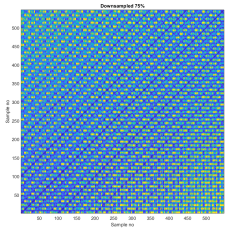
(b) 50% Compressed



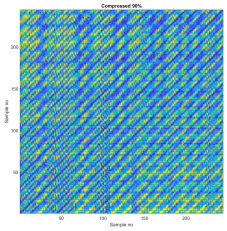
(c) 50% Downsampled



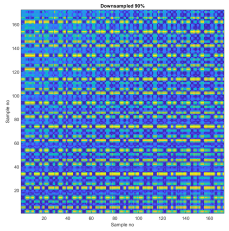
(d) 75% Compressed



(e) 75% Downsampled



(f) 90% Compressed

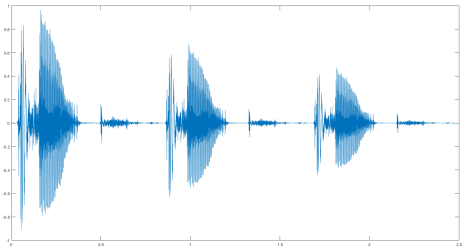


(g) 90% Downsampled

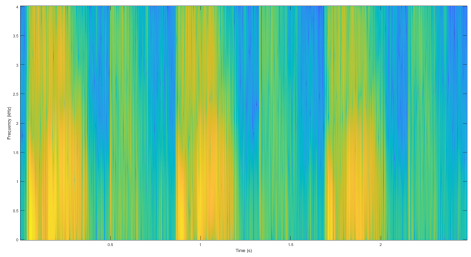
Figure 7: Comparative analysis of Compression vs Downsampling for $cos(3\pi t) + sin(7\pi t) + 0.1t$. (a) RP of the original signal without any compression or downsampling. (b - g) The rows show corresponding compressed and downsampled RPs. It is evident that the compressed RPs are closely similar to the original RP while the downsampled RPs lose features.

6.5 Recurrence Plot vs Spectrograms

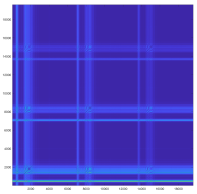
Figure 8 and Figure 9 show the same structure of results as shown previously in Figure 2. Notice that as the decay is increased. The trend is that the brilliance of the spectrogram decreases



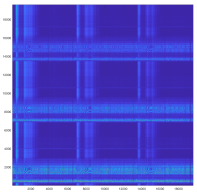
(a) Audio signal with a decay ratio of 30% - each repetition of the word has the amplitude decayed by 30% relative to the previous utterance of the same word.



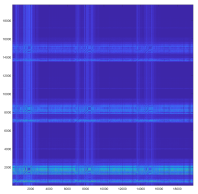
(b) Spectrogram of the perfect repetition 30% decay signal. Notice that the frequency content remains largely the same, but the brilliance of the frequencies is decaying to a severe enough degree to begin questioning if they are the same word or not.



(c) Recurrence plot of the above signal computed with a squared Euclidean distance metric



(d) Recurrence plot of the above signal computed with a β-divergence metric



(e) Recurrence plot of the above signal computed with a Kullback-Leibler-divergence metric

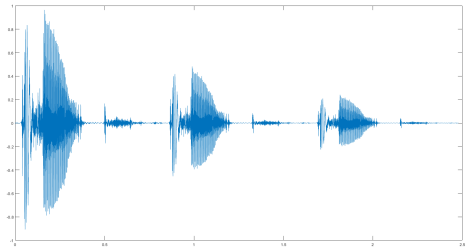
Figure 8: 'Perfect repetition decayed' example with decay of 30%. Observe that the asymmetric RPs produced by the β or KL-divergences in contrast to the symmetric RP produced by the squared Euclidean distance RP that showcases more information about the signal in the same plot window. At this decay level one can somewhat tell that the word being repeated is the same one from the spectrogram. One could also easily mistake is for a different word since the difference in brilliance of the frequencies makes it seem like a different word as we are no in the frequency domain. Meanwhile the RPs having the same structure of thin band followed by thicker band where the band thicknesses stays consistent throughout the repetitions showing that it is indeed the same word being uttered. even with a decay of 30% between signals

significantly. With a decay % of 50% we see the spectrogram becomes very hard to interpret whether the words are the same still or not. The 'classical' RP also has very decayed bands, although these bands are still viable to compute metrics from, they are heavily decayed as a result of the distance not being scale invariant. From Figure 9 we can also see that the RP-KL is showing heavier signs of decay than the RP-BD. Providing a strong case for using a β-divergence over a KL-divergence for this use case. Both are scale-invariant but the β-divergence suffers less from the effects of the signal decay and one can still interpret that the three words are indeed the same words.

From Figures 2, 8, 9 we can deduce that the RP outputs are mostly the same for lower %'s of decay, however, for larger decay values the β-divergence outputs are more robust to the scaling of the audio signals and thus should be used when feasible.

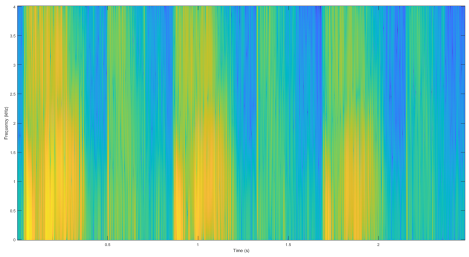
6.5.1 Qualitative Analysis

As discussed in section 5.5.1, the RP for the stutter signal shows patterns of recurring activity and inactivity which is not visible in the RP for the non-stutter signal. A comparative analysis is given in Figure 10. It should be noted that the curated dataset has 4 types of stuttering. However, the Phrase Repetition stuttering type incorporates very long signals. Hence, it was not possible to generate the RPs for this type of stuttering given our available computational resources. Moreover, it can be noticed the RPs in Figure 10e and Figure 10f are quite similar. This happens for the Prolongation type of stuttering. A manual auditory analysis of prolongation reveals that the audios are quite similar. Hence, a visual inspection of the RPs doesn't reveal much difference. However, the RQA reveals a distinct difference in this case which is explained in section 6.5.2.

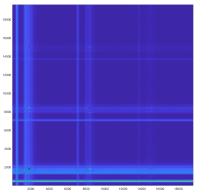


*

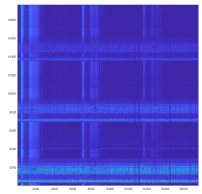
(a) Audio signal with a decay ratio of 50% - each repetition of the word has the amplitude decayed by 50% relative to the previous utterance of the same word.



(b) Spectrogram of the perfect repetition 50% decay signal. Notice that the frequency content remains largely the same, but the brilliance of the frequencies is no heavily decayed.



(c) Recurrence plot of the above signal computed with a squared Euclidean distance metric



(d) Recurrence plot of the above signal computed with a β-divergence metric

Figure 9: 'Perfect repetition decayed' example with decay of 50% Note the asymmetric RPs produced by the β or KL divergences in contrast to the symmetric RP produced by the squared Euclidean distance RP that showcases more information about the signal in the same plot window. The difficulty of gauging whether it is the same word or different similar words in the spectrogram is now much higher. Meanwhile the RPs with scale-invariant metrics having the same structure of thin band followed by thicker band where the band thicknesses stays consistent throughout the repetitions showing that it is indeed the same word being uttered. Also notice that as this degree the lack of the scale invariant property shows in the squared euclidean distance plot of subplot (c), but the scale-invariant divergences still show the structure we desire at this lower brilliance of the signal. Hence, providing a strong argument to use scale invariant divergences like β and KL when generating RPs.

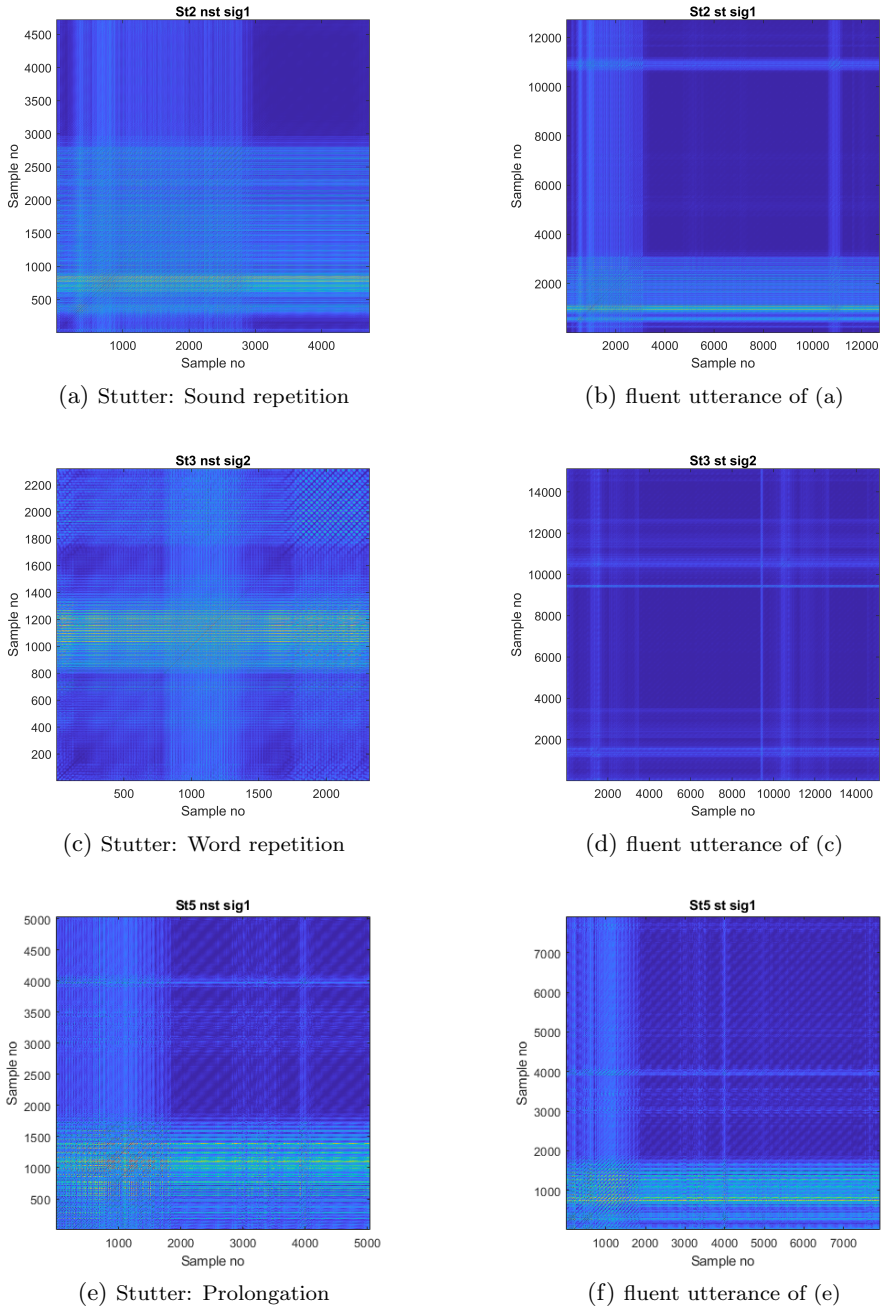
Table 3: RQA outputs of RPs computed using the squared Euclidean distance metric. First column indicates the metric that is being presented and the first row corresponds to the signal decay % of 'perfect repetition decay' signals used to generate the RPs. The table stands to provide a general insight to the trends of the RQAs as the signal has varying decay %s and to compare how similar outputs are in the perspective of RQA.

Euclidean-RQA	**10%**	**20%**	**30%**	**40%**	**50%**
RR	0.0682	0.0579	0.0495	0.0429	0.0378
DET	14.6530	17.2794	20.2102	23.3341	26.4762
L	9801.5000	9801.5000	9801.5000	9801.5000	9801.5000
L_{max}	19601.0000	19601.0000	19601.0000	19601.0000	19601.0000
LAM	14.6530	17.2794	20.2102	23.3342	26.4762
TT	9801.5000	9801.5000	9801.5000	9801.5000	9801.5000
V_{max}	19601.0000	19601.0000	19601.0000	19601.0000	19601.0000
RT_{max}	1.0000	1.0000	1.0000	1.0000	1.0000
$T2$	1.0000	1.0000	1.0000	1.0000	1.0000

6.5.2 Quantitative Analysis

From Tables 3, 4, and 5 we can spot some general trends. As the decay value increases, the RR decreases in all three instances. In addition, for divergences the L decreases but for the Euclidean distance L remains constant. L_{max} differs between the divergences and distance similarly to L, however it is interesting to see that for the Euclidean distance has L_{max} as the size of the plot.

LAM is also $\propto$ the decay %. Suggesting that as the signal decays the system stays in a 'chaotic' state more often. Unfortunately, the last two metrics we hypothesised to help have not produced any variations between the different divergence metrics and we deem this to be due to the noise that remains in the signal. The noise creates smalls recurrence patterns that stop these white



(a) Stutter: Sound repetition (b) fluent utterance of (a)

(c) Stutter: Word repetition (d) fluent utterance of (c)

(e) Stutter: Prolongation (f) fluent utterance of (e)

Figure 10: Comparative analysis of Stutter vs Non-stutter signals. The pairs in each row represent corresponding utterances of fluent speech and stuttered speech. The stutter types are sound repetition, word repetition, and prolongation. The patterns of activity and inactivity are more evident in the RPs of stutter signals. A closer look will reveal that the stutter RPs contain a squeezed version of the non-stutter RPs at the lower left corners.

lines from building into substantial lines within the blocks. This suggests that we require further pre-processing to reduce the background noise of voice recordings before creating the RPs to make use of the RT_{max} and $T2$ metrics more meaningfully. Additionally, the TT and V_{max} don't seem to vary much either and we deem this to be an inherent property resulting from the square structures seen in the RPs of stutter signals. The similarity in the KL and β metrics of TT and RT_{max} remains true up to 4 or 5 decimal places where they begin to differ slightly. Showing that the asymmetric scale-invariant divergences are both very similar but produce different results, further supporting the use of β-divergences for RPs.

Stutter vs Non-Stutter

From Table 6, it is interesting to note that when we have shorter duration stutters that relate to stutters on syllables, such as stutters of type 2 & 5 - sound repetition and prolongation respectively,

Table 4: RQA outputs of RPs computed using the β-divergence with $\beta = 0.5$. First column indicates the metric that is being presented and the first row corresponds to the signal decay % of 'perfect repetition decay' signals used to generate the RPs. The table stands to provide a general insight to the trends of the RQAs as the signal has varying decay %s and to compare how similar outputs are in the perspective of RQA.

β-RQA	**10%**	**20%**	**30%**	**40%**	**50%**
RR	0.09191	0.0834	0.0752	0.0681	0.0612
DET	10.8035	11.9063	13.1160	14.4894	15.9177
L	145.2685	145.2685	84.4588	84.4588	51.1775
L_{max}	3531.0000	3531.0000	3531.0000	3531.0000	3531.0000
LAM	10.8058	11.9088	13.1255	14.4999	15.9602
TT	9801.5070	9801.5070	9801.5130	9801.5130	9801.5244
V_{max}	19601.0000	19601.0000	19601.0000	19601.0000	19601.0000
RT_{max}	1.0000	1.0000	1.0000	1.0000	1.0000
$T2$	1.0000	1.0000	1.0000	1.0000	1.0000

Table 5: RQA outputs of RPs computed using the squared KL-divergence. First column indicates the metric that is being presented and the first row corresponds to the signal decay % of 'perfect repetition decay' signals used to generate the RPs. The table stands to provide a general insight to the trends of the RQAs as the signal has varying decay percentages and to compare how similar outputs are in the perspective of RQA.

KL-RQA	**10%**	**20%**	**30%**	**40%**	**50%**
RR	0.0825	0.0735	0.0651	0.0583	0.0515
DET	12.0343	13.5009	15.1451	16.9118	18.8987
L	140.9998	140.9998	82.5087	84.4588	51.1775
L_{max}	3531.0000	3531.0000	3531.0000	3531.0000	3531.0000
LAM	12.0368	13.5037	15.1560	16.92409	18.9492
TT	9870.5072	9870.5072	9870.5132	9801.5130	9801.0121
V_{max}	19739.0000	19739.0000	19739.0000	19601.0000	19601.0000
RT_{max}	1.0000	1.0000	1.0000	1.0000	1.0000
$T2$	1.0000	1.0000	1.0000	1.0000	1.0000

Table 6: The average RQAs for stutter vs non-stutter signals of three out of four stutter types. Stutter type 4 (phrase repetition) was omitted as the signals were too large to compute RQA metrics on the original signals without downsampling. Stutter 2 and 5 represent stutters of sound repetition and prolongation, while stutter 3 is sound repetition.

	Stutter 2	**Non-stutter 2**	**Stutter 3**	**Non-stutter 3**	**Stutter 5**	**Non-stutter 5**
RR	0,011582048	0,021031714	0,005538256	0,002657063	0,010674865	0,013758692
DET	86,34051562	51,5804154	180,5621872	376,3553872	181,8333211	138,1785551
L	6360,5	2360,5	1160,5	7558	4816	3614,8
L_{max}	12719	4719	2319	15119	9630	7227,6
LAM	86,34051562	51,5804154	180,5621872	376,3553872	181,8333211	138,1785551
TT	6360,5	2360,5	1160,5	7558	4816	3614,8
V_{max}	12719	4719	2319	15119	9630	7227,6
RT_{max}	1	1	1	1	1	1
$T2$	1	1	1	1	1	1

stutter signals have a lower RR but a higher DET. In contrast, for stutters of type 3 - word repetition, stutter signals have a higher RR but a lower DET. This is expected as the word repetition and prolongation will have fewer similar states overall, but the similar states are closer together hence the higher DET for the relatively low RR.

On the other hand, the word repetition case explain the high RR of a stutter signal but a lower DET as the words are further apart in space and thus form smaller diagonal lines as the diagonals form from the same sounds being repeated longer. So from this exploratory experiment we can deduce that higher RR are expected when looking at longer duration repetitions i.e. repetitions of words or phrases, while higher DET for a lower RR is indicative of sound elongation and sound repetition.

7 Discussion

RQ-1: Do RP-BD reveal more information on stutter data when compared to classical RPs?

RP-BDs have been shown in Figure 9 to be more robust than RP-KL to the scaling issue of signals, this aids in the qualitative analysis of the RPs and the quantitative metrics between the RP-KL and RP-BD are largely the same. RPs suffer much less from the scale variance compared to Spectrograms as seen in Figure 9 where the word can no longer be declared the same with certainty as the first and third blotch of the spectrogram vary so significantly.

RP-BDs have also been shown here to store more information in the same plot compared to a 'classical' RP, with brighter lines being present due to the scale-invariant property of β-divergences and differing structures within the thick lines as a result of the asymmetric property of divergences. However, RP-BDs seem to be equally susceptible to noise as the 'classical' counterparts, this was seen from an empirical analysis of the plots generated during the exploration.

RQ-2: How useful are available audio datasets for stuttering in helping to reveal any additional information RP-BDs provide over classical RPs?

As mentioned earlier, the intricacy of audio data introduces several challenges due to the high sampling rate and stuttering being an unexplored territory in terms of the Recurrence Plot (RP) analysis. This led us to find efficient computational and visual methods to implement RPs for speech data. This infers that the available stuttering datasets are not useful for RP analysis. This led us to curate our own small dataset by incorporating corresponding utterances from the LibriSpeech (fluent speech) and LibriStutter (stutter speech) datasets. This comparative analysis helped us interpret the RPs for stutter and non-stutter signals qualitatively and quantitatively. However, a strong standpoint requires further analysis with more signals.

In addition, curating such a dataset of corresponding stutter and non-stutter signals on a non-synthesised is time consuming as individuals who stutter would need to be recorded during the first instance of reading a predetermined sentence, then recorded again after a short time to practise the sentence to get a corresponding stutter and non-stutter signal that then need to be pre-processed and aligned to curate in a similar process to what was done for this study. At the same time, whether RP-BD reveals more information than classical RP requires robust analysis with RQA. To summarise, utilising the available datasets requires further exploration of already found efficient visualisation and computational methods, namely compression and blocking methods respectively.

RQ-3: How can the qualitative analysis of RPs be used for the diagnosis of stuttering?

When looking at a stutter signal in comparison to a non-stutter signal we see that a stutter signal has regions of activity which are more closely related in size, in the case of sound or word repetition. If one is able to compare the stutter signal to it's non stutter counter-part we are able to determine the prolongation case of stuttering from the size of this region of activity. These properties are reflected in the quantitative metrics as well. For stutter and non-stutter signals, we can deduce that higher RR are expected when looking at longer duration repetitions i.e. repetitions of words or phrases, while higher DET for a lower RR is indicative of sound elongation and sound repetition.

RQ-4: Is it possible to develop efficient visualisation techniques and computational methods for RPs given the complex nature of audio signal?

The Blocking method offers an efficient solution for visualising exceptionally large signals with high embedding dimensions, provided that the block size is carefully chosen. It is crucial to select an appropriate block size that balances computational efficiency. For smaller signals consisting of around 3000 samples, a minimum block size of half the signal length or even the full signal length is recommended. In the case of a full signal length block, the Blocking method performs computations similar to the traditional matrix-based approach. However, the true advantage of the Blocking method shines when dealing with larger datasets, where selecting an appropriate block size allows for optimal performance and efficient visualisation of the data.

8 Conclusion

RP-BDs add meaningful value over 'classical' RPs, they are more robust to variations in the scale of the audio signal, and provide different visualisations that carry more 'recurrent' information in relation to 'classical' RPs. Although they are susceptible to background noise, the metrics produced correlate strongly with other scale-invariant metrics such as KL-Divergence while retaining the robustness against scale variance. The RQA both qualitative and quantitative show that cor-

responding stutter and non-stutter signals have trends in the relative values of RR and DET. Current audio datasets present many challenges to be utilised for the construction of RPs. One of the major challenges is the high sampling rate. This causes a computational overload of both memory and run time in MATLAB to process the RP matrix and visualise the RPs. Downsampling significantly reduces the quality of the audio signals. Moreover, the analysis of speech signals specifically stutter signals is an unexplored research field. This made the interpretation of the RPs very difficult. To aid this difficulty, however, we curated a hybrid dataset of stutter vs non-stutter data from the LibriSpeech and LibriStutter datasets for corresponding utterances. This enabled us to interpret the RPs in a qualitative and quantitative way to some extent. However, further exploration is required. A rigorous evaluation of the Blocking method was carried out through a series of experiments, encompassing diverse signal types and multiple distance metrics. The comparison between the Blocking method and the traditional approach showcased no differences in the computed RPs, as evidenced by a plot consisting solely of zero values. These findings demonstrate the reliability and suitability of the Blocking method for generating RPs with the same quality to the conventional method in marginally less time, thereby validating its practical utility.

The current approach has significantly improved computation times and made it possible to visualise previously inaccessibly large signals. We've shown that there are differences in the qualitative and quantitative metrics that need to be explored further with a larger dataset to say with more certainty that these differences aren't a matter of chance. This naturally brings us to the future work that can be conducted using this study as a foundation. The first avenue to explore would be the prospect of parallelisation of the blocked approach to computing the RPs to further improve computation times and make efficient use of multiple cores now available in most computers. A second avenue to explore is the curation of a larger dataset, and a providing a stronger argument with a significance test to ensure that the differences between the RP-BDs of stutter and non-stutter signals are indeed significant with a respective confidence interval. Lastly one could explore the effects of averaging the block information instead of summing blocks together when conducting the compression. This may result in different features of RPs being produced of the original features being more better preserved as the current summation approach begins creating vastly different patterns and features.

References

Bayerl, S., Wolff von Gudenberg, A., Hönig, F., Noeth, E., & Riedhammer, K. (2022, June). Ksof: The kassel state of fluency dataset – a therapy centered dataset of stuttering. In *Proceedings of the language resources and evaluation conference* (pp. 1780–1787). Marseille, France: European Language Resources Association.

Bayerl, S. P., Wagner, D., Nöth, E., Bocklet, T., & Riedhammer, K. (2022). The influence of dataset partitioning on dysfluency detection systems. In *Text, speech, and dialogue: 25th international conference, tsd 2022, brno, czech republic, september 6–9, 2022, proceedings* (pp. 423–436).

Févotte, C., & Idier, J. (2011). Algorithms for nonnegative matrix factorization with the β-divergence. *Neural computation*, *23*(9), 2421–2456.

Froguel, L. B., de Lima Prado, T., Corso, G., dos Santos Lima, G. Z., & Lopes, S. R. (2022). Efficient computation of recurrence quantification analysis via microstates. *Applied Mathematics and Computation*, *428*, 127175. Retrieved from https://www.sciencedirect.com/science/article/pii/S0096300322002491 doi: https://doi.org/10.1016/j.amc.2022.127175

Fukino, M., Hirata, Y., & Aihara, K. (2016, February). Coarse-graining time series data: Recurrence plot of recurrence plots and its application for music. *Chaos: An Interdisciplinary Journal of Nonlinear Science*, *26*(2). Retrieved from https://doi.org/10.1063/1.4941371 doi: 10.1063/1.4941371

Hennequin, R., David, B., & Badeau, R. (2010). Beta-divergence as a subclass of bregman divergence. *IEEE Signal Processing Letters*, *18*(2), 83–86.

Howell, P., Davis, S., Bartrip, J., & Wormald, L. (2004). Effectiveness of frequency shifted feedback at reducing disfluency for linguistically easy, and difficult, sections of speech (original audio recordings included). *Stammering research: an on-line journal published by the British Stammering Association*, *1*(3), 309.

Kourkounakis, T. (2021). *LibriStutter*. Borealis. Retrieved from https://doi.org/10.5683/SP3/NKVOGQ doi: 10.5683/SP3/NKVOGQ

Kourkounakis, T., Hajavi, A., & Etemad, A. (2020). Detecting multiple speech disfluencies using a deep residual network with bidirectional long short-term memory. In *Icassp 2020-2020 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 6089–6093).

Kourkounakis, T., Hajavi, A., & Etemad, A. (2021). Fluentnet: End-to-end detection of stuttered speech disfluencies with deep learning. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *29*, 2986–2999.

Lea, C., Mitra, V., Joshi, A., Kajarekar, S., & Bigham, J. P. (2021). Sep-28k: A dataset for stuttering event detection from podcasts with people who stutter. In *Icassp 2021-2021 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 6798–6802).

Martinovič, T., & Zitzlsberger, G. (2018, April). Highly scalable algorithm for computation of recurrence quantitative analysis. *The Journal of Supercomputing*, *75*(3), 1175–1186. Retrieved from https://doi.org/10.1007/s11227-018-2350-5 doi: 10.1007/s11227-018-2350-5

Marwan, N., & Kraemer, K. H. (2023). Trends in recurrence analysis of dynamical systems. *The European Physical Journal Special Topics*, 1–23.

Marwan, N., Romano, M. C., Thiel, M., & Kurths, J. (2007). Recurrence plots for the analysis of complex systems. *Physics reports*, *438*(5-6), 237–329.

Olaya, J., & Otman, C. (2021, 06). Beta-divergence for nonnegative matrix factorization. In (p. 15-22). doi: 10.1109/ICDATA52997.2021.00013

Pálfy, J., & Pospíchal, J. (2011). Recognition of repetitions using support vector machines. In *Signal processing algorithms, architectures, arrangements, and applications spa 2011* (pp. 1–6).

Panayotov, V., Chen, G., Povey, D., & Khudanpur, S. (2015). Librispeech: an asr corpus based on public domain audio books. In *2015 ieee international conference on acoustics, speech and signal processing (icassp)* (pp. 5206–5210).

Rawald, T., Sips, M., & Marwan, N. (2017). Pyrqa—conducting recurrence quantification analysis on very long time series efficiently. *Computers Geosciences*, *104*, 101-108. Retrieved from https://www.sciencedirect.com/science/article/pii/S0098300416307439 doi: https://doi.org/10.1016/j.cageo.2016.11.016

Rawald, T., Sips, M., Marwan, N., & Dransch, D. (2014). Fast computation of recurrences in long time series. In N. Marwan, M. Riley, A. Giuliani, & C. L. Webber Jr. (Eds.), *Translational recurrences* (pp. 17–29). Cham: Springer International Publishing.

Sheikh, S. A., Sahidullah, M., Hirsch, F., & Ouni, S. (2021). Stutternet: Stuttering detection using time delay neural network. In *2021 29th european signal processing conference (eusipco)* (pp. 426–430).

Sheikh, S. A., Sahidullah, M., Hirsch, F., & Ouni, S. (2022). Machine learning for stuttering identification: Review, challenges and future directions. *Neurocomputing*.

A APPENDIX

A.1 Dataset

We have explored five datasets in total. Two of them are picked as potential datasets we can use for our experiments. A short description of our analysis on the datasets is given below:

1. SEP-28K: A Dataset for Stuttering Event Detection from Podcasts with People Who Stutter.
    - Public podcasts largely consisting of people who stutter interviewing other people who stutter are the source of this dataset (Lea et al., 2021). It consists of a total of 28,177 clips. 3-second clips were extracted near pauses using a voice activity detector as the final representative set of disfluency. In addition, all of the 'FluencyBank interview data' were incorporated with the above data. FluencyBank contains stutter data from 32 adults.

- Clips were annotated by at least three people who received training via written descriptions, examples, and audio clips on how to best identify each disfluency but were not clinicians.
- There are six labels for the types of stuttering, e.g., prolongation, block, sound repetition, word repetition, interjection, and no stuttered words. It also consist of six additional non-stutter labels, e.g., unsure, poor audio quality, difficult to understand, natural pause, music, and no speech.

2. SEP-28K-E: Extension of the SEP-28k

   - The SEP-28k-E dataset by S. P. Bayerl et al. (2022) is an extension of the SEP-28k dataset. It consists of the same data as the original dataset with a few more additions. The additions include speaker and gender information, added number of speakers to expect per podcast episode, named speaker labels for the podcast's hosts, and suggestions for balanced data split, helping with an objective comparison of disfluency detection systems.
   - The labels in the original SEP-28k dataset are non-exclusive, meaning a clip can be labelled as belonging to more than one class. This extended version tried to fix this imbalance by incorporating more information as mentioned above. However, our goal is not to build a classification model but rather to build a diagnosis toolbox. So, we can presumably make do with the original dataset.

3. UCLASS: University College London's Archive of Stuttered Speech
   This dataset consists of three different versions of releases (Howell et al., 2004). The 1st release is a monologue while the second release consists of monologues, reading, and conversation. And the third release focuses on Frequency-Shifted-Feedback. However, the dataset is very small. Additionally, the annotation is not uniform in all the releases and different versions. The dataset also lacks properly labelled speaker information. That is why we decided not to use this dataset.

4. KSoF: Kassel State of Fluency Dataset
   This is the biggest available labelled resource containing German stuttered speech and has fully compatible labels with the SEP-28k dataset. However, this dataset is not publicly available (S. Bayerl et al., 2022). So, we decided against this dataset as well.

5. LibriStutter: Synthesised dataset from the public LibriSpeech ASR corpus

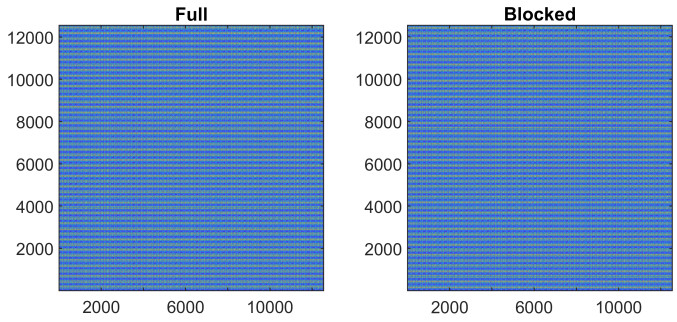
   - This dataset Kourkounakis (2021) contains artificially stuttered speech, as well as time-aligned transcriptions and stutter classification labels for 5 stutter types. It was generated using 20 hours of audio selected from the 'dev-clean-100' section of the original corpus, consisting of "clean" speech.
   - The LibriStutter (English) consists of 50 speakers (23 males and 27 females). For each spoken word, Google Cloud Speech-to-Text (GCSTT) API was used to generate timestamps correspondingly. Random stuttering was inserted within the four-second window of each speech signal.

Final recommendation on the Dataset

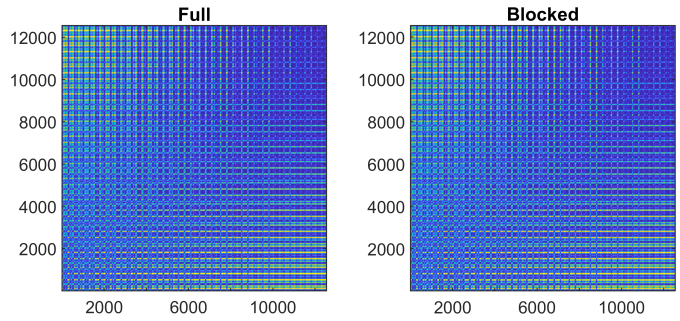
We have mainly considered three aspects namely, i) size, ii) availability, and iii) quality of annotation in the above analysis of the datasets. Finally, we have decided to use the following two datasets in order of preference for the project,

- SEP-28K-E
- LibriStutter

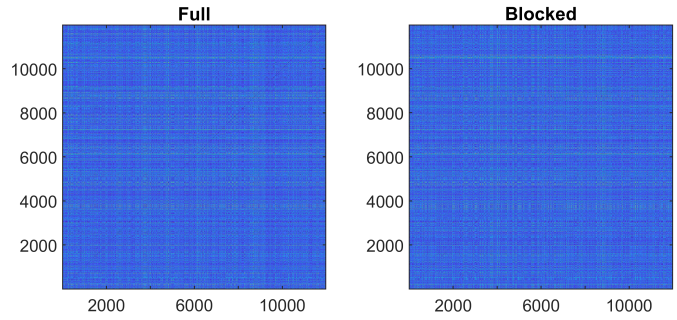
A.2 Validation of Blocking method



(a) Squared Euclidean distance on periodic signal $cos(21) + sin(17)$



(b) IS Divergence on increasing signal $cos(3) + sin(7) + .1t$

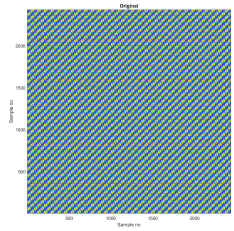


(c) β-divergence on white noise

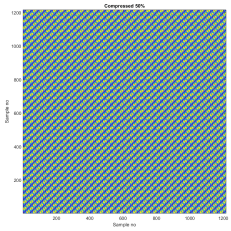
Figure 11: Distance plots comparison between blocked and traditional technique using various type of signals and distance metrics

A.3 Compression vs Downsampling

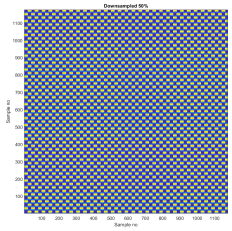
A comparative analysis for a toy signal $sin(5\pi t)$ is illustrated.



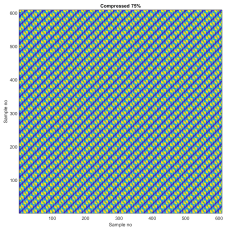
(a) Original RP



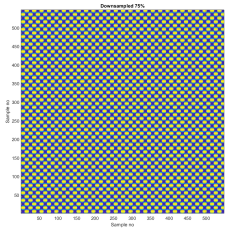
(b) 50% Compressed



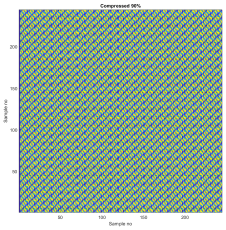
(c) 50% Downsampled



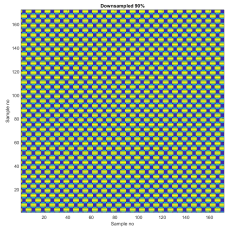
(d) 75% Compressed



(e) 75% Downsampled



(f) 90% Compressed



(g) 90% Downsampled

Figure 12: Comparative analysis of Compression vs Downsampling for $\sin(5\pi t)$. (a) RP of the original signal without any compression or downsampling. (b - g) The rows show corresponding compressed and downsampled RPs. It is evident that the compressed RPs are closely similar to the original RP while the downsampled RPs lose features.