Visualizing EEG data in cloud environments

How to downsample large EEG signals, keeping clinically relevant waveforms while minimizing end-to-end latency

Erik Koprivanacz[1]

Supervisor(s): Arthur-Ervin Avramiea[2]**, Ricardo Marroquim**[1]

[1]**EEMCS, Delft University of Technology, The Netherlands**
[2]**Vrije University Amsterdam, The Netherlands**

A Thesis Submitted to EEMCS Faculty Delft University of Technology,
In Partial Fulfillment of the Requirements
For the Bachelor of Computer Science and Engineering
June 21, 2026

Name of the student: Erik Koprivanacz
Final project course: CSE3000 Research Project
Thesis committee: Arthur-Ervin Avramiea, Ricardo Marroquim, Thomas Abeel

Abstract

Electroencephalography is a widely used non-invasive technique used for measuring brain activity. While relatively cheap, the large volumes of data it produces can make analysis and visualization challenging. NBT Cloud aims to address these challenges by bringing EEG analysis to the cloud and creating an ecosystem for standardized, real-time analysis. Because the application must visualize large data in real time, downsampling the signals is necessary. Modern EEG analysis toolkits, however, focus on downsampling while keeping the data's statistical properties, which introduces computational overhead. This paper investigates whether other downsampling methods, such as Min-Max or Largest Triangle Three Buckets, can achieve better visual fidelity at lower runtime. The results show that simpler candidate algorithms better preserve the visual characteristics of EEG signals while achieving lower runtimes. Among the evaluated algorithms, Min-Max offers the best trade-off between visual similarity and performance, making it the most suitable choice for the use case of NBT Cloud.

1 Introduction

Human cognition remains one of the least understood aspects of human biology. To study how the brain supports perception, attention, and other cognitive functions, researchers rely on non-invasive measurement techniques. Among these, electroencephalography (EEG) is one of the most widely used because of its extremely high temporal resolution and relatively low cost [1]. However, these advantages also introduce significant challenges.

These challenges get particularly apparent in clinical settings. For example, EEGs are the gold standard way of identifying possible brain regions to remove in patients affected by epilepsy. During a typical visit to an epilepsy monitoring unit, where clinicians record patient electrophysiological signals, about 10-20 GB of data is generated. Processing this data requires about 3-4 days of work by specialists [1].

Visualizing the raw signals collected during the patient's stay leads to the best representation of the data. However, due to the high sampling frequencies used by EEGs, not all data points can be directly visualized, as there are considerably more data points than pixels. Visualizing all the raw signal thus incurs computational overhead, greatly increasing the runtime of EEG signal viewers.

An effective way to minimize this overhead is resampling, specifically downsampling. Downsampling is the strategy of reducing the amount of data being processed by keeping fewer samples than in the original. This can be achieved using algorithms such as the M4 algorithm or the Largest-Triangle-Three-Buckets algorithm [2]. A suitable resampling algorithm should preserve clinically and research-relevant waveforms in the data while providing a sufficient speedup so that preprocessing and visualization have lower end-to-end latency than simply visualizing the raw data.

Commonly used EEG analysis toolkits, like EEGLAB or MNE-Python, focus on downsampling the data while keeping its statistical properties consistent with the original signal. The EEG analysis platform, NBT Cloud, developed by researchers at Vrije University Amsterdam, however, ensures that other software features preserve the data's statistical properties. Thus, spending resources to downsample the data in a statistically correct way may be unnecessary. The aim of this research paper is thus the following:

How can we downsample EEG data in cloud environments minimizing end-to-end latency while keeping clinically relevant waveforms visually intact?

End-to-end latency is the time from user input to updating the screen with the new render, i.e., the time to render new data. This paper investigates downsampling that is "visually" closest to the original signal, while not preserving the statistical properties of the data, to quantify whether these methods yield results comparable to already in-use downsampling methods.

2 Background

The following sections will give a review of resampling methods that are currently in use by modern EEG analysis toolkits and how their goal differs from the goal of this paper, an overview of some clinically relevant waveforms that were used for evaluation of the algorithms, and finally a summary of the downsampling algorithms that this paper aims to evaluate.

2.1 EEG Analysis Toolkits

Several widely used EEG analysis toolkits provide built-in support for signal downsampling and resampling. In Python, MNE-Python offers both Fast Fourier Transform (FFT)-based and polyphase resampling methods. [3] FFT-based resampling first computes the signal's frequency spectrum, modifies it to match the target sampling rate, and then computes the inverse transform to return to the time domain. [4] Polyphase resampling is an efficient filter-based approach that splits the filter into phases and combines upsampling and downsampling steps to reorganize the data. [5] SciPy similarly provides several resampling methods, including FFT-based resampling. [6]

In the MATLAB ecosystem, EEGLAB performs resampling by applying an anti-aliasing low-pass filter before reducing the sampling rate. [7] FieldTrip's `ft_resampledata` function applies anti-aliasing filtering by default to prevent aliasing artifacts during resampling. [8]

These approaches are designed to preserve the statistical and spectral properties of the data by removing frequency components that cannot be represented at the lower sampling rate, in accordance with the Nyquist-Shannon sampling theorem. [9] Although these methods are well-suited for subsequent quantitative data analysis, they are not specifically optimized for preserving the visual characteristics of the signal. Anti-aliasing filters can smooth high-frequency components and transient peaks, Fourier-based methods may introduce ringing artifacts around sharp transitions, and averaging

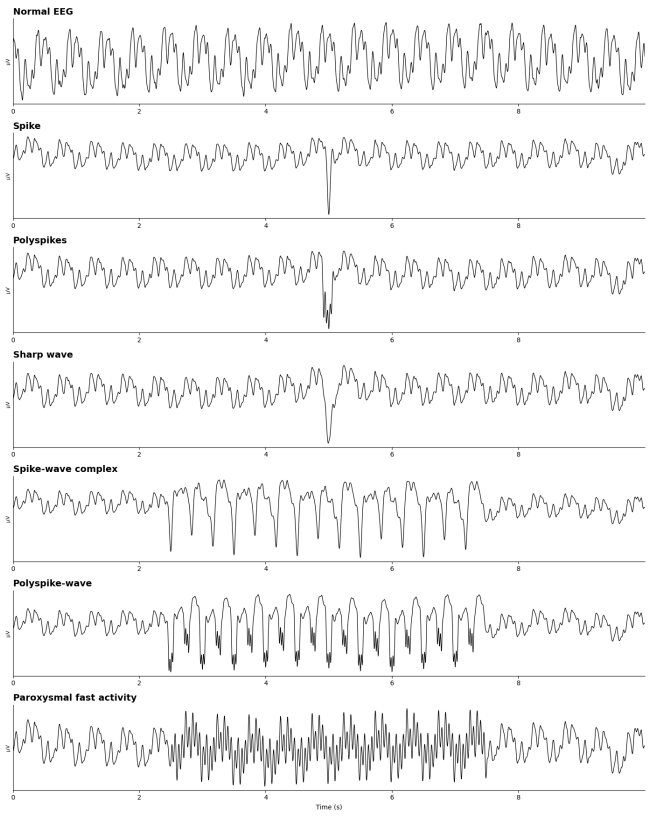


Figure 1: Normal EEG activity and IEDs visualized using a synthetic dataset

effects during resampling can reduce the prominence of local extrema. As a result, features that significantly contribute to a signal's visual appearance may become less noticeable even when the signal remains statistically faithful to the original. Therefore, these methods may not provide optimal results when the primary objective is graphical representation rather than quantitative analysis.

2.2 Clinically Relevant Waveforms

To identify a downsampling algorithm that best preserves the graphical characteristics of EEG signals, the algorithm must maintain not only normal EEG patterns but also pathological waveforms associated with various neurological conditions. Although EEG recordings can contain a wide range of pathological patterns, this study focuses on interictal epileptiform discharges (IEDs) as a representative abnormality for evaluating the performance of the investigated downsampling algorithms.

IEDs are transient EEG patterns that reflect abnormal, synchronized neuronal activity and are widely recognized as biomarkers in the assessment of epileptic and seizure-related phenomena. [10] These events exhibit distinct waveform morphologies, including spikes, sharp waves, polyspikes, and spike-wave or polyspike-wave complexes. Spikes and sharp waves are characterized by brief, high-amplitude transients with pointed morphologies. In contrast, spike-wave complexes consist of a rapid epileptiform component followed by a slower oscillatory wave, a pattern frequently associated with generalized epileptic activity. Synthetic examples of visualized IEDs are presented in fig. 1.

The accurate identification and characterization of IEDs are essential in EEG analysis, as these patterns represent clinically relevant abnormalities. Detecting and annotating such events enables differentiation between physiological and pathological neural activity, providing valuable information for interpreting EEG recordings. Due to their temporal localization and distinct morphological characteristics, epileptiform discharges require precise preservation during signal processing. Therefore, IEDs provide a suitable basis for evaluating how effectively a downsampling algorithm maintains the visual and structural properties of EEG signals.

2.3 Candidate Downsampling Algorithms

Five candidate algorithms were selected for evaluation and analysis. These algorithms cover a wide variety of traditional downsamplers, ranging from naive methods such as uniform stride, which take every $(n+1)$th datapoint, to more nuanced models that aim to preserve some statistical properties. The pseudocode for each algorithm is provided in Appendix B. The five algorithms are:

1. **Uniform Stride**: The Uniform Stride algorithm is a sampling technique that selects every $(n+1)$th measurement by skipping n measurements between consecutive samples, where n is a natural number. As this method does not consider the characteristics of the underlying data, it is used as a baseline against which the other algorithms are evaluated. It is also the current downsampling implementation used in NBT Cloud.

2. **Min-Max**: The Min-Max algorithm divides a signal into n buckets and selects the minimum and maximum value from each bucket while preserving their original order. By explicitly retaining local extrema, the algorithm preserves outliers and prominent peaks, which are often important features in EEG analysis. However, it may fail to accurately represent rapidly oscillating signals, as only two samples are retained per bucket regardless of the signal's complexity. [11]

3. **M4**: M4 operates similarly to the Min-Max algorithm but selects four points per bucket instead of two. Specifically, it extracts the minimum, maximum, first, and last values within each bucket, while preserving their original temporal order. By including boundary samples alongside extrema, M4 retains both local range information and segment continuity. This makes it better suited to capturing short-term structure than Min-Max. [11]

4. **LTTB**: The Largest Triangle Three Buckets (LTTB) algorithm divides the data into n buckets and applies a sliding window over three consecutive buckets. The first bucket contains the previously selected point. The second bucket contains the candidate points from which the next representative point is selected. The third bucket is summarized by a single point representing the average of all samples in the next bucket. The algorithm then selects the point from the second bucket that maximizes the area of the triangle formed by the representative

points from the first and third buckets. LTTB was designed for downsampling large time-series datasets and is particularly effective at preserving the visual structure of the original signal. [2]

5. **Min-Max LTTB**: The Min-Max LTTB algorithm combines the Min-Max and LTTB approaches by first extracting the minimum and maximum values from each bucket and then applying the LTTB algorithm to the resulting reduced set of points. Its performance is comparable to standard LTTB, with slightly lower accuracy in some cases. However, it achieves improved runtime performance by reducing the number of candidate points. [12]

We compare these algorithms using a variety of tests and the metrics outlined in section 3.1 to determine the algorithm that is most suitable in terms of waveform preservation and, secondarily, runtime.

3 Methodology

This project aims to achieve three main objectives. First, each candidate algorithm and toolkit method mentioned in section 2.1 is evaluated to determine how effectively it preserves the original characteristics of the EEG data. Second, the algorithms are compared against each other and against the methods provided by EEG analysis toolkits to assess inter-algorithm agreement. Finally, all evaluated algorithms are benchmarked for computational speed.

3.1 Visual Evaluation

The goal of this phase is to identify the algorithm that achieves the highest visual similarity after downsampling. Visual similarity is defined as the degree to which clinically relevant features, such as the IEDs discussed previously, are preserved while maintaining the overall morphology of the original signal as closely as possible.

To quantify visual similarity, the candidate algorithms and EEG toolkit methods are compared against the original signal using six metrics. These metrics are described as follows:

1. **Spike Preservation**: Spike Preservation measures how well an algorithm retains the amplitude and timing of transient peaks in the signal. In EEG data, such spikes are often associated with clinically relevant events, such as IEDs or artifacts, making them crucial for signal analysis. The metric is computed by comparing the detected peak positions and amplitudes between the original and downsampled signals, using a tolerance window to align the temporal positions.

2. **Complexity Preservation** Complexity preservation measures whether the reconstructed signal retains the oscillatory complexity of the original time series. Retaining oscillations is important because algorithms can oversmooth signals, potentially removing clinically relevant waveforms. Ensuring the downsampled signal has similar complexity, thereby retaining the overall fluctuations of the original. It is based on the complexity estimate (CE), defined as

$$CE = \sqrt{\sum_{t=1}^{T-1} (\Delta x_t)^2}$$

where Δx denotes successive differences in the signal. This metric captures the total variation or "energy" of local fluctuations. A good reconstruction should preserve this value relative to the original signal.

3. **Direction Change Preservation**: Direction change preservation measures how well an algorithm retains the number of local maxima and minima in the signal. Sudden changes in direction, reflected as peaks and troughs, are clinically relevant in EEG analysis and are often associated with transient neural activity or artifacts. While some degree of loss is unavoidable, better retention of local extrema indicates better signal structure preservation. A difference in the number of extrema indicates that the downsampled signal contains fewer oscillatory components than the original, resulting in a loss of detail and reduced visual fidelity.

4. **Euclidean Distance**: Euclidean distance measures the straight-line distance between corresponding points in the original and downsampled signals. [13; 14] Larger distances indicate greater deviation from the original waveform, potentially reducing the fidelity and interpretability of the reconstructed signal.

5. **SBD**: Shape-based distance (SBD) measures the similarity between two time series based on their overall shape rather than their absolute values. [13] It is computed using the normalized cross-correlation between the two signals, where the distance is defined as one minus the maximum correlation over all possible temporal alignments. This allows the metric to account for phase shifts between signals while still capturing structural similarity.

6. **TS3IM**: Time-series structural similarity (TS3IM) evaluates the similarity between two time-series signals using a sliding window approach. [14] It quantifies local structural similarity by comparing three statistical properties within each window: trend, variance, and autocorrelation. The trend component captures directional consistency, variance measures amplitude distribution, and autocorrelation reflects internal temporal dependencies. The score aggregates these components to reflect both global shape similarity and local structural preservation.

3.2 Inter-algorithm Agreement

Beyond comparing each algorithm against the ground-truth reference, the algorithms are also compared with one another and with the EEG toolkit methods to determine whether the observed differences in downsampling results are meaningful. Comparisons are performed using:

1. **Intersection over Union**: IoU is a pixel-level metric that compares the rendered image of the downsampled data and the original signal. [15] Computing IoU is done by:

(a) Rendering both signals using a one-pixel-wide line
(b) Dilating the line on both images by a tolerance range. This is because sub-pixel variations in the signal can produce misleadingly low agreement scores, even when the two graphs look the same.
(c) finally, compute:

$$\text{IoU} = \frac{|\text{intersection}|}{|\text{union}|}$$

2. **Structural Similarity**: SSIM captures perceptual similarity across luminance, contrast, and structural information. [16] It evaluates whether the signal's structural characteristics are preserved, making it suitable for comparing image representations of EEG signals. It is defined as:

$$\text{SSIM}(x, y) = \frac{(2\mu_x\mu_y + c_1)(2\sigma_{xy} + c_2)}{(\mu_x^2 + \mu_y^2 + c_1)(\sigma_x^2 + \sigma_y^2 + c_2)}$$

where μ_x, μ_y are means, σ_x^2, σ_y^2 are variances, σ_{xy} is covariance, and c_1, c_2 are stabilization constants.

3.3 Performance Evaluation

Beyond ensuring that the algorithms' visual fidelity is adequate, another important aspect is the computational efficiency with which they perform downsampling. Therefore, a runtime analysis was performed to evaluate the computational speed of each method. The median runtime and standard deviation were calculated over multiple 10-second evaluation windows extracted from the complete dataset. Only the time required for the downsampling process itself was measured, excluding additional preprocessing and data loading steps, to ensure a fair comparison between algorithms.

4 Experimental Setup

4.1 Dataset

The results of this paper were generated using the SPACE-BAMBI dataset upsampled to 2048Hz using linear interpolation. Due to data privacy constraints, this dataset cannot be made publicly available. However, all code produced is interoperable with different datasets. The code in the Jupyter Notebook of phase one expects Brainvision triplets (`.vhdr`, `.vmrk`, `.eeg`, 1 file each). However, it is possible to create code to read other data types.

Additionally, upon loading the dataset, it is divided into 10-second windows and stratified into five categories based on the properties of the data itself. The five categories are as follows:

- **Artifact:** Windows containing extreme amplitude deviations were classified as artifacts using statistical measures of signal abnormality. A window was marked as an artifact if the maximum absolute z-score exceeded 8 or if the kurtosis exceeded 10. The z-score measures how far individual samples deviate from the mean relative to the signal variability, allowing detection of unusually large amplitudes. Kurtosis measures the distribution's tendency to contain extreme values. High kurtosis indicates heavy-tailed behavior caused by transient spikes or abnormal signal excursions. These windows represent conditions such as signal saturation or electrode malfunction and are used to evaluate algorithm robustness against extreme outliers.
- **Spike:** Windows containing high-frequency, short-duration amplitude transients characteristic of IEDs were identified using spike density greater than 0.3 spikes/second, where spikes were defined as peaks with an absolute z-score above 3.0. This threshold identifies events that significantly exceed the local signal variation. These windows are important for evaluating spike preservation, as downsampling algorithms must retain both the amplitude and temporal location of clinically relevant epileptiform events.
- **Oscillatory High-Frequency:** Windows dominated by rapid oscillatory activity were identified using spectral characteristics. A window was classified in this category if the combined power in the beta (13-30 Hz) and gamma (30-40 Hz) frequency bands exceeded 0.35 or if the spectral edge frequency (the frequency below which 95% of the signal power is contained) exceeded 20 Hz. These measures indicate increased high-frequency activity and allow evaluation of whether algorithms preserve rapid oscillations and waveform morphology during downsampling.
- **Flat:** Windows with minimal signal variation were classified as flat when their baseline variance was below the 10th percentile of the dataset. Low variance indicates limited neural activity and reduced amplitude fluctuations, which can occur during periods of low activity, such as sleep. These windows evaluate whether downsampling algorithms introduce distortions when representing low-amplitude signals, where small deviations may have a larger visual impact.
- **Normal:** Remaining windows were classified as normal and represent typical EEG activity containing mixed-frequency background patterns without dominant artifacts, spikes, or abnormal oscillatory behavior.

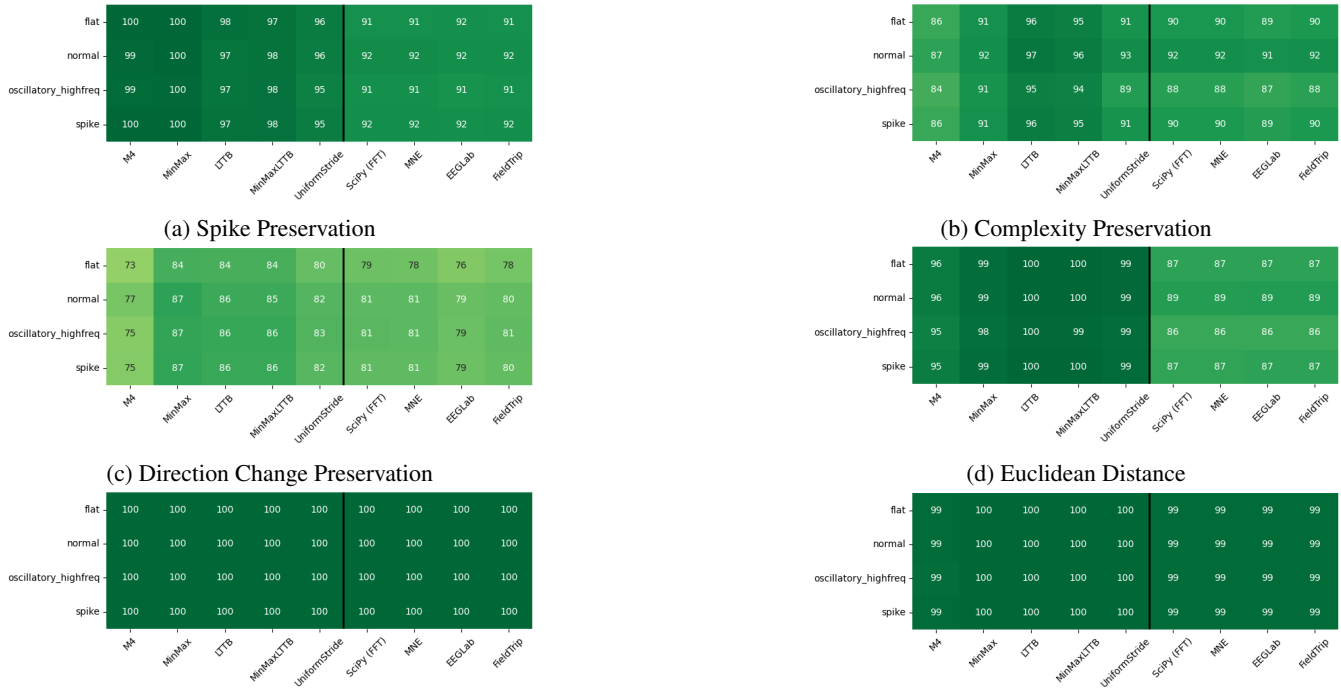
Stratifying the data by these classes enables evaluating whether the candidate algorithms perform differently across distinct EEG signal characteristics. An algorithm may effectively preserve certain features, such as artifacts or high-frequency oscillatory activity, but perform less well on low-amplitude or relatively flat signals. Therefore, this categorization provides insight into the strengths and limitations of each algorithm across diverse EEG signal conditions.

4.2 Code

All code used to generate the results presented in this paper is provided as a Jupyter Notebook. The experiments can be reproduced by executing the notebook in its entirety, provided that the required datasets are available.

4.3 Hardware Setup

To facilitate reproducibility and provide context for the reported performance measurements, the hardware configuration used for benchmarking is reported below. All ex-



(a) Spike Preservation

(b) Complexity Preservation

(c) Direction Change Preservation

(d) Euclidean Distance

(e) TS3IM

(f) SBD

Figure 2: The mean metric scores of each algorithm stratified to flat, normal, oscillatory high frequency, and spiked window patterns. The closer the mean score is to 100, the better the algorithm preserves the original qualities of the data.

periments were conducted on a laptop connected to external power to avoid performance limitations associated with battery-saving modes.

- **Processor:** Intel Core i5-12450H
- **Memory:** 16 GB (2×8) DDR4-3200 RAM
- **Graphics Processor:** NVIDIA GeForce RTX 4050 Laptop
- **Operating System:** Windows 11

5 Results

All of the following results were computed over data downsampled to 800 points from 2048Hz. 800 points were chosen as the target for NBT Cloud, assuming, in the worst case, it would be used on HD monitors (1280×720 resolution), which constrains the displayable data points to 1280 (one per pixel). However, the viewer is displayed alongside different frontend elements and graph axes, which explains the 480-pixel penalty. Visualizations of point amounts are available in Appendix C and D.

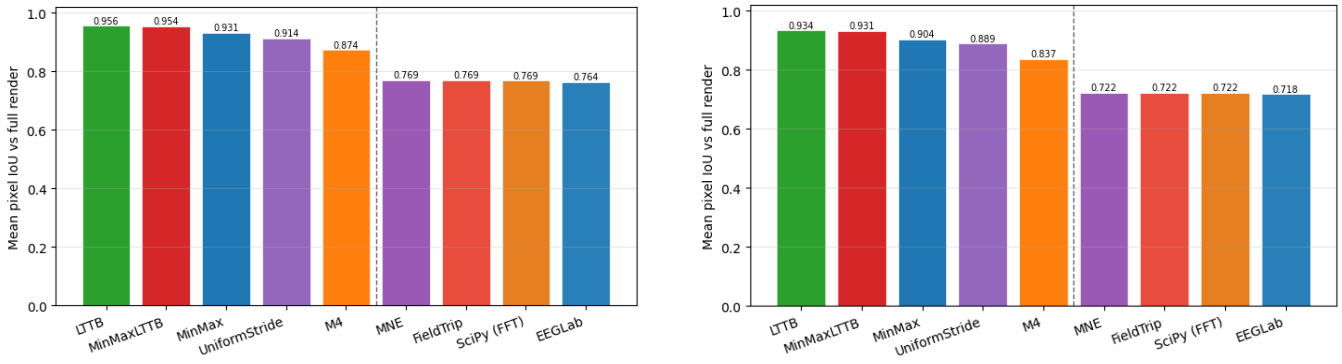
5.1 Visual Evaluation

Visual similarity to the original signal was quantified using the six metrics described previously (fig. 2). Overall, all candidate algorithms achieved high scores across the evaluated metrics, indicating that they are suitable for visually representing EEG data under the chosen downsampling parameters. The only notable exception was direction change preservation (fig. 2c), for which all algorithms exhibited lower scores. However, this result is expected, as high-frequency oscillations may exist at a sub-pixel level and cannot always be preserved after downsampling, even when the reconstructed signal appears visually identical to the original waveform.

Among the candidate algorithms, M4 consistently achieved lower scores than the other methods. This is due to the algorithm's underlying design. Although preserving four points per bucket generally retains more information from the original signal, it also results in fewer buckets being created for a fixed output size. Consequently, the algorithm operates at a lower effective granularity than methods such as Min-Max, leading to reduced preservation of fine-grained signal characteristics.

Compared with the candidate algorithms, the EEG analysis toolkit methods consistently achieved lower scores across most evaluation metrics, with TS3IM (fig. 2e) being the only exception. These results suggest that the candidate algorithms are more effective at preserving the visual characteristics of EEG signals when the objective is faithful graphical representation rather than statistically accurate resampling.

To further validate these findings, visual similarity was also evaluated using IoU against the original signal rendered at both HD (fig. 3a) and 4K resolutions (fig. 3b). The IoU results align with those obtained from the six evaluation metrics. M4 and the EEG analysis toolkit methods again exhibited lower similarity scores, whereas LTTB, Min-Max LTTB, and Min-Max achieved the highest agreement with the original signal. This consistency across multiple evaluation meth-



(a) IoU value at HD resolution

(b) IoU value at 4K resolution

Figure 3: Comparison of IoU values of algorithms compared to the original signal at HD and 4K resolution when downsampling to 800 datapoints. A tolerance range of two pixels at HD resolution and six pixels at 4K resolution is added. The IoU values at 4K are slightly lower as 800 is a lower fraction of the screen at 4K ($\tilde{2}$0.83%) than at HD (62.5%)

ods strengthens the conclusion that these algorithms provide superior visual preservation of EEG data.

5.2 Inter-Algorithm Agreement

The algorithms were also evaluated in terms of inter-algorithm agreement, that is, how similar their outputs are. Pairwise IoU and SSIM scores were computed between all candidate algorithms and the EEG analysis toolkit methods (fig. 4). When comparing the candidate algorithms against each other, they consistently achieved high similarity scores, indicating that their outputs are nearly indistinguishable to the human eye. The only notable exception was M4, which again performed worse for the reasons discussed previously.

In contrast, when the candidate algorithms were compared with the EEG analysis toolkit methods, both SSIM and IoU scores were substantially lower (figs. 4b and 4d). This further supports the observation that the two groups of methods are optimized for different objectives. The toolkit methods appear to preserve the statistical properties of the data better, whereas the candidate algorithms are more effective at preserving the visual appearance of the EEG signal.

5.3 Performance Evaluation

The median runtime of each algorithm was computed over 492 ten-second EEG windows. (table 1) Although Uniform Stride and M4 achieved the lowest runtimes, both methods exhibit notable drawbacks. Due to its sampling strategy, Uniform Stride may fail to preserve clinically relevant features in extreme cases, whereas M4 demonstrated lower visual fidelity than several other candidate algorithms. Considering both visual similarity and computational efficiency, Min-Max provides the most favorable overall trade-off.

The EEG analysis toolkit methods consistently exhibited longer runtimes than the candidate algorithms, with the MATLAB-based toolkits being substantially slower. This is likely attributable to their more sophisticated resampling procedures, which introduce additional computational overhead in exchange for improved preservation of statistical signal properties. Furthermore, the MATLAB-based implementations incur additional performance costs associated with the MATLAB runtime environment and language interoperability. These results suggest that, while the toolkit methods may be appropriate for offline signal processing and analysis, they are less suitable for real-time EEG visualization targeted by NBT Cloud.

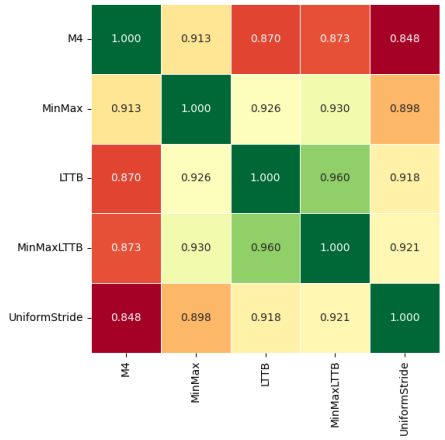
Algorithm	Median (ms)	Std (ms)
Uniform Stride	0.0053	0.0077
M4	0.0361	0.0568
Min-Max	0.0392	0.0259
LTTB	0.0520	0.0297
Min-Max LTTB	0.0876	0.0510
SciPy (FFT)	0.1118	0.0576
MNE	0.4299	0.2049
EEGLab	29.0227	0.7681
FieldTrip	39.1767	9.4338

Table 1: The median run time and standard deviation of each algorithm downsampling data from 2048Hz to 800 datapoints. The data is sorted on decreasing median runtime

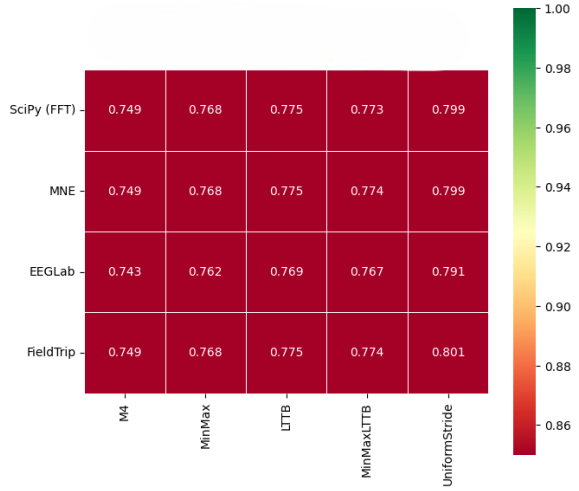
6 Discussion

This research aimed to determine which downsampling algorithm is most suitable for EEG visualization when visual signal preservation, rather than statistical accuracy, is the primary objective. Based on the performed evaluations, Min-Max emerges as the most suitable overall candidate for integration into NBT Cloud. Although LTTB achieved slightly higher visual similarity scores in several evaluations, the difference was small compared to Min-Max's runtime advantage. When both visual fidelity and computational efficiency are considered, Min-Max provides the most favorable trade-off for the intended application.

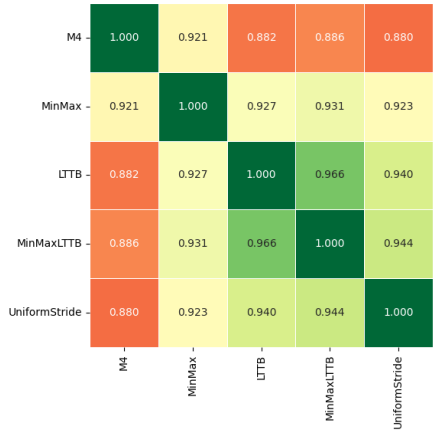
An interesting finding is that the relatively simple Min-Max algorithm performed comparably to, and in some cases better than, LTTB. A likely explanation is the nature of EEG signals themselves. EEG recordings contain frequent local extrema and short-duration transient events, including clinically relevant abnormalities such as IEDs. By explicitly preserving the minimum and maximum values within each



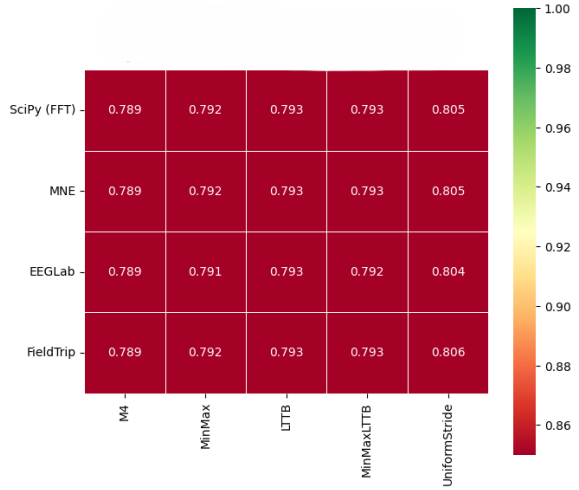
(a) Pairwise IoU against candidate algorithms



(b) Pairwise IoU against EEG toolkit methods



(c) Pairwise SSIM against candidate algorithms



(d) Pairwise SSIM against EEG toolkit methods

Figure 4: Pairwise comparison of IoU and SSIM values of each algorithm. A flat tolerance of two pixels is added to the IoU values. The closer the score is to 1.0, the visually closer the final results of two algorithms are.

bucket, Min-Max retains these features. In contrast, LTTB optimizes point selection based on triangle areas and global visual structure, which may provide less benefit for signals dominated by localized peaks and oscillations. Therefore, the advantages of LTTB become less pronounced in this application, while its increased computational complexity results in longer execution times.

The consistently lower scores observed for direction change preservation require further consideration. Unlike the other evaluation metrics, direction change preservation is highly sensitive to small signal oscillations. When reducing a signal sampled at 2048 Hz to only 800 displayed points, multiple local extrema may occur between neighboring retained samples. As a result, even a visually faithful reconstruction cannot preserve every original direction change. The lower scores, therefore, do not necessarily indicate poor visual quality but rather reflect the unavoidable loss of fine-grained oscillatory information during aggressive downsampling. For this reason, direction change preservation should be interpreted as a measure of retained local detail rather than an absolute measure of visual fidelity.

The comparison with the EEG analysis toolkit methods highlights an important distinction between visualization-oriented and analysis-oriented downsampling approaches. The toolkit methods are primarily designed to preserve statistical and signal-processing properties that are important for subsequent analysis. In contrast, the candidate algorithms evaluated in this work were selected specifically for their ability to preserve visual characteristics. Consequently, the lower visual similarity scores and longer runtimes observed for the toolkit methods should not necessarily be interpreted as inferior performance, but rather as evidence that these methods are optimized for a different objective. For real-time EEG visualization applications such as NBT Cloud, preserving the

visual appearance of the signal is more important than preserving statistical properties of the original data, making the candidate algorithms more suitable for this use case.

Several limitations should be considered when interpreting these results. First, the evaluation focused exclusively on interictal epileptiform discharges, and the conclusions may therefore not generalize to all EEG abnormalities. Second, runtime measurements were obtained on a single hardware platform, meaning that absolute execution times may vary across deployment environments. Finally, the visual similarity metrics employed in this study, while designed to be as comprehensive as possible, cannot fully capture subjective human perception of signal similarity.

Future work should focus on the practical integration of the selected algorithm into NBT Cloud. Several implementation strategies can be considered, including downsampling on the server-side or shifting the computation to the client-side, each with its own advantages and limitations. Additionally, expanding the evaluation to include other EEG abnormalities beyond IEDs would provide a more comprehensive assessment of algorithm performance across a wider range of clinically relevant signal patterns. Ultimately, these improvements could further validate the suitability of downsampling approaches for real-world EEG visualization and analysis applications.

7 Responsible Research

This research was conducted in accordance with the TU Delft Vision on Integrity and the Netherlands Code of Conduct for Research Integrity. Research involving human data and sensitive information raises ethical concerns, particularly regarding data privacy and handling. These considerations can be divided into two areas: data handling and privacy, and the reproducibility of results.

7.1 Data Handling

Privacy The data used to evaluate downsampling algorithms and investigate performance across different platforms comprises real EEG recordings from human participants. These recordings were collected and distributed to research organizations with the explicit consent of the participants and were anonymized prior to distribution.

However, the recordings are associated with metadata that could potentially be used to re-identify participants. Therefore, handling this data confidentially is of utmost importance. The author ensured that all necessary measures were taken to preserve participant anonymity by storing and transferring the datasets exclusively through secure platforms approved by Vrije Universiteit Amsterdam.

Statement on Generative AI Usage The author ensured that appropriate measures were taken to prevent generative AI systems from accessing the datasets used in this project. Additionally, generative AI was used in the following two ways:

1. *Writing Assistance:* Generative AI was used to review the manuscript and suggest edits throughout the project. LLMs were not used to generate independent sections from scratch, but solely to revise and improve human-written text.
2. *Coding Assistance:* Coding agents were used to generate boilerplate code, explain code sections, and debug runtime errors that occurred during the project.

7.2 Reproducibility

The experiments in this study were designed to be easily reproducible, using portable environments such as Jupyter Notebooks whenever possible. Moreover, benchmarking of the NBT platform is integrated into the system and can be executed at any time through the viewer pipeline.

A potential limitation in reproducing the results of this paper is the confidentiality of the datasets used. Due to the sensitive nature of the data, it cannot be made publicly available. However, the metrics and benchmarks were designed to be applicable across diverse datasets. In fact, reproducing the experiments on alternative datasets may provide further insight into the statistical significance of the findings. The author, therefore, encourages other research groups to validate the results using their own data or open-source datasets.

References

[1] Barry Giesbrecht and Jordan Garrett. Electroencephalography. In *Encyclopedia of the Human Brain*, pages 59–75. Elsevier, 2025.

[2] Sveinn Steinarsson. Downsampling Time Series for Visual Representation. May 2013.

[3] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti Hämäläinen. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7, December 2013.

[4] P. Duhamel and M. Vetterli. Fast fourier transforms: A tutorial review and a state of the art. *Signal Processing*, 19(4):259–299, April 1990.

[5] R.E. Crochiere and L.R. Rabiner. Interpolation and decimation of digital signals—A tutorial review. *Proceedings of the IEEE*, 69(3):300–331, March 1981.

[6] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C. J. Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, and Paul van Mulbregt. SciPy 1.0: fundamental algorithms for scientific computing in Python. *Nature Methods*, 17(3):261–272, March 2020.

[7] Arnaud Delorme and Scott Makeig. EEGLAB: an open source toolbox for analysis of single-trial EEG dynamics including independent component analysis. *Journal of Neuroscience Methods*, 134(1):9–21, March 2004.

[8] Robert Oostenveld, Pascal Fries, Eric Maris, and Jan-Mathijs Schoffelen. FieldTrip: Open Source Software for Advanced Analysis of MEG, EEG, and Invasive Electrophysiological Data. *Computational Intelligence and Neuroscience*, 2011(1):156869, 2011. _eprint: https://onlinelibrary.wiley.com/doi/pdf/10.1155/2011/156869.

[9] C.E. Shannon. Communication in the Presence of Noise. *Proceedings of the IRE*, 37(1):10–21, January 1949.

[10] Juan Luis Alcala-Zermeno, Roohi Katyal, Birgit Frauscher, Donald Schomer, Michael R. Sperling, Roy Strowd, William O. Tatum, Elaine Wirrell, Sándor Beniczky, and Fábio A. Nascimento. Seminar in Epileptology: Normal awake and sleep patterns, interictal abnormalities, and ictal patterns on scalp EEG. *Epileptic Disorders: International Epilepsy Journal with Videotape*, 27(5):803–866, October 2025.

[11] Uwe Jugel, Zbigniew Jerzak, Gregor Hackenbroich, and Volker Markl. M4: a visualization-oriented time series data aggregation. *Proceedings of the VLDB Endowment*, 7(10):797–808, June 2014.

[12] Jeroen Van Der Donckt, Jonas Van Der Donckt, Michael Rademaker, and Sofie Van Hoecke. MinMaxLTTB: Leveraging MinMax-Preselection to Scale LTTB. In *2023 IEEE Visualization and Visual Analytics (VIS)*, pages 21–25, Melbourne, Australia, October 2023. IEEE.

[13] John Paparrizos and Luis Gravano. k-Shape: Efficient and Accurate Clustering of Time Series. In *Proceedings of the 2015 ACM SIGMOD International Conference on Management of Data*, SIGMOD '15, pages 1855–1870, New York, NY, USA, May 2015. Association for Computing Machinery.

[14] Yuhan Liu and Ke Tu. TS3IM: Unveiling Structural Similarity in Time Series through Image Similarity Assessment Insights. In *2024 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 567–573, October 2024.

[15] Maxim Berman, Amal Rannen Triki, and Matthew B. Blaschko. The Lovasz-Softmax Loss: A Tractable Surrogate for the Optimization of the Intersection-Over-Union Measure in Neural Networks. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4413–4421, June 2018. ISSN: 2575-7075.

[16] Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing: a publication of the IEEE Signal Processing Society*, 13(4):600–612, April 2004.

Appendix

A Jupyter Notebook

The Jupyter Notebook containing the implementation of the metrics and the code used to produce the results is accessible through the following link: https://github.com/kire-github/research-project

B Algorithm Pseudocodes

Algorithm 1 Uniform Stride Sampling

Require: Data sequence D, stride parameter n
Ensure: Downsampled sequence S
$S \leftarrow \emptyset$
for $i \leftarrow 0$ to $|D| - 1$ step $n + 1$ **do**
    Append $D[i]$ to S
end for
return S

Algorithm 2 Min-Max Downsampling

Require: Data sequence D, number of buckets n
Ensure: Downsampled sequence S
$S \leftarrow \emptyset$
$b \leftarrow \lfloor |D|/n \rfloor$
for $i \leftarrow 0$ to $n - 1$ **do**
    $B \leftarrow D[i \cdot b : (i + 1) \cdot b - 1]$
    $p_{min} \leftarrow$ index of minimum value in B
    $p_{max} \leftarrow$ index of maximum value in B
    if $p_{min} < p_{max}$ **then**
        Append $B[p_{min}]$ to S
        Append $B[p_{max}]$ to S
    else
        Append $B[p_{max}]$ to S
        Append $B[p_{min}]$ to S
    end if
end for
return S

Algorithm 3 M4 Downsampling

Require: Data sequence D, number of buckets n
Ensure: Downsampled sequence S
$S \leftarrow \emptyset$
$b \leftarrow \lfloor |D|/n \rfloor$
for $i \leftarrow 0$ to $n - 1$ **do**
    $B \leftarrow D[i \cdot b : (i + 1) \cdot b - 1]$
    $p_{min} \leftarrow \arg\min(B)$
    $p_{max} \leftarrow \arg\max(B)$
    $p_{first} \leftarrow$ first index in B
    $p_{last} \leftarrow$ last index in B
    $C \leftarrow \{p_{first}, p_{min}, p_{max}, p_{last}\}$
    Sort C by original time index
    for each p in C **do**
        Append $B[p]$ to S
    end for
end for
return S

Algorithm 4 Largest Triangle Three Buckets (LTTB)

Require: Data sequence D, target size k
Ensure: Downsampled sequence S
$S \leftarrow \emptyset$
$bucketSize \leftarrow \frac{|D|}{k}$
Append first point $D[0]$ to S
for $i \leftarrow 1$ to $k - 2$ **do**
    $a \leftarrow S[i - 1]$ ▷ previous selected point
    $start \leftarrow \lfloor (i - 1) \cdot bucketSize \rfloor$
    $end \leftarrow \lfloor i \cdot bucketSize \rfloor$
    $nextEnd \leftarrow \lfloor (i + 1) \cdot bucketSize \rfloor$
    $B_2 \leftarrow D[start : end]$
    $B_3 \leftarrow D[end : nextEnd]$
    $c \leftarrow$ mean(B_3) ▷ centroid of next bucket
    $bestArea \leftarrow -\infty$
    $bestPoint \leftarrow \emptyset$
    for each point b in B_2 **do**
        $area \leftarrow$ triangleArea(a, b, c)
        if $area > bestArea$ **then**
            $bestArea \leftarrow area$
            $bestPoint \leftarrow b$
        end if
    end for
    Append $bestPoint$ to S
end for
Append last point $D[-1]$ to S
return S

Algorithm 5 Min-Max LTTB Downsampling

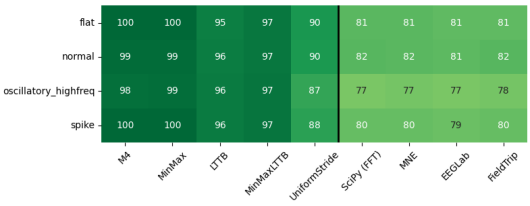
Require: Data sequence D, target size k
Ensure: Downsampled sequence S

$S \leftarrow \emptyset$
$bucketSize \leftarrow \frac{|D|}{k}$
Partition D into k buckets $B_1, B_2, \ldots, B_k$
Extract candidates $C \leftarrow \emptyset$
for each bucket B_i **do**
    $p_{min} \leftarrow \arg\min(B_i)$
    $p_{max} \leftarrow \arg\max(B_i)$
    Append p_{min} and p_{max} to C
end for
Append first and last point of D to C
Sort C by original time index
$S \leftarrow \emptyset$
Append first point of C to S
for $i \leftarrow 1$ to $k - 2$ **do**
    $a \leftarrow S[i-1]$
    $B_2 \leftarrow$ candidates in window C
    $B_3 \leftarrow$ next window in C
    $c \leftarrow \text{mean}(B_3)$
    $bestArea \leftarrow -\infty$
    $bestPoint \leftarrow \emptyset$
    for each point b in B_2 **do**
        $area \leftarrow \text{triangleArea}(a, b, c)$
        if $area > bestArea$ **then**
            $bestArea \leftarrow area$
            $bestPoint \leftarrow b$
        end if
    end for
    Append $bestPoint$ to S
end for
Append last point of C to S
return S

C Results for downsampling at 400 points

Algorithm	Median (ms)	Std (ms)
UniformStride	0.0049	0.0024
M4	0.0341	0.0151
MinMax	0.0353	0.0171
LTTB	0.0459	0.0163
MinMaxLTTB	0.0592	0.0256
SciPy (FFT)	0.1117	0.0541
MNE	0.4087	0.1695
EEGLab	28.7481	2.6263
FieldTrip	36.6869	0.5829

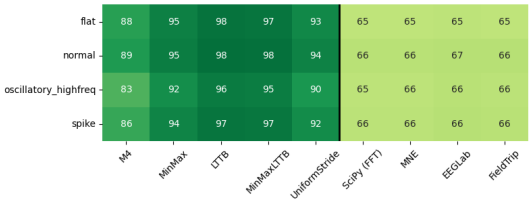
Table 2: The median run time and standard deviation of each algorithm downsampling data from 2048Hz to 400 datapoints. The data is sorted on decreasing median runtime



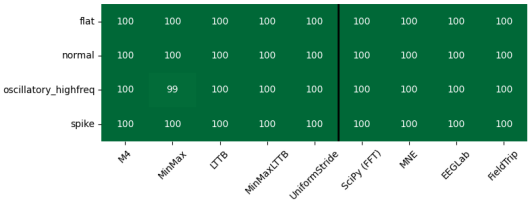
(a) Spike Preservation

(b) Complexity Preservation

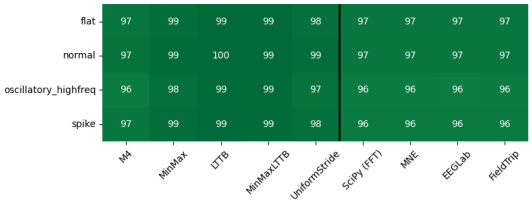
(c) Direction Change Preservation



(d) Euclidean Distance

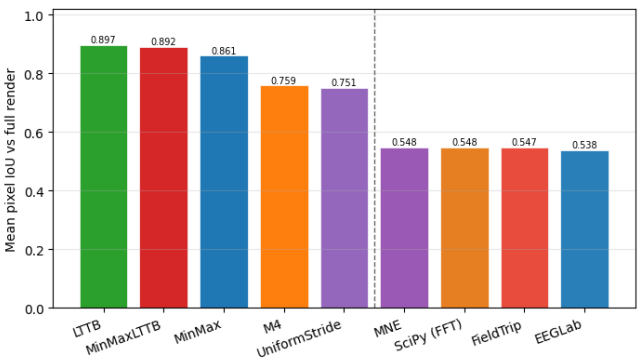


(e) TS3IM

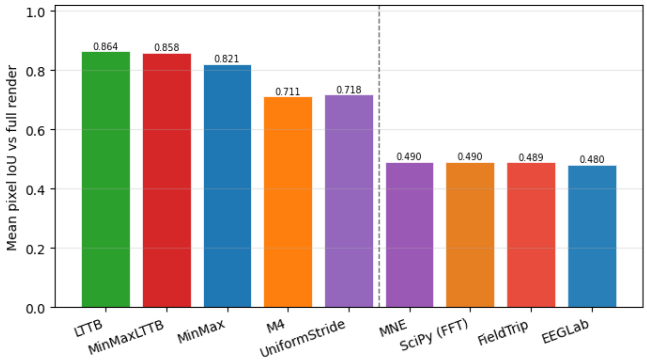


(f) SBD

Figure 5: The mean metric scores of each algorithm stratified to flat, normal, oscillatory high frequency, and spiked window patterns. The closer the mean score is to 100, the better the algorithm preserves the original qualities of the data.

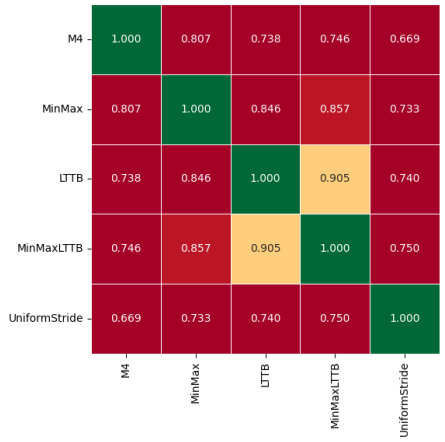


(a) IoU value at HD resolution

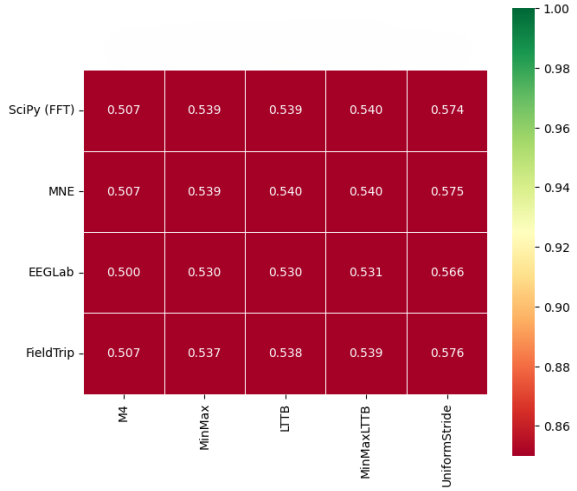


(b) IoU value at 4K resolution

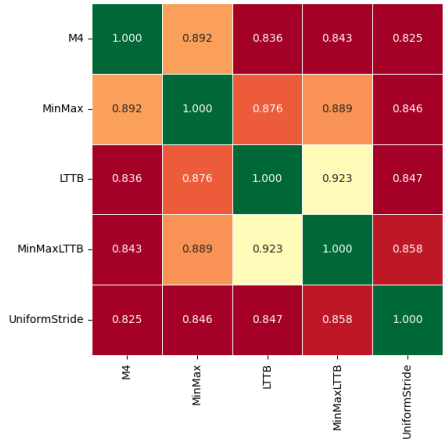
Figure 6: Comparison of IoU values of algorithms compared to the original signal at HD and 4K resolution when downsampling to 400 datapoints. A tolerance range of two pixels at HD resolution and six pixels at 4K resolution is added.



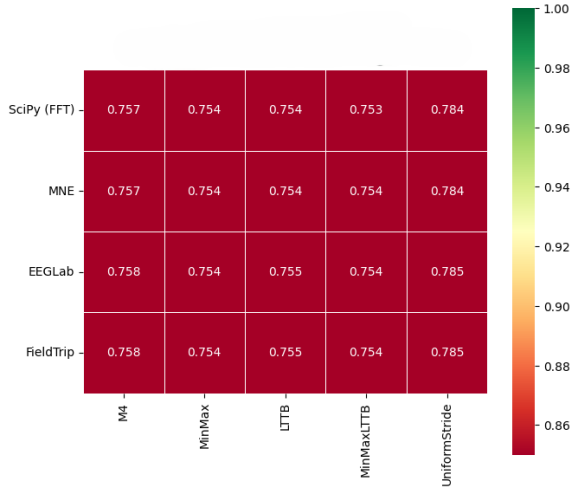
(a) Pairwise IoU against candidate algorithms



(b) Pairwise IoU against EEG toolkit methods



(c) Pairwise SSIM against candidate algorithms



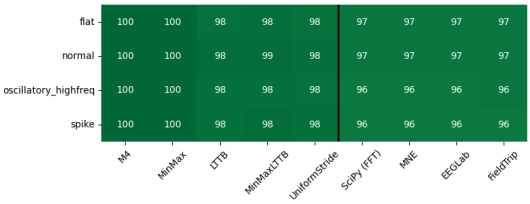
(d) Pairwise SSIM against EEG toolkit methods

Figure 7: Pairwise comparison of IoU and SSIM values of each algorithm. A flat tolerance of two pixels is added to the IoU values. The closer the score is to 1.0, the visually closer the final results of two algorithms are.

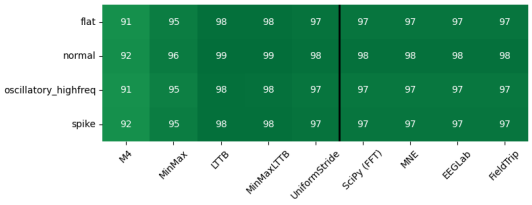
D Results for downsampling at 1600 points

Algorithm	Median (ms)	Std (ms)
UniformStride	0.0063	0.0031
M4	0.0392	0.0190
MinMax	0.0503	0.0644
LTTB	0.0592	0.0166
MinMaxLTTB	0.1274	0.0389
SciPy (FFT)	0.1147	0.0335
MNE	0.4128	0.1700
EEGLab	28.7918	0.6880
FieldTrip	40.9605	1.7978

Table 3: The median run time and standard deviation of each algorithm downsampling data from 2048Hz to 1600 datapoints. The data is sorted on decreasing median runtime

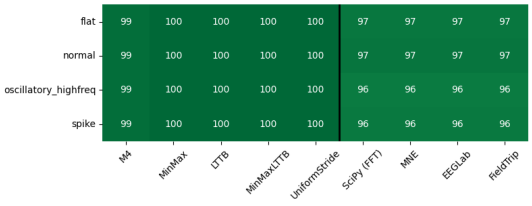


(a) Spike Preservation

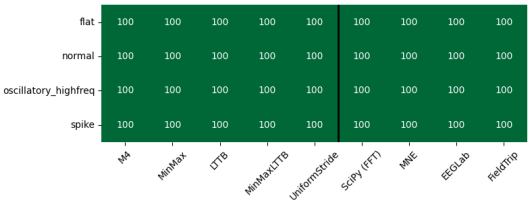


(b) Complexity Preservation

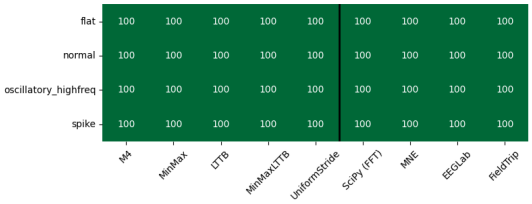
(c) Direction Change Preservation



(d) Euclidean Distance

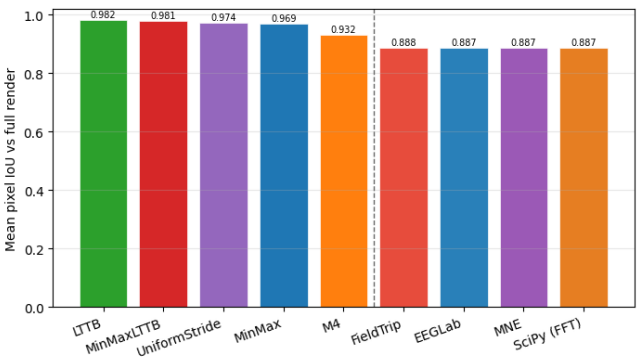


(e) TS3IM

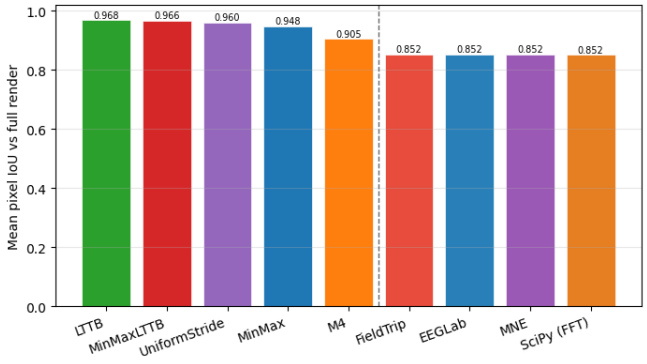


(f) SBD

Figure 8: The mean metric scores of each algorithm stratified to flat, normal, oscillatory high frequency, and spiked window patterns. The closer the mean score is to 100, the better the algorithm preserves the original qualities of the data.

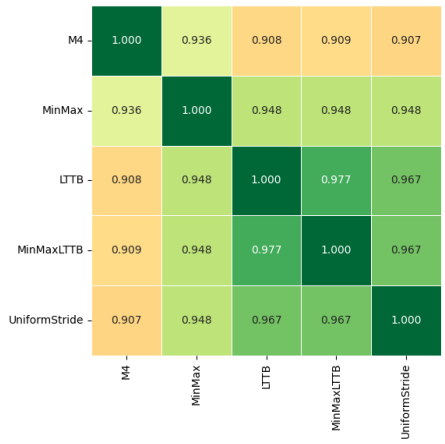


(a) IoU value at HD resolution

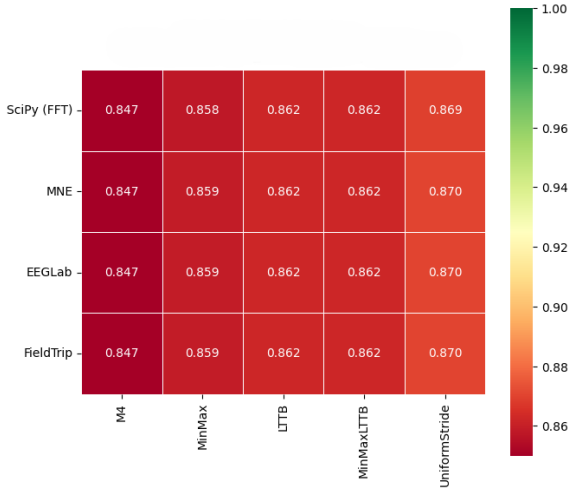


(b) IoU value at 4K resolution

Figure 9: Comparison of IoU values of algorithms compared to the original signal at HD and 4K resolution when downsampling to 1600 datapoints. A tolerance range of two pixels at HD resolution and six pixels at 4K resolution is added.

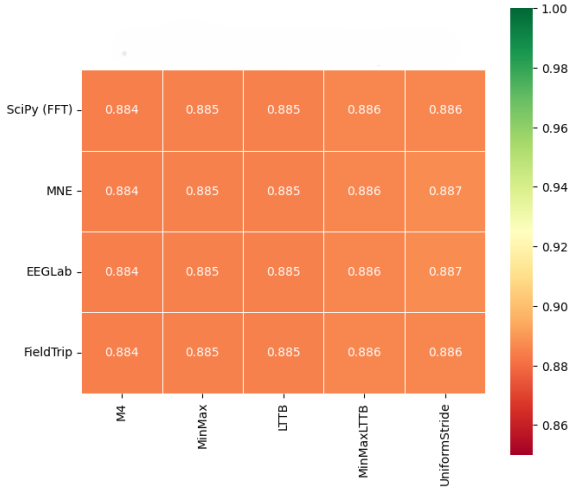


(a) Pairwise IoU against candidate algorithms



(b) Pairwise IoU against EEG toolkit methods

(c) Pairwise SSIM against candidate algorithms



(d) Pairwise SSIM against EEG toolkit methods

Figure 10: Pairwise comparison of IoU and SSIM values of each algorithm. A flat tolerance of six pixels is added to the IoU values. The closer the score is to 1.0, the visually closer the final results of two algorithms are.