Neuralink Compression Challenge: Web-based Interactive Visualization of N1 Implant Signals and Inter-Channel Correlations in Python

Martin Kukrál[a]

[a]*Department of Computer Science and Engineering, Faculty of Applied Sciences, University of West Bohemia, Plzeň, Czech Republic*

Abstract

Neuralink is an American company focused on developing highly advanced brain-computer interaction (BCI) devices and methods of their invasive implantation. Currently, their N1 implant faces issues with signal compression, leading the company to start the public Neuralink Compression Challenge, which contains signal samples for the compression experiments. This seminar work focuses on the interactive visualization of these signals, as they are quite large and complex, with the goal of fascilitating their comprehension for users. The web-based application written in Python utilizes visualizations of N1 signals themselves and their inter-channel correlation, making it suitable for compression researchers to gain insight into the large and complex data.

Keywords: Neuralink, interactive visualization, correlation analysis, Python, web application

1. Introduction

Neuralink is an American neurotechnology company focusing on brain computer interface (BCI) devices, with the mission to "Create a generalized brain interface to restore autonomy to those with unmet medical needs today and unlock human potential tomorrow." [1]. Musk & Neuralink[2] provide a detailed description of not just their proposed invasive N1 implant (Figure 1), but also the neurosurgical robot developed to accurately insert the device into the patient's brain.

Figure 1: Neuralink's N1 implant [3]

Neuralink currently faces the problem with signal compression, as implied by the ongoing Neuralink Compression Challenge[4]. The same issue is also being solved in the field of commonly used electroencephalography (EEG), which also produces immense amounts of data due to the use of multiple channels and high sampling rates, resulting in the problem with signal transmission and long-term storage [5].

This seminar work does not focus on the compression of N1 signals. Instead, it aims to facilitate their comprehension via web-based interactive visualization, which may allow the researchers to gain otherwise inaccessible insight into their behavior, leading to potentially better compression algorithms. Inter-channel correlation is deemed the most significant property to be visualized.

2. Related Works

No academic article or website that deals with the visualization of Neuralink's N1 signals was found, making the presented application first of its kind. However, the used visualization methods are very well known, and therefore no novelty is claimed in the field of data visualization. Related EEG signals are commonly visualized using the following methods: time-domain plots of individual channels, scalp topography, and epoch averages in the context of event-related potential (ERP) experiment [6]. Naturally, any other method of representing multivariate time series or matrices could be used.

3. Materials and Methods

Section provides brief description of the visualized Neuralink data, used software tools, as well as used vi-

Email address: `kukrma@students.zcu.cz` (Martin Kukrál)

sualization techniques and methods of providing interactivity to the user.

3.1. Software Tools

The entire work is programmed in Python 3.11.4. Used libraries are listed in Table 1.

LIBRARY	VERSION	PURPOSE
`dash`	3.0.3	web application
`numpy`	1.25.2	nD arrays
`os`	built-in	file manipulation
`pandas`	2.1.0	correlation analysis
`plotly`	5.22.0	various plots
`scipy`	1.11.2	hierarchical clustering
`tqdm`	4.66.1	loop progress bar

Table 1: Used libraries in Python

In its current state, the application is not published on a web server, but has to be accessed locally once launched via `http://127.0.0.1:8050/` on any web browser.

3.2. Dataset

The publicly available N1 signal dataset can be downloaded from here: `https://content.neuralink.com/compression-challenge/README.html`. Although the N1 implant contains 1,024 electrodes, readings from only 743 electrodes are provided, each roughly 5 seconds long. Sampling rate should be 20 kHz, but based on the provided WAV files, it is actually much closer to 19.5 kHz.

3.3. Preprocessing

The data preprocessing consisted of padding all channels to uniform size (i.e., the longest channel size) with zeroes and forming an array with dimensions $channels \times samples$. This form can be directly used for signal visualization.

Next, the correlation matrices were computed in `pandas` for the following correlation measures: Pearson's correlation coefficient, Spearman's rank correlation coefficient, and Kendall's Tau. The correlation matrices can be directly visualized.

Finally, the correlation matrices were used in the hierarchical clustering via `scipy` to reorder the channels in a way that places more correlated channels next to each other. Used methods were: Nearest Point, UPGMA, UPGMC, and Ward. This information can be used to reorder both signal and correlation visualizations.

The signals, correlation matrices, and indices to reorder arrays are all pre-computed and saved to decrease computational demands on the application.

3.4. Visualization Techniques

Generally, the visualization techniques used can be listed as follows:

1) **time-series plot** – common line plot (used to visualize N1 signals);
2) **heatmap** – visualizing matrices by coloring each cell (used in an alternative visualization of N1 signals and to visualize correlation matrices).

The N1 signal time-series plots can be shown separately or overlayed on top of each other and each channel gets assigned random color. The heatmap-based N1 signal visualization adapted from Kukrál et al.[7] and the correlation matrices utilize two diverging color scales, as they both contain positive and negative values.

3.5. Interactive Elements

The following elements of interactivity are integrated into the application:

1) **navigation** – consists of zoom and pan (synchronized among multiple N1 signal subplots);
2) **hovering** – shows channel, sample, and voltage;
3) **sliders** – used to select a subsection of the channels and/or samples to visualize (can be also specified directly via a popup);
4) **sort** – change the order of channels using the pre-computed reordering indices;
5) **buttons** – used to switch between different visualizations and to select a method of visualizing N1 signals (separate subplots or overlay in one plot).

The navigation and hovering is natively implemented in `plotly`. Sliders, sorting, and buttons were programmed additionally in `dash`. The amount of interactivity is deemed sufficient to explore the data comfortably.

4. Web Application

The web application itself is divided into two main parts: the sidebar and the canvas. The sidebar contains the majority of interface, i.e., buttons to switch between visualizations, sorting options, sliders, and additional options. The canvas is used to visualized selected plot, which can be interactively navigated. The elements of the application are provided on Figure 2.

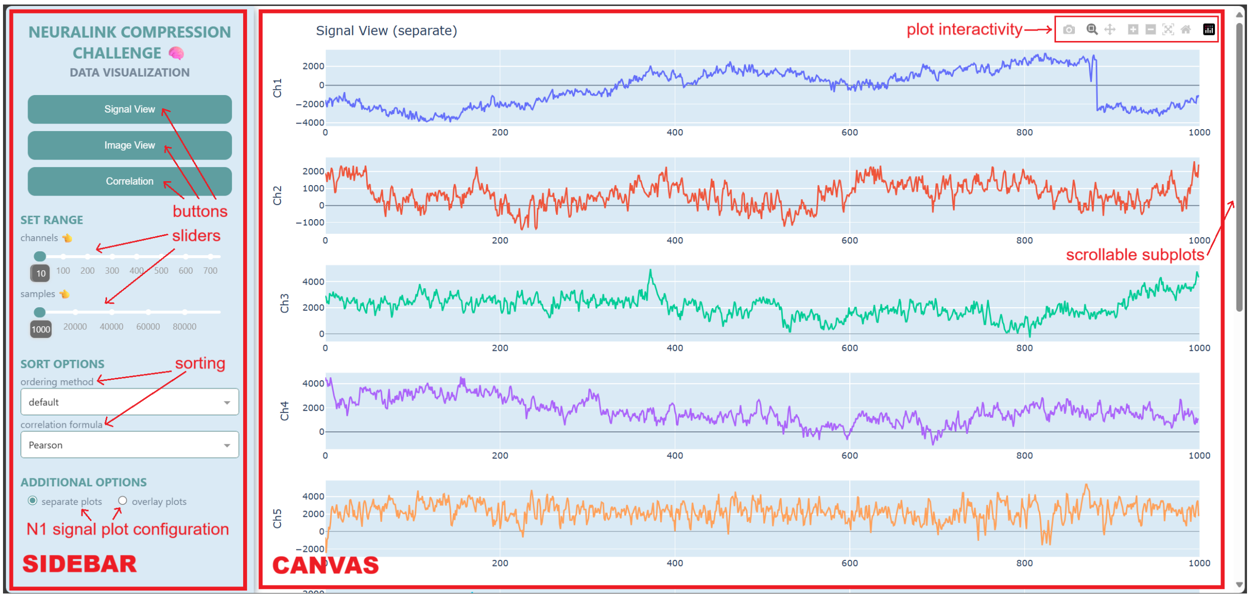

Figure 2: Description of the application (default view upon launch)

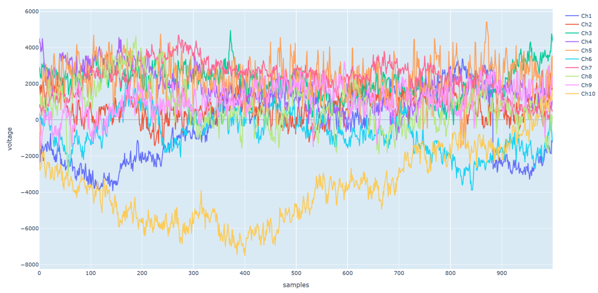

Figure 3: Overlay mode of N1 signal view (extent same as above)

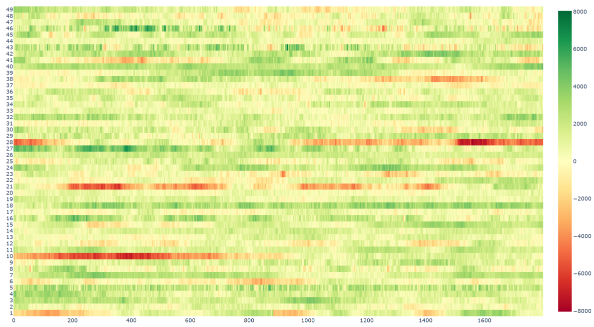

Figure 4: Image view of N1 signals

The "Signal View" button visualizes the N1 signals either as multiple separate plots (one channel each), as seen on Figure 2, or in an overlay mode, showing all selected N1 signals in one plot (Figure 3).

The "Image View" button visualizes the N1 signals in a form of image (Figure 4). The channel labels disappear if their number exceeds 50 due to readability issues. Additionally, if more than 50,000 samples are shown, the heatmap is created from the downsampled signal (every second sample), as plotting all channels and more than approx. 50,000 samples led to issues.

The "Correlation" button visualizes the correlation matrices (Figure 5). Similarly to the image view, the number of labels is limited to 50.

All hover tooltips are provided on Figure 6. They are mainly useful when dealing with heatmaps showing 50+ channels due to the stated removal of labels.

5. Results

The main result of the semetral work is the web application itself. It allows the user to explore the N1 signals and their inter-channel correlations in an interactive manner, which may not be possible by using just static plots.

Additonally, by using the provided application, it is possible to gain better insight into the inter-channel dependencies in the N1 signals. The most direct and natural way is to explore the various correlation matrices, which directly provide 3 measures of similarity between channels. Thanks to the fact that the channel reordering takes place across all visualizations, the channels can be directly observed for similarity upon reordering (Figure 7). As such, the hypothesis about channel similarities

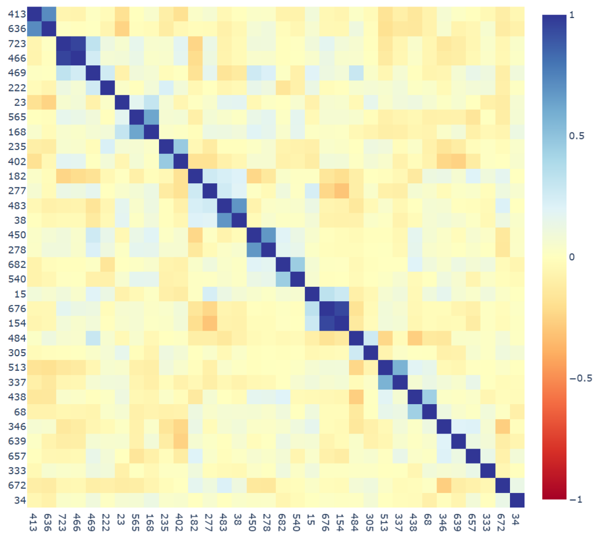

Figure 5: Correlation matrix visualization (Spearman's rank correlation coefficient, reordered using UPGMA)

can be directly tackled by using the visualization application.

6. Discussion

Although the goals of the seminar work are fulfilled and the resulting application is deemed quite useful, it is by no means perfect.

The first limitation of the work is the data itself, which comes from a disk-shaped array of electrodes without a predefined topography, and so the signals could not be visualized spatially (as is common in e.g. EEG recordings with a standardized electrode placement). As such, all visualizations had to be spatially-unaware.

The visualization techniques are deemed appropriately selected and used. When it comes to the used colors, the potential point of debate is the use of random colors when plotting separate N1 channels, as the color here does not carry any information. However, given the fact that the channels can be overlayed (the color is kept) and that scrolling through hundreds of subplots with the same color can be visually hard to orient in, the colors were applied to the subplots as well. Both diverging color scales used in the image view and correlation visualization are conceptually similar, with the main difference being the positive value (green in image view and blue in correlation view). The scale in image view adjusts dynamically to the current minimum and maximum, the scale in correlation view is fixed to range $[-1, 1]$, as it is intrinsically bounded by the correlation

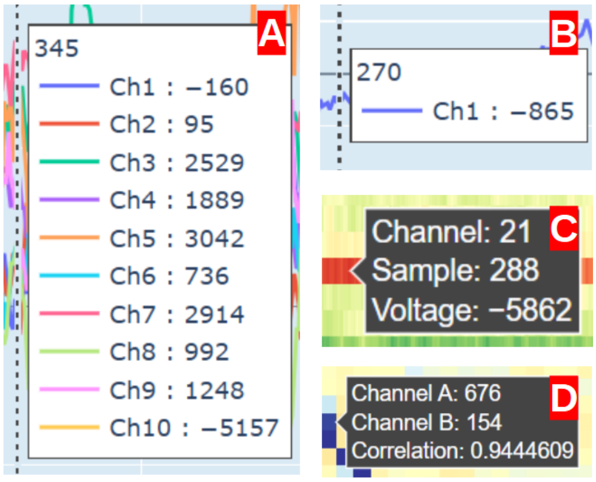

Figure 6: Hover tooltips: overlay signal view (A), separate signal view (B), image view (C), correlation (D)

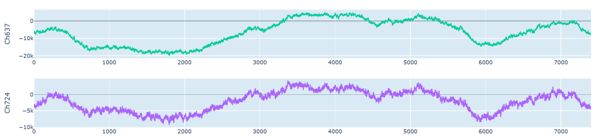

Figure 7: Example of observed similarities between CH637 and CH724 (Spearman's rank correlation coefficient, reordered using UPGMA)

coefficients. The interactive elements are deemed sufficient.

When working with large amounts of channels and samples (potentially all of them), it is possible to notice a significant increase of time needed to visualize the data. This is particularly true when plotting all N1 channels as subplots, as this is the most computation-intensive visualization. For those interested in navigating all channels at the same time, the use of image view might prove faster and more convenient. Nevertheless, the main point of the application is not to visualize the entire data as is, but to explore it using provided sliders and reordering options.

One limitation in the navigation can be noticed, and that is the cutoff of signals after panning out of the currently selected signal subsets via sliders. The reasoning behind this feature is the fact that visualizing the entire signals (and keeping them that way) is definitely possible, but results in the flattening of signals, particularly in the case of channels that exhibit large amplitude changes over time, because the plots are stretched based on the channels' maximum and minimum. Possible solution would be to dynamically adjust the y axis of each subplot based on the current extent, but still keep the entire channels plotted in the background (might lead to unpleasant jumps when the y axis gets updated).

Currently, the main shortcoming is the suboptimal responsiveness of the website, as it is assumed that the application will be used on a desktop computer. However, adapting this desktop-first design philosophy makes sense, as the application is aimed at compression researchers, which are expected to have easy access to their desktop computer.

Originally, the frequency-domain characteristics of the channels were supposed to be visualized as well. The reason for excluding these plots is not the lack of ability to do so, but the fact that the vast majority of them was very similar and not as informative as initially expected. Nevertheless, exploring the frequency domain properties of N1 signals might still be a worthwhile endeavor in the future.

Given that the provided plots are generally usable in all sort of brain-activity measurements (most notably in EEG), the application could be extended to allow the user to import the data from their own experiments and visualize them. This approach would likely be quite straightforward in the signal view, but may prove very challenging if the entire correlation analysis and the subsequent channel reordering was to be available to the user as well, as it takes a really long time to compute (approx. 2 hours for the N1 signals). A possible not so user-friendly solution is to ask the user to also provide the pre-computed correlation matrices and reordering indices as well. Another possibility would be to leverage parallel computing (e.g. in C++) on the server side to compute the correlations much faster than in Python via `pandas`.

7. Future Work

The main improvements to be made is the addition of a fully-responsive layout, making the application potentially usable on phones and tablets as well. New features could be added, such as additional plotting options and new features (possibly from the frequency domain). Optimizing the application to make it run faster is always beneficial. An option to import user data would make the application more generally usable, but would require the solution to the issue of computational complexity when computing the correlation analysis.

References

[1] Neuralink, Neuralink — Pioneering Brain Computer Interfaces, 2025. URL: `https://neuralink.com/`.

[2] E. Musk, Neuralink, An Integrated Brain-Machine Interface Platform With Thousands of Channels, Journal of Medical Internet Research 21 (2019) e16194. doi:`10.2196/16194`.

[3] J. A. Lanz, Brain Implant Testing: Neuralink Begins Recruiting for Human Thought-Reading Trials, 2023. URL: `https://decrypt.co/198110/neuralink-brain-implant-chip-human-testing-fda`.

[4] Neuralink, Neuralink Compression Challenge, 2024. URL: `https://content.neuralink.com/compression-challenge/README.html`.

[5] L. Shaw, D. Rahman, A. Routray, Highly Efficient Compression Algorithms for Multichannel EEG, IEEE Transactions on Neural Systems and Rehabilitation Engineering 26 (2018) 957–968. doi:`10.1109/TNSRE.2018.2826559`.

[6] Cognitive Computational Neuroscience Group, University of Bern, Data Visualization, 2021. URL: `https://neuro.inf.unibe.ch/AlgorithmsNeuroscience/Tutorial\_files/DataVisualization.html`.

[7] M. Kukrál, D. T. Pham, J. Kohout, Štefan Kohek, M. Havlík, D. Grygarová, Near-lossless eeg signal compression using a convolutional autoencoder: Case study for 256-channel binocular rivalry dataset, Computers in Biology and Medicine 189 (2025) 109888. doi:`10.1016/j.compbiomed.2025.109888`.