universität wien

MASTERARBEIT | MASTER'S THESIS

Titel | Title

Frequency mixing: An Effective Approach for Neurophysiological Signal Augmentation

verfasst von | submitted by

Jakob Prager BSc

angestrebter akademischer Grad | in partial fulfilment of the requirements for the degree of

Master of Science (MSc)

Wien | Vienna, 2024

Studienkennzahl lt. Studienblatt | Degree programme code as it appears on the student record sheet:	UA 066 910
Studienrichtung lt. Studienblatt | Degree programme as it appears on the student record sheet:	Masterstudium Computational Science
Betreut von | Supervisor:	Univ.-Prof. Dipl.-Ing. Dr. -Ing. Moritz Grosse-Wentrup

Abstract

Brain-Computer-Interfaces (BCIs) ermöglichen eine direkte Interaktion mit externen Geräten ausschließlich durch das Scannen von Gehirnwellen. Da jedoch die Gehirnaufzeichnungen zwischen Individuen drastisch variieren können, brauchen viele BCIs benutzerspezifisches Training, um gut zu funktionieren. Dies schränkt deren Benutzerfreundlichkeit ein. Ein Ansatz, um dieses Problem anzugehen, ist Data Augmentation. Dabei werden Daten auf vielfältige Weise manipuliert, damit maschinelle Lernsysteme sich an Variabilitäten anpassen und allgemeinere Modelle erstellen können. In dieser Studie präsentiere ich Frequency mixing, einen effektiven Ansatz zur Augmentation neurophysiologischer Signale, der sich insbesondere auf SSVEP (Steady-State Visually Evoked Potential) und EEG2Code-basierte BCIs konzentriert. Mein Hauptziel war es, eine Technik zur Data Augmentation zu entwickeln, die die Klassifizierung von SSVEP-Daten verbessert, ohne auf benutzerspezifisches Training angewiesen zu sein. Zudem sollte ihre Wirksamkeit bei EEG2Code-Daten bewertet werden. Frequency mixing erzeugt neue Datensätze, indem der Durchschnitt der Frequenzspektren verschiedener Probanden berechnet wird. Die Ergebnisse zeigen eine signifikante Verbesserung der Klassifizierungsgenauigkeit von SSVEP-Daten, wobei eine maximale Verbesserung von 9 % bei der Verwendung begrenzter Trainingsdaten erzielt wurde. Obwohl Frequency mixing keine Verbesserungen bei der EEG2Code-Genauigkeit erzielte, zeigt diese Studie sein Potenzial als einfacher und dennoch effektiver Algorithmus zur Augmentation neurophysiologischer Signale.

Schlagwörter: Neurophysiologische Signale, Gehirn-Computer-Schnittstellen, Frequency mixing, SSVEP, EEG2Code, c-VEP, Daten Augmentation

Abstract

Brain-computer interfaces (BCIs) enable interaction with external devices entirely by scanning brain waves. However, as brain recordings can vary drastically among individuals, many BCIs struggle to perform well without user-specific training, limiting their usability. One approach to tackle this problem is through Data augmentation. This involves manipulating data in manifold ways so machine learning systems can adjust to variabilities and create more generalizing models. In this study I introduce Frequency mixing, an effective approach to augment neurophysiological signals, particularly focusing on steady-state visual evoked potential (SSVEP) and EEG2Code-based BCIs. My primary objective was to develop a data augmentation technique that improves SSVEP data classification without the need for user-specific training. Furthermore, I assessed their effectiveness on EEG2Code data. Frequency mixing generates new datasets by mixing and averaging frequency spectra from different subjects. The results demonstrate a significant improvement in SSVEP data classification accuracy, with a maximum enhancement of 9% observed when using limited training data. Even though Frequency mixing did not yield improvements in EEG2Code accuracy, this study shows its potential as a simple yet effective algorithm for augmenting neurophysiological signals.

Keywords: Neurophysiological signals, Brain-computer interfaces (BCIs), Frequency mixing, SSVEP, EEG2Code, c-VEP, Data augmentation

Index

1 Introduction

Brain-computer interfaces (BCIs) make it possible for their users to control external devices entirely by scanning their brain signals without the use of extremities like hands or feet. The aspect of directly communicating with machines, bypassing the physical hindrances, makes them not only a very interesting research topic but also a helpful device for disabled users or possibly industrial use.

To this day there exist three types of recording modalities for BCI control that are commonly used, which are electroencephalography (EEG), Electrocorticography (ECoG), and intracortical microelectrode arrays (MEAs) (Miller et al., 2020).
EEG can be understood as the recording of the scalp of the BCI user. By wearing a non-invasive EEG cap, the electric field potentials of the scalp can be measured and used for BCI control. A cap typically used for medical studies features an array of 64 evenly distributed electrodes according to the 10/20 convention with a reference electrode at the zenith of the head (Beres, 2017). EEG is very flexible and combines low cost and high temporal resolution, however, it suffers from poor spatial resolution (Wolpaw et al., 2002) (Liu et al., 2020).To this day EEG is not only used in a medical context but is also starting to be used commercially as gaming controllers or educational support (Portillo-Lara et al., 2021).

ECoG can be seen as the extension of EEG as the electrodes are placed directly under the skull and on the surface of the brain. Typical ECoG arrays consist of electrodes of 2.3 mm diameter spaced 1 cm apart. The array measures the sum of local field potentials directly under each electrode and provides localized cortical signals. ECoG has a high signal-to-noise ratio and is mostly used for medical purposes such as the treatment of epilepsy but several proof-of-concept studies have already shown it to be useful for BCI control (Miller et al., 2020).

Going even further are MEAs which penetrate the cortex of the brain and can measure the activity of the single neurons near the electrodes. Intracortical microelectrode arrays feature the highest spatial resolution leading to the most precise recording of the desired brain region, however, they are highly invasive. A typical microelectrode array features 100 electrodes of 1.5-mm in length in a 4 mm x 4-mm grid (M. Wang & Guo, 2020). Microelectrodes are still in development and feature issues such as infections, biological rejection, and signal instability (Miller et al., 2020). As of now, the MEAs that are implanted for long-term use are still

succumbing to deterioration over time, leading to few clinical trials running for longer than 4 years.

While not the most precise technology of those three, EEG is by far the most used for BCI control due to its ease of use and cost-effectiveness. Over time researchers developed many different EEG-based BCI paradigms which can be broadly divided into “motor imagery paradigms” and “external stimulation paradigms”. Motor imagery paradigms use brain responses from imagining a movement. External stimulation paradigms on the other hand use brain responses stimulated by sound or flickering lights (Abiri et al., 2019). In this thesis, I will focus on one very popular external stimulation paradigm, namely steady-state visual evoked potentials (SSVEPs).

1.1 SSVEPs and c-VEP

Steady-state visual evoked potentials are brain signals that occur in response to a constantly flickering visual stimulus. When stimulated by a signal oscillating at a fixed frequency between 3.5 and 75 Hz the human brain will start to produce brain waves at the same frequency (Beverina et al., 2003). To measure SSVEPs, most EEG setups use electrodes located at the occipital lobe which is primarily responsible for visual processing. In this area, setups can use one to multiple electrodes, however, electrodes between Pz and Oz seem to be most optimal (Marx et al., 2019). By recording the SSVEPs one can infer the original stimulus frequency and use the information for BCI control. Typically, SSVEP setups present grid structures on LCD screens where each cell is flickering with a distinct frequency and is imbued with a certain meaning. By looking at different cells and matching the recorded EEG pattern with the stimulus frequency one can create visual keyboards on computer screens. The BCI´s user can then write by simply looking at the letters. As the displayed objects don´t need to be letters but can be virtually anything, SSVEP is an extremely versatile BCI paradigm. To match EEG signals with the activation frequency, different classification methods can be used ranging from Support-vector machines (SVMs) to deep-leaning-based convolutional neural networks (CNN) (Bassi et al., 2021).

1.1.1 c-VEP and EEG2Code

Code-modulated visual evoked potentials (c-VEP) are used in a similar way to SSVEPs but instead of a fixed range of frequencies, they use pseudorandom activation patterns of black and white "flashes" that aim to have a minimal cross-correlation (Bin et al., 2009). Although a different code could define each target, it is not easy to find a group of codes with suitable cross-correlation (Martínez-Cagigal et al., 2021). Instead, a typical approach is to find a bit-sequence with low autocorrelation and then time-shift the sequence, so that each target is defined by a unique pattern (Bin et al., 2009). A type of sequence often employed is 63-bit maximum length sequences (m-sequences) that are then displayed in a loop and commonly shifted by two bits for each additional target. Their autocorrelation is 1 if not shifted and 1/N otherwise (Holmes, 2007) (Martínez-Cagigal et al., 2021).

A paper written by Nagel & Spüler (2018a) proposed a method called EEG2Code which completely discards the idea of predicting the complete patterns as a whole and instead classifies the singular bits. It then successively creates the predicted sequence. The most prominent part of a visual evoked potential (VEP) after a flash lasts about 250 ms (Odom et al., 2016). It is possible to classify this time window as either black or white and therefore as zero or one of a pseudorandom bit-sequence. Using this approach reduces the classes to predict to only two and makes it possible to classify any arbitrary pattern (Nagel & Spüler, 2018b). In a more recent paper, they showed that by combining this method with deep learning it is possible to discriminate between 500,000 stimulation patterns from a two-second recording of data (Nagel & Spüler, 2019). This makes EEG2Code a very powerful tool for setups with high amounts of targets.

1.1.2 Constraints

SSVEP, c-VEP, and EEG2Code setups are relatively cheap, versatile, and easy to use, however, as good as they already are, they still have limitations. Due to the different anatomies of human heads and brains, it is hard to classify the stimuli without proper training on the subject's specific brain response. This leads to unwanted training times before one can use the system for experiments.

In the current state of VEP-based BCIs, a user needs to feed training data to the system so that it can get a "feel" for the user's brain response. In the case of SSVEP, a user would need to look at every frequency for a span of a few seconds or longer while the system knows the respective target frequency. It then uses the gathered data to calibrate. Even though this period of training

does not seem like a hindrance for the singular user, it can be very time-consuming in large-scale studies and impedes its ease of use.

There are ways in which people try to address the problem, but as of now, there are no “plug and play” builds of a VEP-based BCI setup. One paper by Thielen et al. (2021) presented a c-VEP setup not completely unlike EEG2code which does not require training pre-session but calibrates itself on the session data while the subject is already using the system. While this is arguably very close, it is still not a true “plug-and-play” setup that requires zero user-specific training.

1.2 In search for plug-and-play

“Zero-training” plug-and-play Brain-Computer Interfaces could (BCIs) revolutionize neurotechnology, making brain-machine interfaces accessible and user-friendly. The key advantages include simplifying setup for users without technical expertise and broadening BCI adoption for clinicians, researchers, and end-users. Furthermore, they could massively increase experimentation speeds in general, enabling better healthcare applications. All those benefits lead to “plug-and-play” BCIs being very sought after; however, they are only rarely achieved.

One of the main reasons zero training BCIs are hard to implement, especially if the BCIs rely on training data to be calibrated, is the lack of training data to begin with. If a machine learning system is trained only on small amounts of data or if the data comes from only one single subject, the system will likely not generalize well on data of new subjects.

1.2.1 Large datasets needed

One naïve approach to solving the problem of generalization would be to simply gather more data. In theory, if one had information on the system response of millions of heads and for each calibration of the EEG cap, the trained system should be able to classify the data of a new person much better than a system with low amounts of training data. The problem here is that there is not enough training data available. Even if it were, most of it would possibly not be compatible due to differences in the setups of data gathering. Considering EEG-SSVEP recordings, differences can go from other choices of channels, recording lengths, and sampling frequencies to entirely other sets of stimulation frequencies. To overcome this problem, two approaches have been most prominent in the last years: Transfer Learning and Data augmentation.

1.2.2 Transfer Learning

Transfer Learning is a machine learning paradigm that leverages knowledge gained from solving one task to improve the performance of a different but related task. Instead of training a model from scratch for a specific task, Transfer Learning involves using a pre-trained model, typically trained on a large dataset for a much broader task. The idea is that the knowledge gained during the initial training can be beneficial for solving a new task. The pre-trained model's learned features or representations are “transferred” to the new model, which is then fine-tuned on the target task. Transfer Learning is particularly powerful in scenarios where labeled data for the target task is limited or expensive to obtain, allowing for more efficient and effective model training. This approach has found success across various domains, including computer vision, natural language processing, and speech recognition, accelerating progress, and improving the performance of machine learning models on a wide range of applications (Hosna et al., 2022).

1.2.3 Data augmentation

Data augmentation (DA) is a technique widely employed in machine learning to artificially expand and diversify a training dataset by applying various transformations to the existing data. By introducing variations such as rotations, flips, zooms, and changes in brightness or contrast, Data augmentation aims to expose the model to a broader range of scenarios, enhancing its generalization and robustness. This method is particularly valuable when the available dataset is limited, as it effectively increases the size of the training data without requiring additional labeled examples. Data augmentation is extensively used in computer vision tasks, such as image classification and object detection, as well as in natural language processing, where it helps improve the performance and resilience of machine learning models when faced with real-world variations in input data (Mumuni & Mumuni, 2022).

1.3 Goal and approach

The research group Neuroinformatics at the University of Vienna uses a VEP-based BCI setup that uses the EEG2Code approach of Nagel & Spüler (2019). As this is a relatively new approach to VEP-based BCIs, as of now, there are no studies that have tried to implement a zero-training approach and improve on it by using Data augmentation techniques. Therefore, my supervisor and I thought this would be a very interesting topic for my Master´s thesis.

There are only sparse amounts of EEG2Code data distributed online, and the dataset used by the research group is also quite small, so testing the statistical significance of my found methods with this dataset would be unreasonable. With this in mind, we devised an alternative approach. As there are plenty of large SSVEP datasets online, I planned to create Data augmentation methods that improve on a zero-training approach for SSVEP data. In the case of success, I would test if the respective methods also worked on the small set of EEG2Code data collected by the research group Neuroinformatics at the University of Vienna.

In the two-step approach of my thesis, creating at least one novel way of augmenting SSVEP data would be the main goal, while testing the method on EEG2Code data would be the subsequent step. Depending on the result, it could either be a good starting point for further research or give us insight into the differences between SSVEP and EEG2Code.

I need to mention that there are already some studies that used Data augmentation for SSVEP, however as the topic is very broad, many things are still unexplored. Luo et al. (2023) for example, used DA to boost the performance of SSVEP classification. However, they did not make it a zero-training system but used DA to augment a single trial of one subject that was recorded for training.

1.3.1 Difference between SSVEP and EEG2code

While EEG2Code and SSVEP share the same base paradigm, they are still different from one another. Here I will give an overview of the differences between both datatypes, how they are sampled and can be used for BCI control.

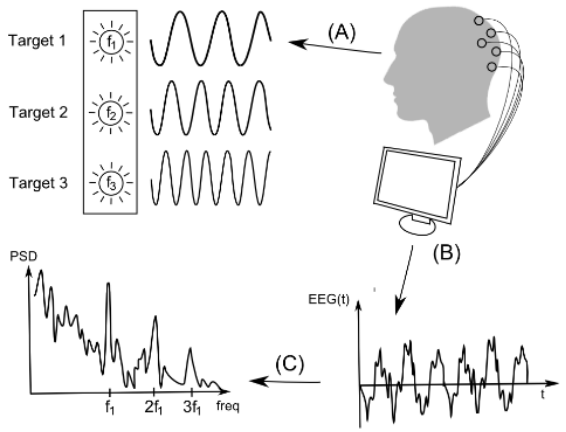

Figure 1: An approach of decoding SSVEP signals (Robben et al., 2011). The approach involves looking at frequencies, recording the brain signal, converting it into the frequency domain and recording the highest peaks.

Steady-state visual evoked potentials, as the name suggests, use stable oscillations as the input signal. The brain response reflects these oscillations and makes it possible to detect the stimulation frequency (Beverina et al., 2003). Naturally, an SSVEP signal measured via EEG does not only consist of the input frequency but also a lot of additional frequencies that are part of the brainwaves. When trying to detect the stimulation frequency, a way of doing so would be to create a Power spectral density (PSD) and look for the frequencies with the most power. Figure 1(Robben et al., 2011) shows one possible setup where the highest peak in the PSD indicates the stimulation frequency. I want to note that

this example is only one of a few ways to classify SSEVP signals. Depending on the setup, one does not need to create the PSD but can also train the system on raw data or spectrograms of SSVEPs. I will dive into this topic later as it will tie together with possible approaches to augmenting the data.

When recording SSVEP data for BCI control, one needs to decide on the length of the recording window as it not only defines how fast the user can convey their intentions but also influences how well the system can classify the input. While longer windows generally lead to better accuracies, they slow down the system. The performance of an SSVEP setup is typically measured as its information transfer rate (ITR) in bits per second and includes its speed and accuracy (Arslan & Sinha, 2023). The length of a window can vary drastically from 0.5 to 4 seconds, however, nowadays window lengths of 0.5 to 2 seconds seem to be standard (Pan et al., 2023).

A c-VEP signal, on the other hand, does not use distinct frequencies for activation but rather uses a pseudorandom sequence of flashes. It uses the fact that the brain produces a visual evoked potential (VEP) to any given flash. This makes c-VEP like a fingerprint of the random sequence. EEG2Code works with the same concept but makes use of the fact that the complete VEP of one flash lasts for around 250 ms (Nagel & Spüler, 2018b). While capturing this timeframe, many other flashes occur and contribute to the signal, however, the first bit of the time window (flash or no flash) is still detectable. For example, when using a 60 Hz monitor and displaying a new bit of the sequence every 1/60 seconds, in a 250 ms recording of the first bit, there will be an overlap of 15 bits. Figure 2 shows how this works in practice. The black line shows the underlying bit sequence that represents either one or zero, and the colored straight lines, show the start and the timeframe for each bit. The blue line represents the brain signal corresponding to the stimulation pattern.

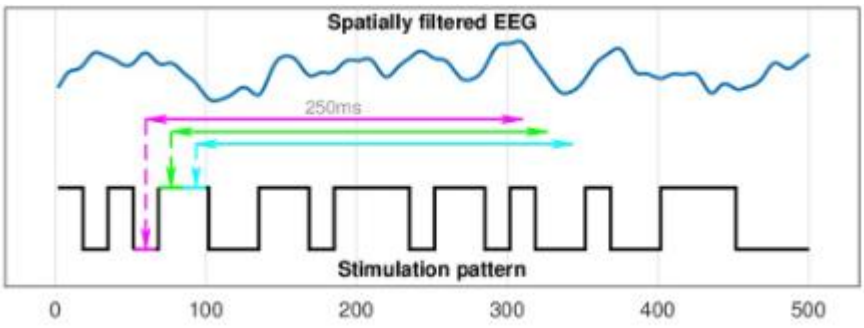

Figure 2: Depiction of EEG2Code (Nagel & Spüler, 2018b). The black line shows the underlying stimulation pattern of bits (zeros and ones). The blue line shows the brain response of a subject, and the arrowed lines indicate the length and overlap of the recording windows for each bit.

To use EEG2Code for BCI control one would first create a codebook of unique sequences for each target. The next step would be to train a classifier to distinguish between black and white bits. When predicting the targets, the classifier first only classifies the singular bits of the activation sequence. To complete the prediction, the system keeps track of the classified bits, compares them to all predefined code sequences, and chooses the best-fitting sequence. As the window length for EEG2Code is set to 250 ms for each bit, the prediction speed of the system depends on the trial length I.e. the number of predicted bits. One can either set fixed trial lengths or let the system run until a similarity threshold is surpassed. A typical trial length in Nagel und Spüler´s paper was 1 second which amounted to 60 predicted bits. To conclude, by directly predicting the activation bit of each 250 ms window and not the entire sequence, the system only needs to deal with two classes while still being able to distinguish between immense amounts of targets (Nagel & Spüler, 2019).

1.4 Types of Data augmentation and related work

As the realm of Data augmentation is already quite vast, in this section I will give an overview of some of the most used basic types of DA methods.

Data augmentation is commonly used for two kinds of data, images, and time series. The most basic image data consists of pixel values arranged in a 2-dimensional grid, representing spatial information. For images, the spatial arrangement of pixels is crucial and cannot be changed too much before losing a lot of information. Common DA methods involve rotating and cropping the images to introduce variance that also retains their relative position. Other methods involve a certain amount of shuffling of pixels to give them a “blurry” effect or the cropping and patching together of multiple images (Shorten & Khoshgoftaar, 2019).

Time series data, on the other hand, can be interpreted as a one-dimensional stream of information that has a chronological order and strong sequential dependencies. Flipping the data along the x-axis assumes it to be symmetrical in the context of “up” and “down” (Wen et al., 2021) while flipping it along the y-axis would change the sequentiality. Both properties make flipping a suboptimal tool for augmenting time series data. While rotating the data is also possible, it will change the information and can make it lose all meaning (Iglesias et al., 2023).

As SSVEP and EEG2Code are time series data, I want to present some basic DA methods for time series that I encountered during my research. Time series can be augmented in three domains: Time domain, Frequency domain, and Time-frequency domain (Wen et al., 2021).

1.3.1 Time domain

The time domain can be understood as the raw data stream that shows how a signal changes over time. Two key DA techniques in the time domain are time warping and amplitude warping. Time warping involves stretching or compressing the temporal dimension of a time series, altering the speed or duration of parts of the signal. The introduced variability can be interpreted as a change in the timestamps of events. Amplitude warping, on the other hand, focuses on modifying the magnitudes of data points, allowing for the simulation of diverse scenarios with varying intensities. Jittering is another key technique when it comes to DA. It involves adding small random fluctuations to data points by introducing Gaussian noise (Wen et al., 2021). Another proposed method is the permutation of time windows within a signal (Um et al., 2017). While possibly being useful, the main problem of applying permutation is not preserving time dependencies, which can lead to invalid samples (Iglesias et al., 2023).

1.3.2 Frequency domain

Compared to the time domain, a graph of a signal in the frequency domain shows the distribution of frequencies comprising the signal. The maximum displayable frequency is half the sampling frequency of the signal, also known as the Nyquist frequency (Levesque, 2014). To access the frequency domain, one can use the Fourier Transform.

The Fourier Transform is a mathematical operation that can convert a signal from the time domain into the frequency domain. When putting in a real-valued signal, the operation yields a complex-valued signal that contains the amplitude information for every frequency that is necessary to create the signal and its respective phase (Müller, 2015). One can extract both properties, modify one or both, and convert the signal back into the time domain. The resulting signal then differs from the original signal depending on how much its properties were altered. Gao et al. (2021) did this in a paper by introducing noise into both phase and frequency spectra. Another possible application in this domain is frequency warping which is another version of amplitude warping (Iglesias et al., 2023). One of the most popular fields for frequency warping is speech recognition. The frequency domain is also used to create filters to remove certain ranges of frequencies e.g. low-pass or high-pass filters.

1.3.3 Time-Frequency Domain

The time-frequency domain is essentially a split-up frequency domain that shows which frequencies occurred at what timestamp during a signal (Boashash, 2003) and is represented as spectrograms. In a spectrogram, the x and y axes represent time and frequency, whereas the color represents the power of its frequency band at a given time. For further visualization, Figure 3 shows one SSVEP signal in all three domains. Two prominent DA methods in this domain are SpecAugment (Park et al., 2019) and SpecMix (Kim et al., 2021). SpecAugment was designed for speech recognition models and directly acts on the spectrogram. The method performs time warping, frequency, and time masking, creating an incomplete and slightly altered spectrogram. Bassi et al. (2021) used this method for SSVEPs and managed to slightly improve classification accuracy, following a zero-training approach. SpecMix on the other hand can be seen as an extension of SpecAugment. It uses two different spectrograms and masks one of them with the other along time and frequency axes, essentially filling the parts SpecAugment would cut with the other spectrogram.

It was shown that SSVEPs can be classified in all three domains (Sheykhivand et al., 2017) (Attia et al., 2018) (Bassi et al., 2021) however, EEG2Code was only ever worked with in the time domain. For the sake of having comparable results between datasets, I opted to classify the SSVEP signals in the time domain as well. I want to note that it is possible to transform a signal into any domain, alter it there, and transform it back.

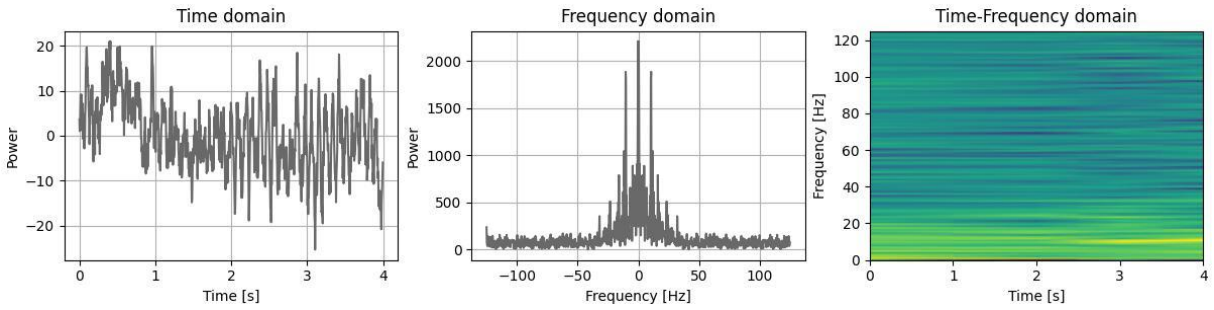

Figure 3: Depiction of an SSVEP signal in the time domain, frequency domain and time-frequency domain. The power (color) of the time-frequency plot is scaled logarithmically for better visualization.

2 Material

For my thesis, I used two datasets, one SSVEP dataset for developing Data augmentation techniques and one EEG2Code dataset to test the techniques.

2.1 SSVEP dataset

The first one is an SSVEP dataset that is freely available online and was gathered by Y. Wang et al. (2017). It consists of 64-channel EEG data from 35 healthy people. The subjects were asked to gaze upon a visual keyboard that presented 40 different visual stimuli that ranged from 8 Hz to 15.8 Hz displayed on a 60 Hz LCD monitor. Every two frequencies had an interval of 0.2 Hz and were coded, using a joint frequency and phase modulation approach. Furthermore, adjacent frequencies were phase-shifted by $\pi/2$.

For each subject, the data consists of 6 blocks of 40 trials, one for each frequency. Between every block, there was a resting break of several minutes. One trial is 6 seconds long with 0.5 seconds on and offset and 5 seconds of frequency data between them. The team recorded data with a Synamps2 EEG system (Neuroscan, Inc.) and a sampling frequency of 1000 Hz. To reduce storage costs, they subsequently downsampled it to 250 Hz. When loading the data of one person into a program, the dimensions of the resulting array are (40 classes x 1500 sample length x 64 channels x 6 blocks). In this dataset, each subject has the same amount of data with the same labels.

2.2 EEG2Code dataset

The second dataset is an EEG2Code dataset that was recorded by the research group Neuroinformatics at the University of Vienna (Meunier et al., 2024). The dataset consists of 6-channel EEG data from 5 subjects. The data was recorded with a sampling frequency of 1000 Hz while the bits of the code were flashing at 20 Hz. For this dataset, the brain response was continuously recorded but for each bit of the activation code, the 250 ms of the VEP were stored separately. This leads to an array of many short snippets of overlapping data that are labeled after the activation bit e.g. 0 or 1 (black/white). The overlap of one sample to each adjacent sample is 4 bits. One factor to note is that for this dataset neither the activation sequence nor sample size is the same across the subjects. Due to this fact, I could only classify the bits and not the encoded targets.

3 Methods

After trying to use basic DA methods in the time domain like jittering and masking of channels without any meaningful success, I changed my focus of domain and delved deeper into the realm of frequency augmentation. With this new direction, my goal was not just to randomly perturb the data but to expand the dataset in meaningful ways, reflecting possible variations between the subjects.
Along with my studies, I developed the method Frequency mixing that significantly improved the decoding accuracy of the SSVEP dataset which I subsequently tested on the EEG2Code dataset.

3.2 Frequency mixing

Frequency mixing operates in the frequency domain and takes advantage of the functionality of the Fourier transform. As mentioned earlier, the Fourier transform can take in virtually any signal and create a complex signal consisting of frequency spectrum and phase spectrum. While the frequencies correspond to the absolute value of the Fourier transform, the phases correspond to its angle. Both properties can be separated, individually modified, and put together to create a new time domain signal.

Frequency mixing assumes that the frequency spectrum of an SSVEP signal of each subject lies in the space of all valid SSVEP signals. Under this assumption, between every two frequency spectra, there should be at least one additional spectrum that also contains the frequencies of a valid SSVEP signal.
With this concept in mind, Frequency mixing takes two signals of the same channel with the same activation frequency of two subjects, converts them into the frequency domain, and computes the average of the frequency spectra. This new spectrum lies exactly between the old ones and forms a hypothetically valid spectrum. It then takes the phase information of both original signals and uses the inverse Fourier transform to convert them back to the time domain using the averaged spectrum as the frequency information. This operation creates two conceptually new signals for every two signals put into the function. The output of the function scales almost quadratically with a factor of $n(n-1)$, where n represents the number of original signals. This quadratic growth enables the generation of larger amounts of data as the quantity of original data increases.

Algorithm 1 Frequency Mixing

Input: two multichannel signals Sig1, Sig2 with dimensions (signal length/channels)

Output: two multichannel signals $\tilde{\text{Sig}}1$, $\tilde{\text{Sig}}2$ with dimensions (signal length/channels)

```
for channel do
    s1, s2 = Sig1(channel), Sig2(channel)
    S1, S2 = FFT(s1), FFT(s2)
    m1, m2 = abs(S1), abs(S2)
    p1, p2 = angle(S1), angle(S2)
    m3 = (m1 + m2)/2
    s̃1, s̃2 = iFFT(m3, p1), iFFT(m3, p2)
    S̃ig1.add(s̃1), S̃ig2.add(s̃1)
end for
return S̃ig1, S̃ig2
```

The pseudocode only shows the operations regarding actual data, not the labels. As the output of Frequency mixing should have the same labels as the input, one could simply copy the labels for each batch of data fed into the system and append them to the originals.

As I mainly worked with multichannel data, I included the iteration over the channels in the pseudocode. As the concept is quite simple one could easily omit the for-loop and use the inner part for single-channel signals. In fact, during my research, I found that almost exactly this algorithm was proposed in a paper in 2023, however for the topic of time series forecasting (Chen et al., 2023). In this paper, the algorithm was used to create new data to train models to predict the flow of time series like traffic or weather. While being far from BCI classification, the fact that the same algorithm was already shown to work in creating useful augmented data, makes me even more confident about its application for BCI setups. As they proposed the name Frequency mixing and developed it before me, I simply adopted the name. Figure 4 shows the visual depiction of the algorithm´s workflow and gives a deeper understanding of how the signals might look during the individual steps of the workflow.

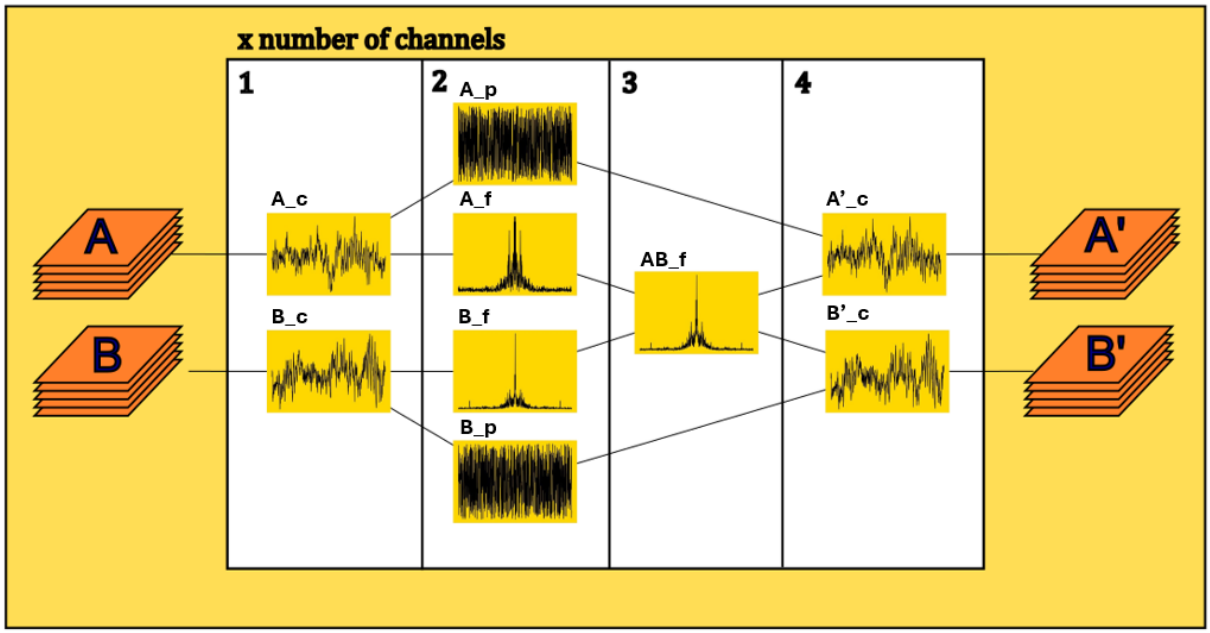

*Figure 4: A Visual depiction of the workflow of Frequency mixing. **A** and **B** are multichannel signals. In 1, one channel gets selected **(A_c, B_c)**, and in 2, the frequency spectrum and Phase spectrum get extracted **(A_p, A_f, B_p, B_f)** by performing an FFT. In 3, both frequency spectra **(A_f, B_f)** get averaged **(AB_f)** and in 4, the average gets reverted by using the phase spectra **(A_p, B_p)** and an iFFT. By performing 1-4 for each channel **A'** and **B'** are created.*

3.3 Graphs

3.3.1 SSVEP graph

Figure 5 shows the signals of one instance of Frequency mixing applied to two SSVEP signals, from randomly chosen singular channels of the same frequency. As the output of this method varies massively with the input signals, it is essential to note that this is just a snapshot of many possible plots.

With the input being two signals of length 1000 and a sampling rate of 250 ms, the first graph shows their frequency spectrum from -125 to 125 Hz. I did not apply any filtering to the signals and one can see the maximum of the power resides in very low frequencies.

Despite having the same activation frequency, both spectra present a very different set of amplitudes. In this example specifically, the first signal seems to have much more power in terms of amplitudes than the second one. These differences show the variability of SSVEP signals across multiple subjects.

Subsequent plots show the signals after performing Frequency mixing, overlayed with the original SSVEP signal. Visually, both signals are not substantially different from their originals and generally follow the same path. This can be explained by the fact that their phase

information is still the same, which can have a huge impact on the shape of the signal in the time domain.
On a closer look, however, both signals differ strongly in the amplitude of the power. Despite them being closely related to each other, they are interpretable as independent signals. Furthermore, in this specific example, as the overall power of one signal is much higher than the other, one can observe an exchange in the power of both signals. Frequency mixing makes the first signal lose power while the second signal gains power compared to its original.
It is important to remember that for effective Data augmentation, one cannot simply distort data to a point where it loses all meaning but overall needs to keep the semantics intact. From the visuals alone it cannot be determined if this is the case but rather needs to be tested in practice.

3.3.2 EEG2Code graph

Figure 6 shows a snapshot of one instance of Frequency mixing of two EEG2Code signals of a random channel of two samples of two subjects. In this case, both the original signals were stimulated by the same bit (black or white), however, due to the dataset not being uniform, the surrounding bits are very likely not the same. Considering that baseline EEG2code already works regardless of the surrounding bits, this additional perturbation should not make a big difference.
With signals of 250 ms length, the first graph shows an overlay of the frequency spectrum of both signals from -500 to 500 Hz, with the amplitude limited to 2000.
Similar to the SSVEP dataset, the frequency spectra have no clear similarities other than having roughly the same shape with a lot of power in the lower frequencies and less power in the high frequencies.
The subsequent plots show both the original and augmented signals in the time domain as an overlay. Similar to SSVEP, the signals are dissimilar from one another but generally follow the same path. Again, this is due to the same phase information being present in the signals.

In conclusion, data augmented by Frequency mixing can introduce high variability in the amplitudes of the signal compared to the original, while maintaining a strong correlation in the time axis. The amount of variation introduced is dependent on the spectral differences of both signals. By sourcing the variation from already existing data, the new data has a high chance of keeping its semantic value while also introducing enough variation to ensure more generalizability.

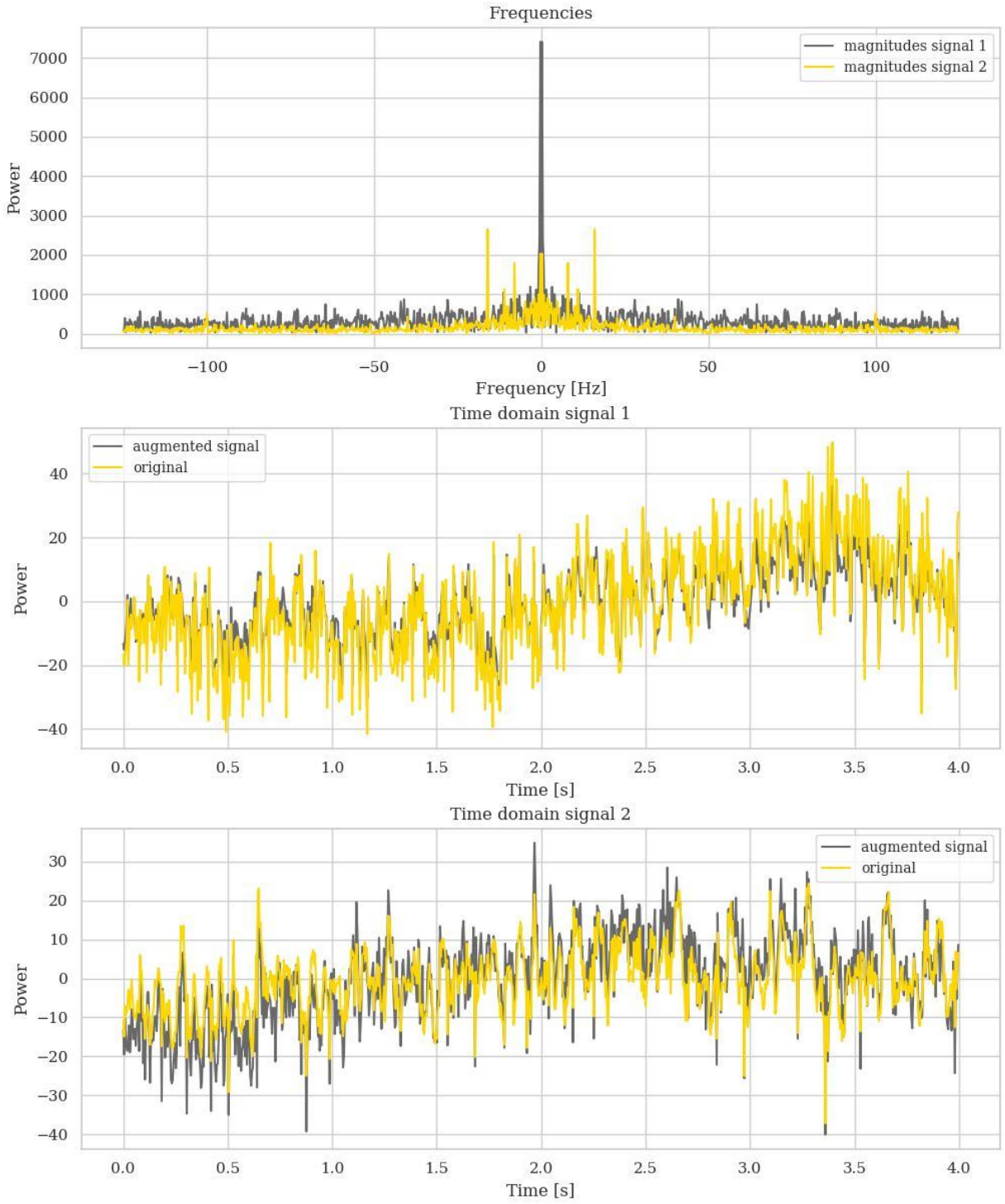

Figure 5: Graph of application of Frequency Mixing on two SSVEP signals of 4 seconds length and a sampling frequency of 250 Ms. The first one shows the frequency spectrum of both signals. The second and third show both the original signals and their augmented versions as an overlay.

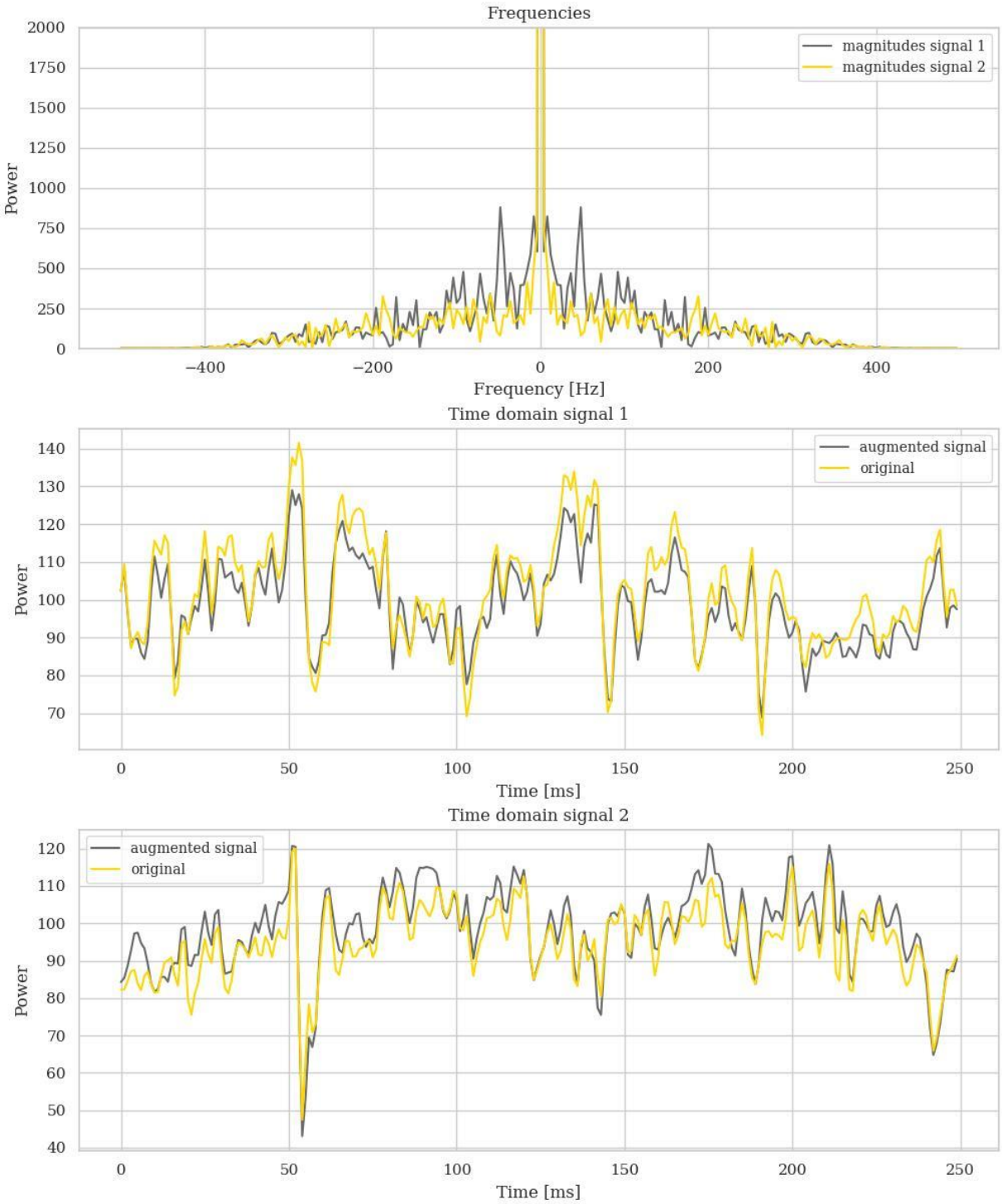

Figure 6: Graph of application of Frequency Mixing on two EEG2Code signals of 250 seconds length and a sampling frequency of 1000 Ms. The first shows the frequency spectrum of both signals. The second and third show both the original signals and their augmented versions as an overlay.

4 Methodology

To write the code for my thesis I used Python 3.10 (*Python Release Python 3.10.0*, n.d.), as this is one of the most used programming languages in data science and machine learning. I made use of several packages such as NumPy, SciPy, and TensorFlow to streamline the use of computing the FFT and setting up the augmentation pipeline.
The objective of my thesis was to enhance SSVEP and EEG2Code classification using a zero-training approach. In this context, the system operates without receiving any prior information about the brain data of the specific subjects it is used for. To simulate that, I needed to exclude the test/validation subjects from the training set. For both datasets my approach was to make one test run with original data and then one with augmented data. As the two datasets differed in size, I approached them differently as well.

4.2 Methodology for SSVEP data

4.2.1 Original data

I began by using the whole dataset with 64 channels and 40 classes but quickly ran into the problem of long calculation times. I reduced the number of both classes and channels gradually and settled on the combination of 4 classes and 4 channels. For my thesis, I only needed to validate the performance of my DA techniques on the original datasets, so I was free to choose the exact configurations. I chose to use the last 4 channels in the dataset which were O1, O2, Oz, and CB2 because all of them are placed at the occipital lobe. For the number of frequencies, I chose to use 8 Hz, 10.2 Hz, 12 Hz, and 13.8 Hz as they were almost evenly spaced over the set of 40 possible targets. I chose this number as I wanted to use more than two targets but also could not use too many as I needed to take training times into account. The signal length I used was four seconds. As four seconds is quite long, to test the possible real-world application of my DA method, I later on additionally made a test run using the same setup, however with signal lengths of one second.
My choices for frequencies and channels were not optimized by any means but can be seen as educated guesses on what would make sense for real-world purposes.

To evaluate the performance of the CNN on each subject, I trained the CNN 35 times iterating over the dataset and taking new training and validation sets and a new test subject. I did this by iterating over a Toeplitz matrix of subject integers, using the number of training subjects

needed, and then taking the next subject as the test subject and the rest of the data as validation data.

After loading the respective datasets, I trained the system on the training set and recorded the validation loss of the classifier on the validation set. I then chose the weights corresponding to the smallest loss as this corresponds to the closest fit for the validation data. With the best weights, I then evaluated the system on the test set and recorded the classification accuracy.
I also wanted to assess the classification accuracy for varied sizes of training data, so I opted to use training sizes of 3,5,10 and 20 subjects, while the validation set had the sizes 31, 29, 24, and 14 subjects respectively. For proper comparability and statistical tests, I did not shuffle the data before splitting it, ensuring the same test subjects for the runs with the same training sets. The whole evaluation of this dataset required the CNN to retrain 140 times and produce 35 evaluations for each of the four training set sizes.

For my approach of recording the weights of the best-fitted model, overfitting was of little concern. The model approached about 100% training accuracy after about 200 epochs regardless of the amount of training data. Normally after reaching this point, validation loss does not significantly decrease but rather increases and overfitting occurs. I set the CNN to run for 1000 epochs with early stopping and patience of 300 epochs, giving the system more than enough time to find the best weights. It often reached epoch numbers of 600+ epochs but always stopped early before reaching the end. While trying to find a suitable amount of training epochs and patience, I tried using more epochs and more patience, but the results did not improve.

4.2.2 Augmented data

To evaluate how well my DA method works on the data, I simply applied it to the training set before starting training and added the augmented data to the original data. Here again, I want to emphasize that DA was only performed after the splitting of data into the different sets and that it was only ever applied to the training set, leaving the test set completely unknown to the classifier.

The SSVEP dataset I used is distributed with each subject's information stored in a separate file. Furthermore, the files had the same order for each subject with information on stimulation frequency and trial block on the same index. To apply Frequency mixing, I looped over each unique pair of subjects, mixed the signals of each of the channels one by one, and stored the resulting data as two "new" subjects. By doing so I created n*(n-1) augmented subjects, where

n represents the number of subjects. For the four different subject sizes of training data 3,5,10 and 20, the method created 6, 20, 90, and 380 “new” subjects.

Finally, I added the newly created dataset to the original set and fed it into the classification system. During my testing phase, I experimented with mixing data from different augmentation methods which did not improve results but left me using a double copy of augmented data in the training set. This did not change the results, but it is necessary to mention as it affected the number of epochs to be used. For the evaluation of each subject size, I used a training time of 500 epochs with early stopping and patience of 100.

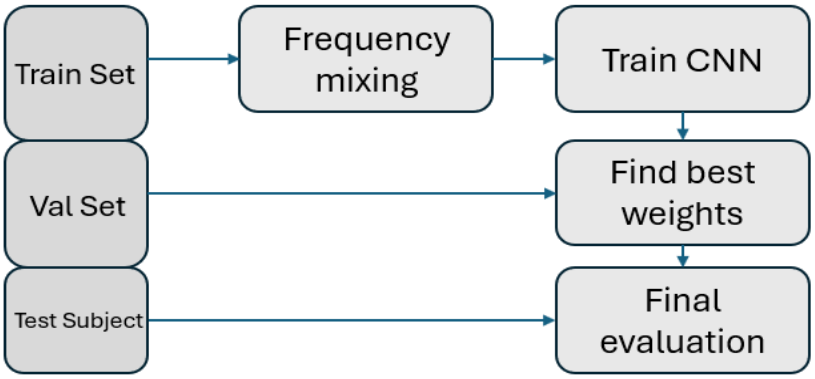

Figure 7: Diagram of the augmented SSVEP training workflow. The data is split into training set, validation set and one test subject. The training set gets augmented, the best weights evaluated using the validation set and the CNNs performance evaluated on the test subject.

4.3 Methodology for EEG2Code

4.3.1 Original data

As mentioned earlier, the EEG2Code dataset featured a very diverse set of activation sequences and sample lengths. In my classification process, I focused solely on the “bit-level” rather than the “full” classification of an EEG2Code target. Since my approach to Data augmentation aimed to enhance the classification of the bits anyway, this strategy was perfectly suitable.

To work with the 5-subject EEG2Code dataset, I opted for a 5-fold cross-validation approach. For the SSVEP approach, taking the best weights from a validation set and evaluating it on a test subject was a reasonable procedure as the dataset was large enough. However, splitting five subjects into three sets is not advisable practice.

K-fold Cross-validation is a resampling technique that splits the data into different training and test sets k-times and evaluates the model's performance each time. By evaluating the model more than once one can be more confident about its performance even with small datasets.
Mimicking the SSVEP set approach, I also wanted to investigate the model's performance with different amounts of data, so I made 3 separate cross-validation runs with a rising amount of training subjects. The spreads of subjects in the 3 runs looked like this: (2 training / 3 test), (3 training / 2 test), (4 training / 1 test). To summarize: for each spread, this approach computes the performance for the 5 folds and gives a good estimate of the average accuracy.
As the data only featured 6 channels and two classes, my laptop was able to handle the amount of data and I did not have to reduce it. For each run, I trained the CNN for 200 epochs as this was the point where training accuracy generally stopped increasing.

4.3.2 Augmentation

To augment this dataset, for all 3 runs of different data spreads, I simply applied Frequency mixing to the training set when creating a new fold for the 5-fold cross-validation.
The augmentation process was similar to the SSVEP set, however, due to the different sample sizes and activation sequences I could not simply align them as for the former set. That meant to usefully mix data with the same underlying activation bit, zero or one, I first ordered the datasets by their activation bits. After ordering them, I had differently sized arrays of the same class which I augmented using Frequency mixing. As the arrays never had the same size, I only augmented a certain amount of them, e.g. the size of the smaller set. For example, with datasets of size 2000 and 4000, I only mixed the first 2000 of the second set with the first one, omitting the rest. I theoretically could have mixed the rest as well, but I chose not to do it as I thought the resulting amount of new data would suffice. For the different spreads, Frequency mixing created 2, 6, and 12 "new" subjects respectively. I always trained the CNN for 200 epochs. Figure 8 shows the workflow for the augmented training of EEG2Code data.

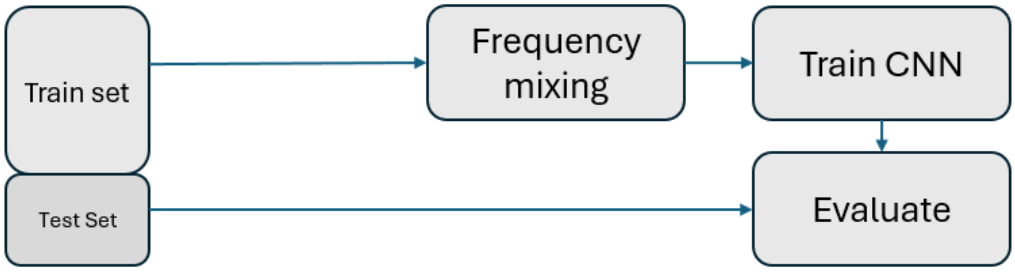

Figure 8: Diagram of the augmented EEG2Code training workflow. The data is split into training set and test set. The training set gets augmented and the CNNs performance evaluated on the test set.

4.1 Classifier

As a classifier, I used a convolutional neural network (CNN) employed by Nagel & Spüler (2019) that is based on a CNN by Gusteren (2018) used for another EEG paradigm. The research group Neuroinformatics at the university also used this network to classify data for a separate project (Meunier et al., 2024). Given its already-established functionality and considering that creating a new classifier was beyond the scope of my thesis, I chose to use this CNN for my work. The CNN processes data of the dimensions (sample length x channels x 1). Theoretically, the network should be able to use data without the additional dimension, however, I included it as I got error messages. As the signal length and the number of channels can be easily changed, I could use the CNN for both SSVEP and EEG2Code without any problems. Figure 9 shows the architecture of the network.

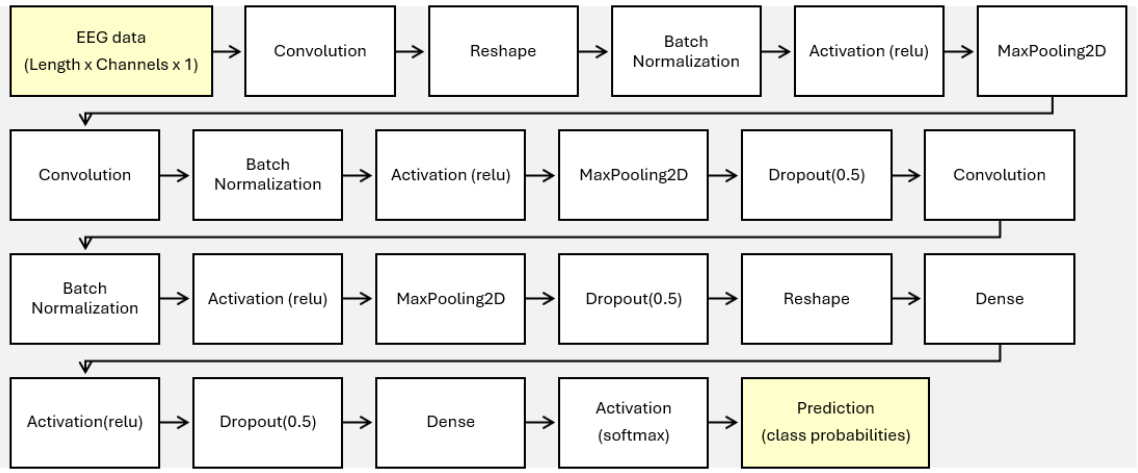

Figure 9: Architecture of the employed CNN.

4 Results

In this section, I will present the results for the 4-second and the 1-second SSVEP datasets, as well as for the EEG2Code dataset. The statistical methodology will be discussed only once in the section for the 4-second SSVEP data but was the same for all three tests.

4.1 SSVEP

In Table 1 I present the results of the original SSVEP dataset. The table columns indicate the different numbers of training subjects. The table shows the test accuracies with a maximum achievable accuracy of 1 (100%). The rows indicate the different subjects taken as the test subjects. The last row shows the mean accuracy of all 35 subjects for all sets of training set sizes. The results for the augmented dataset are visible in Table 2, with the same structure as Table 1.

When looking at the results of Table 1, it is apparent that the classification accuracy is steadily rising with larger training sets. This is to be expected as a larger sample set often improves the generalization of a model. Furthermore, the amount of added accuracy from one set to the other gets smaller the larger the set already was, indicating that the amount of new information is also getting smaller with each added subject.

The highest mean accuracy of 91% was achieved for the training set with 20 subjects, only closely followed by 90% with a training set of 10 subjects. The highest jump in added accuracy was from 3 to 5 training subjects, with an increase of 16%. The number of times that the system classified all signals of the subject with 100 percent accuracy also rises according to the number of training subjects.

Now looking at Table 2, it is visible that the results for data augmented with Frequency mixing exhibit the same behavior as the original data but consistently showcase higher mean accuracies across all sets. The highest mean accuracy of 94% was achieved for the training set with 20 subjects, while 10 and 5 subjects led to 91% and 88% respectively. The highest jump in added accuracy was again achieved for 3 and 5 training subjects, with an increase of 13%.

Table 1: Classification accuracy of unaltered SSVEP data. The table shows classification accuracies for each Subject after training using the respective number of subjects, 3,5, 10, and 20. The final column shows the mean accuracy for each column.

Test Subject	**3 train subjects**	**5 train subjects**	**10 train subjects**	**20 train subjects**
1	0,75	0,96	0,92	0,96
2	0,42	0,92	0,96	1,00
3	0,92	1,00	0,67	0,88
4	0,92	0,96	0,04	1,00
5	0,92	0,88	1,00	1,00
6	0,71	0,88	0,92	1,00
7	1,00	0,92	1,00	0,96
8	0,83	0,63	1,00	0,96
9	1,00	0,79	0,71	0,67
10	0,75	1,00	1,00	1,00
11	0,50	0,67	0,75	1,00
12	0,88	0,75	1,00	1,00
13	0,29	0,92	0,83	0,46
14	0,83	0,71	1,00	0,96
15	0,96	0,96	0,92	1,00
16	0,38	0,83	1,00	0,92
17	0,46	1,00	0,96	0,88
18	0,25	0,67	0,96	1,00
19	0,71	0,96	0,75	1,00
20	0,92	0,79	0,96	0,88
21	1,00	0,96	1,00	1,00
22	0,63	1,00	1,00	1,00
23	0,92	0,92	0,50	0,96
24	0,88	0,33	1,00	0,92
25	0,88	0,96	1,00	0,92
26	0,38	1,00	0,79	0,92
27	0,25	1,00	1,00	0,96
28	0,29	0,25	0,92	0,75
29	0,29	0,67	1,00	0,00
30	0,46	1,00	0,96	1,00
31	0,42	0,75	1,00	0,96
32	0,88	0,75	1,00	1,00
33	0,58	0,67	0,96	1,00
34	0,71	0,88	1,00	0,96
35	0,79	1,00	0,92	1,00
Mean	**0,68**	**0,84**	**0,90**	**0,91**

Table 2: Classification accuracy of 4-second augmented SSVEP data. The table shows classification accuracies for each Subject after training using the respective number of subjects, 3,5, 10, and 20, and adding augmented data using Frequency mixing. The final column shows the mean accuracy for each column.

Test Subject	**3 train subjects**	**5 train subjects**	**10 train subjects**	**20 train subjects**
1	0,96	1,00	0,96	0,92
2	0,88	1,00	0,96	1,00
3	1,00	1,00	0,75	0,83
4	0,96	0,88	0,17	1,00
5	0,96	0,92	1,00	1,00
6	0,92	0,83	0,96	0,96
7	1,00	0,83	0,92	0,96
8	0,83	0,67	1,00	0,96
9	0,88	0,79	0,92	0,67
10	0,54	1,00	1,00	1,00
11	0,46	0,83	0,79	1,00
12	0,42	0,96	1,00	1,00
13	0,75	1,00	0,92	0,75
14	0,88	0,54	1,00	0,96
15	0,96	1,00	1,00	1,00
16	0,33	0,83	1,00	0,96
17	1,00	1,00	0,96	0,96
18	0,71	0,75	0,96	1,00
19	0,88	1,00	0,63	1,00
20	0,75	0,79	0,96	0,96
21	0,92	0,96	1,00	1,00
22	0,92	0,96	1,00	1,00
23	0,83	0,92	0,42	1,00
24	0,92	0,42	1,00	0,96
25	0,92	1,00	1,00	1,00
26	0,54	1,00	0,96	0,96
27	0,42	1,00	1,00	0,96
28	0,25	0,42	1,00	0,88
29	0,25	0,88	1,00	0,29
30	0,33	1,00	1,00	1,00
31	0,92	0,79	1,00	1,00
32	0,79	1,00	1,00	1,00
33	0,67	1,00	0,96	1,00
34	0,58	0,88	0,96	1,00
35	0,92	0,96	0,88	1,00
Mean	**0,75**	**0,88**	**0,91**	**0,94**

While Figure 10 displays the performances of original and augmented data in graphical comparison, Table 3 shows their statistical significance. Both visual elements show that Frequency mixing significantly improved the zero-training classification of most SSVEP datasets.

Across all training-set sizes, there was a consistent improvement in mean classification accuracy compared to the baseline. The biggest improvement was achieved with the smallest dataset of 3 subjects, for which accuracy improved by approximately 7%. For datasets consisting of 5, 10, and 20 subjects, Frequency mixing led to improvements of 4%, 1%, and 3%, respectively. The diminishing rate of improvement with larger datasets can be attributed to the increasing amount of information present before augmentation. The only set that could not be significantly improved was the 10-subject set. Even though its p-value with p = 0.0643 is close to the threshold, it is definitively far from the others.

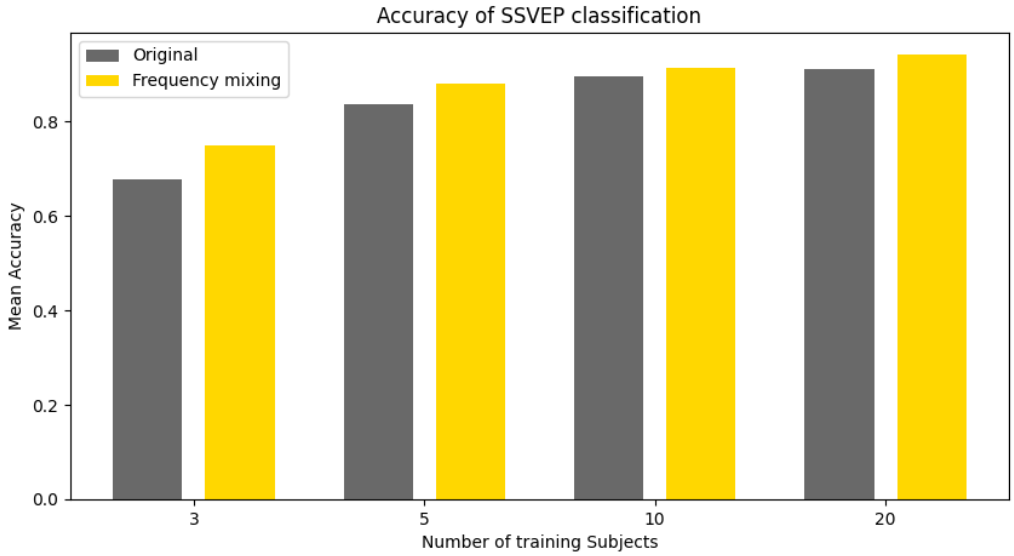

Figure 10: Mean classification accuracy for the original dataset and the dataset augmented with Frequency mixing. The figure shows 4 different blocks, each signifying the number of training subjects.

Table 3: p-values of permutation tests of results for the original and augmented 4-second SSVEP dataset. The test compared the accuracies for each pair of subjects. The threshold for significance is set to 0.05. Significant results are shown in bold.

	Augmented data	**3 subjects**	**5 subjects**	**10 subjects**	**20 subjects**
Original data	**Mean accuracy**	0,75	0,88	0,91	0,94
3 subjects	0,68	**p = 0.0349**			
5 subjects	0,84		**p = 0.0077**		
10 subjects	0,90			p = 0.0643	
20 subjects	0,91				**p = 0.0047**

4.1.1 Statistics

To determine the significance of my results, the statistical test I used was a permutation test with a p-value threshold of 0.05. The permutation test is used for two or more sets of samples and uses the null hypothesis that all samples come from the same distribution. It takes two sets of samples and records the difference of their respective means. Then it randomly shuffles the dataset for each pair of data and again records the difference. By recording this thousands of times and comparing the difference to the unaltered sets, the permutations test calculates the likelihood of them coming from the same distribution (Pesarin & Salmaso, 2010).

I used this test as an alternative to a standard paired t-test as it would need the data to be normally distributed (Pesarin & Salmaso, 2010). As visible in Tables 1 and 2, the CNN generally performed well, often reaching test accuracies of 80% and higher. With a maximum achievable accuracy of 100%, this led to a skewing of the distribution and therefore a loss of normality. Figure 11 shows the graphic visualization of the distribution of accuracies, and it is apparent that none of them are normally distributed but skewed at the higher percentages. I applied a Shapiro-Wilk test to test for normality (Shapiro & Wilk, n.d.), but it was negative for every distribution. An alternative for a t-test would be a Wilcoxon signed rank test, as it does not rely on normality, but it lacks statistical power compared to the t-test (Siegel & Castellan, 1988). The permutation test on the other hand does not need a normal distribution and is more powerful than the Wilcoxon signed-rank test. It does however need more computing time than the others, which in my case was not a problem due to the small set of comparable data.

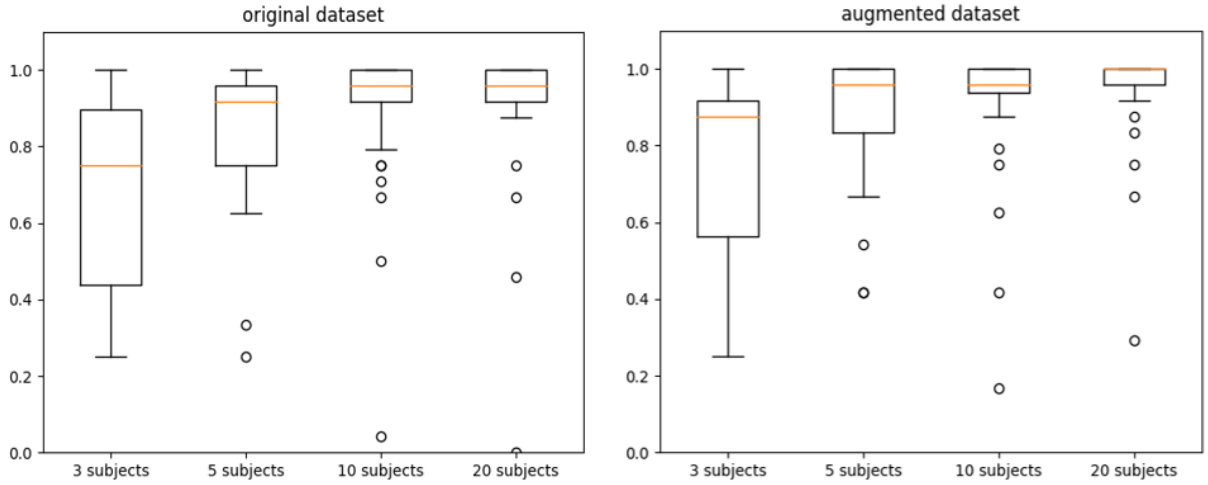

Figure 11: Distribution of accuracies for original and augmented dataset, for original training set sizes of 3,5,10 and 20 subjects.

4.2 SSVEP 1-second length

For most of my research, I worked with long SSVEP signals of 4 seconds length where Frequency mixing was performing well. For real-world applications, a 4-second window would be too slow as this would lead to a maximum of less than 15 letters per minute when using a virtual keyboard setup. A relatively reasonable signal length is about 1 second as it also mirrors the time needed to encode a 63-bit m-sequence for c-VEP and EEG2Code. To test how well the DA method would work for real-world data lengths, I made a second test run where I cut the signals into four sets of one-second lengths.

Table 4 shows the mean accuracies as well as the statistical significance of the results. Frequency mixing performed very well for signals of one-second length. The DA method improved the baseline by 9% and 3% for 3 and 5 training subjects respectively. The results are highly significant as the p-values for both are way below the 0.05 threshold. For the training size of 10 subjects, the DA method could not improve the baseline accuracy. Similar to longer signals, the effectiveness of the DA method seems to get smaller with bigger datasets. By using a shorter signal length, the results confirm the applicability of Frequency mixing for real-world SSVEP signals.

For completeness, Table 5 shows the exact results for each test subject of the short signal test for both datasets. Like the upper tables, the columns indicate the number of training subjects, while the rows indicate the specific test subject. The final row shows the mean test accuracy of each column.

Table 4: P-value of permutation tests of results of augmented and original 1-second SSVEP dataset. The test compared the accuracies for each pair of subjects. The threshold for significance is set to 0.05. Significant results are shown in bold.

	Augmented data	**3 Subjects**	**5 Subjects**	**10 Subjects**
Original data	**Mean accuracy**	0,48	0,68	0,77
3 subjects	0,57	**p = 0.0026**		
5 subjects	0,71		**p = 0.0081**	
10 subjects	0,77			p = 0.4888

Table 5: Classification accuracies of 1 second SSVEP data. The table shows classification accuracies for each Subject after training using the respective number of subjects, 3,5, and 10. The three columns on the left side represent the augmented dataset using Frequency mixing, while the three columns on the right side represent the unaltered dataset. The final column shows the mean accuracy for each column.

	Original			**Augmented**		
Test Subject	**3 Subjects**	**5 Subjects**	**10 Subjects**	**3 Subjects**	**5 Subjects**	**10 Subjects**
1	0,55	0,78	0,74	0,59	0,79	0,74
2	0,63	0,9	0,56	0,75	0,94	0,59
3	0,73	0,94	0,57	0,82	0,98	0,6
4	0,9	0,57	0,57	0,84	0,6	0,34
5	0,92	0,93	0,94	0,77	0,9	0,89
6	0,52	0,73	0,76	0,63	0,75	0,72
7	0,77	0,51	0,96	0,54	0,38	0,89
8	0,52	0,58	0,84	0,65	0,49	0,84
9	0,35	0,28	0,31	0,38	0,31	0,33
10	0,2	0,65	0,77	0,43	0,74	0,86
11	0,3	0,57	0,31	0,28	0,67	0,45
12	0,43	0,25	0,86	0,23	0,23	0,88
13	0,3	0,73	0,7	0,42	0,73	0,73
14	0,53	0,35	0,89	0,57	0,33	0,94
15	0,71	0,86	0,7	0,73	0,9	0,69
16	0,26	0,43	0,81	0,35	0,51	0,89
17	0,24	0,78	0,89	0,84	0,9	0,95
18	0,3	0,74	0,8	0,55	0,76	0,83
19	0,26	0,75	0,45	0,38	0,9	0,39
20	0,41	0,5	0,81	0,68	0,54	0,92
21	0,39	0,84	1	0,52	0,94	0,99
22	0,49	0,75	1	0,4	0,79	1
23	0,52	0,82	0,4	0,75	0,77	0,34
24	0,65	0,43	0,92	0,72	0,36	0,86
25	0,7	0,63	0,99	0,73	0,83	0,99
26	0,3	0,88	0,75	0.30	0,99	0,69
27	0,25	1	0,68	0,29	0,94	0,81
28	0,41	0,26	0,95	0,33	0,34	0,92
29	0,25	0,85	0,79	0,24	0,84	0,8
30	0,35	0,97	0,91	0,3	1	0,77
31	0,65	0,59	0,9	0,92	0,59	0,91
32	0,36	0,71	0,94	0,84	0,74	0,96
33	0,36	0,83	0,96	0,45	0,91	0,97
34	0,57	0,72	0,71	0,61	0,72	0,68
35	0,8	0,83	0,96	0,85	0,86	0,94
Mean	**0,48**	**0,68**	**0,77**	**0,57**	**0,71**	**0,77**

4.2 Results for EEG2Code

In Table 6 I show the classification accuracies for the EEG2code dataset. The Table is divided into the results of the different data spreads for both the original and the augmented data. The columns indicate the training set sizes while the rows indicate the folds for the cross-validation. The last row shows the mean accuracies.
Inspecting only the accuracies for the original data, one can see that, even though only by a little, the accuracy gets higher with more training data. From this, it can be inferred that the CNN could profit from the increase in training data and create a more generalized model.

Contrary to SSVEP signals, the augmented dataset did not perform better than the original but performed worse most of the time. Taking a closer look at the results, the augmented set performed slightly worse on the 3 and 4-subject training sets while it performed equally for the 2-subject training set.
With the knowledge that the CNN can profit from a bigger dataset one might infer that it should be possible to augment EEG2Code data. However, given the lack of performance of Frequency mixing, I need to assume that this specific DA method is not able to effectively augment this type of neurophysiological signal. I will talk about potential reasons for this outcome in the discussion part of this thesis.

Table 6: Classification accuracy for original and augmented EEG2Code data. The rows show the classification accuracy for each test subject. The last row shows the mean accuracy.

	Original			**Augmented**		
K-folds	**2 Subjects**	**3 Subjects**	**4 Subjects**	**2 Subjects**	**3 Subjects**	**4 Subjects**
1	0,61	0,60	0,59	0,61	0,59	0,60
2	0,59	0,56	0,54	0,58	0,57	0,55
3	0,59	0,61	0,67	0,58	0,61	0,63
4	0,64	0,68	0,71	0,65	0,68	0,70
5	0,64	0,69	0,71	0,64	0,67	0,70
Mean	**0,61**	**0,63**	**0,65**	**0,61**	**0,62**	**0,64**

5 Discussion

In this section, I will discuss the implications of the obtained results. The goal of this thesis was to develop a novel DA method that succeeds in enhancing the performance of zero-training SSVEP-based BCIs and see if it is possible to extend its applicability to EEG2Code.

Frequency mixing was able to successfully augment SSVEP signals for long and short signal lengths and showed to be even useful for diverse sets of training data sizes. While it managed to improve accuracy even in data-rich settings, its main power lies in low data regimes where it could boost classification accuracy by 7% and 9% compared to the baselines.

As the augmented signals don´t look drastically different from their originals I would dub the method a “conservative” augmentation method that doesn´t introduce variation unless it already exists in the dataset. This may be a main selling point as well as its main drawback. While being able to create massive amounts of data, it only reflects what was already existent, therefore failing to adapt to outliers in the future.

This may also be one reason why it was not able to work for EEG2Code. Comparing Figure 5 and Figure 6, the augmented signals show the same characteristics for both datatypes, however, this might just not be the right type of perturbation for EEG2Code. While Frequency mixing did not improve on the baseline of EEG2Code it also did not substantially decrease the classification accuracy making it appear irrelevant. One could argue that the method was just not able to create data that was semantically interesting enough to produce a positive effect.

Another reason for the lack of performance could be the opposite of the latter. As I could not align the datasets, I had to reorder the signals according to the singular bits and align them afterward. However, the alignment was not perfect as the signals themselves were stimulated by different surrounding bits. To explain this, I want to refer to the notion that while the impact of one bit can be measured for 250 ms, all the other bits before and after the respective stimulation bit should be present in a signal as well and somehow contribute to its recognizability. Now, when using Frequency mixing on signals with different ambient activation patterns the only constant is the first activation bit, while the rest will be scrambled together, possibly changing the signal too drastically. While this may have affected the performance of Frequency mixing, I am skeptical, given that Nagel & Spüler (2018) already effectively utilized completely random stimulation patterns for training an EEG2Code BCI.

5.2 Limits of Frequency mixing

Even though the amount of data created by Frequency mixing can rise quadratically with the amount of training data available, one soon gets to a point where the added accuracy gets insignificantly small. This phenomenon is well known in the context of machine learning as “diminishing return” (Thompson et al., 2021) and occurs if a model already has a high accuracy. The main reason for this is that if a model already has an excellent understanding of the task, the further addition of data might not provide enough novel information to refine the model's capabilities.

In my case, this phenomenon could be observed for the bigger training sets where Frequency mixing struggled to significantly improve the accuracies. Even though for the 4-second samples the augmented 20 subject-dataset performed significantly better, the improvement of the 10 subject run was insignificant. One could argue the former to be overperforming or the latter to be underperforming but the mainline is that with rising amounts of original data after some point augmenting a dataset may not result in any improvement. For the test with the 1-second cuts of SSVEP data, this point was reached already with 10 subjects where the augmented dataset could not even improve the accuracy but performed equally to the baseline.

Another limit of Frequency mixing is its performance. While the improvements were significant most of the time, Frequency mixing only once provided better accuracy than the baseline of the higher subject value. Namely, 10 subjects augmented (0,91428571) versus 20 subjects original (0,90952381) for the 4-second SSVEP data. Every other time the performance was lower than the next higher subject class. This observation shows that the method is not to be seen as a substitution for real data and emphasizes the value of original data if available.

5.2.1 Implications of the best performances

One thing I hardly ever mentioned was the significance of the best performances I achieved during this thesis. As it was never really important to achieve the best accuracies possible but to increase the baselines, it might have gone under that the underlying idea for my thesis was to take something unviable (zero-training BCI usage) and test if it can be made viable via data augmentation.

Now that I have effectively shown the significance of my method, I still need to address that no performance ever lived up to anything truly usable in the real world. As 94% for 4 targets seems to be quite good, it was only achieved for data lengths of 4 seconds which would lead to very slow BCIs. While augmentation worked for the 1-second data, the maximum achieved accuracy

was 77% for 4 targets. This means that almost every fourth command would be wrong and therefore as well not suited for real-world scenarios.

Concluding this part, the results show that, while Frequency mixing can significantly increase a model's generalizability, it is preferably applied to small datasets, with a low possibility of gathering more data. To make zero-training BCIs truly viable, Frequency mixing might not be the answer but at least one step in the right direction.

5.1 Extensions of Frequency mixing

During my work, I explored some other aspects of mixing data and want to briefly mention what I tried, what already exists, and what could be an interesting topic going into the future.

5.1.1 Phase mixing

Initially, when coming up with the idea of Frequency mixing, inspired by its potential, I alternatively thought of mixing the phase information of two signals while leaving the frequencies untouched. I tried this method but as it did not perform well, I omitted this idea in early development. After Frequency mixing did not work for EEG2Code, I tried out Phase mixing as well but again with no success. However, despite its shortcomings, I can imagine that this method still holds use for some kind of data, so I want to emphasize this idea for further research.

5.5.2 Adjustable mixing

An easy way of extending Frequency mixing would be to change the mixing factor. For my algorithm, I always take the mean of two complete signals, but it would be easy to take only 50% of the power of one signal and 150% of the other signal making the mixing adjustable. I did not explore this, but this idea was implemented in the paper of Chen et al (2023) which is the same paper that also proposed Frequency mixing as an augmentation technique.
The possible advantage of masking the powers would be making even more nuanced signals, which might lead to even better results.

5.5.3 Wider range of mixing targets

In this work I only mixed signals of the same channel and only for two different subjects at a time. I want to suggest looking into the mixing of signals from the same subject taken in multiple blocks or sessions and the mixing of signals from different channels. Another idea

would be to mix more than two subjects, creating an overall mean frequency spectrum. As established, Frequency mixing seems very robust as long as both signals have the same activation frequency. Using this property and some of the suggestions above, combined with adjustable mixing one might be able to create even more data.

5.5.4 Perturbing spectra before mixing

As mentioned earlier, Frequency mixing is rather conservative when it comes to introducing perturbations but balances this out with a high output of data. One could easily try to introduce perturbations as filtering out or empowering whole frequency ranges. As the signals get averaged in the end, even with many perturbations the resulting signals may still be sensible, while being a bit more “creative”.

3.1 Powershifting

Powershifting was a DA method developed relatively early during my research. As it did not work in the beginning, I omitted further research into the concept and developed Frequency mixing as a successor. However, late in the process, I found out that it can positively augment SSVEP data. Due to exceeding the time limit of my work, I chose not to run deeper test runs but instead will present it here as a stepping stone for future works.

The idea of Powershifting was to relocate the power of the frequencies in the frequency spectrum of a signal by broadening and squishing the spectrum inwards and outwards. By doing this, I thought the method would be able to mimic the signal response of different subjects. As each subject has a different distribution of powers of frequencies, taking real data and changing the distribution slightly seemed feasible. While I tested both, shifting the power inwards to the low frequencies and outwards to the high frequencies, only the latter was able to effectively augment the dataset.

I accomplished this by creating the Fourier transform of a given signal and then multiplying a mask to the frequency spectrum that rescaled it from 1 to x where x would be the shifting value given. The mask closely resembles a “V” with the steepness of the slope determining the shift. I tested the method for different shifting factors and found that even very drastic augmentation worked well for the SSVEP dataset. Figure 11 shows such a shifting mask with a shifting value of 1%.

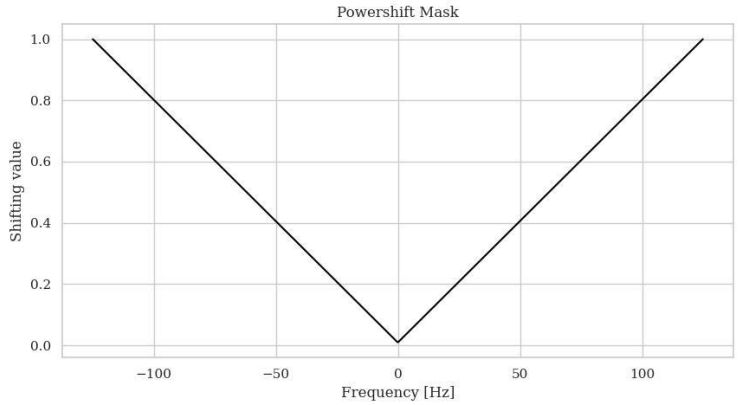

Figure 12: Image of a powershifting mask that is multiplied with the frequency spectrum to change the intensities of the frequencies along a gradient. This mask is used for signals with sampling frequency of 250 Hz and has a shifting value of 1% to boost high frequencies.

The functions workflow can be summarized as this: When taking a multichannel sample of a subject, for each channel the frequency spectrum is created via a Fourier Transform. Then one creates a shifting mask with the desired slope which is multiplied by the spectrum. After that, the spectrum is normalized to have the same amount of power as before the shifting. The result is a changed frequency spectrum that is then returned to the time domain via an inverse Fourier Transform. The newly created signals are then concatenated and added to the original data. This method essentially doubles the size of the dataset if used only with one mask, however, one could try to use this method multiple times with different shifting values.

Figure 12 shows how data augmented by this method is changed. The upper graph shows the spectrum of the original and augmented signals, while the lower graph shows them in the time domain. For this graph, the shifting factor was 1%. In the plot, one can see the drastic effect of applying the shifting mask onto the frequency spectrum in that the low frequencies almost get canceled out (black) compared to the original signal (yellow). The higher frequencies, which are less dominant in the original get emphasized in the augmented signal. One can see this effect in the time domain as well. While the original signal (yellow) is made up of a lot of short waves that themselves fluctuate as longer waves, the augmented signal (black) does not feature the longer waves, however a pronunciation of the short waves.

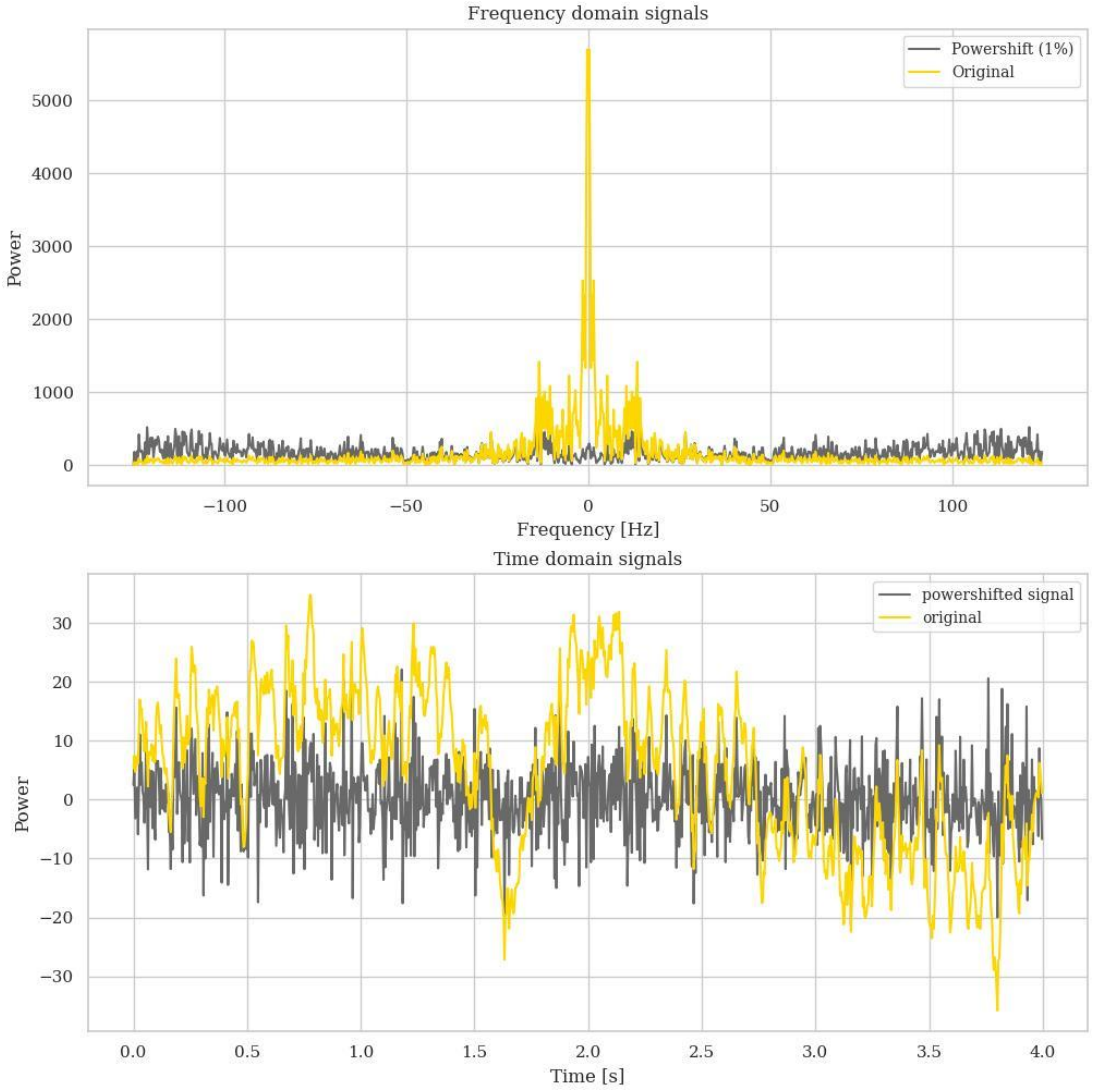

Figure 13: Graph of application of Powershifting on a SSVEP signal of 4 seconds length and a sampling frequency of 250 Hz. The first window shows the frequency spectrum of the original signal and its augmented version. The second window shows both the original signal and its augmented versions in the time domain. The shifting factor is 1%.

6 Conclusion

In this master thesis, the primary objective was to develop a novel Data augmentation technique that enhances the performance of zero-training SSVEP-based brain-computer interfaces and investigate its applicability to EEG2Code BCI setups. During my research, I developed a method called Frequency mixing, which averages the frequency spectra of two subjects and creates new time domain signals. I applied the method to both datasets and examined the experimental results.

6.1 Findings

The results demonstrated a significant improvement in the classification of SSVEP data when using Frequency mixing. I attribute the positive effect to the high output of semantically qualitative data. However, the application of Frequency mixing to EEG2Code resulted in no improvement in accuracy, suggesting a limitation of this DA method to certain types of data. While there are many possible explanations for this outcome, the main point is the methods fail to add useful information to the dataset.

6.4 Limits

Even though Frequency mixing showed much potential in augmenting the SSVEP dataset, it still has some limitations that might hamper its real-world application:
The effectiveness of the DA method diminishes with larger initial datasets and the accuracies achieved with it rarely surpass the baseline of higher subject values. Both findings suggest that it is preferably applied to small datasets where gathering more data is challenging.

6.5 Concluding Remarks

In conclusion, this study contributes to the understanding of Data augmentation techniques in the context of SSVEP and EEG2Code-based BCIs. As Frequency mixing succeeded in augmenting one datatype but failed for the other, the results call for further research into Data augmentation methods specifically built for different neurophysiological signals. The insights gained from this work provide a foundation for future research in improving the robustness and generalization of BCIs.

References

Abiri, R., Borhani, S., Sellers, E. W., Jiang, Y., & Zhao, X. (2019). A comprehensive review of EEG-based brain–computer interface paradigms. *Journal of Neural Engineering*, *16*(1), 011001. https://doi.org/10.1088/1741-2552/aaf12e

Arslan, S. S., & Sinha, P. (2023). *Information Transfer Rate in BCIs: Towards Tightly Integrated Symbiosis* (arXiv:2301.00488). arXiv. https://doi.org/10.48550/arXiv.2301.00488

Attia, M., Hettiarachchi, I., Hossny, M., & Nahavandi, S. (2018). A time domain classification of steady-state visual evoked potentials using deep recurrent-convolutional neural networks. *2018 IEEE 15th International Symposium on Biomedical Imaging (ISBI 2018)*, 766–769. https://doi.org/10.1109/ISBI.2018.8363685

Bassi, P. R. A. S., Rampazzo, W., & Attux, R. (2021). Transfer Learning and SpecAugment applied to SSVEP Based BCI Classification. *Biomedical Signal Processing and Control*, *67*, 102542. https://doi.org/10.1016/j.bspc.2021.102542

Beres, A. M. (2017). Time is of the Essence: A Review of Electroencephalography (EEG) and Event-Related Brain Potentials (ERPs) in Language Research. *Applied Psychophysiology and Biofeedback*, *42*(4), 247–255. https://doi.org/10.1007/s10484-017-9371-3

Beverina, F., Palmas, G., Silvoni, S., Piccione, F., & Giove, S. (2003). User adaptive BCIs: SSVEP and P300 based interfaces. *PsychNology Journal*, *1*, 331–354.

Bin, G., Gao, X., Wang, Y., Hong, B., & Gao, S. (2009). VEP-based brain-computer interfaces: Time, frequency, and code modulations [Research Frontier]. *IEEE Computational Intelligence Magazine*, *4*(4), 22–26. https://doi.org/10.1109/MCI.2009.934562

Boashash, B. (Ed.). (2003). Chapter 2 - Heuristic Formulation of Time-Frequency Distributions00Author: Boualem Boashash, Signal Processing Research Centre, Queensland University of Technology, Brisbane, Australia. Reviewers: K. Abed-Meraim, A. Beghdadi, M. Mesbah, G. Putland and V. Sucic. In *Time Frequency Analysis* (pp. 29–57). Elsevier Science. https://doi.org/10.1016/B978-008044335-5/50023-1

Chen, M., Xu, Z., Zeng, A., & Xu, Q. (2023). *FrAug: Frequency Domain Augmentation for Time Series Forecasting* (arXiv:2302.09292). arXiv. https://doi.org/10.48550/arXiv.2302.09292

Gao, J., Song, X., Wen, Q., Wang, P., Sun, L., & Xu, H. (2021). *RobustTAD: Robust Time Series Anomaly Detection via Decomposition and Convolutional Neural Networks* (arXiv:2002.09545). arXiv. https://doi.org/10.48550/arXiv.2002.09545

Holmes, J. (2007). *Spread Spectrum Systems for GNSS and Wireless Communications*.

Hosna, A., Merry, E., Gyalmo, J., Alom, Z., Aung, Z., & Azim, M. A. (2022). Transfer learning: A friendly introduction. *Journal of Big Data*, *9*(1), 102. https://doi.org/10.1186/s40537-022-00652-w

Iglesias, G., Talavera, E., González-Prieto, Á., Mozo, A., & Gómez-Canaval, S. (2023). Data Augmentation techniques in time series domain: A survey and taxonomy. *Neural Computing and Applications*, *35*(14), 10123–10145. https://doi.org/10.1007/s00521-023-08459-3

Kim, G., Han, D. K., & Ko, H. (2021). *SpecMix: A Mixed Sample Data Augmentation method for Training withTime-Frequency Domain Features* (arXiv:2108.03020). arXiv. https://doi.org/10.48550/arXiv.2108.03020

Levesque, L. (2014). Nyquist sampling theorem: Understanding the illusion of a spinning wheel captured with a video camera. *Physics Education*, *49*, 697. https://doi.org/10.1088/0031-9120/49/6/697

Liu, X., Makeyev, O., & Besio, W. (2020). Improved Spatial Resolution of Electroencephalogram Using Tripolar Concentric Ring Electrode Sensors. *Department of Electrical, Computer, and Biomedical Engineering Faculty Publications*. https://doi.org/10.1155/2020/6269394

Luo, R., Xu, M., Zhou, X., Xiao, X., Jung, T.-P., & Ming, D. (2023). Data Augmentation of SSVEPs Using Source Aliasing Matrix Estimation for Brain-Computer Interfaces. *IEEE Transactions on Bio-Medical Engineering*, *70*(6), 1775–1785. https://doi.org/10.1109/TBME.2022.3227036

Martínez-Cagigal, V., Thielen, J., Santamaría-Vázquez, E., Pérez-Velasco, S., Desain, P., & Hornero, R. (2021). Brain–computer interfaces based on code-modulated visual evoked potentials (c-VEP): A literature review. *Journal of Neural Engineering*, *18*(6), 061002. https://doi.org/10.1088/1741-2552/ac38cf

Marx, E., Benda, M., & Volosyak, I. (2019). Optimal Electrode Positions for an SSVEP-based BCI. *2019 IEEE International Conference on Systems, Man and Cybernetics (SMC)*, 2731–2736. https://doi.org/10.1109/SMC.2019.8914280

Meunier, A., Žák, M. R., Munz, L., Garkot, S., Eder, M., Xu, J., & Grosse-Wentrup, M. (2024). *A Conversational Brain-Artificial Intelligence Interface* (arXiv:2402.15011). arXiv. https://doi.org/10.48550/arXiv.2402.15011

Miller, K. J., Hermes, D., & Staff, N. P. (2020). The current state of electrocorticography-based brain–computer interfaces. *Neurosurgical Focus*, *49*(1), E2. https://doi.org/10.3171/2020.4.FOCUS20185

Müller, M. (2015). *The Fourier Transform in a Nutshell* (pp. 39–57).

Mumuni, A., & Mumuni, F. (2022). Data augmentation: A comprehensive survey of modern approaches. *Array*, *16*, 100258. https://doi.org/10.1016/j.array.2022.100258

Nagel, S., & Spüler, M. (2018a). Modelling the brain response to arbitrary visual stimulation patterns for a flexible high-speed Brain-Computer Interface. *PLOS ONE*, *13*(10), e0206107. https://doi.org/10.1371/journal.pone.0206107

Nagel, S., & Spüler, M. (2018b). Modelling the brain response to arbitrary visual stimulation patterns for a flexible high-speed Brain-Computer Interface. *PLOS ONE*, *13*(10), e0206107. https://doi.org/10.1371/journal.pone.0206107

Nagel, S., & Spüler, M. (2019). World's fastest brain-computer interface: Combining EEG2Code with deep learning. *PLOS ONE*, *14*(9), e0221909. https://doi.org/10.1371/journal.pone.0221909

Odom, J. V., Bach, M., Brigell, M., Holder, G. E., McCulloch, D. L., Mizota, A., Tormene, A. P., & International Society for Clinical Electrophysiology of Vision. (2016). ISCEV standard for clinical visual evoked potentials: (2016 update). *Documenta Ophthalmologica. Advances in Ophthalmology*, *133*(1), 1–9. https://doi.org/10.1007/s10633-016-9553-y

Pan, Y., Li, N., Zhang, Y., Xu, P., & Yao, D. (2023). *Short-length SSVEP data extension by a novel generative adversarial networks based framework* (arXiv:2301.05599). arXiv. http://arxiv.org/abs/2301.05599

Park, D. S., Chan, W., Zhang, Y., Chiu, C.-C., Zoph, B., Cubuk, E. D., & Le, Q. V. (2019). SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition. *Interspeech 2019*, 2613–2617. https://doi.org/10.21437/Interspeech.2019-2680

Pesarin, F., & Salmaso, L. (2010). *Permutation Tests for Complex Data: Theory, Applications and Software*. John Wiley & Sons.

Portillo-Lara, R., Tahirbegi, B., Chapman, C. A. R., Goding, J. A., & Green, R. A. (2021). Mind the gap: State-of-the-art technologies and applications for EEG-based brain–

computer interfaces. *APL Bioengineering*, *5*(3), 031507. https://doi.org/10.1063/5.0047237

Python Release Python 3.10.0. (n.d.). Python.Org. Retrieved 1 April 2024, from https://www.python.org/downloads/release/python-3100/

Robben, A., Chumerin, N., Manyakov, N., Combaz, A., Vliet, M., & Van Hulle, M. (2011). *Combining object detection and brain computer interfacing: Towards a new way of subject-environment interaction*. 1–6. https://doi.org/10.1109/MLSP.2011.6064548

Shapiro, S. S., & Wilk, M. B. (n.d.). *An Analysis of Variance Test for Normality (Complete Samples)*.

Sheykhivand, S., Yousefi Rezaii, T., Naderi, A., & Romooz, N. (2017). *Comparison Between Different Methods of Feature Extraction in BCI Systems Based on SSVEP*.

Shorten, C., & Khoshgoftaar, T. M. (2019). A survey on Image Data Augmentation for Deep Learning. *Journal of Big Data*, *6*(1), 60. https://doi.org/10.1186/s40537-019-0197-0

Siegel, S., & Castellan, N. J. (1988). *Nonparametric statistics for the behavioral sciences* (2. ed.). MacGraw-Hill.

Thielen, J., Marsman, P., Farquhar, J., & Desain, P. (2021). From full calibration to zero training for a code-modulated visual evoked potentials for brain–computer interface. *Journal of Neural Engineering*, *18*(5), 056007. https://doi.org/10.1088/1741-2552/abecef

Thompson, N. C., Greenewald, K., Lee, K., & Manso, G. F. (2021). Deep Learning's Diminishing Returns: The Cost of Improvement is Becoming Unsustainable. *IEEE Spectrum*, *58*(10), 50–55. https://doi.org/10.1109/MSPEC.2021.9563954

Um, T. T., Pfister, F. M. J., Pichler, D., Endo, S., Lang, M., Hirche, S., Fietzek, U., & Kulić, D. (2017). Data Augmentation of Wearable Sensor Data for Parkinson's Disease Monitoring using Convolutional Neural Networks. *Proceedings of the 19th ACM*

International Conference on Multimodal Interaction, 216–220. https://doi.org/10.1145/3136755.3136817

Wang, M., & Guo, L. (2020). *Intracortical Electrodes* (pp. 67–94). https://doi.org/10.1007/978-3-030-41854-0_4

Wang, Y., Chen, X., Gao, X., & Gao, S. (2017). A Benchmark Dataset for SSVEP-Based Brain–Computer Interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, *25*(10), 1746–1752. https://doi.org/10.1109/TNSRE.2016.2627556

Wen, Q., Sun, L., Yang, F., Song, X., Gao, J., Wang, X., & Xu, H. (2021). Time Series Data Augmentation for Deep Learning: A Survey. *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence*, 4653–4660. https://doi.org/10.24963/ijcai.2021/631

Wolpaw, J. R., Birbaumer, N., McFarland, D. J., Pfurtscheller, G., & Vaughan, T. M. (2002). Brain–computer interfaces for communication and control. *Clinical Neurophysiology*, *113*(6), 767–791. https://doi.org/10.1016/S1388-2457(02)00057-3