Data augmentation for learning predictive models on EEG: a systematic comparison

Cédric Rommel, Joseph Paillard, Thomas Moreau, Alexandre Gramfort

Data augmentation for learning predictive models on EEG: a systematic comparison

Cédric Rommel, Joseph Paillard, Thomas Moreau & Alexandre Gramfort

Université Paris-Saclay, Inria, CEA, Palaiseau, 91120, France

E-mail: {firstname.lastname}@inria.fr

Abstract.

Objective: The use of deep learning for electroencephalography (EEG) classification tasks has been rapidly growing in the last years, yet its application has been limited by the relatively small size of EEG datasets. Data augmentation, which consists in artificially increasing the size of the dataset during training, can be employed to alleviate this problem. While a few augmentation transformations for EEG data have been proposed in the literature, their positive impact on performance is often evaluated on a single dataset and compared to one or two competing augmentation methods. This work proposes to better validate the existing data augmentation approaches through a unified and exhaustive analysis.

Approach: We compare quantitatively 13 different augmentations with two different predictive tasks, datasets and models, using three different types of experiments.

Main results: We demonstrate that employing the adequate data augmentations can bring up to 45% accuracy improvements in low data regimes compared to the same model trained without any augmentation. Our experiments also show that there is no single best augmentation strategy, as the good augmentations differ on each task.

Significance: Our results highlight the best data augmentations to consider for sleep stage classification and motor imagery brain-computer interfaces. More broadly, it demonstrates that EEG classification tasks benefit from adequate data augmentation.

Keywords: Data augmentation, sleep stage classification, brain-computer interface
Submitted to: *J. Neural Eng.*

1. Introduction

Decoding the brain electrical activity is a great scientific challenge for both clinicians and researchers seeking a better understanding of brain dynamics. Recent attempts to leverage deep learning for this difficult task have shown promising results [38]. These new methods have led to performance gains in a wide range of clinically relevant tasks, such as the automatic sleep stage classification from polysomnographic recordings [8, 32, 31, 7]. The ambition of deep learning models is to automatically learn relevant representations from high dimensional data such as EEG [2, 15], while previously used brain decoding methods relied on prior knowledge and handcrafted features [33]. Doing so, deep learning approaches require a less sharp understanding of the underlying neurophysiology and are thus more versatile. Yet, this comes at the cost of requiring large training datasets.

Indeed, most of the breakthroughs in deep learning have been enabled by large datasets such as *ImageNet* [39]. Unfortunately, similar datasets do not exist in neuroscience as labeled brain data remains comparatively scarce. Labelling EEG recordings requires a high expertise, is time consuming and can sometimes be inaccurate due to the bias introduced by the human annotator [37]. A second obstacle to the application of deep learning in neuroscience is the high inter-subject variability that is inherent in brain signals [11]. Along with the lack of data, this property makes the generalization on unseen subjects particularly difficult. Without proper regularization, both of these problems can hinder generalization performance and lead to overfitting.

To mitigate the small scale of the neuroscience databases, a promising direction is the use of data augmentation [23, 28, 14]. Data augmentation allows to increase artificially the size of the training set by adding new synthetic examples. These examples are generated by randomly transforming existing ones in a label-preserving way. Doing so, data augmentation helps the decision function to become invariant to the transformation enforced, thus softly reducing the hypothesis space of the training problem. Consequently, it can be interpreted as a regularization method that induces a useful bias by preventing the model from focusing on irrelevant features [9], which in the end makes it less prone to overfitting [43].

Although data augmentation is a well-established method in computer vision, and despite a number of recent studies, data augmentation is still under-explored for EEG data. Among the recent studies listed in Table 1, some propose to perturb the data in either the spatial domain of sensors [22, 12, 40], the frequency domain of signals [42, 10] or the time domain [44, 27, 36]. While some reviews cover data augmentation for EEG [38, 25, 19], they mainly focus on summarizing results from the literature without carrying out extensive new experiments.

Contributions In this paper, we propose to better validate the main existing EEG data augmentation methods listed in Table 1 through a unified and exhaustive analysis in the context of sleep stage classification and motor imagery brain-computer interfaces (BCI) [11]. In total, we compare 13 different augmentations with two different predictive tasks, datasets and models. The objectives of our experiments are three-fold: (i) to evaluate the impact of the magnitude of each transformation, (ii) to compare the benefit of each augmentation with different training set

sizes, and (iii) to highlight how the effects of augmentations vary across data classes. We organize our analysis as follows. First, we outline in section 2 the experimental setting and protocol used to tune and compare all data augmentations. Then, we describe in more details the rationale of each augmentation and present the experimental results for time domain augmentations (section 3), frequency domain augmentations (section 4) and spatial domain augmentations (section 5). Finally, we summarize our findings and draw more general conclusions in section 6. The code used in our experiments has been available in an open-source repository‡.

2. Experimental protocol

The experiments are conducted on two different tasks: sleep stage classification and motor imagery classification in the context of BCI. Both tasks and datasets are described in Section 2.1. Section 2.2 presents the common experimental protocols used to study each data augmentation and Section 2.3 provides implementation details for reproducibility purposes.

2.1. EEG classification tasks

2.1.1. Sleep stage classification

Dataset and preprocessing Sleep stage classification is essential to diagnose sleep disorders such as sleep apnea or insomnia. It consists in classifying EEG windows of 30 seconds into 5 stages defined by the *American Academy of Sleep Medecine* (*AASM*) manual [4]: Wake (W), Rapid Eye Mouvement (REM) and Non REM stages 1, 2 and 3 (respectively N1, N2 and N3). This task is usually performed by sleep experts, which can be very time consuming and subjective, motivating the use of automatic classification systems when possible. To evaluate the impact of data augmentation on automatic sleep stage classification, we used the *SleepPhysionet* dataset [16], which contains whole night polysomnographic recordings from 78 healthy subjects using two EEG channels: Fpz-Cz and Pz-Oz. In this dataset, each signal's window has been annotated by trained technicians according to the *Rechtschaffen and Kales* manual [34], which are re-asigned to the more recent stages from the *AASM* manual. As suggested in [7], a minimal data preprocessing is applied, consisting in a low-pass filter with a cutoff frequency of 30 Hz, followed by a standardization step (each channel's signal is centered and scaled to have unit variance).

Splitting strategy EEG recordings have a strong inter-subject variability [11]. Subsequently, to test the trained models in real conditions, some subjects must be set aside and used only for testing in order to avoid subject related information leakage [38]. To take this into account, the following splitting strategy has been defined. First, the dataset is separated into k-folds, each of them containing different subjects. One fold is left out for testing and among the remaining $(k-1)$, 20% of the subjects are used for validation. Finally, a subset of the remaining dataset is extracted using a stratified split and used for training. This last step allows to train the model in low data regime while maintaining the distribution of classes comparable to the validation and test sets.

Model and training details Experiments are performed using a deep convolutional neural network that has been designed for sleep stage

‡ https://github.com/eeg-augmentation-benchmark/eeg-augmentation-benchmark-2022

Augmentation	Type	Reference	Short description
`FTSurrogate`	F	Schwabedal et al. [42]	Randomize Fourier phases of all channels.
`BandstopFilter`	F	Mohsenvand et al. [27], Cheng et al. [10]	Randomly filter a small frequency band of all channels.
`FrequencyShift`	F	Rommel et al. [36]	Randomly translate all channels PSD by small shift.
`GaussianNoise`	T	Wang et al. [44]	Add Gaussian white noise to the signals.
`SmoothTimeMask`	T	Mohsenvand et al. [27]	Randomly pick a portion of the signal and set it to zero.
`SignFlip`	T	Rommel et al. [36]	Randomly flip the sign of all channels.
`TimeReverse`	T	Rommel et al. [36]	Randomly reverse the axis of time in all channels.
`ChannelsSymmetry`	S	Deiss et al. [12]	Randomly swap signals from right hemisphere to left hemisphere and *vice-versa*.
`ChannelsDropout`	S	Saeed et al. [40]	Randomly pick a given number of channels and set their signals to zero.
`ChannelsShuffle`	S	Saeed et al. [40]	Randomly pick a given number of channels and permute their signals.
`SensorsRotation`	S	Krell and Kim [22]	Interpolate channels signals on randomly rotated positions.

Table 1: Data augmentation methods studied in this work. Types stand for Frequency (F), Time (T) and Spatial (S) transformations.

classification tasks [7]. It is trained using the Adam optimizer [21] with a learning rate of 10^{-3}. The weighed cross-entropy loss is used to take into account the class-imbalance of the dataset. The batch size is set to 16 to preserve the stochasticity of the gradient descent in very low data regimes. We train the model for 300 epochs using early-stopping on the validation loss with a patience of 30 epochs [7].

2.1.2. BCI

Dataset and preprocessing Likewise, experiments are carried out with the *BCI IV 2a* dataset [5]. It consists of recordings from 9 subjects using 22 EEG electrodes. The subjects were asked to perform four motor imagery tasks, namely to imagine the movement of the left hand (class 1), right hand (class 2), both feet (class 3), and tongue (class 4). This dataset was preprocessed using a band-pass filter between 4 Hz and 38 Hz followed by an exponential moving standardization as in [41]. Then trials of 4.5 seconds are used as inputs. Each trial starts 0.5 seconds before the cue that tells the subject to perform the motor imagery task and ends when the cue disappears.

Splitting strategy According to the rules of the BCI competition [5], for each subject, our model is trained on the first session (or a fraction of it) and evaluated on the second session. The experiment is repeated across all

nine subjects.

Model and training details Experiments are performed using a generic deep convolutional network [41], as implemented in the library BRAINDECODE [41]. This architecture is inspired by the success of Common Spatial Patterns (CSP) methods [33]. It can take advantage of the spatio-temporal structure of the data using spatial filters and convolutions across time. Following the work of [41], the network's training uses the AdamW optimizer [26] with a learning rate of 6.25×10^{-4}, batch size 64, maximum of 1600 epochs and early stopping on the validation error (patience of 160 epochs).

2.2. Experiments

In this section, we describe the three types of experiments carried out with each EEG augmentation considered.

2.2.1. Parameters selection Several augmentations have a parameter that can be adjusted to control how strongly the inputs are transformed. For example, the Gaussian noise augmentation has a parameter σ corresponding to the standard deviation of the distribution from which the noise is sampled. The choice of the value of such a parameter is as important as the choice of the augmentation itself, as later depicted in our results (sections 4, 3 and 5).

This experiment unfolds in two steps: 1) narrowing down the range of parameter values and 2) carrying out a grid-search. For the first step, an upstream manual exploration allows to estimate an upper bound above which the augmentation distorts too much the relevant information contained in the signal. For instance, in the case of Gaussian noise, with σ values greater than 0.2, EEG signals become so noisy that the augmentation is systematically detrimental to the learning.

For the second step, a grid-search is carried out using 11 linearly spaced values within the aforementioned interval. Since data augmentation is all the more efficient in low data regimes (as shown in our experiments from Sections 4.2.2, 3.2.2 and 5.2.2), we carried our parameter selection using a small balanced fraction of the initial datasets (e.g. 2^{-7} for *SleepPhysionet* dataset) to make potential improvements more apparent. For each value in the grid, the accuracy metric is computed using a 10-fold cross-validation.

2.2.2. Learning curves The second experiment aims at comparing the benefits brought by different augmentation methods. To this end, for each augmentation operation, we compute a learning curve which shows the model's performance when it is trained on increasing fractions of the training set. Note that the same validation and test set are used for model selection and evaluation for all training set fractions, to ensure the learning curve points are comparable. The results obtained with each data augmentation are then compared to a baseline, consisting in the same model trained with no data augmentation.

2.2.3. Per class analysis Finally, to get a deeper understanding of the effects of data augmentations, we take a closer look at single points from the learning curve and analyze how effects vary depending on the classes. This observation is guided by the intuition that the invariances encoded by data augmentations might be more relevant for some classes than others. For example, the channel symmetry augmentation, which switches EEG channels from left and right hemispheres, is much more relevant for non-lateralized brain activities

such as imagining tongue movements, whereas it is likely detrimental for lateralized functions, such as right- or left-hand movements.

The first experiments on the *SleepPhysionet* dataset reveal that augmentations are systematically more helpful in low data regimes (*cf.* Section 4.2.2), and thus have a greater effect on underrepresented classes. Since we are seeking to assess the effects of transformations on learned representations for each class, these experiments require class proportions to be equalized before training. To do this, a subsampling step is added to the preprocessing pipeline, allowing to work with balanced data. We choose to report results here in terms of 10-fold cross-validated F1-scores, which is a natural metric choice in class-wise analysis.

2.3. Implementation

2.3.1. Data augmentation Data augmentation consists in randomly applying an output-preserving transform to training samples. More formally, for each input-output pair (x, y) sampled from the training set, there is a probability p_{aug} of transforming the input using some transformation T chosen *a priori*. T should in theory be such that $P(y|T(x))$ is similar to $P(y|x)$. We then feed the transformed input $T(x)$ to the model and train it to predict the original output y. This procedure is carried on-the-fly, for each example of every mini-batch. In all our experiments, we use a probability $p_{\text{aug}} = 0.5$ of augmenting each input and $1 - p_{\text{aug}} = 0.5$ of leaving them unchanged.

2.3.2. Reproducible code The implementation of all data augmentations has been added to the open-source package Braindecode [41]. The data are automatically fetched thanks to MOABB [20] for the *BCI IV 2a* dataset and MNE [17] for sleep physionet. Moreover, the code used in all our experiments has been made available in an open-source repository§.

3. Time domain augmentations

3.1. Rationale of time domain augmentations

Gaussian noise

The `GaussianNoise` augmentation, proposed for example in [44], consists in adding Gaussian white noise to the recorded EEG signals.

In practice, a perturbation $E(t) \sim \mathcal{N}(0, \sigma^2)$ is sampled independently for each channel and acquisition time, and is added to the original signal X:

$$\texttt{GaussianNoise}[X](t) = X(t) + E(t) \ .$$

Here σ denotes the standard-deviation of the noise distribution. This parameter is interpreted as this transformation's magnitude, as the larger it is the more the original signal is distorted.

The motivation behind this augmentation is make the model more robust to noise in EEG recordings, as they are known to suffer from limited signal-to-noise ratio (SNR). As EEG signal power decreases with the frequency, this augmentation mostly preserves power band ratios at lower frequencies, which are instrumental in many EEG decoding tasks, such as sleep stage classification [24, 7]. On the contrary, by adding the same amount of power to all frequencies, the addition of white noise hides the information contained in high frequency bands, as depicted in Figure 1.

§ https://github.com/eeg-augmentation-benchmark/eeg-augmentation-benchmark-2022

From this perspective, the effect of this transformation is somehow analogous to a low-pass filter where the parameter σ plays the role of a cut-off frequency.

Smooth time mask

The `SmoothTimeMask` augmentation, proposed in [27], consists in replacing by zeros a portion of length Δt of all channels, starting from a randomly sampled instant t_{cut}.

In practice, the computation is carried out by multiplying the signal X by a mask m_λ made out of two opposing sigmoid functions of temperature λ:

$$\texttt{SmoothTimeMask}[X](t) := X(t) \cdot m_\lambda(t),$$
$$m_\lambda(t) := \sigma_\lambda(t - t_{\text{cut}}) + \sigma_\lambda(t_{\text{cut}} + \Delta t - t)$$
$$\sigma_\lambda(t) := \frac{1}{1 + \exp(-\lambda t)} \ ,$$
$$t_{\text{cut}} \sim \mathcal{U}[t_{\min}, t_{\max} - \Delta t] \ .$$

This allows to set the signal smoothly to zero and avoid creating discontinuities, as shown in Figure 2. The magnitude of this transformation is controlled by the length of masked signal Δt.

The motivation of this augmentation is to teach the deep network to be less driven by isolated transient events. Indeed, as described in the *AASM* scoring manual [4], sleep stages are most often characterized by the global information contained in a time window. For example, the sleep stage N1 is scored when more than 15 seconds ($\geq 50\%$) of a time window is dominated by theta activity ($4 - 7\,\text{Hz}$). The representations learned by the model should thus encapsulate the global information of the signal and avoid to rely on transient patterns. Consequently, one would expect two almost identical EEG windows differing only for a few seconds to be very close in the representation space [10]. By masking part of the signal, we assign the same label (*e.g.,* sleep stage or action) to windows differing in the time domain only inside a short time span.

Time reverse

The `TimeReverse` augmentation (also noted `TRev`) was proposed in [36] and consists in randomly flipping the time axis in all channels.

More practically, we implement this augmentation by reversing the time indexing of the input signals X with probability p_{aug}:

$$\texttt{TRev}[X](t) := \begin{cases} X(t_{\max} - t) & \text{with } p_{\text{aug}}, \\ X(t) & \text{with } 1 - p_{\text{aug}}, \end{cases}$$

where $t_{\max}$ is the length of a time window.

The motivation for this data augmentation method is that the frequency power ratios contain a substantial part of the information useful for many EEG classification tasks. In sleep stage classification for example, some stages, such as N1, are scored based on the dominant rhythms observed (theta waves). Considering that the orientation of the time axis has no effect on the signal's power spectral densities (PSD), it can be hypothesized that flipping the time axis generates a new input while preserving most of the information. Moreover, a large part of the time-domain information is also preserved by this transformation, since symmetric waveforms are merely shifted along the time axis and only asymmetric patterns are modified, as illustrated in Figure 3. This should be a useful property for instance to score the sleep stage N2, which is characterized by more than two occurrences of K-complexes throughout the stage [4].

Sign flip

The `SignFlip` augmentation, introduced

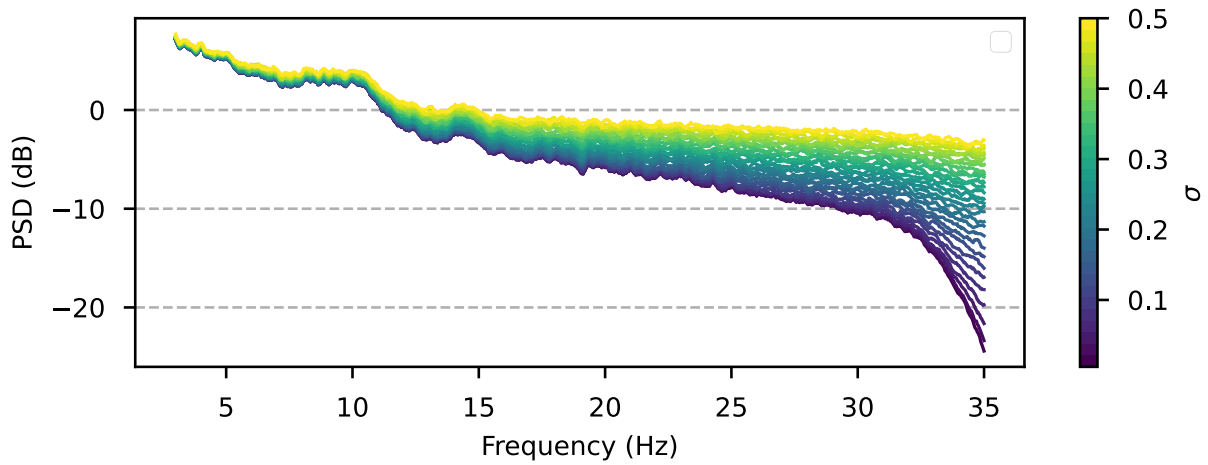


Figure 1: Effects of the addition of `GaussianNoise` on the PSD. Power spectra were averaged over N1 windows for one night of sleep from the *SleepPhysionet* dataset. In polysomnograms, the power globally decreases as the frequency increases. Consequently, the `GaussianNoise`, which adds a constant amount of power across all frequencies, has a greater relative impact on higher frequencies. As we increase σ, a greater portion of the signal is hidden by the added noise.

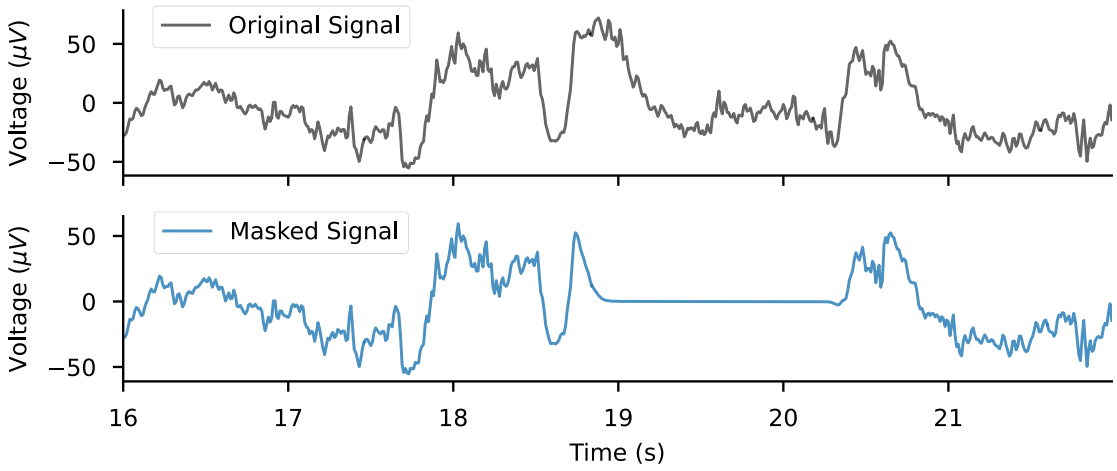


Figure 2: Effect of `SmoothTimeMask` on a time window from the *SleepPhysionet* dataset. The mask length is 1.6*sec* and the transition between unchanged parts and the masked portion is smooth.

in [36], consists in randomly inverting the sign of all EEG channels.

In practice, each new input signal X is multiplied by -1 with probability p_{aug} (*cf.* Section 2.1):

$$\texttt{SignFlip}[X](t) := \begin{cases} -X(t) & \text{with } p_{\text{aug}}, \\ X(t) & \text{with } 1 - p_{\text{aug}}. \end{cases}$$

The motivation behind this augmentation

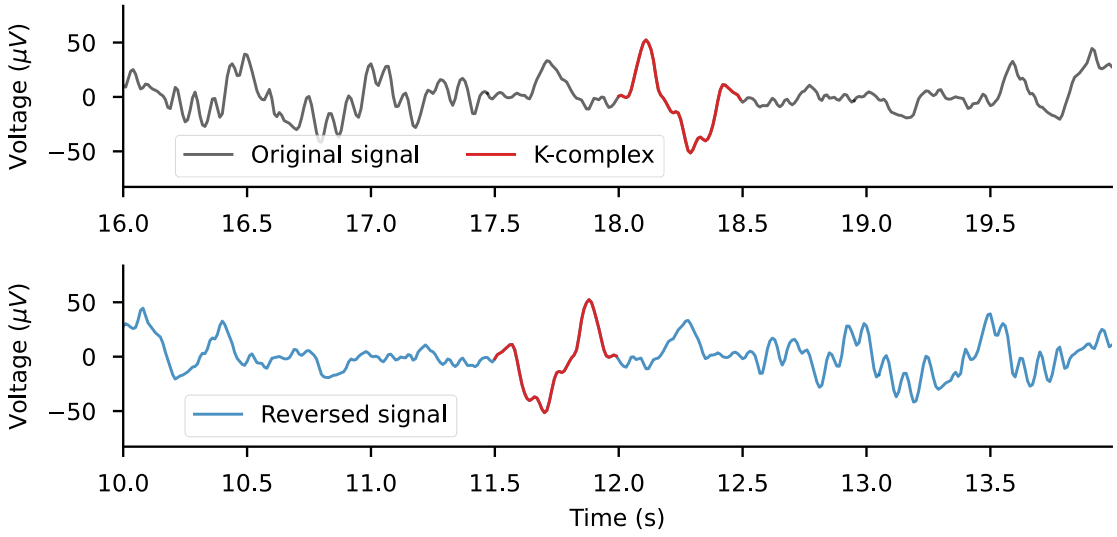


Figure 3: `TimeReverse` augmentation on an EEG signal from the *SleepPhysionet* dataset. A large part of the signal is not deeply affected. Wave patterns are merely translated along the time axis. Some specific asymmetric EEG patterns such as K-complexes are reversed by this transformation.

is that it preserves the topographical properties of the electric field potential in terms of location and intensity, while changing its polarity. Indeed, the electric potentials measured with EEG are driven by post-synaptic potentials along dendrites of pyramidal neurons. Depending on the geometric alignment of active neurons, the current produced can add up to be measured non-invasively with EEG. The group of neurons can be well modeled as electric current dipole, characterized by a moment vector at a given location in the brain. For most analysis, the moment strength (norm) and location (origin) are sufficient. While the moment direction is often unused, it is responsible for the sign of the potential measured by the EEG device. Our guess is that changing the sign of EEG channels will preserve the instrumental information contained in the strength and location of the dipole. In terms of physiology, it corresponds to current flowing from superficial cortical layers to deep layers or vice versa.

3.2. Empirical comparison of time domain augmentations

3.2.1. Parameters selection Here we investigate the effects of the transformations' magnitude on the classification performance. As shown in Table 2, the magnitude of `GaussianNoise` is controlled by its standard deviation σ, while `SmoothTimeMask` is governed by the length of the mask Δt. Note that there is no notion of strength or magnitude for `TimeReverse` and `SignFlip`, which are hence not studied in this subsection.

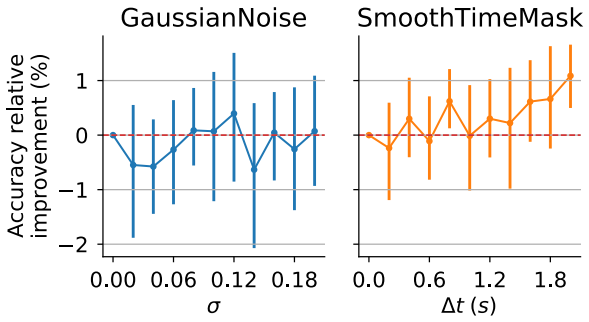
SleepPhysionet As illustrated in Figure 4, the impact of the magnitude is quite different between `SmoothTimeMask` and `GaussianNoise`. While increasing the magnitude of `SmoothTimeMask` seems beneficial, no clear trend is observed for `GaussianNoise`. For `SmoothTimeMask`, we re-

stricted our experiment to masks of less than two seconds to avoid removing too much crucial information from the signal. It seems that the augmentation is more efficient with masks of maximum length.

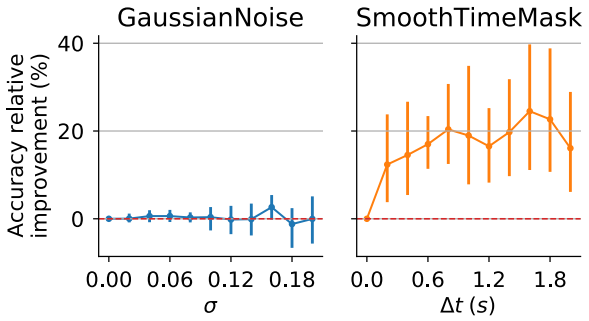
BCI IV 2a The grid search on the *BCI IV 2a* dataset presented in Figure 4b shares similarities with its counterpart on the *SleepPhysionet* dataset (Figure 4a). In both cases, increasing the mask length enhances the performance of `SmoothTimeMask`, whereas `GaussianNoise` does not seem to lead to any robust improvement. `SmoothTimeMask` appears to be much more useful on the BCI task though, since it produces relative improvements of up to 20% when only 60 windows per class are used for the training.

3.2.2. Learning curves

SleepPhysionet As expected from the previous experiment, Figure 5a confirms that `GaussianNoise` and `SmoothTimeMask` have a negligible effect on the sleep staging task and perform on par with the baseline model. The most significant improvements are brought by `SignFlip` and `TimeReverse` with the latter outperforming the former. This suggests that symmetries are notably relevant invariances for the sleep stage classification task. They preserve both the frequencies and some of the transient patterns occurring during polysomnographies, such as sleep spindles, which presents a symmetry both along the x and y axis. This observation also supports the claim that sleep scoring heavily relies on frequency domain features since the two augmentations that leave the PSD of the signal unchanged outperform the others.



(a) *SleepPhysionet*



(b) *BCI IV 2a*

Figure 4: Time augmentations parameters selection on the *SleepPhysionet* (a) and *BCI IV 2a* (b) datasets. Models were trained on respectively 350 and 60 windows using augmentations parametrized with 10 different linearly spaced values. Validation accuracies are reported relatively to a model trained without data augmentation. The error bars correspond to the 95% confidence intervals based on a 10-fold cross-validation.

BCI IV 2a Figure 5b contains the learning curve plots for the *BCI IV 2a* dataset. It suggests that `TimeReverse` and `SmoothTimeMask` are the most suited time domain augmentations for the BCI task in low and high data regimes respectively. An intuitive interpretation of this result is that, in motor imagery, subjects are asked to mentally simulate the same physical action during the whole trial and the information encoded in brain signals

Augmentation	Parameter	Interval	Unit	Best value (sleep staging)	Best value (BCI)
`GaussianNoise`	σ	$[0, 0.2]$	-	0.12	0.16
`SmoothTimeMask`	Δt	$[0, 2]$	s	2 s	1.6 s

Table 2: Adjustable parameter for each time domain augmentation.

is hence invariant to local time distortions. Another striking observation has to do with the learning curve of `SignFlip`, which is consistently equivalent to the baseline's while it is the second most efficient augmentation in *SleepPhysionet*. This happens because, unlike in the sleep staging experiments, the network architecture used here includes a layer which squares the activations, thus already encoding the sign invariance.

3.2.3. Per class analysis

SleepPhysionet Figure 6a introduces the results per class of the time domain augmentations for the sleep staging task. The REM stage benefits more than the others from this class of transformations. This stage is characterized by low amplitude mixed frequency and bursts of eye activity. The scoring thus relies heavily on global signal amplitudes which are mostly preserved with time domain augmentations (except for `GaussianNoise` which performs the worst among them). `TimeReverse` especially yields the best results for this class as well as for all others. Note also that while `SmoothTimeMask` leads to improvements for REM and W stages, it appears to be detrimental for learning to recognize non-REM sleep. A possible interpretation of this observation is that this transformation may erase important waves, such as K-complexes and spindles, which strongly characterize stages N2 and N3.

BCI IV 2a Figure 6b introduces the results per class of the time domain augmentations for the BCI motor imagery task. Here again, we observe that `SignFlip` is equivalent to the baseline with no augmentation, as explained in Section 3.2.2. While `GaussianNoise` also does not help for any imagined movement, `TimeReverse` and `SmoothTimeMask` greatly improve the predictive accuracy for most classes, specially left-hand, right-hand and foot.

3.3. Conclusion of time augmentations experiments

The `TimeReverse` augmentation consistently yielded the largest performance improvements with small training sets relative to the baseline trained without data augmentation: up to +13% for sleep stage classification and +25% for motor imagery experiments‖. With larger training sets, `SmoothTimeMask` seems like the best data augmentation for motor imagery, while it appears to be detrimental in sleep staging. The same can be observed for `GaussianNoise`, although it does not bring the same level of improvements as `SmoothTimeMask` in our BCI experiments.

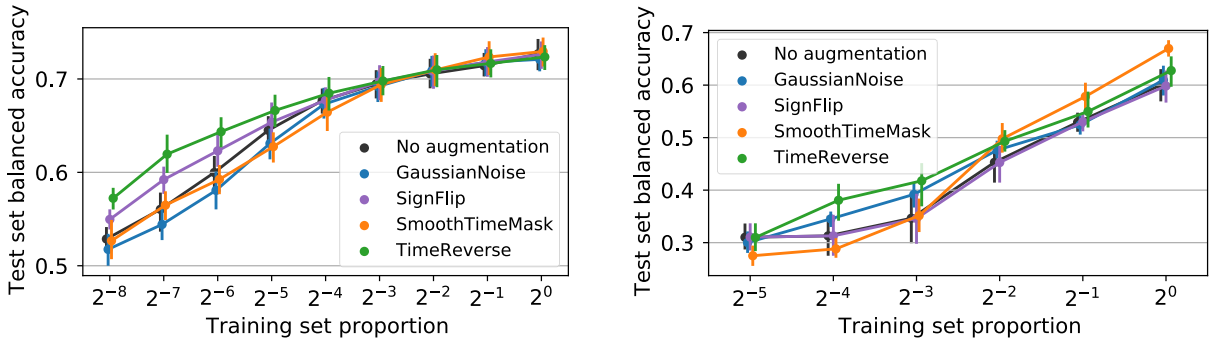
4. Frequency domain augmentations

4.1. Rationale of the frequency augmentations

Frequency shift

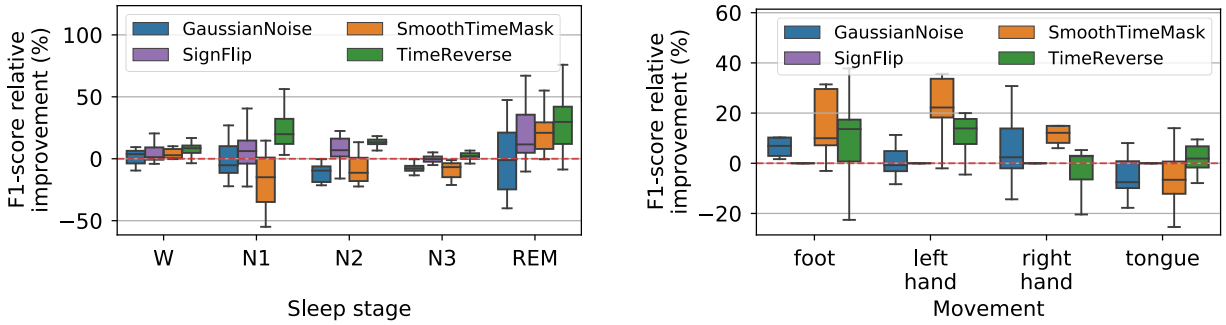
The `FrequencyShift` augmentation, proposed in [36], consists in shifting the spec-

‖ At training set fractions of 2^{-7} and 2^{-4} respectively.



(a) *SleepPhysionet* (b) *BCI IV 2a*

Figure 5: Learning curves for time domain augmentations along with the baseline trained with no augmentation. For each transformation, the same model is trained on fractions of the dataset of increasing size. After each training, the average balanced accuracy score on the test set is reported with error bars representing the 95% confidence intervals estimated from 10-fold cross-validation.



(a) *SleepPhysionet* (b) *BCI IV 2a*

Figure 6: Per-class F1-score for time domain transformations. Scores are reported as relative improvement over a baseline trained without data augmentation. Models were trained on 180 and 230 time windows for *SleepPhysionet* and *BCI IV 2a* datasets respectively. Boxplots were estimated using 10-fold cross-validation.

trum of the signals of all EEG channels by a random frequency Δf.

In practice the shift is performed on the complex analytic signal associated to the EEG signal X, $X_a = X + j\mathcal{H}(X)$, where $\mathcal{H}$ denotes the Hilbert transform:

$$\texttt{FrequencyShift}[X](t) := \mathrm{Re}(X_a(t) \cdot e^{2i\pi\Delta f \cdot t})$$

Here one takes the real part to recover the shifted signal. The shift value Δf is randomly sampled with uniform probability in an interval $[-\Delta f_{\max}, +\Delta f_{\max}]$ each time an EEG window is augmented. The parameter $\Delta f_{\max}$ is used to set the magnitude of this transformation.

The motivation for this data augmentation method is that a substantial part of the information useful for many EEG classification tasks lies in the frequency domain of the signal. In sleep scoring for example, most sleep stages

are characterized by the occurrence of specific brain rhythms in a given frequency range. The stage N2 can be characterized by so-called sleep spindles with frequencies located between 12 and 15 Hz, while stage N3 is characterized by slow waves in the delta band [4]. The predominance of certain brain rhythms is well captured by the power spectral density (PSD) of the EEG signals, which have peaks at specific frequency ranges, as depicted in Figure 7. Due to strong inter-subject variability, the location of these peaks for a given sleep stage can be slightly different from one subject to another, and the `FrequencyShift` augmentation aims to mimic this variability.

Fourier transform surrogate

The Fourier Transform surrogate augmentation, proposed in [42] and denoted `FTSurrogate` hereafter, consists in randomizing the phase of the Fourier coefficients of EEG signals.

In practice, this transformation is performed by computing the Fourier coefficients of all EEG channels and adding random noise to their phase:

$$\mathcal{F}[\texttt{FTSurrogate}(X)](f) = \mathcal{F}[X](f)e^{i\Delta\varphi} \ ,$$

where $\mathcal{F}$ is the Fourier transform operator, f is a frequency and $\Delta\varphi$ is a frequency-specific random phase perturbation. The augmented signals are obtained with the inverse Fourier transform. In our implementation, the values of $\Delta\varphi$ for each frequency f are uniformly sampled in an interval $[0, \Delta\varphi_{\max}]$, where $\Delta\varphi_{\max} \in [0, 2\pi)$ controls the magnitude of the transformation. This transformation can be independently applied to each channel, or by enforcing that the phase shift is common for all channels. While EEG channels are perturbed independently in our sleep classification experiments, we found that perturbing all channels equally is crucial in BCI experiments to preserve cross-channel correlations and obtain good results.

The motivation behind this augmentation is that it preserves the frequency-bands power ratios, yet it leads to a global change of the signal's representation in the time-domain. This is illustrated on Figure 8, which shows that characteristic patterns in the time-domain such as K-complexes are erased by this operation. As a result, it decreases the model's reliance on the signal's waveform, encouraging the model to leverage the PSD information.

The Fourier transform surrogate method is based on the assumption that EEG signals are well described by stationary linear stochastic processes [42]. As such, they must be uniquely characterized by the amplitudes of their Fourier coefficients and must have random phases in $[0, 2\pi)$. This implies that each frequency in the signal is considered independent, which might appear as a strong assumption. Indeed, neural signals contain transient events, such as K-complexes, which spread across multiple frequencies. This creates some dependence among the phases of multiple Fourier coefficients (cf. Figure 8). Besides transient events, phenomena known as cross-frequency coupling (CFC) have been reported in human electrophysiology signals [6, 13]. Nevertheless, the hypothesis of stationary linear stochastic processes have been shown to be compatible with EEG signals in [1]. Plausible arguments in favor of this hypothesis are the huge number of neurons included in an EEG recording, the complicated structure of the brain and the possible blur of dynamical structures due to the different conductivities of the skull and other intermediate tissues. Thus, while the Fourier transform surrogate method might be useful on the context of EEG, it is likely to be less relevant for intracranial

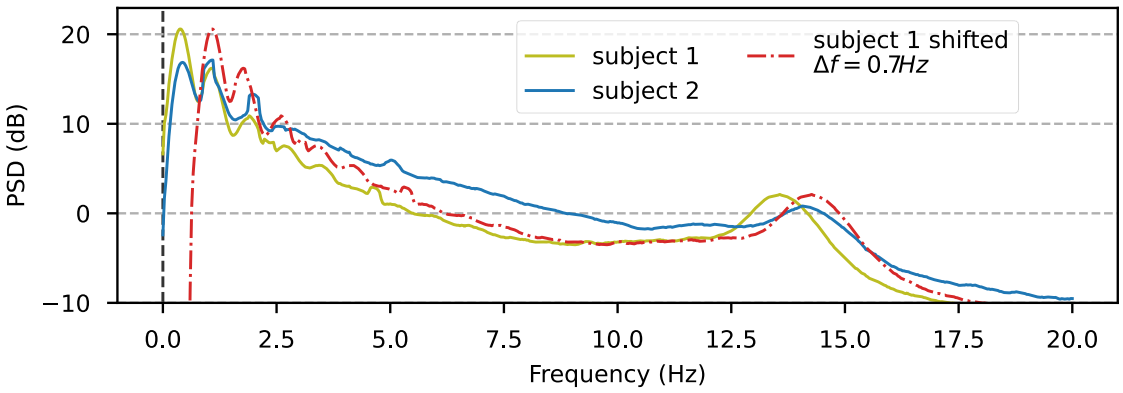


Figure 7: Averaged power spectral density of windows corresponding to the sleep stage N2, for two different subjects from the *SleepPhysionet* dataset. The red dash-dot curve corresponds to the recording of subject 1 transformed using the `FrequencyShift` augmentation. It allows to translate the PSD peak close to subject 2.

recordings, which have a better signal-to-noise ratio.

Band-stop filter

The `BandstopFilter` augmentation, proposed for example in [10, 27], consists in filtering out from all EEG channels a given frequency band selected at random.

Our implementation of this augmentation uses the finite impulse response notch filter from *MNE-Python* [18]. For each augmented EEG signal, the center of the frequency band is randomly picked from a uniform distribution between 0 and 38 Hz, which is the approximate low-pass filtering frequency used to preprocess the datasets. The width of the filter allows to control the magnitude of this transform.

The motivation of this augmentation is to prevent machine learning models from overfitting on subject specific features and from relying too much on a few narrow frequency regions. Indeed, some patterns in EEG signals, such as the K-complex, have frequency signatures that spread over multiple frequency bands. By filtering randomly selected regions of the PSD, one can promote models that use the full spectral information of these patterns. In a sense, this transformation can be compared to the commonly used dropout layer [43], which prevents artificial neural networks from relying too much on certain neurons.

4.2. Empirical comparison of frequency augmentations

4.2.1. Parameters selection The parameters to be set for frequency domain augmentations are listed in Table 3.

SleepPhysionet The results on the sleep staging task presented in Figure 9 and Table 3 reveal that the optimal magnitude varies significantly from one transformation to the other. The `FTSurrogate` augmentation benefits from the larger possible range of $\Delta\varphi$ values, corresponding to a nearly completely random phase selection within the interval $[0, 2\pi)$. On the contrary, `FrequencyShift` works better for small $\Delta f_{\max}$ values. An interpretation for this result could be that small frequency shifts

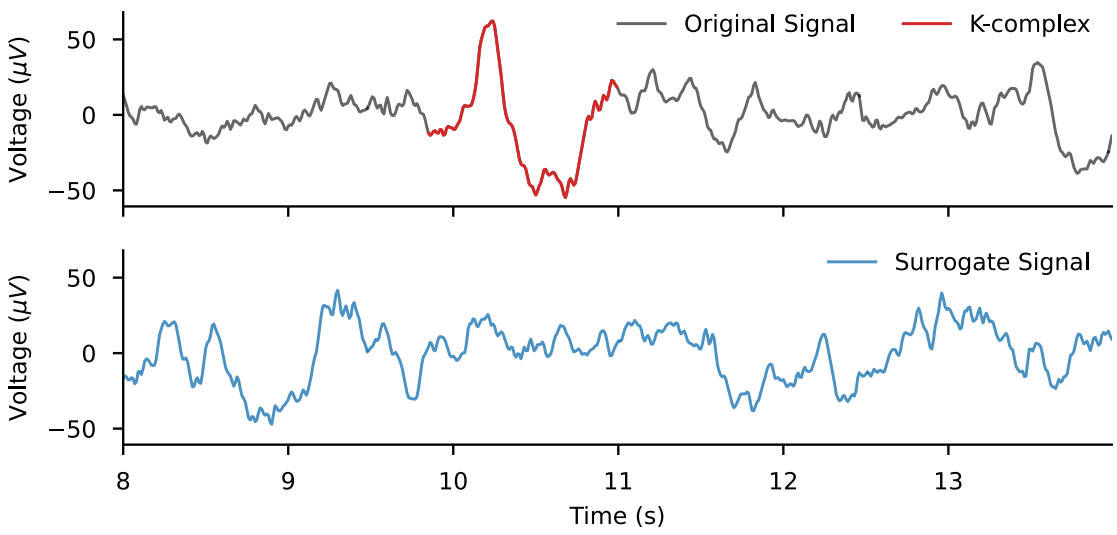


Figure 8: Effect of `FTSurrogate` on transient patterns. The original extract from a window scored as N2 presents a highly localized K-complex, whereas the surrogate does not.

allow to capture the inter-subject variability whereas larger values mix-up the frequency bands characterizing different classes. Finally, `BandstopFilter` shows marginally better results for a bandwidth of 1.2 Hz, although performance gains compared to the baseline are not significant.

BCI IV 2a The results for *BCI IV 2a* presented in Figure 9b reveal several differences compared to *SleepPhysionet*. Unlike for the sleep stage classification task, smaller bandwidths seem to work better for `BandstopFilter`. However, this transformation does not lead to statistically significant improvements here again. Maximum frequency shifts $\Delta f_{\max}$ privileged for the BCI task are also higher than for sleep staging. Finally, the `FTSurrogate` augmentation shows a pattern similar to the sleep staging task, as it benefits from higher $\Delta\varphi_{\max}$ values.

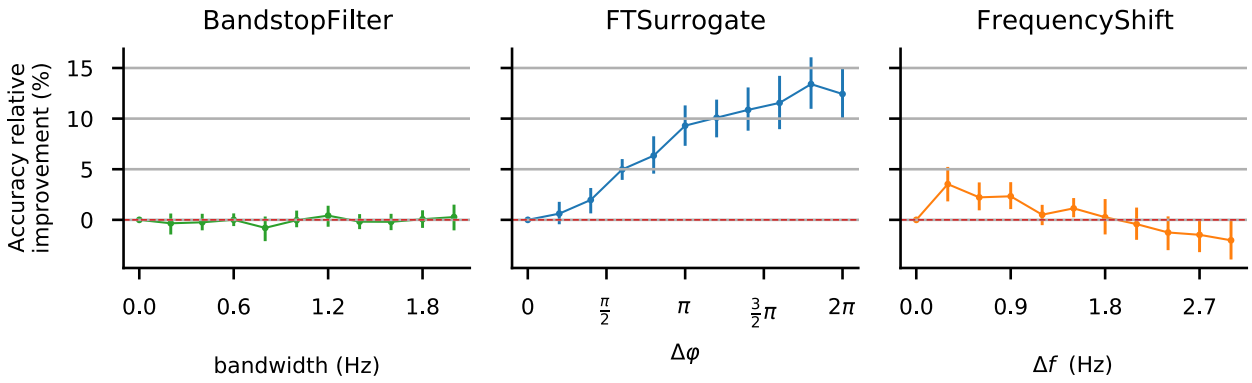
4.2.2. Learning curves

SleepPhysionet As expected, the first learning curve experiment depicted in Figure 10a reveals that data augmentation methods are more helpful in low data regimes. It helps to mitigate the lack of data by artificially increasing the training set size. For example, a model trained with `FTSurrogate` on a small fraction of training data (2^{-4}) achieves performances comparable to a model trained without augmentation on four times as many data points (2^{-2}). Moreover, this augmentation appears as the top performer in this task, yielding balanced accuracy relative gains of up to 12% in low data regimes compared to the baseline. While `FrequencyShift` also brings some smaller performance improvements, `BandstopFilter` is ineffective on this dataset. This is evidence that preserving the frequencies' power ratios is important in sleep stage classification.

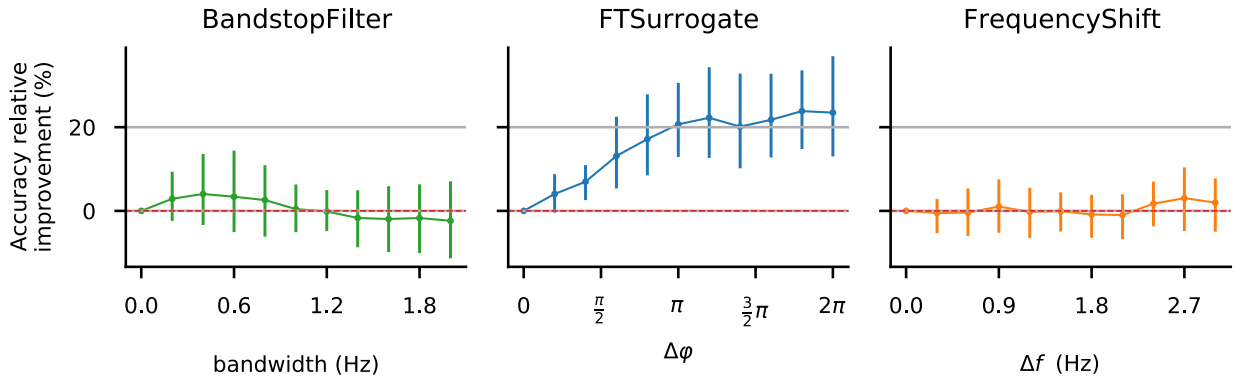
BCI IV 2a As in the sleep stage classification task, `FTSurrogate` outperforms other frequency domain augmentations on the *BCI IV*

Augmentation	Parameter	Interval	Unit	Best value (sleep staging)	Best value (BCI)
`BandstopFilter`	bandwidth	$[0, 2]$	Hz	1.2 Hz	0.4 Hz
`FTSurrogate`	$\Delta\varphi_{\max}$	$[0, 2\pi)$	rad	$\frac{9}{10}\pi$	$\frac{9}{10}\pi$
`FrequencyShift`	$\Delta f_{\max}$	$[0, 3]$	Hz	0.3 Hz	2.7 Hz

Table 3: Adjustable parameter of each frequency domain augmentation.



(a) *SleepPhysionet*



(b) *BCI IV 2a*

Figure 9: Frequency augmentations parameters selection on the *SleepPhysionet* (a) and *BCI IV 2a* (b) datasets. Models were trained on respectively 350 and 60 windows using augmentations parametrized with 10 different linearly spaced values. Validation accuracies are reported relatively to a model trained without data augmentation. The error bars correspond to the 95% confidence intervals based on a 10-fold cross-validation.

2a dataset, as shown in Figure 10b. Namely, a model trained with `FTSurrogate` on half of the training set reaches a test accuracy higher than the same model trained without data augmentation on the whole training set. It also leads to a 45% relative improvement in accuracy compared to the model trained without augmentation on a training subset 2^{-3} times smaller than the original one. Moreover, this augmentation seems to yield significant improvements over all training set sizes considered in this experiments, unlike in sleep

stage classification where it mostly helped with smaller training sets.

Another similarity with the sleep stage classification task are the good results of `FrequencyShift`, whose performance improvements even exceed those observed on the *Sleep-Physionet* dataset. This is surprising given that this transformation was originally designed to simulate the inter-subject variability observed in sleep stages. Despite these similarities, a major difference between sleep stage classification and BCI is that `BandstopFilter` appears as a relevant data augmentation technique for the latter, since it is shown to improve accuracy by up to 17% in low data regimes. Overall, these results suggest that a critical part of the information lie in the frequency domain for both classification tasks.

4.2.3. Per class analysis Figure 11a shows the F1-score improvements per class when using each frequency domain data augmentation in the sleep staging task. First, it can be seen that all frequency data augmentation methods only produce marginal improvements for sleep stages W and N3. These results can be interpreted in light of the performance of the baseline model presented in Figure 12a. Indeed, the highest F1-scores are reached for these classes and it is probably difficult to improve over the representations extracted by the baseline. On the contrary, REM stage appears as benefiting the most from the frequency domain augmentations. This cannot be completely explained by the low baseline accuracy for this stage, since a similar baseline performance is obtained for the N2 stage, which benefits less from data augmentation.

BCI IV 2a Per class results for the BCI task are presented in Figure 11b. Frequency augmentations seem to bring the smallest improvements for right hand movements, which might be due to the high performance reached by the baseline for this class (*cf.* Figure 12b). `FTSurrogate` consistently leads to larger performance improvements than other augmentations for 3 out of 4 classes.

4.3. Conclusion of frequency augmentations experiments

In both sleep stage classification and motor imagery experiments, the `FTSurrogate` augmentation seems to consistently lead to the most significant performance boost. For instance, this augmentation reached 45% relative improvement compared to the baseline when training on a small portion of the BCI dataset. Although performance boosts are not as impressive in our sleep stage classification experiments, we demonstrate that using `FTSurrogate` can sometimes be equivalent to training on a dataset four times larger. Using the maximum magnitude for this augmentation, as proposed in the original publication [42], seems to yield the best results in both datasets. Surprisingly, `FrequencyShift` seems to be useful not only for sleep stage classification, but also for BCI, with improvements sometimes comparable to `FTSurrogate`. The `BandstopFiltering` augmentation, however, seems beneficial only for BCI applications and not for sleep staging.

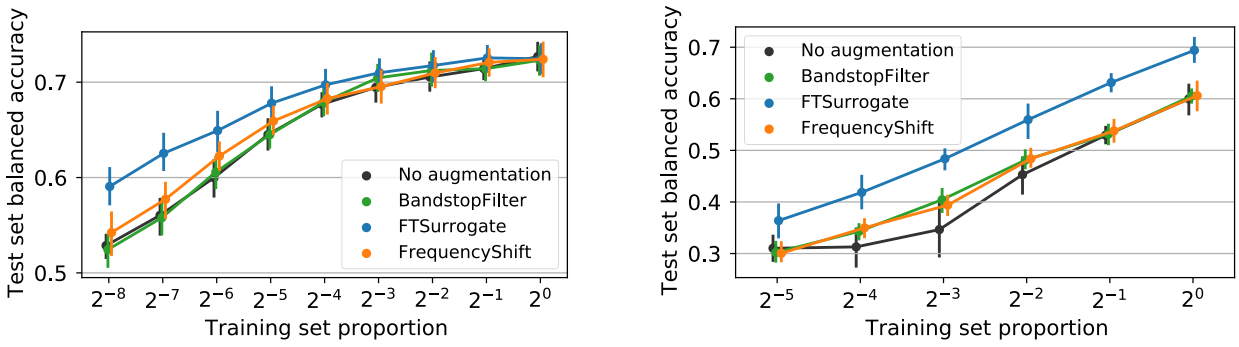
5. Spatial domain augmentations

In this section we study augmentations exploiting the sensors spatial positions.

5.1. Rationale of spatial augmentations

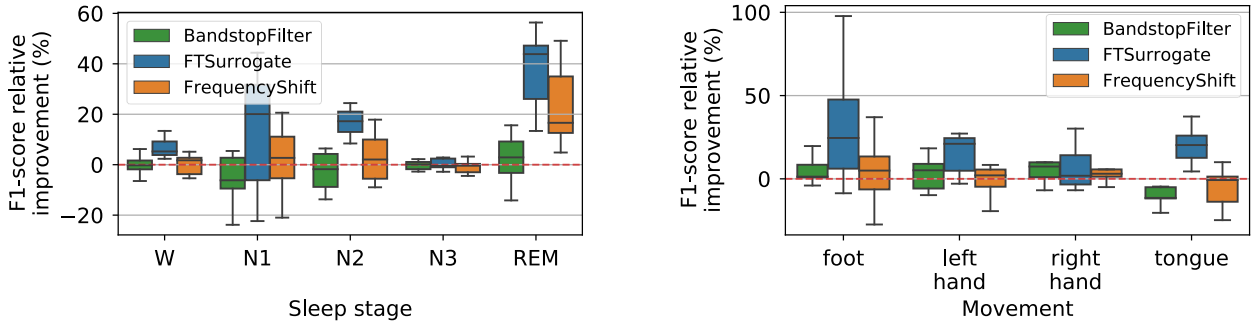
Channels symmetry

The `ChannelsSymmetry` augmentation,



(a) *SleepPhysionet* (b) *BCI IV 2a*

Figure 10: Learning curves for frequency domain augmentations along with the baseline trained with no augmentation. For each transformation, the same model is trained on 8 fractions of the dataset of increasing size. After each training, the average balanced accuracy score on the test set is reported with error bars representing the 95% confidence intervals estimated from 10-fold cross-validation.



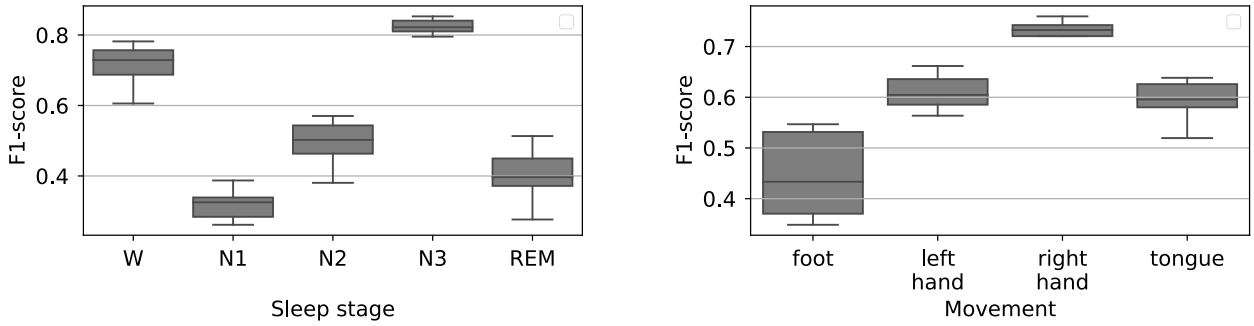
(a) *SleepPhysionet* (b) *BCI IV 2a*

Figure 11: Per-class F1-score for frequency domain transformations. Scores are reported as relative improvement over a baseline trained without data augmentation. Models were trained on 180 and 230 time windows for *SleepPhysionet* and *BCI IV 2a* datasets respectively. Boxplots were estimated using 10-fold cross-validation.

proposed in [12], simulates a swap between EEG sensors placed on the right and left hemispheres.

More practically, this is obtained by permuting the rows of the input signal X in a particular way. Each row or channel of X corresponds to a known sensor position, *e.g.* $[C3, C4, F3, F4, O1, O2]$ in a standard 10-20 system. In this setting, odd indices correspond to the left hemisphere and even indices to the right hemisphere. Hence, we can obtain the augmented signal by swapping rows corresponding to the same electrodes in left and right sides: $C3 \leftrightarrow C4$, $F3 \leftrightarrow F4$, $O1 \leftrightarrow O2$.

This augmentation is motivated by the observation that several brain activities monitored using EEG involve a sagittal plane symmetry (left-right). For example, it has been evidenced that tongue movements stem from



(a) *SleepPhysionet* (b) *BCI IV 2a*

Figure 12: Per-class F1-score for a baseline model trained without data augmentation. Boxplots were estimated using 10-fold cross-validation.

an activation of the primary and supplementary sensorimotor areas without any significant lateralization [45]. Yet, motor imagery tasks involving hand movements are strongly lateralized and dominated by contralateral activations [29], suggesting that applying the `ChannelsSymmetry` transformation in such a context would degrade model performance.

Channels dropout

The `ChannelsDropout` augmentation, initially proposed in [40], randomly sets some channels of the EEG recording to zero with a given probability p_{drop}.

More precisely, for $X \in \mathbb{R}^{C \times T}$ an EEG window of T samples collected on C channels, the augmented signal takes the form

$$\texttt{ChannelsDropout}[X]_c := d_c \cdot X_c,$$

where d_c's are sampled from a Bernoulli distribution $\mathcal{B}$ of probability p_{drop}, and X_c denotes the c^{th} row of X corresponding to channel $c \in \{1, \ldots, C\}$.

As the widely used dropout layer [43], the motivation of this augmentation is to prevent the model from relying too heavily on a given input channel, which could lead to overfitting and poor generalization on different datasets. The transformation naturally improves the robustness of the model to corrupted channels, which is a major hurdle for the analysis of EEG signals. For example, in polysomnography, changes in the subject's position during sleep might result in a loss of contact between several electrodes and the scalp. Beyond sleep applications, the spread of mobile wearable EEG devices raises new challenges, as they are more prone to noise and missing channels [3]. Finally, the EEG and machine learning communities consider with great interest the question of transferability across datasets, which raises major challenges regarding inconsistent numbers of channels or channels ordering [35]. This augmentation hence drives the model to learn from the global information available from all channels instead of relying on a single one.

Channels shuffle

The `ChannelsShuffle` augmentation, also proposed in [40], consists in randomly permuting the rows of EEG input matrices.

Given an input signal $X \in \mathbb{R}^{T \times C}$, the augmented signal is defined as

$$\texttt{ChannelsShuffle}[X]_c := X_{\tau(c)},$$

where τ is uniformly sampled from all the possible permutations of a random subset of channels I. Although all channels are shuffled in the original formulation proposed in [40] ($I = \{1, \dots, C\}$), we implement this augmentation with a variable subset of permuted channels:

$$I = \{c | s_c = 1, s_c \sim \mathcal{B}(p_{\text{shuffle}})\},$$

where $\mathcal{B}$ is a Bernoulli distribution and p_{shuffle} is the probability of adding each channel to the permutation set I. The parameter p_{shuffle} hence sets the magnitude of this transformation as it defines how strongly the input signals will be distorted. The original formulation from [40] can be obtained by choosing $p_{\text{shuffle}} = 1$.

In addition to helping the transfer to datasets with different channels ordering, this augmentation induces invariance to the absolute and relative positioning of EEG sensors, since it prevents the decision function from relying on it. Such a transformation can make sense for several EEG classification tasks for which the precise localization of the cerebral activity is not strongly predictive, such as sleep staging. Indeed, sleep experts hardly consider the sensors position of the channel they observe (*e.g.,* Fpz-Oz) but rather rely on the waveforms and spectral characteristics (*e.g.,* theta activity, K-complex). If this augmentation seems well suited for the aforementioned task, it is expected to be less efficient in a context where sensors positions and source localization features have a greater impact, such as BCI.

Sensors rotations

The `SensorsRotation` augmentation, proposed in [22], approximates what would have been recorded with a device slightly rotated by a random angle along a given axis (x, y or z).

To achieve this, a random angle θ is drawn for each new input X within a chosen range $[-\theta_{\text{rot}}, \theta_{\text{rot}}]$. Then, the sensors 3D coordinates in a standard 10-20 montage are rotated by θ along the desired axis. Finally, the electrical potentials in X are interpolated from the original sensor positions to the new rotated ones. While in [22] a radial-basis functions interpolator is used, we implemented this transformation using spherical splines from the MNE-Python library [18], as commonly done for bad EEG channels preprocessing [30]. Following the MNE software head coordinate convention, the X axis goes from the left to the right ear, the Y axis goes from the back of the head to the nose, while the Z axis goes upwards. The magnitude of this transformation is set through the maximum rotation angle θ_{rot}.

The main idea of this augmentation is to promote robustness to perturbations of the EEG sensors positions. It is motivated by the fact that, between different recording sessions, the EEG cap can move over the subjects head, resulting in slightly shifted sensors locations. Compared to `ChannelsShuffle`, which encourages *global* invariance to electrodes positions, `SensorsRotation` induces robustness to small and local variations.

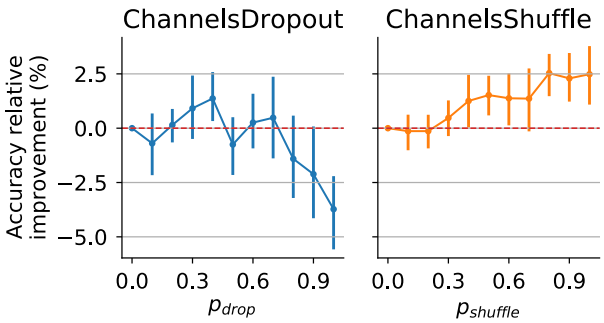
5.2. Empirical comparison of spatial augmentations

5.2.1. Parameters selection The parameters listed in Table 4 control the magnitude of the transformations, namely: the probability to drop channels p_{drop}, the probability to shuffle channels p_{shuffle} and the angle of rotation θ_{rot} respectively for `ChannelsDropout`, `ChannelsShuffle` and `SensorsRotations`. Regarding the sensors rotations, we restricted the range of possible angles θ_{rot} to $[0, 30]$ de-

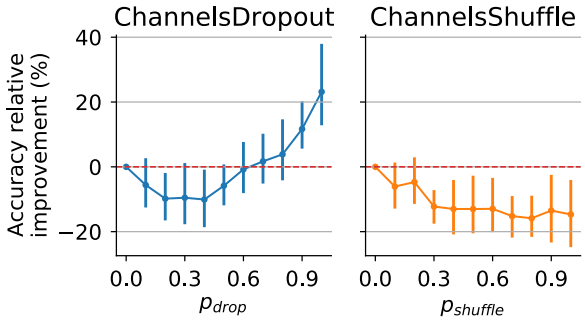
grees as done in [22]. The magnitude of the `ChannelsSymmetry` augmentation cannot be adjusted.

SleepPhysionet The results of the grid search for the parameters of spatial augmentations are presented in Figures 13a and 14a. We can see that `SensorsRotations` consistently lead to poor performances. This can be explained by the scarce number of EEG sensors available in this dataset, resulting in imprecise interpolation. This same reason might also explain why high probabilities of dropping channels in `ChannelsDropout` appear to be detrimental to learning, the best results being obtained with $p_{\text{drop}} = 0.4$. Indeed, as there are only two channels in this dataset, a probability of $p_{\text{drop}} = 0.7$ would erase all channels for 1 out of 4 windows on average (the probability to augment $p_{\text{aug}} = 0.5$, multiplied by p_{drop} squared). On the contrary, for `ChannelsShuffle`, higher probability to shuffle channels yields the best results. This was to be expected, since sleep stage information is not very spatially localized.

BCI IV 2a As shown in Figure 13b, the magnitude of the spatial domain augmentations impacts the performance on the motor imagery task in a very different way than on the sleep staging task. Indeed, the patterns for `ChannelsDropout` and `ChannelsShuffle` are reversed here: larger values of p_{drop} yield better performances, while `ChannelsShuffle` is consistently harmful, with stronger shuffling probabilities yielding the worst performances. The order of magnitude of the impact of these augmentations is also different compared to the sleep staging case, with up to 20% improvement for `ChannelsDropout`. Moreover, it is surprising to obtain the best results with a probability $p_{\text{drop}} = 1$, given that it corresponds to all channels being dropped for 1 out of 2 windows. This might indicate that our model is overfitting the data, since gradients computed from fully dropped examples only update the biases of the model. Regarding the rotational augmentations, parameter selection results are depicted in Figure 14b. Due to very large error bars and mean values close to zero, practical usefulness of random channel rotations is hard to assess here.



(a) *SleepPhysionet*



(b) *BCI IV 2a*

Figure 13: Spatial augmentations parameters selection on the *SleepPhysionet* (a) and *BCI IV 2a* (b) datasets. Models were trained on respectively 350 and 60 windows using augmentations parametrized with 10 different linearly spaced values. Validation accuracies are reported relatively to a model trained without data augmentation. The error bars correspond to the 95% confidence intervals based on a 10-fold cross-validation.

Augmentation	Parameter	Interval	Unit	Best value (sleep staging)	Best value (BCI)
`ChannelsDropout`	p_{drop}	$[0, 1]$	-	0.4	1
`ChannelsShuffle`	p_{shuffle}	$[0, 1]$	-	0.8	0.1
`SensorXRotations`	θ_{rot}	$[0, 30]$	degree	25^o	3^o
`SensorYRotations`	θ_{rot}	$[0, 30]$	degree	9^o	12^o
`SensorZRotations`	θ_{rot}	$[0, 30]$	degree	30^o	3^o

Table 4: Potential and selected values for the adjustable parameter of each spatial domain augmentation.

5.2.2. Learning curves

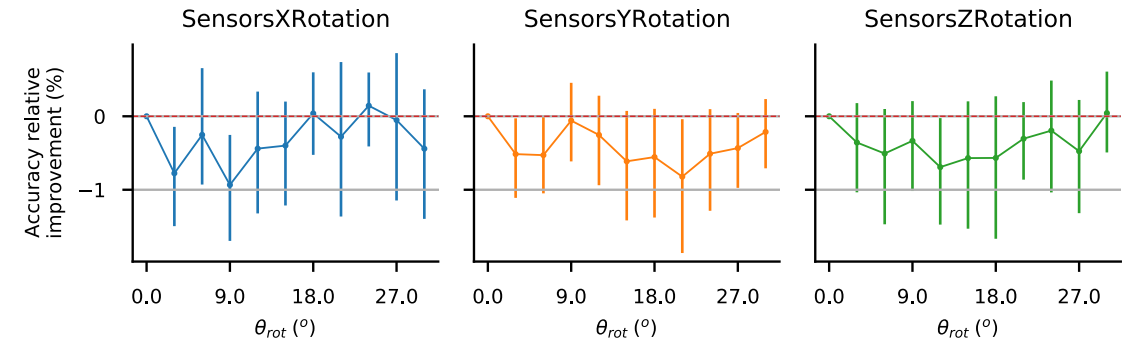
SleepPhysionet The learning curves of spatial domain augmentations in sleep staging are plotted on Figure 15a. All augmentations seem to globally perform on par with the baseline, suggesting they are not particularly relevant for sleep stage classification. These poor results could also be explained by the fact that the *SleepPhysionet* dataset comprises only 2 EEG electrodes in the sagital plane. Note that, for this reason, the `ChannelsSymmetry` augmentation was omitted for these experiments as it would correspond to the Identity mapping here.

BCI IV 2a The learning curves obtained with spatial data augmentations are presented on Figure 15b. They confirm that mixing channels up with `ChannelsShuffle` and `ChannelsSymmetry` might be detrimental for motor imagery tasks. These results confirm that the spatial information in multivariate EEG recordings is crucial for BCI. On the contrary, `ChannelsDropout` significantly enhance the performance, specially on larger training sets. Unlike the two previous spatial augmentations, `ChannelsDropout` helps to learn more robust motor imagery features by hiding part of the spatial information without misleading the model. Finally, its seems `SensorsRotation` augmentations can be helpful although performance improvements depicted in Figure 16 are not statistically significant, as already observed in Figure 14b.

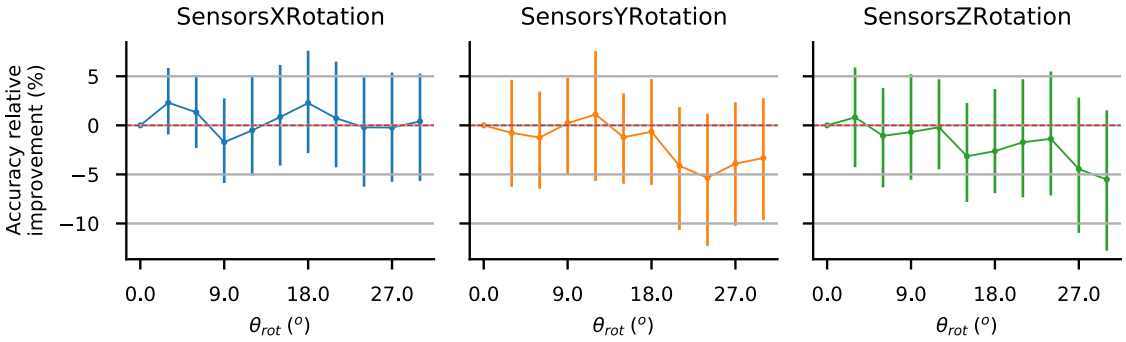
5.2.3. Per class analysis

SleepPhysionet While Figure 17a confirms that spatial augmentations have no significant impact on the classification of stages W, N1 and N2, it also brings more nuance to the previous results from Figure 15a. Indeed, it seems that dropping channels can significantly harm the recognition of N3 and REM stages, suggesting they are more easily identified on one of the two available channels. Likewise, shuffling seems to degrade the performance for the N3 stage, while yielding 10% median improvement for REM stages. This might indicate that N3 stages are partly characterized by spatial patterns which are lost when channels are shuffled. It also seems to confirm that REM stages are not localized and rather correspond to a global brain activity, similar to the awake state.

BCI IV 2a The results per class for the motor imagery task presented in Figure 17b confirm that `ChannelsSymmetry` is particularly detrimental to the classification of right and left hand movements, which are charac-



(a) *SleepPhysionet*



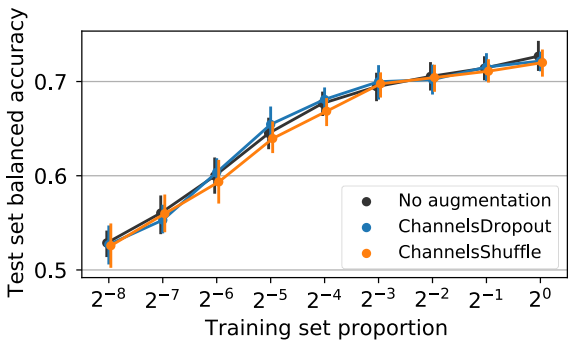
(b) *BCI IV 2a*

Figure 14: Rotational augmentations parameters selection on the *SleepPhysionet* (a) and *BCI IV 2a* (b) datasets. Models were trained on respectively 350 and 60 windows using augmentations parametrized with 10 different linearly spaced values. Validation accuracies are reported relatively to a model trained without data augmentation. The error bars correspond to the 95% confidence intervals based on a 10-fold cross-validation.

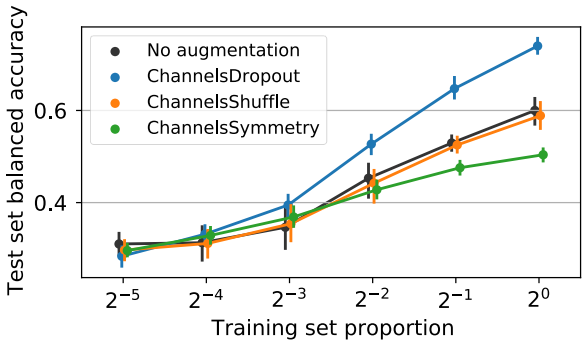
terised by a heavily lateralised brain activity. Meanwhile, this augmentation slightly helps to learn to recognize tongue movements, which are associated with non-lateralised brain activities. In fact, this last class seems to be the least affected by all three augmentations. These results confirm that tongue movements are not spatially characterized and lateralised as heavily as the other movements considered [45]. Concerning spatial rotations, Figure 18 also helps to nuance the previous aggregated results from Figure 16. Indeed, although boxes' whiskers reach negative values, rotations around the Z axis consistently improve the performance in at least 75% of cases by a significant amount.

5.3. Conclusion of spatial augmentations experiments

While some spatial domain augmentations lead to promising results on our motor imagery BCI experiments, none of them seems very helpful for sleep stage classification. `ChannelsDropout` appears to be particularly interesting for BCI, leading to up to 25% accuracy boosts. `SensorsRotations` can also lead to interesting performance increases, specially around axis Z (longitudinal axis point-



(a) *SleepPhysionet*



(b) *BCI IV 2a*

Figure 15: Learning curves for spatial domain augmentations (except rotations) along with the baseline trained with no augmentation. For each transformation, the same model is trained on fractions of the dataset of increasing size. After each training, the average balanced accuracy score on the test set is reported with error bars representing the 95% confidence intervals estimated from 10-fold cross-validation.

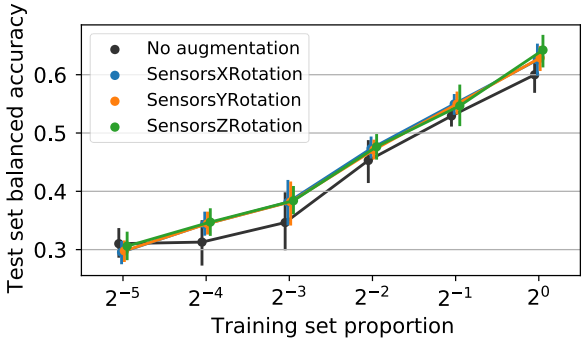
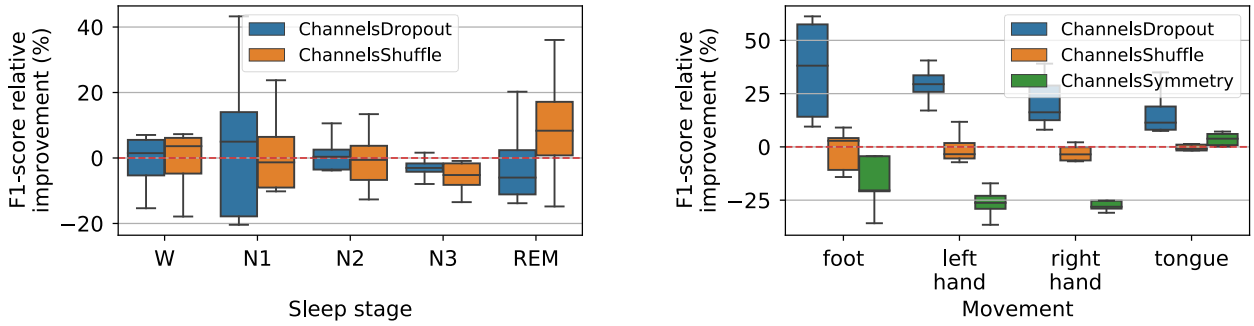


Figure 16: Learning curves for sensors rotations augmentations on the *BCI IV 2a* dataset, along with the baseline trained with no augmentation. For each transformation, the same model is trained on fractions of the dataset of increasing size. After each training, the average balanced accuracy score on the test set is reported with error bars representing the 95% confidence intervals estimated from 10-fold cross-validation.

ing up). On the contrary, the sensors permutations done in `ChannelsShuffle` and `ChannelsSymmetry` can actually harm the performance when training to classify lateralised brain activities.

6. General discussion and findings summary

The findings of our experiments are well-summarized on Figure 19, which shows the learning curves of the two best augmentations of each group for the two datasets studied. It becomes apparent from Figure 19a that time and frequency augmentations are preferable for sleep stage classification tasks. In this case, data augmentations mainly help when training on small datasets, leading to improvements between 5-12%, as opposed to 1-2% boosts when training on larger training set sizes. We did not find that spatial domain augmentations are relevant for sleep stage classification, although this might be due to the low spatial resolution of the two-electrode dataset considered. In contrast, Figure 19b indicates that motor imagery can benefit from all three groups of augmentations. On smaller training sets, we find the same winning time and frequency augmentations as for sleep staging (`TimeReverse`



(a) *SleepPhysionet* (b) *BCI IV 2a*

Figure 17: Per-class F1-score for spatial domain transformations (except for rotations). Scores are reported as relative improvement over a baseline trained without data augmentation. Models were trained on 180 and 230 time windows for *SleepPhysionet* and *BCI IV 2a* datasets respectively. Boxplots were estimated using 10-fold cross-validation.

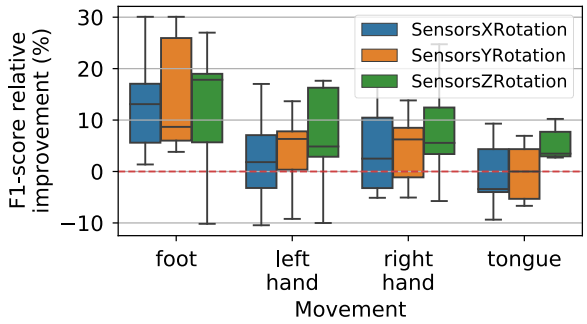


Figure 18: Per-class F1-score for rotation transformations. Scores are reported as relative improvement over a baseline trained without data augmentation. Models were trained on 230 time windows of the *BCI IV 2a* dataset. Boxplots were estimated using 10-fold cross-validation.

and `FTSurrogate`), while *well chosen* spatial augmentations (`ChannelsDropout`) lead to the best results on larger training sets. In general, the BCI task seems to benefit more from data augmentation than the sleep stage classification task, with performance boosts reaching a 45% increase in small data regimes.

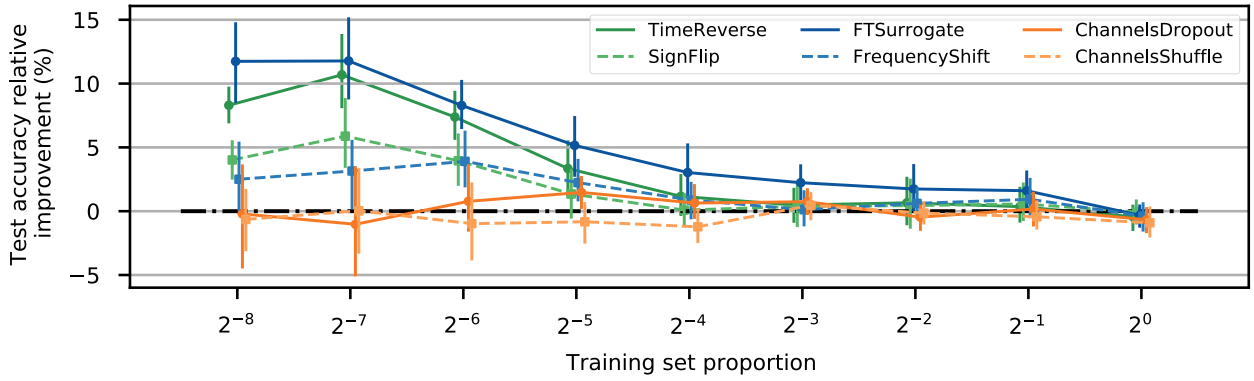
Importantly, one would like to stress that these gains in predictive performance are obtained with a negligible extra computation time. Indeed data augmentation is carried at the data loading level, which is done asynchronously and in parallel during neural networks' training.

7. Conclusion

By allowing to increase the training data during learning, data augmentation limits the need for large annotated datasets that are required to fully leverage the potential of deep learning models. In this paper we carried out a unified and quasi-exhaustive analysis of existing data augmentation methods for EEG signals. To this end, we have presented the rationale and assumptions behind each augmentation considered, both from the perspective of the underlying neurophysiology and of the experimental setups. Overall, our experimental results demonstrate that the use of data augmentation is beneficial for the training of EEG classifiers, both for sleep staging and BCI tasks.

Our experiments on two very different

(a) *SleepPhysionet*



(b) *BCI IV 2a*

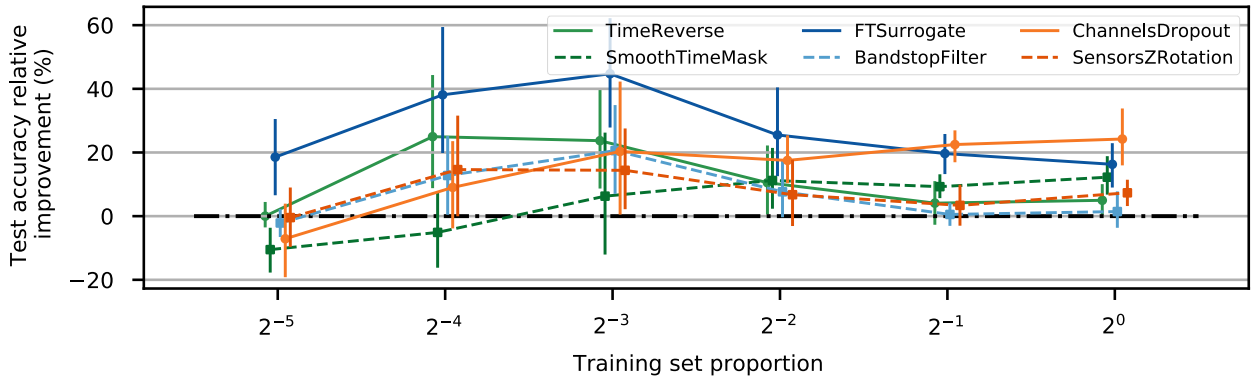


Figure 19: Comparison between the three groups of augmentations. Each curve corresponds to the same model trained with a different augmentation on fractions of the dataset of increasing size. After each training, the average balanced accuracy score on the test set is computed and reported as an improvement relative to the baseline model trained without data augmentation. Error bars represent the 95% confidence intervals estimated from 10-fold cross-validation. Time augmentations are plotted in green, frequency augmentations in blue and spatial augmentations in orange. Full lines with round markers correspond to the best augmentation of each group and dataset, while dashed lines with square markers represent the second best augmentations.

datasets demonstrate the importance of the selection of both the right transformation and magnitude for each different type of task considered. While time-frequency transformations appear to be preferable for sleep stage classification, spatial augmentations also seem competitive for motor imagery tasks. Moreover, our per-class analysis allowed to identify structural differences between different imagined actions and sleep stages, thus also demonstrating the descriptive usefulness of augmentations. These results also align with the claims and experiments presented in [36], showing the relevance of class-dependant data augmentation for neuroscience predictive tasks.

While this study is not completely exhaustive and could be enriched by the addition of other datasets and new augmentations,

we believe that the methodological framework presented, along with our reproducible code¶, should allow to conduct similar analysis on any other EEG dataset and augmentation. We believe that this systematic analysis of EEG data augmentation will help practitioners improve their predictive models and foster new research works aiming to better understand augmentation methods for EEG signals.

Acknowledgments

We would like to thank Apolline Mellot and Hubert J. Banville for their valuable feedback on this manuscript, as well as Martin Wimpff and Bruno Aristimunha for reviewing our augmentation implementations in the braindecode software. This work was supported by the ANR BrAIN (ANR-20-CHIA-0016) and ANR AI-Cog grants (ANR-20-IADJ-0002). It was also granted access to the HPC resources of IDRIS under the allocation 2021-AD011012284 and 2021-AD011011172R2 made by GENCI.

References

[1] Ralph G Andrzejak, Klaus Lehnertz, Florian Mormann, Christoph Rieke, Peter David, and Christian E Elger. Indications of nonlinear deterministic and finite-dimensional structures in time series of brain electrical activity: Dependence on recording region and brain state. *Physical Review E*, 64(6):061907, 2001.

[2] Kai Keng Ang, Zheng Yang Chin, Chuanchu Wang, Cuntai Guan, and Haihong Zhang. Filter bank common spatial pattern algorithm on BCI competition IV datasets 2a and 2b. *Frontiers in neuroscience*, 6:39, 2012.

[3] Hubert Banville, Sean U.N. Wood, Chris Aimone, Denis-Alexander Engemann, and Alexandre Gramfort. Robust learning from corrupted EEG with dynamic spatial filtering. *NeuroImage*, 251:118994, 2022.

[4] R.B. Berry, R. Brooks, C.E. Gamaldo, S.M. Harding, R.M. Lloyd, C.L. Marcus, B.V. Vaughn, and American Academy of Sleep Medicine. *The AASM Manual for the Scoring of Sleep and Associated Events: Rules, Terminology and Technical Specifications : Version 2.3*. American Academy of Sleep Medicine, 2015.

[5] Clemens Brunner, Robert Leeb, Gernot Müller-Putz, Alois Schlögl, and Gert Pfurtscheller. BCI Competition 2008–Graz data set A. *Institute for Knowledge Discovery (Laboratory of Brain-Computer Interfaces), Graz University of Technology*, 16:1–6, 2008.

[6] R. T. Canolty et al. High gamma power is phase-locked to theta oscillations in human neocortex. *Science*, 313(5793): 1626–8, 2006.

[7] Stanislas Chambon, Mathieu Galtier, Pierrick Arnal, Gilles Wainrib, and Alexandre Gramfort. A deep learning architecture for temporal sleep stage classification using multivariate and multimodal time series. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 26(4):758–769, 2018.

[8] Junjian Chen, Zhuliang Yu, Zhenghui Gu, and Yuanqing Li. Deep temporal-spatial feature learning for motor imagery-based brain–computer interfaces. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(11):2356–2366, 2020.

¶ https://github.com/eeg-augmentation-benchmark/eeg-augmentation-benchmark-2022

[9] Shuxiao Chen, Edgar Dobriban, and Jane H. Lee. A Group-Theoretic Framework for Data Augmentation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2020.

[10] Joseph Y. Cheng, Hanlin Goh, Kaan Dogrusoz, Oncel Tuzel, and Erdrin Azemi. Subject-Aware Contrastive Learning for Biosignals. *arXiv:2007.04871*, 2020.

[11] Maureen Clerc, Laurent Bougrain, and Fabien Lotte. *Brain-Computer Interfaces 1*. Wiley, July 2016.

[12] Olivier Deiss, Siddharth Biswal, Jing Jin, Haoqi Sun, M. Brandon Westover, and Jimeng Sun. HAMLET: Interpretable Human And Machine co-LEarning Technique. *arXiv:1803.09702*, 2018.

[13] Tom Dupré la Tour, Lucille Tallot, Laetitia Grabot, Valérie Doyère, Virginie van Wassenhove, Yves Grenier, and Alexandre Gramfort. Non-linear auto-regressive models for cross-frequency coupling in neural time series. *PLOS Computational Biology*, 13(12):1–32, 12 2017.

[14] Steven Y. Feng, Varun Gangal, Jason Wei, Sarath Chandar, Soroush Vosoughi, Teruko Mitamura, and Eduard Hovy. A survey of data augmentation approaches for NLP. *arXiv:2105.03075*, 2021.

[15] Lukas A. W. Gemein, Robin T. Schirrmeister, Patryk Chrabąszcz, Daniel Wilson, Joschka Boedecker, Andreas Schulze-Bonhage, Frank Hutter, and Tonio Ball. Machine-learning-based diagnostics of EEG pathology. *NeuroImage*, 220:117021, 2020.

[16] Ary L Goldberger, Luis AN Amaral, Leon Glass, Jeffrey M Hausdorff, Plamen Ch Ivanov, Roger G Mark, Joseph E Mietus, George B Moody, Chung-Kang Peng, and H Eugene Stanley. PhysioBank, PhysioToolkit, and PhysioNet: components of a new research resource for complex physiologic signals. *Circulation*, 101(23):e215–e220, 2000.

[17] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti S. Hämäläinen. MEG and EEG data analysis with MNE-Python. *Frontiers in Neuroscience*, 7 (267):1–13, 2013.

[18] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A. Engemann, Daniel Strohmeier, Christian Brodbeck, Roman Goj, Mainak Jas, Teon Brooks, Lauri Parkkonen, and Matti S. Hämäläinen. MEG and EEG Data Analysis with MNE-Python. *Frontiers in Neuroscience*, 7 (267):1–13, 2013.

[19] Chao He, Jialu Liu, Yuesheng Zhu, and Wencai Du. Data augmentation for deep neural networks model in EEG classification task: A review. *Frontiers in Human Neuroscience*, page 747, 2021.

[20] Vinay Jayaram and Alexandre Barachant. MOABB: trustworthy algorithm benchmarking for BCIs. *Journal of neural engineering*, 15(6):066011, 2018.

[21] Diederik P. Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *International Conference on Learning Representations (ICLR)*, 2015.

[22] Mario Michael Krell and Su Kyoung Kim. Rotational data augmentation for electroencephalographic data. In *2017 39th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 471–474. IEEE, 2017.

[23] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E. Hinton. ImageNet classification with deep convolutional neural networks. In *Advances in neural information processing systems (NeurIPS)*, 2012.

[24] Tarek Lajnef, Sahbi Chaibi, Perrine Ruby, Pierre-Emmanuel Aguera, Jean-Baptiste Eichenlaub, Mounir Samet, Abdennaceur Kachouri, and Karim Jerbi. Learning Machines and Sleeping Brains: Automatic Sleep Stage Classification using Decision-Tree Multi-Class Support Vector Machines. *Journal of Neuroscience Methods*, 250:94–105, 2015.

[25] Elnaz Lashgari, Dehua Liang, and Uri Maoz. Data augmentation for deep-learning-based electroencephalography. *Journal of Neuroscience Methods*, 346:108885, 2020.

[26] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. In *International Conference on Learning Representations (ICLR)*, 2019.

[27] Mostafa Mohsenvand, Mohammad Rasool Izadi, and Pattie Maes. Contrastive Representation Learning for Electroencephalogram Classification. In *Machine Learning for Health*, 2020.

[28] Daniel S. Park, William Chan, Yu Zhang, Chung-Cheng Chiu, Barret Zoph, Ekin D. Cubuk, and Quoc V. Le. SpecAugment: A simple data augmentation method for automatic speech recognition. In *Interspeech*. ISCA, 2019.

[29] Wilder Penfield and Herbert Jasper. *Epilepsy and the functional anatomy of the human brain*. Epilepsy and the functional anatomy of the human brain. Little, Brown & Co., Oxford, England, 1954. Pages: xv, 896.

[30] François Perrin, Jacques Pernier, Olivier Bertrand, and Jean Francois Echallier. Spherical splines for scalp potential and current density mapping. *Electroencephalography and clinical neurophysiology*, 72(2):184–187, 1989.

[31] Mathias Perslev, Sune Darkner, Lykke Kempfner, Miki Nikolic, Poul Jørgen Jennum, and Christian Igel. U-sleep: resilient high-frequency sleep staging. *NPJ digital medicine*, 4(1):1–12, 2021.

[32] Huy Phan, Oliver Y. Chen, Minh C. Tran, Philipp Koch, Alfred Mertins, and Maarten De Vos. XSleepNet: Multi-view sequential model for automatic sleep staging. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[33] H. Ramoser, J. Müller-Gerking, and G. Pfurtscheller. Optimal spatial filtering of single trial EEG during imagined hand movement. *IEEE transactions on rehabilitation engineering*, 8(4):441–446, 2000.

[34] Allan Rechtschaffen and Anthony Kales. *A Manual of Standardized Terminology, Techniques and Scoring System for Sleep Stages of Human Subjects*. Brain Information Service/Brain Research Institute, University of California, 1973.

[35] Pedro Luiz Coelho Rodrigues, Christian Jutten, and Marco Congedo. Riemannian procrustes analysis: Transfer learning for brain–computer interfaces. *IEEE Transactions on Biomedical Engineering*, 66(8):2390–2401, 2019. doi: 10.1109/TBME.2018.2889705.

[36] Cédric Rommel, Thomas Moreau, Joseph Paillard, and Alexandre Gramfort. CADDA: Class-wise Automatic Differentiable Data Augmentation for EEG

Signals. In *International Conference on Learning Representations (ICLR)*, 2022.

[37] Richard S. Rosenberg and Steven Van Hout. The American Academy of Sleep Medicine Inter-scorer Reliability Program: Sleep Stage Scoring. *Journal of Clinical Sleep Medicine : JCSM : Official Publication of the American Academy of Sleep Medicine*, 9(1):81–87, 2013.

[38] Yannick Roy, Hubert Banville, Isabela Albuquerque, Alexandre Gramfort, Tiago H Falk, and Jocelyn Faubert. Deep learning-based electroencephalography analysis: a systematic review. *Journal of Neural Engineering*, 16(5):051001, 2019.

[39] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. ImageNet large scale visual recognition challenge. *International Journal of Computer Vision*, 2015.

[40] Aaqib Saeed, David Grangier, Olivier Pietquin, and Neil Zeghidour. Learning from heterogeneous EEG signals with differentiable channel reordering. In *International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.

[41] Robin Tibor Schirrmeister, Jost Tobias Springenberg, Lukas Dominique Josef Fiederer, Martin Glasstetter, Katharina Eggensperger, Michael Tangermann, Frank Hutter, Wolfram Burgard, and Tonio Ball. Deep learning with convolutional neural networks for EEG decoding and visualization. *Human Brain Mapping*, 38 (11):5391–5420, 2017.

[42] Justus T. C. Schwabedal, John C. Snyder, Ayse Cakmak, Shamim Nemati, and Gari D. Clifford. Addressing Class Imbalance in Classification Problems of Noisy Signals by using Fourier Transform Surrogates. *arXiv:1806.08675*, 2019.

[43] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15 (56):1929–1958, 2014.

[44] Fang Wang, Sheng-hua Zhong, Jianfeng Peng, Jianmin Jiang, and Yan Liu. Data Augmentation for EEG-Based Emotion Recognition with Deep Convolutional Neural Networks. In *MultiMedia Modeling*, volume 10705, pages 82–93. Springer, 2018.

[45] Jobu Watanabe, Motoaki Sugiura, Naoki Miura, Yoshihiko Watanabe, Yasuhiro Maeda, Yoshihiko Matsue, and Ryuta Kawashima. The human parietal cortex is involved in spatial processing of tongue movement-an fMRI study. *NeuroImage*, 21(4):1289–1299, 2004.