Robust learning from corrupted EEG with dynamic spatial filtering

Hubert Banville*[1,2], Sean U.N. Wood[2], Chris Aimone[2], Denis-Alexander Engemann†[1,3], and Alexandre Gramfort†[1]

[1]Université Paris-Saclay, Inria, CEA, Palaiseau, France
[2]InteraXon Inc., Toronto, Canada
[3]Max Planck Institute for Human Cognitive and Brain Sciences, Department of Neurology, Leipzig, Germany

Abstract

Building machine learning models using EEG recorded outside of the laboratory setting requires methods robust to noisy data and randomly missing channels. This need is particularly great when working with sparse EEG montages (1-6 channels), often encountered in consumer-grade or mobile EEG devices. Neither classical machine learning models nor deep neural networks trained end-to-end on EEG are typically designed or tested for robustness to corruption, and especially to randomly missing channels. While some studies have proposed strategies for using data with missing channels, these approaches are not practical when sparse montages are used and computing power is limited (*e.g.,* wearables, cell phones). To tackle this problem, we propose dynamic spatial filtering (DSF), a multi-head attention module that can be plugged in before the first layer of a neural network to handle missing EEG channels by learning to focus on good channels and to ignore bad ones. We tested DSF on public EEG data encompassing ∼4,000 recordings with simulated channel corruption and on a private dataset of ∼100 at-home recordings of mobile EEG with natural corruption. Our proposed approach achieves the same performance as baseline models when no noise is applied, but outperforms baselines by as much as 29.4% accuracy when significant channel corruption is present. Moreover, DSF outputs are interpretable, making it possible to monitor the effective channel importance in real-time. This approach has the potential to enable the analysis of EEG in challenging settings where channel corruption hampers the reading of brain signals.

Keywords Electroencephalography, mobile EEG, deep learning, machine learning, noise robustness

1 Introduction

Electroencephalography (EEG) enables investigations into brain function and health in an economical manner and for a wide array of purposes, including sleep monitoring, pathology screening, neurofeedback, brain-computer interfacing and anaesthesia monitoring [1, 2, 3, 4, 5, 6]. Thanks to recent advances in mobile EEG technology, these applications can now be more easily

*correspondence: hubert.jacob-banville@inria.fr
†joint senior authors

translated from the lab and clinic to contexts such as at-home or ambulatory assessments. This carries the potential of democratizing EEG applications and revolutionizing the study of brain health in real-world settings. However, in these new settings, the number of electrodes available is often limited and signal quality is much harder to control. Moreover, with the increasing availability of these devices, the amount of data generated now exceeds the capacity of human experts (*e.g.,* neurologists, sleep technicians, etc.) to analyze and manually annotate every single recording, as is traditionally done in research and clinical settings. Novel methods facilitating clinical and research applications in real-world settings, especially with sparse EEG montages, are therefore needed.

The use of machine learning for automating EEG analysis has been the subject of much research in recent decades [7, 8]. However, state-of-the-art EEG prediction pipelines are generally benchmarked on datasets recorded in well-controlled conditions that are relatively clean when compared to data from mobile EEG. As a result, it is unclear how models designed for laboratory data will cope with signals encountered in real-world contexts. This is especially critical for mobile EEG recordings that may contain a varying number of usable channels as well as overall noisier signals, in contrast to most research- and clinical-grade recordings. In addition, the difference in number of channels between research and mobile settings also means that interpolating bad channels offline (as is commonly done in recordings with dense electrode montages) is likely to fail on mobile EEG devices given their limited spatial information. It is an additional challenge that the quality of EEG data is not static but can vary significantly within a given recording. This suggests that predictive models should handle noise dynamically. Ideally, not only should machine learning pipelines produce predictions that are robust to (changing) sources of noise in EEG, but they should also do so in a way that is interpretable. For instance, if noise is easily identifiable, corrective action can be quickly taken by experimenters or users during a recording.

It is important to consider that not all sources of noise affect EEG recordings in the same way [9]. Physiological artifacts are large electrical signals that are generated by current sources outside the brain such as heart activity, eye or tongue movement, muscle contraction, sweating, etc. Depending on the EEG electrode montage and the setting of the recording (*e.g.,* eyes open or closed), these artifacts can be more or less disruptive to measuring the brain activity of interest. Movement artifacts, on the other hand, are caused by the relative displacement of EEG electrodes with respect to the scalp, and can introduce noise of varying spectral content in the affected electrodes during movement. If an electrode cannot properly connect with the skin (*e.g.,* after a movement artifact or because it was not correctly set up initially), its reading will likely contain little or no physiological information and instead pick up instrumentation and environmental noise. These are commonly referred to as "bad" or "missing" channels in the literature. In the context of this work, we refer to them as "corrupted channels" to explicitly include the case where a signal corruption mechanism (*e.g.,* active noise sources in uncontrolled environments) must be accounted for by predictive models. While channel corruption affects EEG recordings in all contexts, it is more likely in real-world mobile EEG recordings than in controlled laboratory settings where trained experimenters can monitor and remedy bad electrodes during the recording. Therefore, special care must be given to the problem of channel corruption in sparse mobile EEG settings.

In this paper, we propose and benchmark an attention mechanism module designed to handle corrupted channel data, based on the concept of "scaling attention" [10, 11]. This module can be inserted before the first layer of any convolutional neural network architecture in which activations have a spatial dimension [12, 13, 14], and then be trained end-to-end for the prediction task at

hand.

The rest of the paper is structured as follows. Section 2 presents an overview of the EEG noise handling literature, then describes the attention module and denoising procedure proposed in this study. The neural architectures, baseline methods and data used in our experiments are introduced in Section 3. Next, Section 4 reports the results of our experiments on sleep and pathology EEG datasets. Lastly, we examine related work and discuss the results in Section 5.

2 Methods

2.1 State-of-the-art approaches to noise-robust EEG processing

Existing strategies for dealing with noisy data can be divided into three categories (Table 1): (1) ignoring or rejecting noisy segments, (2) implicit denoising, *i.e.*, methods that allow models to work despite noise, and (3) explicit denoising, *i.e.*, methods that rely on a separate preprocessing step to handle noise or missing channels before prediction. We now discuss existing methods employing these strategies in more detail.

The simplest way to deal with noise in EEG is to assume that it is negligible or to simply discard bad segments [8]. For instance, a manually selected amplitude or variance threshold [15, 16, 17] or a classifier trained to recognize artifacts [2] can be used to identify segments to be ignored. This approach, though commonplace, is ill-suited to mobile EEG settings where noise cannot be assumed to be negligible, but also to online applications where model predictions need to be continuously available. Moreover, this approach is likely to discard windows due to a small fraction of bad electrodes, potentially losing usable information from other channels.

Implicit denoising approaches can be used to design noise-robust processing pipelines that do not contain a specific noise handling step. First, implicit denoising approaches can use representations of EEG data that are robust to missing channels. For instance, multichannel EEG can be transformed into topographical maps ("topomaps") that are less sensitive to the absence of a few channels. This representation is then typically fed into a standard convolutional neural network (ConvNet) architecture. While this approach can gracefully handle missing channels in dense montages (*e.g.,* 16 to 64 channels in [18, 19, 20]), it is likely to perform poorly on sparse montages (*e.g.,* 4 channels) as spatial interpolation might fail if channels are missing. Moreover, this approach requires computationally demanding preprocessing and feature extraction steps, undesirable in online and low-computational resources contexts. In the traditional machine learning setting, Sabbagh *et al.* [21] showed that representing input windows as covariance matrices and using Riemannian geometry-aware models did not require common noise correction steps to reach high performance on a brain age prediction task. However, the robustness of this approach has not been evaluated on sparse montages. Also, its integration into neural network architectures is not straightforward with geometry-aware deep learning remaining an active field of research [22]. Signal processing techniques can also be used to promote invariance to certain types of noise. For instance, the Lomb-Scargle periodogram can be used to extract spectral representations that are robust to missing samples [23, 24]. However, this approach fails when channels are completely missing. Finally, implicit denoising can be achieved with traditional machine learning models that are inherently robust to noise. For instance, random forests trained on handcrafted EEG features were shown to be notably more robust to low SNR inputs than univariate models on a state-of-consciousness prediction task [25]. Although promising, this approach is limited by its feature engineering step, as features (1) rely heavily on domain knowledge, (2) might not be optimal to the task, and (3) require an

additional processing step which can be prohibitive in limited resource contexts.

Multiple studies have explicitly handled noise by correcting corrupted signals or predicting missing or additional channels from available ones. Spatial projection approaches aim at projecting the input signals to a noise-free subspace before projecting the signals back into channel-space, *e.g.,* using independent component analysis (ICA) [26, 27, 28] or principal components analysis (PCA) [29, 30]. While approaches such as ICA are powerful tools to mitigate artifact and noise components in a semi-automated way, their efficacy can diminish when only few channels are available. For instance, in addition to introducing an additional preprocessing step, these approaches are likely to discard important discriminative information during preprocessing because they are decoupled from the prediction task. Also, the fact that preprocessing is done independently from the supervised learning task, or the statistical testing procedure, actually makes the selection of preprocessing parameters (*e.g.,* number of good components) challenging. Motivated by the challenge of parameter selection, fully automated denoising pipelines have been proposed. FASTER [31] and PREP [32] both combine artifact correction, noise removal and bad channel interpolation into a single automated pipeline. Autoreject [33] is another recently developed pipeline that uses cross-validation to automatically select amplitude thresholds to use for rejecting windows or flagging bad channels. These approaches are well-suited to offline analyses where the morphology of the signals is of interest, however they are typically computationally demanding and are also decoupled from the statistical modeling. Additionally, it is unclear how interpolation can be applied when using bipolar montages (*i.e.*, that do not share a single reference), as is often the case in *e.g.,* polysomnography [34] and epilepsy monitoring [35].

Finally, generic machine learning models have been proposed to recover bad channels. For instance, generative adversarial networks (GANs) have been trained to recover dense EEG montages from a few electrodes [36, 37]. Other similar methods have been proposed, *e.g.,* long short-term memory (LSTM) neural networks [38], autoencoders [39], or tensor decomposition and compressed sensing [40, 41]. However, these methods postulate that the identity of bad channels is known ahead of time, which is a non-trivial assumption in practice.

In contrast to the existing literature on channel corruption handling in EEG, we introduce an interpretable end-to-end denoising approach that can learn implicitly to work with corrupted sparse EEG data, and that does not require additional preprocessing steps.

2.2 Dynamic spatial filtering: Second-order attention for learning on noisy EEG signals

The key goal behind dynamic spatial filtering (DSF) is to help neural networks focus on the most important channels, at each time instant, given a specific machine learning task on EEG. To do so, we introduce a spatial attention mechanism that dynamically reweights channels according to their predictive power. This idea is inspired by recent developments in attention mechanisms, most specifically the "scaling attention" approach proposed in computer vision [10, 11]. Notably, DSF leverages second-order information, *i.e.*, spatial covariance, to capture dependencies between EEG channels. In this section, we detail the learning problem under study, the proposed attention architecture and a data augmentation transform designed to help train noise-robust models.

Notation We denote by $[\![q]\!]$ the set $\{1, \dots, q\}$. The index t refers to time indices in the multivariate time series $S \in \mathbb{R}^{C \times M}$, where M is the number of time samples and C is the number of EEG channels. S is further divided into non-overlapping windows $X \in \mathbb{R}^{C \times T}$ where T is the

Table 1: Existing methods for dealing with noisy EEG data.

	Approach	Examples	Notes
Ignore or reject noise	No denoising	[12, 13, 42, 43, 44, 45, 46, 47, 48]	Might not work in real-life applications (out of the lab/clinic)
	Removing bad epochs	[15, 2, 16, 17]	Doesn't allow online predictions; Might discard useful information
Implicit denoising	Robust input representations	Covariance matrices in Riemannian tangent space [21]	Might not work if too few channels available
		Topomaps [18, 19, 20]	Expensive preprocessing step; Might not work if too few channels available
	Robust signal processing techniques	Lomb-Scargle periodogram [23, 24]	Only useful for missing samples, not missing channels
	Robust machine learning classifiers	Handcrafted features and random forest [25]	Requires feature engineering step
Explicit denoising	Spatial projection-based approaches	Signal Space Separation (SSS) for MEG [49]	Might not work if too few channels available; Additional preprocessing step; Preprocessing might discard important information for learning task
		ICA-based denoising [26, 27, 28]	
	Automated correction	Autoreject [33], FASTER [31], PREP [32]	Expensive preprocessing step
	Model-based interpolation/ reconstruction	Deep learning-based superresolution (GAN, LSTM, AE, etc.) [50, 51, 36, 37, 39]	Separate training step; Additional inference step to reconstruct at test time; Requires separate procedure to detect corrupted channels
		Tensor decomposition, compressed sensing [41, 40]	
Interpretable denoising	**Channel corruption-invariant architecture**	**Dynamic Spatial Filtering (this work)**	**Trained end-to-end, no additional preprocessing, interpretable, works with sparse montages**

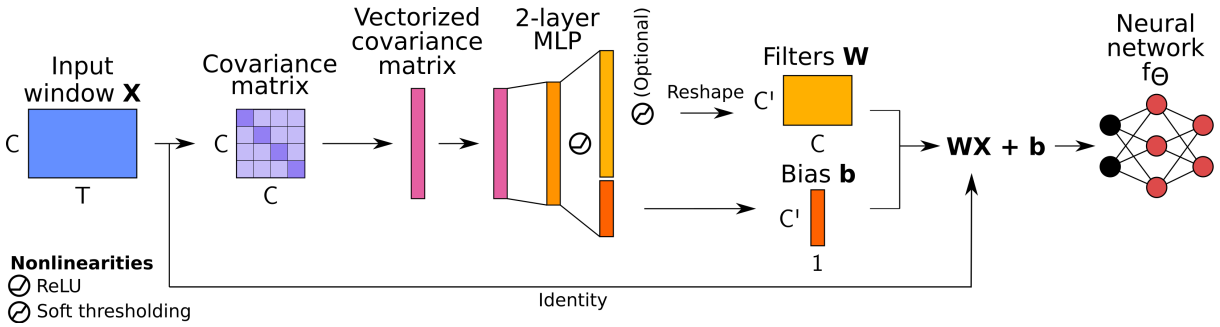

Figure 1: Visual description of the Dynamic Spatial Filtering (DSF) attention module. An input window $\mathbf{X}$ with C spatial channels is processed by a 2-layer MLP to produce a set of C' spatial filters $\mathbf{W}$ and biases $\mathbf{b}$ that dynamically transform the input $\mathbf{X}$. This allows the subsequent layers of a neural network to ignore bad channels and focus on the most informative ones.

number of time samples in the window. We denote by $y \in \mathcal{Y}$ the target used in the learning task. Typically, $\mathcal{Y}$ is $[\![L]\!]$ for a classification problem with L classes.

We perform experiments in the supervised classification setting. A model $f_\Theta : \mathcal{X} \to \mathcal{Y}$ with parameters Θ (*e.g.,* a convolutional neural network) is trained to predict the class y of EEG windows X. For this, we train f_Θ to minimize the loss $\mathcal{L}$, *e.g.,* the categorical cross-entropy loss, over the example-label pairs (X_i, y_i):

$$\hat{f_\Theta} = \arg\min_\Theta \mathbb{E}_{X_i, y_i \in \mathcal{X}\times\mathcal{Y}}[\mathcal{L}(f_\Theta(X_i), y_i)] \ . \tag{1}$$

In particular, we are interested in the performance of f_Θ when random channels are corrupted and more specifically when channel corruption occurs at test time (*i.e.*, when training data is mostly clean). Toward this goal, we insert an attention-based module $m_{\text{DSF}} : \mathbb{R}^{C\times T} \to \mathbb{R}^{C'\times T}$ into f_Θ which performs a (fixed) transformation $\Phi(X)$ to extract relevant spatial information from X, followed by a reweighting mechanism for the input signals.

In order to implicitly handle noise in neural network architectures, we design an attention module where second-order information is extracted from the input and used to predict weights of a linear transformation of the input EEG channels, that are optimized for the learning task (Fig. 1). Applying such linear transforms to multivariate EEG signals is commonly referred to as "spatial filtering", a technique that has been widely used in the field of EEG [52, 53, 54, 55, 56, 57, 58]. This enables the model to learn to ignore noisy outputs and/or to reweight them, while still leveraging any remaining spatial information. We now show how this module can be applied to the raw input X.

We define the dynamic spatial filter (DSF) module m_{DSF} as:

$$m_{\text{DSF}}(X) = W_{\text{DSF}}(X)X + b_{\text{DSF}}(X) \ , \tag{2}$$

where $W_{\text{DSF}} \in \mathbb{R}^{C'\times C}$ and $b_{\text{DSF}} \in \mathbb{R}^{C'}$ are obtained by reshaping the output of a neural network, *e.g.,* a multilayer perceptron (MLP), $h_{\Theta_{DSF}}(\Phi(X)) \in \mathbb{R}^{C'\times(C+1)}$ (see Fig. 1). Under this formulation, each row in W_{DSF} corresponds to a spatial filter that linearly transforms the input signals into another virtual channel. Here, C' can be set to the number of input spatial channels C or considered a hyperparameter of the attention module[1]. When $C' = C$, if the

[1]In which case it can be used to increase the diversity of input channels in models trained on sparse montages

diagonal of W_{DSF} is 0, W_{DSF} corresponds to a linear interpolation of each channel based on the $C-1$ others, as is commonly done in the classical EEG literature [59] (see Appendix Ffor an in-depth discussion). Heavily corrupted channels can be ignored by giving them a weight of 0 in W_{DSF}. To facilitate this behavior, we can further apply a soft-thresholding element-wise nonlinearity to W_{DSF}:

$$W'_{\text{DSF}} = \text{sign}(W_{\text{DSF}}) \max(|W_{\text{DSF}}| - \tau, 0) \ , \tag{3}$$

where τ is a threshold empirically set to 0.1, $|\cdot|$ is the element-wise absolute value and both the sign and max operators are applied element-wise.

In our experiments, the spatial information extracted by the transforms $\Phi(X)$ was either (1) the log-variance of each input channel or (2) the flattened upper triangular part of the matrix logarithm of the covariance matrix of X (see Appendix A)[2]. When reporting results, we denote models as *DSFd* and *DSFm* when DSF takes the log-variance or the matrix logarithm of the covariance matrix as input, respectively. We further add the suffix "-st" to indicate the use of the soft-thresholding nonlinearity, *e.g., DSFm-st*.

Interestingly, the DSF module can be seen as a multi-head attention mechanism [60] with real-valued attention weights and where each head is tasked with producing a linear combination of the input spatial signals.

Finally, we can inspect the attention given by m_{DSF} to each input channel by computing the "effective channel importance" metric [3] $\boldsymbol{\phi} \in \mathbb{R}^C$ where

$$\phi_j = \sqrt{\sum_{i=1}^{C'} {W_{ij}}^2} \ . \tag{4}$$

Intuitively, $\boldsymbol{\phi}$ measures how much each input channel is used by m_{DSF} to produce the output virtual channels. A normalized version

$$\hat{\boldsymbol{\phi}} = \frac{\boldsymbol{\phi}}{\max_i \boldsymbol{\phi}_i} \tag{5}$$

can also be used to obtain a value between 0 and 1. This straightforward way of inspecting the functioning of the DSF module facilitates the identification of important or noisy channels.

To further help our models learn to be robust to noise, we design a data augmentation procedure that randomly corrupts channels. Specifically, channel corruption is simulated by performing a masked channel-wise convex combination of input channels and Gaussian white noise $Z \in \mathbb{R}^{C \times T}$:

$$\tilde{X} = (1-\eta)\,\text{diag}(\boldsymbol{\nu})X + \eta\,\text{diag}(\boldsymbol{\nu})Z + \text{diag}(1-\boldsymbol{\nu})X \ , \tag{6}$$

($C' > C$) or perform dimensionality reduction to reduce computational complexity ($C' < C$).

[2] In practice, if a channel is "flat-lining" (has only 0s) inside a window and therefore has a variance of 0, its log-variance is replaced by 0. Similarly, if a covariance matrix eigenvalue is 0 when computing the matrix logarithm (see Appendix A), its logarithm is replaced by 0.

[3] "Effective channel importance" measures how useful the **actual data** of a channel is. It is not to be confused with the theoretical importance of a channel, *i.e.*, the fact that in theory some channels (given good signal quality) might be more useful for some tasks than other channels. Therefore, in this work, when we measure or discuss the "importance" of a channel, we refer to the usefulness of the actual signal collected with that channel with respect to the task. For instance, a corrupted channel will likely have low "importance", although the neurophysiological information available at that location would be useful should the channel not be corrupted. The use of the word *importance* in the present context is in line with the literature in statistical machine learning referring to "feature importance" as quantified for example using "permutation importance" [61].

where $Z_{i,j} \sim \mathcal{N}(0, \sigma_n^2)$ for $i \in [\![T]\!]$ and $j \in [\![C]\!]$, $\eta \in [0, 1]$ controls the relative strength of the noise, and $\boldsymbol{\nu} \in \{0, 1\}^C$ is a masking vector that controls which channels are corrupted. The operator $\text{diag}(x)$ creates a square matrix filled with zeros whose diagonal is the vector x. Here, $\boldsymbol{\nu}$ is sampled from a multinouilli distribution with parameter p. Each window X is individually corrupted using random parameters $\sigma_n \sim \mathcal{U}(20, 50)$ μV, $\eta \sim \mathcal{U}(0.5, 1)$, and a fixed p of 0.5.

2.3 Computational considerations

We set the following hyperparameters when training deep neural networks: optimizer, learning rate schedule, batch size, regularization strength (number of training epochs, weight decay, dropout) and parameter initialization scheme. In all experiments, we used the AdamW optimizer [62] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, a learning rate of 10^{-3} and cosine annealing. The parameters of all neural networks were randomly initialized using uniform He initialization [63]. Dropout [64] was applied to f_Θ's fully connected layer at a rate of 50% and weight decay was applied to the trainable parameters of all layers of both f_Θ and $h_{\Theta_{DSF}}$. Moreover, during training, the loss was weighted to optimize balanced accuracy. Some hyperparameters were tuned on a dataset-specific basis and are described along with the datasets (*i.e.*, weight decay and batch size).

Deep learning and baseline models were trained using a combination of the braindecode [12], MNE-Python [65], PyTorch [66], pyRiemann [67], mne-features [68] and scikit-learn [69] packages.[4] Finally, deep learning models were trained on 1 or 2 Nvidia Tesla V100 or P4 GPUs for anywhere from a few minutes to 7 hours, depending on the amount of data, early stopping and GPU configuration.

3 Experiments

3.1 Downstream tasks

We studied noise robustness through two common EEG classification downstream tasks: sleep staging and pathology detection. First, sleep staging, a critical step in sleep monitoring, allows the diagnosis and study of sleep disorders such as apnea and narcolepsy [70]. This 5-class classification problem consists of predicting which sleep stage (W (wake), N1, N2, N3 (different levels of sleep) or R (rapid eye movement periods)) an individual is in, in non-overlapping 30-s windows of overnight recordings. While a large number of machine learning approaches have been proposed to perform sleep staging [71, 14, 8, 48], the handling of corrupted channels has not been addressed in a comprehensive manner yet, as channel corruption is less likely to occur in clinical and laboratory settings than in the real-world settings we consider here[5].

Second, the pathology detection task aims at detecting neurological conditions such as epilepsy and dementia from an individual's EEG [73, 74]. In a simplified formulation this gives rise to a binary classification problem where recordings have to be classified as either pathological or non-pathological. Such recordings are typically carried out in well-controlled settings (*e.g.,* in a hospital [75]) where sources of noise can be monitored and mitigated in real-time by experts. To test pathology detection performance in the context of mobile EEG acquisition, we used a limited set of electrodes, in contrast to previous work [76, 43, 44].

[4] Code used for data analysis can be found at `https://github.com/hubertjb/dynamic-spatial-filtering`.

[5] A recent study reported training a neural network on artificially-corrupted sleep EEG data, with a goal similar to ours [72]; however, this study only appears as a Supplement with little information on the methods and results.

These two tasks are further described in Section 3.3 when discussing the data used in our experiments.

3.2 Compared methods

We compared the performance of the proposed DSF and data augmentation method to other established approaches. In total, we contrasted combinations of three machine learning pipelines and three different noise-handling strategies.

We consider the following machine learning pipelines: (1) end-to-end deep learning (with and without the DSF module) from raw signals, (2) filter-bank covariance matrices with Riemannian tangent space projection and logistic regression [67, 77, 78, 21] (which we refer to as "Riemann"), and (3) handcrafted features and random forest (RF) [44].

We used ConvNet architectures as f_Θ in deep learning pipelines (Appendix B). For pathology detection, we used the ShallowNet architecture from [12] which parametrizes the frequency-band common spatial patterns (FBCSP) pipeline [44]. We used it without modifying the architecture, yielding a total of 13,482 trainable parameters when $C = 6$. For sleep staging, we used a 3-layer ConvNet which takes 30-s windows as input [14, 79], with a total of 18,457 trainable parameters when $C = 4$ and an input sampling frequency of 100 Hz. Finally, when evaluating DSF, we added modules m_{DSF} before the input layer of each neural network. The input dimensionality of m_{DSF} depends on the chosen spatial information extraction transform $\Phi(X)$: either C (log-variance) or $C(C+1)/2$ (vectorized covariance matrix). We fixed the hidden layer size of m_{DSF} to C^2 units, while the output layer size depended on the chosen C'. The DSF modules added between 420 and 2,864 trainable parameters to those of f_Θ depending on the configuration.

The Riemann pipeline first applied a filter bank to the input EEG, yielding narrow-band signals in the 7 bands bounded by (0.1, 1.5, 4, 8, 15, 26, 35, 49) Hz. Next, covariance matrices were estimated per window and frequency band using the OAS algorithm [80]. The covariance matrices were then projected into their Riemannian tangent space exploiting the Wasserstein distance to estimate the mean covariance used as the reference point [81, 82]. The vectorized covariance matrices with dimensionality of $C(C+1)/2$ were finally z-score normalized using the mean and standard deviation of the training set, and fed to a linear logistic regression classifier.

The handcrafted features baseline, inspired by [44] and [25], relied on 21 different feature types: mean, standard deviation, root mean square, kurtosis, skewness, quantiles (10, 25, 75 and 90th), peak-to-peak amplitude, frequency log-power bands between (0, 2, 4, 8, 13, 18, 24, 30, 49) Hz as well as all their possible ratios, spectral entropy, approximate entropy, SVD entropy, Hurst exponent, Hjorth complexity, Hjorth mobility, line length, wavelet coefficient energy, Higuchi fractal dimension, number of zero crossings, SVD Fisher information and phase locking value. This resulted in 63 univariate features per EEG channel, along with $\binom{C}{2}$ bivariate features, which were concatenated into a single vector of size $63 \times C + \binom{C}{2}$ (*e.g.,* 393 for $C = 6$). In the event of non-finite values in the feature representation of a window, we imputed missing values feature-wise using the mean of the feature computed over the training set. Finally, feature vectors were fed to a random forest model.

When applying traditional pipelines to pathology detection experiments, we aggregated the input representations recording-wise as each recording has a single label (*i.e.*, pathological or not). To do so, we used the geometric mean on covariance matrices and the median on handcrafted features. Deep learning models, on the other hand, were trained on non-aggregated windows, but their performance was evaluated recording-wise by averaging the predictions over windows within each recording. Hyperparameter selection for logistic regression and random forest models

Table 2: Description of the datasets used in this study.

	TUAB [85, 75]	PC18 (train) [83, 84]	MSD
Recording settings	Hospital	Sleep clinic	At-home
# recordings	2,993	994	98
# unique subjects	2,329	994	67
Sampling frequency (Hz)	250, 256 or 512	200	256
# EEG channels	27 to 36	6	4
Reference	Common average	M1 or M2	Fpz
Labels	Normal, abnormal	W, N1, N2, N3, R	W, N1, N2, N3, R

is described in Appendix C.

We combined the machine learning approaches described above with the following noise-handling strategies: (1) no denoising, *i.e.*, models are trained directly on the data without explicit or implicit denoising, (2) Autoreject [33], an automated correction pipeline, and (3) data augmentation, which randomly corrupts channels during training.

Autoreject is a denoising pipeline that explicitly handles noisy epochs and channels in a fully automated manner [33]. First, using a cross-validation procedure, it finds optimal channel-wise peak-to-peak amplitude thresholds to be used to identify bad channels in each window separately. If more than κ channels are bad, the epoch is rejected. Otherwise, up to ρ bad channels are reconstructed using the good channels with spherical spline interpolation. In pathology detection experiments, we allowed Autoreject to reject bad epochs, as classification was performed recording-wise. For sleep staging experiments however, we did not reject epochs as one prediction per epoch was needed, but still used Autoreject to automatically identify and interpolate bad channels. In both cases, we used default values for all parameters as provided in the Python implementation[6], except for the number of cross-validation folds, which we set to 5.

Finally, data augmentation consists of artificially corrupting channels during training to promote invariance to missing channels. When training neural networks, the data augmentation transform was applied on-the-fly to each batch. For feature-based methods, we instead precomputed augmented datasets by applying the augmentation multiple times to each window (10 for pathology detection, 5 for sleep staging), and then extracting features from the augmented windows.

3.3 Data

Approaches were compared on three datasets (Table 2): for pathology detection on the TUH Abnormal EEG dataset [75] and for sleep staging on both the Physionet Challenge 2018 dataset [83, 84] and an internal dataset of mobile overnight EEG recordings.

The TUH Abnormal EEG dataset v2.0.0 (TUAB) [85, 75] contains 2,993 recordings of 15 minutes or more from 2,329 different patients who underwent a clinical EEG exam in a hospital setting. Each recording was labeled as "normal" (1,385 recordings) or "abnormal" (998 recordings) based on detailed physician reports. Most recordings were sampled at 250 Hz and comprised between 27 and 36 electrodes. The corpus is already divided into a training and an evaluation set with 2,130 and 253 recordings each. The mean age across all recordings is 49.3 years (min: 1, max: 96) and 53.5% of recordings are of female patients. The TUAB data was preprocessed in

[6] `https://github.com/autoreject/autoreject`

the following manner. The first minute of each recording was cropped to remove noisy data that occurs at the beginning of recordings [44]. Longer files were cropped such that a maximum of 20 minutes was used from each recording. Then, 21 channels common to all recordings were selected (Fp1, Fp2, F7, F8, F3, Fz, F4, A1, T3, C3, Cz, C4, T4, A2, T5, P3, Pz, P4, T6, O1 and O2). EEG channels were downsampled to 100 Hz and clipped at $\pm 800\,\mu V$. Finally, non-overlapping windows of 6 seconds were extracted, yielding windows of size (600×21). Deep learning models were trained on TUAB with a batch size of 256 and weight decay of 0.01.

Physionet Challenge 2018 dataset (PC18) The Physionet Challenge 2018 (PC18) dataset [83, 84] contains recordings from a total of 1,983 different individuals with (suspected) sleep apnea whose EEG, EOG, chin EMG, respiration airflow and oxygen saturation were monitored overnight. Bipolar EEG channels F3-M2, F4-M1, C3-M2, C4-M1, O1-M2 and O2-M1 were recorded at 200 Hz. Sleep stage annotations were obtained from 7 trained scorers following the AASM manual [34] (W, N1, N2, N3 and R). We focused our analysis on a subset of 994 recordings for which these annotations are publicly available. In this subset of the data, mean age is 55 years (min: 18, max: 93) and 33% of participants are female. For PC18, the EEG was first filtered using a 30 Hz FIR lowpass filter with a Hamming window to reject higher frequencies that are not critical for sleep staging [14, 86]. The EEG channels were then downsampled by a factor of two to 100 Hz to reduce the dimensionality of the input data. Finally, non-overlapping 30-second windows (3000×6) were extracted. Experiments on PC18 used a batch size of 64 and weight decay of 0.001.

Muse Sleep Dataset (MSD) We lastly tested our approach on real-world mobile EEG data, in which channel corruption is likely to occur naturally. We used an internal dataset of overnight sleep recordings collected with the Muse S EEG headband from InteraXon Inc. (Toronto, Canada). This data was collected in accordance with the privacy policy (July 2020) users must agree to when using the Muse headband[7] and which ensures their informed consent concerning the use of EEG data for scientific research purposes. The Muse S is a four-channel dry EEG device (TP9, Fp1, Fp2, TP10, referenced to Fpz), sampled at 256 Hz. The Muse headband has been previously used for event-related potentials research [87], brain performance assessment [6], research into brain development [88], sleep staging [89], and stroke diagnosis [90], among others. A total of 98 partial and complete overnight recordings (mean duration: 6.3 h) from 67 unique users were selected from InteraXon's anonymized database of Muse customers, and annotated by a trained scorer following the AASM manual. Despite the derivations being different from the common montage used in polysomnography, the typical microstructure necessary to identify sleep stages, *e.g.,* sleep spindles, k-complexes and slow waves, can be easily seen in all four channels. Therefore, sleep stage annotations were obtained from actual EEG activity rather than ocular or muscular artifacts. Mean age across all recordings is 37.9 years (min: 21, max: 74) and 45.9% of recordings are of female users. Preprocessing of MSD data was the same as for PC18, with the following differences: (1) channels were downsampled to 128 Hz, (2) missing values (occurring when Bluetooth packets are lost) were replaced by linear interpolation using surrounding valid samples, (3) after filtering and downsampling, samples which overlapped with the original missing values were replaced by zeros, and (4) channels were zero-meaned window-wise. We used a batch size of 64 and weight decay of 0.01 for MSD experiments.

[7] https://choosemuse.com/legal/privacy/

We split the available recordings from TUAB, PC18 and MSD into training, validation and testing, such that recordings used for testing were not used for training or validation. For TUAB, we used the provided evaluation set as the test set. The recordings in the development set were split 80-20% into a training and a validation set. Therefore, we used 2,171, 543 and 276 recordings in the training, validation and testing sets. For PC18, we used a 60-20-20% random split, meaning there were 595, 199 and 199 recordings in the training, validation and testing sets respectively. Finally, for MSD, we retained the 17 most corrupted recordings for the test set (Appendix D) and randomly split the remaining 81 recordings into training and validation sets (65 and 16 recordings, respectively). This was done to emulate a situation where training data is mostly clean, and strong channel corruption occurs unexpectedly at test time. We performed hyperparameter selection on each of the three datasets using a cross-validation strategy on the combined training and validation sets.

We repeated training on different training-validation splits (two for PC18, three for TUAB and MSD). Neural networks and random forests were trained three times per split on TUAB and MSD (two times on PC18) with different parameter initializations. Training ran for at most 40 epochs or until the validation loss stopped decreasing for a period of a least 7 epochs on TUAB and PC18 (a maximum of 150 epochs with a patience of 30 for MSD, given the smaller size of the dataset).

Finally, accuracy was used to evaluate model performance for pathology detection experiments, while balanced accuracy (bal), defined as the average per-class recall, was used for sleep staging due to important class imbalance (the N2 class is typically much more frequent than other classes).

3.4 Evaluation under conditions of noise

The impact of noise on downstream performance and on the predicted DSF filters was evaluated in three steps. First, we artificially corrupted the input EEG windows of TUAB and PC18 by using a similar process to our data augmentation strategy (Equation 6). We used the same values for η, σ and p, but used a single mask $\boldsymbol{\nu}$ per recording, such that the set of corrupted channels remained the same across a recording. Before corrupting, we subsampled a few EEG channels to recreate the sparse montage settings of TUAB (Fp1, Fp2, T3, T4, Fz, Cz) and PC18 (F3-M2, F4-M1, O1-M2, O2-M1). We then analyzed downstream performance under varying noise level conditions. Second, we ran experiments on real corrupted data (MSD) by training our models on the cleanest recordings and evaluating their performance on the noisiest recordings. Finally, we analyzed the distribution of DSF filter weights predicted by a subset of the trained models.

4 Results

4.1 Performance of existing methods degrades under channel corruption

How do standard EEG classification methods fare against channel corruption? If channels have a high probability of being corrupted at test time, can noise be compensated for by adding more channels? To answer these questions, we measured the performance of three baseline approaches (Riemannian geometry, handcrafted features and a "vanilla" net, *i.e.*, ShallowNet without attention) trained on a pathology detection task on three different montages as channels were artificially corrupted. Results are presented in Fig. 2.

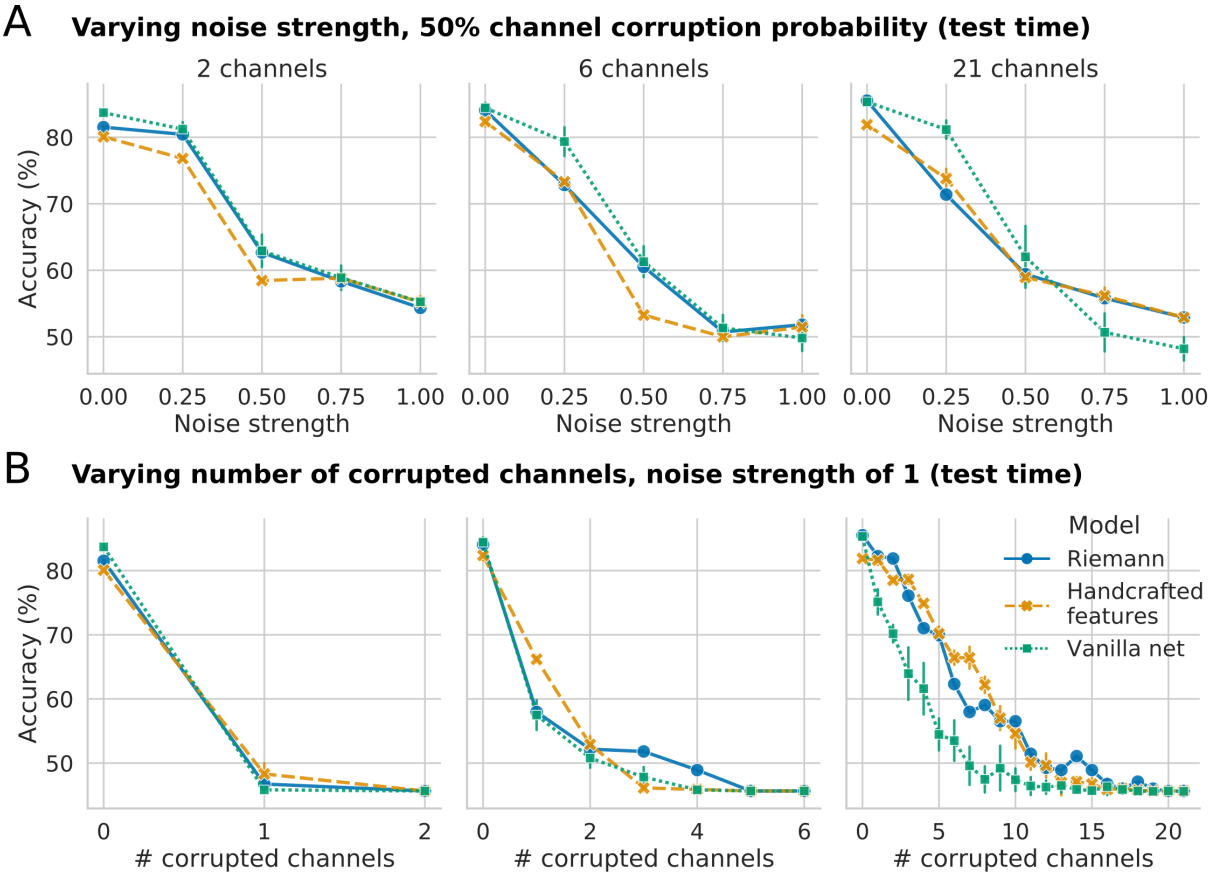

Figure 2: Impact of channel corruption on pathology detection performance of standard models. We trained a filter-bank Riemannian geometry pipeline (blue), a random forest on handcrafted features (orange) and a standard ShallowNet architecture (green) on the TUAB dataset, given montages of 2 (T3, T4), 6 (Fp1, Fp2, T3, T4, Fz, Cz) or 21 (all available) channels. Performance was then evaluated on artificially corrupted test data under two scenarios: (A) the η noise strength parameter was varied given a constant channel corruption probability of 50%, and (B) the number of corrupted channels was varied given a constant noise strength of 1. Error bars show the standard deviation over 3 models for handcrafted features and 6 models for neural networks. While traditional feature-based models fared slightly better than a vanilla neural network in some cases (bottom right), adding noise predictably degraded the performance of all three models.

All three baseline methods performed similarly and suffered considerable performance degradation as stronger noise was added (Fig. 2A) and as more channels were corrupted (Fig. 2B). First, under progressively noisier conditions, adding more channels did not generally improve performance. Strikingly, adding channels even hampered the ability of the models to handle noise. Indeed, the impact of noise was much less significant for 2-channel models than for 6- or 21-channel models. The vanilla net performed slightly better than the other methods in low noise conditions, however it was less robust to heavy noise when using 21 channels.

Second, when an increasing number of channels was corrupted (Fig. 2B), using denser montages did improve performance, although by a much smaller factor than what might be expected. For instance, losing one or two channels with the 21-channel models only yielded a minor decrease in performance, while models trained on sparser montages lost as much as 30% accuracy. However, even when as many as 15 channels were still available (*i.e.*, six corrupted channels), models trained on 21 channels performed worse than 2- or 6-channel models without any channel corruption, despite having access to much more spatial information on average. Interestingly, when models were trained on 21 channels, traditional feature-based methods were more robust to corruption than a vanilla net up to a certain point, however this did not hold for sparser montages.

These results suggest that standard approaches cannot handle significant channel corruption at a satisfactory level, even when denser montages are available. Therefore, better tools are necessary to train noise-robust models.

4.2 Attention and data augmentation mitigates performance loss under channel corruption

If including additional EEG channels does not by itself resolve performance degradation under channel corruption, what can be done to improve the robustness of standard EEG classification methods? We evaluated the performance of our models when combined with three denoising strategies (Section 3.2) for a fixed 6-channel montage[8]. Results on pathology detection (TUAB) are presented in Fig. 3.

Without denoising, all methods showed a steep performance decrease as noise became stronger (Fig. 3A) or more channels were corrupted (Fig. 3B). Automated noise handling (second column) reduced differences between methods when noise strength was increased (Fig. 3A), and helped marginally improve robustness when only one or two channels were corrupted (Fig. 3B). However, it is only with data augmentation that clear performance improvements could be obtained, allowing all methods to perform considerably better in the noisiest settings (third column). Performance of traditional baselines was degraded however in low noise conditions. Neural networks, in contrast, saw their performance increase the most across noise strengths and numbers of corrupted channels. Whereas their performance decreased by at least 34.6% when going from no noise to strongest noise with the other strategies, training neural networks with data augmentation reduced performance loss to 5.3-10.5% on average. The DSF models improved performance further still over the vanilla ShallowNet by yielding an improvement of *e.g.,* 1.8-7.5% across noise strengths. Finally, adding the matrix logarithm and the soft-thresholding nonlinearity (DSFm-st, in magenta) yielded marginal improvements over DSFd. Under strong noise corruption ($\eta = 1$) our best performing model (DSFm-st + data augmentation) yielded an

[8]This 6-channel montage (Fp1, Fp2, T3, T4, Fz, Cz) performed similarly to a 21-channel montage in no-corruption conditions (Fig. 2) while being more representative of the sparse montages likely to be found in mobile EEG devices.

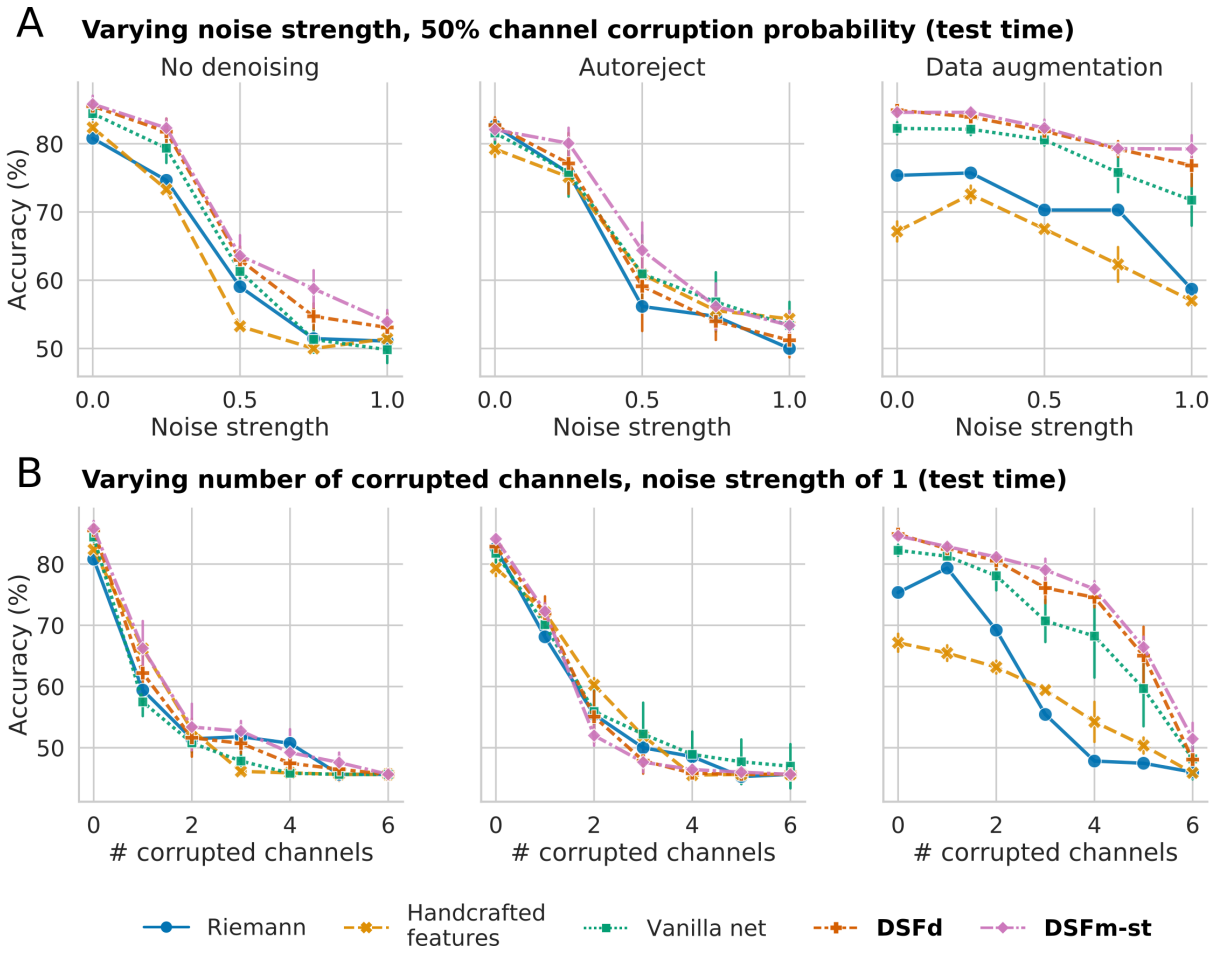

Figure 3: Impact of channel corruption on pathology detection performance for models coupled with (1) no denoising strategy, (2) Autoreject and (3) data augmentation. We compared the per recording accuracy on the TUAB evaluation set (6-channel montage) as (A) the η noise strength parameter was varied given a constant channel corruption probability of 50%, and (B) the number of corrupted channels was varied given a constant noise strength of 1. Error bars show the standard deviation over 3 models for handcrafted features and 6 models for neural networks. Using an automated noise handling method (Autoreject; second column) provided some improvement in noise robustness over using no denoising strategy at all (first column). Data augmentation benefited all methods, but deep learning approaches and in particular DSF (third column, in red and magenta) yielded the best performance under channel corruption.

accuracy improvement of 29.4% over the vanilla net without denoising. Overall, this suggests that learning end-to-end to both predict and handle channel corruption at the same time is key to successfully improving robustness.

Next, we repeated this analysis on a sleep staging task using the PC18 dataset (Fig. 4). As above, not using a denoising strategy led to a steep decrease in performance. Once more, Autoreject leveled out differences between the different methods and boosted performance under single-channel corruption, but otherwise did not improve or degrade performance as compared to training models without denoising. Data augmentation, in contrast, again helped improve the robustness of all methods. Interestingly, it benefited non-deep learning approaches more than in pathology detection, yielding for instance a similar performance for both handcrafted features and the vanilla StagerNet. DSF remained the most robust though with both DSFd and DSFm-st consistently outperforming all other methods. The performance of these two methods was highly similar, producing mostly overlapping lines (Fig. 4).

Finally, do these results hold under more intricate, naturally occurring corruption such as found in at-home settings? To verify this, we trained the same sleep staging models as above on the cleanest recordings of MSD (4-channel mobile EEG), and evaluated their performance on the 17 most corrupted recordings of the dataset. Results are presented in Fig. 5. As above, the Riemann approach did not perform well, while the handcrafted features approach was more competitive with the vanilla StagerNet without denoising. However, contrary to the above experiments, noise handling alone did not improve the performance of our models. Data augmentation was even detrimental to the Riemann and vanilla net models on average (see Fig. S3). Combined with dynamic spatial filters (DSFd and DSFm-st) though, data augmentation helped improve performance over other methods. For instance, DSFm-st with data augmentation yielded a median balanced accuracy of 65.0%, as compared to 58.4% for a vanilla network without denoising. Performance improvements were as high as 14.2% when looking at individual sessions. Importantly, all recordings saw an increase in performance, showing the ability of our proposed approach to improve robustness in noisy settings.

Taken together, our experiments on simulated and natural channel corruption indicate that a strategy combining an attention mechanism and data augmentation yields higher robustness than traditional baselines and existing automated noise handling methods.

4.3 Attention weights are interpretable and correlate with signal quality

Our experiments above demonstrated that DSF with data augmentation led to higher classification performance than "no denoising" and Autoreject baselines on both pathology detection and sleep staging tasks, under simulated and real-world channel corruption. Given the validated benefit of using DSF, can we explain the behavior of the module by inspecting its internal functioning? If so, in addition to improving robustness, DSF could also be used to monitor the effective importance of each incoming EEG channel, providing an interesting "free" insight into signal quality. To test this, we analyzed the effective channel importance ϕ_i of each EEG channel i to the spatial filters over the TUAB evaluation set. Results are shown in Fig. 6.

Overall, the attention weights behaved as expected: the more usable (*i.e.*, noise-free) a channel was, the higher its effective channel importance ϕ_i was relative to those of other channels. For instance, without any additional corruption, the DSF module focused most of its attention on channels T3 and T4 (Fig. 6A, first column), known to be highly relevant for pathology detection [43, 44]. However, when channel T3 was replaced with white noise, the DSF module reduced its attention to T3 and instead further increased its attention on other channels (second column).

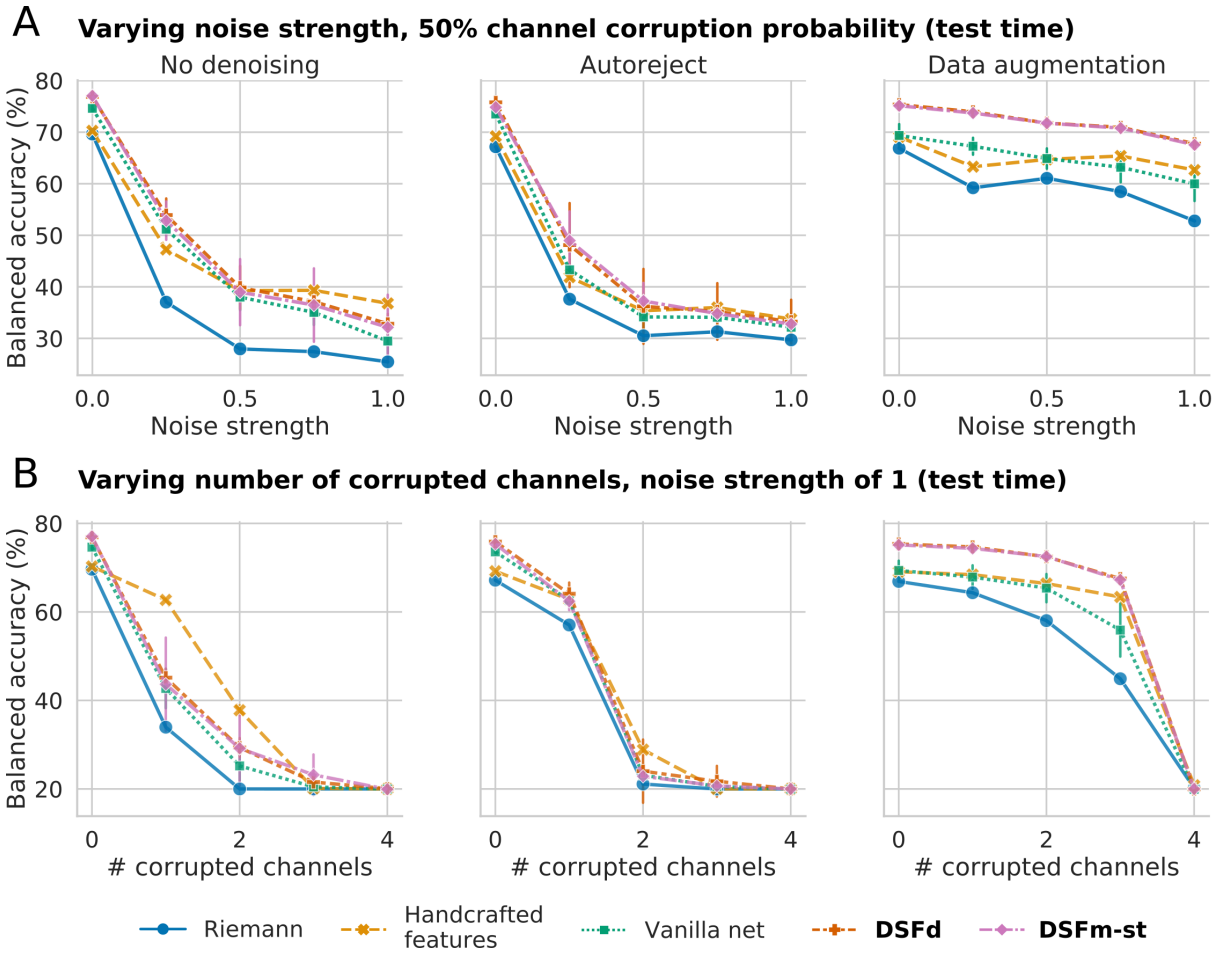

Figure 4: Impact of channel corruption on sleep staging performance for models coupled with (1) no denoising strategy, (2) Autoreject and (3) data augmentation. We compared the test balanced accuracy on PC18 (4-channel montage) as (A) the η noise strength parameter was varied given a constant channel corruption probability of 50%, and (B) the number of corrupted channels was varied given a constant noise strength of 1. Error bars show the standard deviation over 3 models for handcrafted features and 4 models for neural networks. Similarly to Fig. 3, automated noise handling provided a marginal improvement in noise robustness in some cases, data augmentation yielded a performance boost for all methods, while a combination of data augmentation and DSF (third column, red and magenta lines which overlap) led to the best performance under channel corruption.

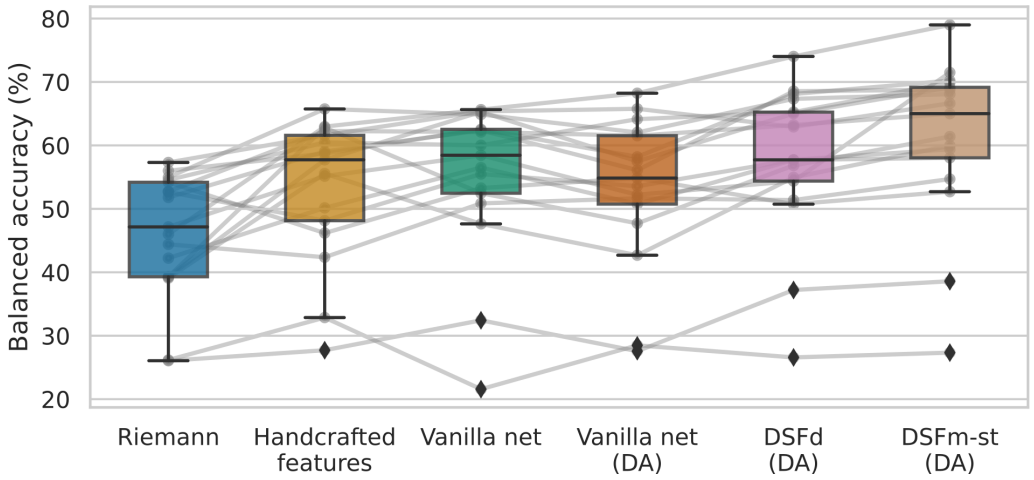

Figure 5: Recording-wise sleep staging results on MSD. Test balanced accuracy is presented for the Riemann, handcrafted features and vanilla net models without a denoising strategy, and for the vanilla net, DSFd and DSFm-st models with data augmentation (DA). Each point represents the average performance obtained by models with different random initializations (1, 3 and 9 initializations for Riemann, handcrafted features and deep learning models, respectively) on each recording from the test set of MSD. Lines represent individual recordings. The best performance was obtained by combining data augmentation with DSF with logm(cov) and soft-thresholding (DSFm-st).

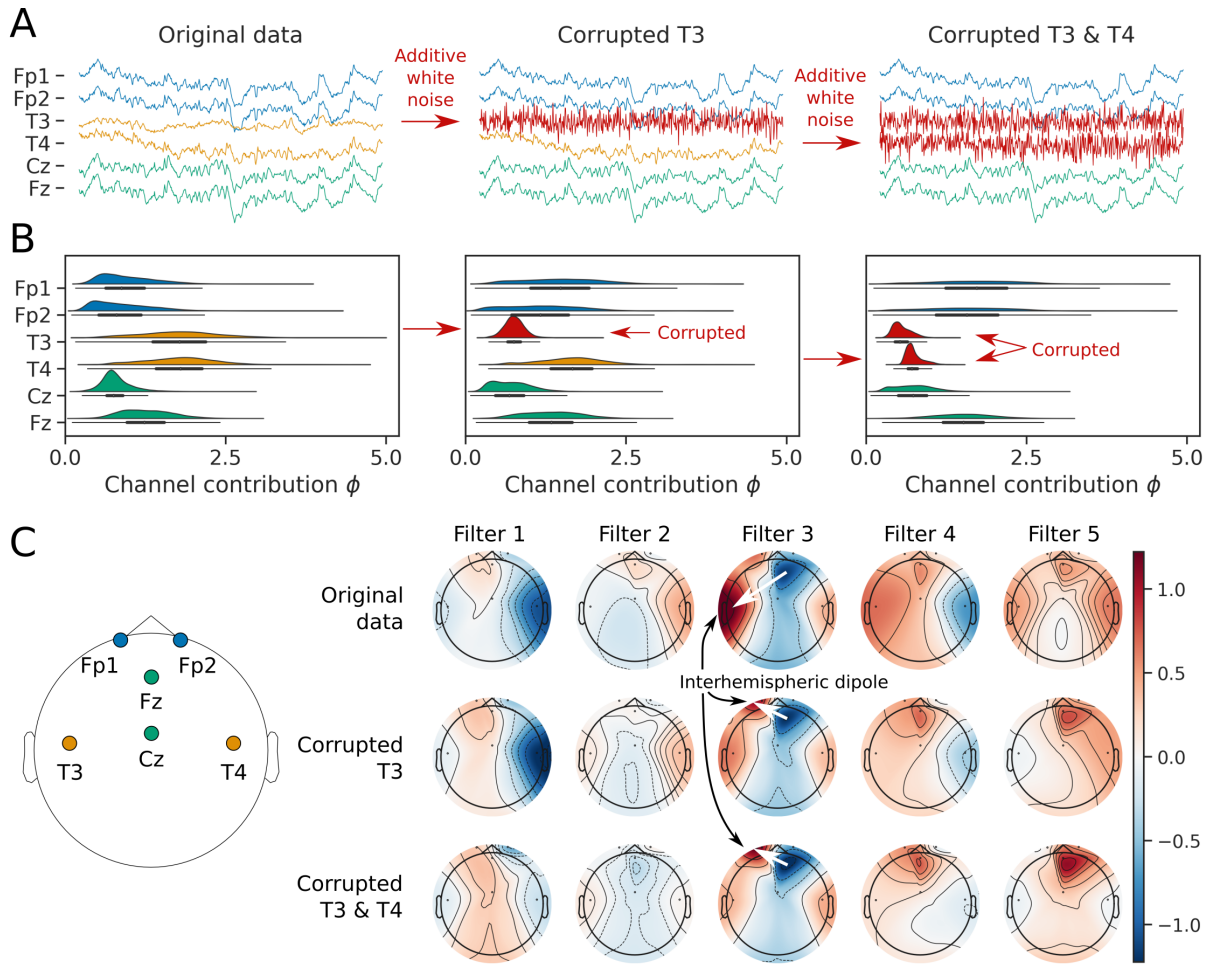

Figure 6: Effective channel importance and spatial filters predicted by the DSF module trained on pathology detection. We compared three scenarios on the TUAB evaluation set: no added corruption, only T3 is corrupted and both T3 and T4 are corrupted. (A) The corruption process was carried out by replacing a channel with white noise ($\sigma \sim \mathcal{U}(20, 50)$ μV), as illustrated with a single 6-s example window (first row). (B) The distribution of effective channel importance values ϕ is presented using density estimate and box plots. Corrupted channels are significantly down-weighted in the spatial filtering. (C) A subset of the spatial filters (median across all windows) are plotted as topomaps for the three scenarios. Corrupting T3 overall reduced the effective importance attributed to T3 and slightly boosted T4 values, while corrupting both T3 and T4 led to a reduction of ϕ for both channels, but to an increase for the other channels. This change was also reflected in the overall topography: dipole-like patterns (indicated by white arrows) were dynamically modified to focus on clean channels (*e.g.,* Filter 3).

Similarly, when both T3 and T4 were corrupted the module reduced its attention on both channels and leveraged the remaining channels instead, *i.e.*, mostly Fp1 and Fp2 (third column). Interestingly, this change is reflected by the topography of the predicted filters W_{DSF} (Fig. 6B): for instance, some dipolar filters computing a difference between left and right hemispheres were dynamically adapted to rely on Fp1 or Fp2 instead of T3 or T4 (*e.g.,* filters 1, 3 and 5). Intuitively, the network has learned to ignore corrupted data and to focus its attention on the good EEG channels, and to do so in a way that preserves the meaning of each virtual channel.

To further verify the interpretability of DSF's attention weights on naturally-corrupted real-world EEG data, we visualized the normalized effective channel importance metric alongside a time-frequency representation of the raw EEG in Fig. 7. As expected, the metric dropped to values close to zero when a channel suffered heavy corruption, e.g. Fp1 throughout the recording (left column) and TP9 intermittently (right column). These results again illustrate the capacity of DSF to ignore corrupted data, but also highlight its capacity to dynamically adapt to changing noise characteristics.

4.4 Deconstructing the DSF module

What might explain the capacity of the DSF module to improve robustness to channel corruption and provide interpretable attention weights? By comparing DSF to simpler interpolation-based methods, DSF can be understood as a more complex version of a simple attention-based model that decides how much each input EEG channel should be replaced by its interpolated version (details provided in Appendix F). With this connection in mind, we performed an ablation study to understand the importance of each additional mechanism leading to the formulation of the DSF module. Fig. 8 shows the performance of the different attention module variations trained on the pathology detection task with data augmentation, under different noise strengths.

Naive interpolation of each channel based on the $C-1$ others (orange) performed similarly to or worse than the vanilla ShallowNet model (blue) across noise strengths. Introducing a single attention weight (green) to control how much channels should be mixed with their interpolated version only improved performance for noise strengths above 0.5. Using one attention weight per channel (red) further improved performance, this time across all noise strengths. The addition of dynamic interpolation (magenta), in which both the attention weights and an interpolation matrix are generated based on the input EEG window, yielded an additional substantial performance boost. Relaxing the constraints on the interpolation matrix and adding a bias vector to obtain DSFd (brown) led to very similar performance. Finally, the addition of the soft-thresholding non-linearity and the use of the matrix logarithm of the covariance matrix (DSFm-st, pink) further yielded performance improvements.

Together, these results show that combining channel-specific interpolation and dynamic prediction of interpolation matrices is necessary to outperform simpler attention module formulations. Performance can be further improved by providing the full covariance matrix as input to the attention module and encouraging the model to produce 0-weights with a nonlinearity.

5 Discussion

We introduced Dynamic Spatial Filtering (DSF), a new method to handle channel corruption in EEG based on an attention mechanism architecture and a data augmentation transform. Plugged into a neural network whose input has a spatial dimension (*e.g.,* EEG channels), DSF predicts spatial filters that allow the model to dynamically focus on important channels and

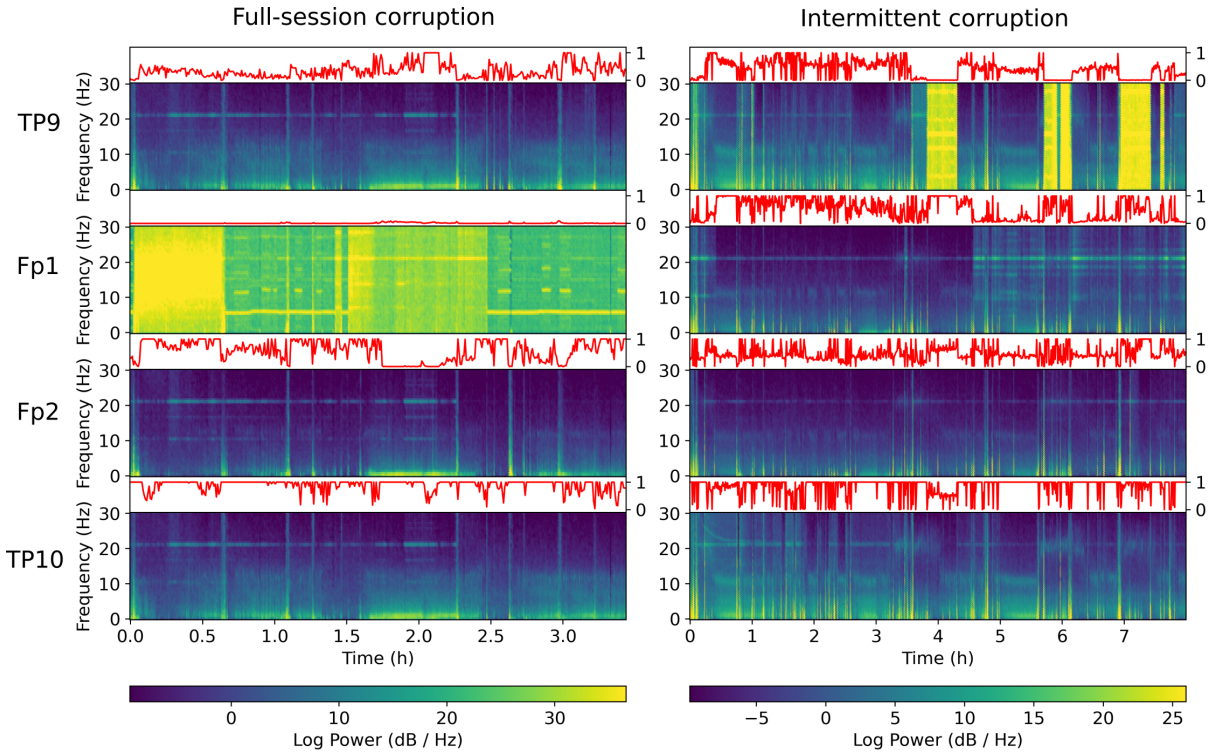

Figure 7: Normalized effective channel importance $\hat{\boldsymbol{\phi}}$ predicted by the DSF module on two MSD sessions with naturally-occurring channel corruption. Each column represents the log-spectrogram of the four EEG channels of one recording (Welch's periodogram on 30-s windows, using 2-s windows with 50% overlap). The red line above each spectrogram is the normalized effective channel importance $\hat{\phi}_i$ (see Eq. 5), between 0 and 1, computed using a DSFm-st model trained on MSD. When a channel is corrupted throughout the recording (left column, second row, as indicated by broad spectrum high power noise), DSF mostly "ignores" it by predicting small weights for that channel. This results in $\hat{\phi}_i$ values close to 0 for Fp1. When the corruption is intermittent (right column, first row), DSF dynamically adapts its spatial filters to only ignore important channels when they are corrupted. This is the case for channel TP9 around hours 4, 6, and 7, where $\hat{\phi}_i$ is again close to 0.

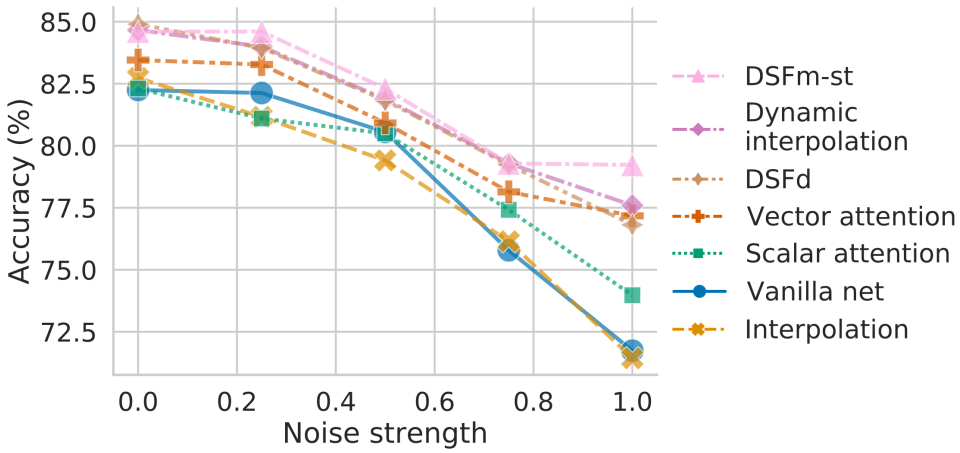

Figure 8: Performance of different attention module architectures on the TUAB evaluation set under increasing channel corruption noise strength. Each line represents the average of 6 models (2 random initializations, 3 random splits). Models that dynamically generate spatial filters, such as DSF, outperform simpler architectures across noise levels.

ignore corrupted ones. DSF shares links with interpolation-based methods traditionally used in EEG processing but in contrast does not require separate preprocessing steps that are often expensive with dense montages or poorly adapted to sparse ones. DSF outperformed feature-based approaches and automated denoising pipelines under simulated corruption on two large public datasets and in two different predictive tasks. Similar results were obtained on a smaller dataset of mobile sparse EEG with strong natural corruption, demonstrating the applicability of our approach to challenging at-home recording conditions. Finally, the inner functioning of DSF can easily be inspected using a simple measure of effective channel importance and topographical maps. Overall, DSF is computationally lightweight, easy to implement, and improves robustness to channel corruption in sparse EEG settings.

5.1 Handling EEG channel loss with existing denoising strategies

As opposed to the more general problem of "noise handling" (Table 1), we focused our experiments on the problem of channel corruption in sparse montages. In light of our results, we explain why existing strategies are not well suited for handling channel corruption, while DSF is.

Our first experiment (Section 4.1) demonstrated that adding more EEG channels does not necessarily make a classifier more robust to channel loss. In fact, we observed the opposite: a model trained on two channels can outperform 6- and 21-channel models under heavy channel corruption (Fig. 2A). This can be explained by two phenomena. First, increasing the number of channels increases the input dimensionality of classifiers, making them more likely to overfit the training data. Tuning regularization hyperparameters can help with this, but does not solve the problem by itself. Second, in vanilla neural networks, the weights of the first spatial convolution layer, *i.e.*, the spatial filters applied to the input EEG, are fixed. If one of the spatial filter relies mostly on one specific (theoretically) important input channel, *e.g.,* T3, and this input channel is corrupted, all successive operations on the resulting virtual channel will carry noise as well. This highlights the importance of dynamic reweighting: with DSF, we can find alternative spatial filters when a theoretically important channel is corrupted, and even completely ignore a

corrupted channel if it contains no useful information.

Since adding channels is not on its own a solution, can traditional EEG denoising techniques help handle the channel corruption problem? A seemingly simple approach would be to use a fixed threshold on a relevant descriptor of signal quality (*e.g.,* amplitude, variance or spectral slope) to identify bad channels window-by-window. While this approach may appear straightforward, it requires making non-trivial choices: Which descriptor should we use? How should we select threshold values? How do we handle bad channels once they have been identified? Moreover, this approach is likely to perform suboptimally as different EEG hardware, channel and reference positions, preprocessing steps and recording conditions, especially in out-of-the-lab settings, all have an impact on the power and morphology of the signals. As a result, fixed threshold values will work well in some cases, but fail to catch actual noise (or be too strict) in others.

Instead, it would make sense to adapt thresholds in a data-driven manner. This is the basis for Autoreject [33] which selects amplitude thresholds using a cross-validation procedure and interpolates bad channels using head geometry. In our experiments, automated denoising did help but only marginally (middle column of Fig. 3 and 4). The relative ineffectiveness of this approach can be explained by the very low number of available channels in our experiments (4 or 6) which likely harmed the quality of the interpolation. Our results therefore do not invalidate the use of interpolation-based methods (whose performance has been demonstrated multiple times on denser montages and in challenging noise conditions [31, 32, 33]) but only expose their limitations when working with few channels. Still, there are other reasons why interpolation-based methods might not be optimal in settings like the ones studied in this paper. For instance, completely replacing a noisy channel by its interpolated version means that any remaining usable information in this channel will be discarded and that any noise contained in the other (non-discarded) channels will end up in the interpolated channel.

Finally, an interesting case to consider is when tasks can be performed accurately with a single good channel, *e.g.,* sleep staging [91]. In such a case, could a single-channel model perform as well as a multi-channel model, without the need to worry about the challenges discussed above? While this may be true if we have access to a reliably good channel, as soon as it is corrupted (*e.g.,* in real-world mobile EEG settings) it can no longer be used by the model. An ensemble of single-channel models might be an interesting solution; however this requires knowing both which channel to focus on and when, which is not trivial and requires additional logic and processing pipeline components. Moreover, to improve upon such a model by making use of spatial information [14] the model should be trained on all possible combinations of good channels, which can quickly become prohibitive. DSF offers a compelling solution to the challenges encountered with single-channel models thanks to its end-to-end dynamic reweighting capabilities.

5.2 Impact of the input spatial representation

The representation used by the DSF module constrains the types of patterns that can be leveraged to produce spatial filters. For instance, using the log-variance of each channel allows detecting large-amplitude corruption or artifacts, however this makes the DSF model blind to more subtle kinds of interactions between channels. These interactions can be very informative in certain cases, *e.g.,* when one channel is corrupted by a noise source which also affects other channels but to a lesser degree.

Our experiments suggested that models based on log-variance (DSFd) or vectorized covariance matrices (DSFm-st) were roughly equivalent in simulated noise conditions (Fig. 3-4). This is

likely because the additive white noise we used was not spatially correlated and therefore no spatial interactions could be leveraged by the DSF modules to identify noise. On naturally corrupted data however, using the full spatial information along with soft-thresholding was critical to outperforming other methods (Fig. 5). This is likely because the noise in at-home recordings was often correlated spatially and because corrupted channels, often containing mostly noise (Appendix D), could be completely ignored by DSF.

Related attention block architectures have used average-pooling [10] or a combination of average- and max-pooling [11] to summarize channels. Intuitively, average pooling should not yield a useful representation of the input, as EEG channels are often assumed to have zero-mean, or are explicitly highpass filtered to remove their DC offset. Max-pooling, on the other hand, does capture amplitude information that overlaps with second-order statistics, however it does not allow differentiating between large transient artifacts and more temporally consistent corruption. Experiments on TUAB (not shown) confirmed this: a combination of min- and max-pooling was less robust to noise than covariance-based models. From this perspective, vectorized covariance matrices or similar representations (Appendix A) are an ideal choice of spatial representation. Ultimately, DSF could be fed with any learned representations with a spatial dimension, *e.g.,* filter-bank representations.

5.3 Impact of the data augmentation transform

Data augmentation was critical to developing invariance to corruption (Section 4.2). For instance, under simulated corruption, a vanilla neural network trained with our data augmentation transform gained considerable robustness, even without an attention mechanism. Does this mean that data augmentation is the key ingredient to DSF? In fact, our results on naturally corrupted data (Fig. 5) showed that data augmentation without attention negatively impacted performance and that adding an attention mechanism was necessary to improve performance. Moreover, traditional pipelines generally did not benefit from data augmentation as much as neural networks did, and even saw their performance degrade considerably in certain cases, *e.g.,* in low noise conditions in pathology detection experiments and on the real-world data for the Riemann models.

Nonetheless, these results highlight the role of data augmentation transforms in developing robust representations of EEG. Recently, work in self-supervised learning for EEG [79, 92, 93] has further suggested the importance of well-characterized data augmentation transforms for representation learning. Importantly though, the motivation behind the use of data augmentation in our experiments was not primarily to reduce overfitting due to limited sample sizes like commonly done in deep learning, but rather to evaluate methods under controlled corruption of experimental data. Ultimately, our additive white noise transform could be combined with channel masking and shuffling [94] and other potential corruption processes such as those described in [92, 93].

5.4 Interpreting dynamic spatial filters to measure effective channel importance

The results in Fig. 6 demonstrated that visualizing the spatial filters produced by the DSF module can reveal the spatial patterns a model has learned to focus on (Section 4.3).

As observed in our experiments, a higher ϕ indicates higher effective importance of a channel for the downstream task. For instance, temporal channels were given a higher importance in the

pathology detection task, which is consistent with previous work [43, 44]. Similarly, in real-world data, low $\boldsymbol{\phi}$ values were given to a channel whenever it was corrupted (Fig. 7).

However, $\boldsymbol{\phi}$ is not a strict measure of signal quality but more of channel usefulness: there could be different reasons behind the boosting or attenuation of a channel by the DSF module. Naturally, if a channel is particularly noisy, its contribution might be brought down to zero to avoid contaminating virtual channels with noise. Conversely though, if the noise source behind a corrupted channel is also found (but to a lesser degree) in other channels, the corrupted channel could also be used to regress out noise and recover clean signals [95]. In other words, $\boldsymbol{\phi}$ reflects the importance of a channel conditionally to others.

Finally, using DSF to obtain a measure of channel usefulness actually opens the door to DSF being used in non-machine learning settings. For instance, once a neural network is trained with DSF, its effective channel importance values can be reused as an indicator of signal quality on similar data (*e.g.,* data collected with the same or similar hardware). Such a signal quality metric can be helpful during data collection, or to know which parts of the recording should be kept for analysis.

5.5 Practical considerations

When faced with channel corruption in a predictive task, which modelling and denoising strategies should be preferred? This choice should depend on the number of available channels, as well as on assumptions about the stationarity of the noise. When using sparse montages, as in this paper, different solutions can lead to good results. For instance, handcrafted features with random forests can perform well when spatial information is not critical (*e.g.,* sleep staging, Section 4.2) or noise is stationary [25], although they require a non-trivial feature engineering step. However, when less can be assumed about the predictive task, *e.g.,* corruption might be non-stationary or spatial information is likely important, DSF with data augmentation is an effective way to make a neural network noise-robust. Although we did not test denoising approaches on dense montages, we can expect different methods to work well in these settings. For instance, under stationary noise, Riemmanian geometry-based approaches were shown to be robust to the lack of preprocessing in MEG data [21]. If, on the other hand, noise is not stationary and the computational resources allow it, interpolation-based methods might be used to impute missing channels before applying a predictive model (*e.g.,* [33]). In cases where introducing a separate preprocessing step is not desirable, DSF with data augmentation might again be a promising end-to-end solution.[9]

5.6 Related work

Deep learning and noise robustness for audio data Noise robustness is of particular interest to the speech recognition community. For example, "noise-aware training" was proposed to train deep neural networks on noisy one-channel speech signals by providing an estimate of the noise level as input to the network [96]. Noise-invariant representations of speech signals were also developed by training a classifier to perform well on the speech recognition task but badly on signal quality classification [97] or by penalizing the distance between the internal representations of clean and noisy signals [98, 99]. Methods have also been designed to leverage the spatial information of multiple audio channels similarly to our proposed DSF approach. Deep

[9] In this case, the number of parameters of the module can be controlled by *e.g.,* selecting log-variance as the input representation or reducing dimensionality by using fewer spatial filters than there are input channels.

beamforming networks were used to dynamically reweight different audio channels to improve robustness to noise, for instance with filter prediction subnetworks [100, 101, 102]. In a fashion similar to ours, recent work also used spatial attention to reweight beamformed input speech signals to decide which filters to focus on [103].

Attention mechanisms for EEG processing Recent efforts in the deep learning and EEG community have led to various applications of attention mechanisms to end-to-end EEG processing. First, some studies used attention to improve performance on a specific task by focusing on different dimensions of an EEG representation. For instance, natural language processing-inspired attention modules were used in sleep staging architectures to improve processing of temporal dependencies [47, 104, 46, 48, 105]. Attention was also applied in the spatial dimension to dynamically combine information from different EEG channels [106, 107] or even from heterogeneous channel types [104]. In one case, spatial and temporal attention were used simultaneously in a BCI classification task [108]. Second, attention mechanisms have been used to enable transfer learning between different datasets with possibly different montages. In [109], two parallel attention mechanisms allowed a neural network to focus on the channels and windows that were the most transferable between two datasets. Combined with an adversarial loss, this approach improved domain adaptation performance on a cross-dataset sleep staging task. Similarly to DSF, a spatial attention block was used in [105] to recombine input channels into a fixed number of virtual channels and allow models to be transferred to different montages. A Transformer-like spatial attention module was also proposed to dynamically re-order input channels [94]. In contrast to DSF, though, these approaches used attention weights in the $[0, 1]$ range, breaking the conceptual connection between channel recombination and spatial filtering.

5.7 Limitations

Our experiments on sleep data focused on window-wise decoding, *i.e.,* we did not aggregate larger temporal context but directly mapped each window to a prediction. However, modeling these longer-scale temporal dependencies was recently shown to help sleep staging performance significantly [45, 14, 47, 104, 46, 48, 105]. Despite a slight performance decrease, window-wise decoding offered a simple but realistic setting to test robustness to channel corruption, while limiting the number of hyperparameters and the computational cost of the experiments. In practice, the effect of data corruption by far exceeded the drop in performance caused by using slightly simpler architectures.

The data augmentation and the noise corruption strategies exploited in this work employ additive Gaussian white noise. While this approach helped develop noise robust models, spatially non-correlated additive white noise represents an "adversarial scenario". Indeed, under strong white noise, the information in higher frequencies is more likely to be lost than with *e.g.,* pink or brown noise. Additionally, the absence of spatial noise correlation means that spatial filtering can less easily leverage multi-channel signals to regress out noise (Section 5.4). Exploring more varied and realistic types of channel corruption could further help clarify the ability of DSF to work under different conditions. Despite this, our experiments on naturally corrupted sleep data showed that additive white noise as a data augmentation does help improve noise robustness.

Finally, we focused our empirical study of channel corruption on two clinical problems that are prime contenders for mobile EEG applications: pathology screening and sleep monitoring. Interestingly, these two tasks have been shown to work well even with limited spatial information (*i.e.*, single-channel sleep staging [91]) or to be highly correlated with simpler spectral power

representations [43]. Therefore, future work will be required to validate the use of DSF on tasks where fine-grained spatial patterns might be critical to successful prediction, *e.g.,* brain age estimation [110]. Other common EEG-based prediction tasks such as seizure detection might benefit from DSF and will require further validation.

6 Conclusion

We presented Dynamic Spatial Filtering (DSF), an attention mechanism architecture that improves robustness to channel corruption in EEG prediction tasks. Combined with a data augmentation transform, DSF outperformed other noise handling procedures under simulated and real channel corruption on three datasets. Moreover, DSF enables efficient end-to-end handling of channel corruption, works with few channels, is interpretable and does not require expensive preprocessing. We hope that our method can be a useful tool to improve the reliability of EEG processing in challenging non-traditional settings such as user-administered, at-home recordings.

Acknowledgements

This work was supported by Mitacs (project number IT14765) and InteraXon Inc. (graduate funding support) for HB and by the BrAIN (ANR-20-CHIA-0016) and AI-cog (ANR-20-IADJ-0002) ANR grants for AG and DAE.

This work was performed using HPC resources from GENCI-IDRIS.

We would like to thank the Python [111, 112] community for developing many of the tools used in the making of this article: NumPy [113], SciPy [114], matplotlib [115], seaborn [116], pandas [117], scikit-learn [69], MNE-Python [65], PyTorch [66], hydra [118], pyRiemann [119] and braindecode [12].

Declarations of interest

HB receives graduate funding support from InteraXon Inc. SUNW and CA are employees of InteraXon Inc.

References

[1] Vojkan Mihajlović, Bernard Grundlehner, Ruud Vullers, and Julien Penders. Wearable, wireless EEG solutions in daily life applications: what are we missing? *IEEE Journal of Biomedical and Health Informatics*, 19(1):6–21, 2014.

[2] Kiret Dhindsa. Filter-bank artifact rejection: High performance real-time single-channel artifact detection for EEG. *Biomedical Signal Processing and Control*, 38:224–235, 2017.

[3] Matthias Kreuzer. EEG based monitoring of general anesthesia: taking the next steps. *Frontiers in Computational Neuroscience*, 11:56, 2017.

[4] Kristina T Johnson and Rosalind W Picard. Advancing neuroscience through wearable devices. *Neuron*, 108(1):8–12, 2020.

[5] Matthias R Hohmann, Lisa Konieczny, Michelle Hackl, Brian Wirth, Talha Zaman, Raffi Enficiaud, Moritz Grosse-Wentrup, and Bernhard Schölkopf. MYND: Unsupervised evaluation of novel BCI control strategies on consumer hardware. In *Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology*, pages 1071–1084, 2020.

[6] Olave E Krigolson, Mathew R Hammerstrom, Wande Abimbola, Robert Trska, Bruce W Wright, Kent G Hecker, and Gordon Binsted. Using Muse: Rapid mobile assessment of brain performance. *Frontiers in Neuroscience*, 15, 2021.

[7] Fabien Lotte, Marco Congedo, Anatole Lécuyer, Fabrice Lamarche, and Bruno Arnaldi. A review of classification algorithms for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, 4(2):R1, 2007.

[8] 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.

[9] Riitta Hari and Aina Puce. *MEG-EEG Primer*. Oxford University Press, 2017.

[10] Jie Hu, Li Shen, and Gang Sun. Squeeze-and-excitation networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 7132–7141, 2018.

[11] Sanghyun Woo, Jongchan Park, Joon-Young Lee, and In So Kweon. CBAM: Convolutional block attention module. In *Proceedings of the European conference on computer vision (ECCV)*, pages 3–19, 2018.

[12] 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*, aug 2017.

[13] Vernon J Lawhern, Amelia J Solon, Nicholas R Waytowich, Stephen M Gordon, Chou P Hung, and Brent J Lance. EEGNet: a compact convolutional neural network for EEG-based brain–computer interfaces. *Journal of Neural Engineering*, 15(5):056013, 2018.

[14] Stanislas Chambon, Mathieu N Galtier, Pierrick J 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.

[15] Ran Manor and Amir B Geva. Convolutional neural network for multi-category rapid serial visual presentation BCI. *Frontiers in Computational Neuroscience*, 9:146, 2015.

[16] Ryan Hefron, Brett Borghetti, Christine Schubert Kabban, James Christensen, and Justin Estepp. Cross-participant EEG-based assessment of cognitive workload using multi-path convolutional recurrent neural networks. *Sensors*, 18(5):1339, 2018.

[17] 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 *International Conference on Multimedia Modeling*, pages 82–93. Springer, 2018.

[18] Pouya Bashivan, Irina Rish, Mohammed Yeasin, and Noel Codella. Learning representations from EEG with deep recurrent-convolutional neural networks. *arXiv preprint arXiv:1511.06448*, 2015.

[19] Pierre Thodoroff, Joelle Pineau, and Andrew Lim. Learning robust features using deep learning for automatic seizure detection. In *Machine learning for Healthcare Conference*, pages 178–190, 2016.

[20] Juan Lorenzo Hagad, Kenichi Fukui, and Masayuki Numao. Deep visual models for EEG of mindfulness meditation in a workplace setting. In *International Workshop on Health Intelligence*, pages 129–137. Springer, 2019.

[21] David Sabbagh, Pierre Ablin, Gaël Varoquaux, Alexandre Gramfort, and Denis A Engemann. Predictive regression modeling with MEG/EEG: from source power to signals and cognitive states. *NeuroImage*, page 116893, 2020.

[22] Michael M. Bronstein, Joan Bruna, Yann LeCun, Arthur Szlam, and Pierre Vandergheynst. Geometric deep learning: Going beyond euclidean data. *IEEE Signal Processing Magazine*, 34(4):18–42, 2017.

[23] Junhua Li, Zbigniew Struzik, Liqing Zhang, and Andrzej Cichocki. Feature learning from incomplete EEG with denoising autoencoder. *Neurocomputing*, 165:23–31, 2015.

[24] Yaqi Chu, Xingang Zhao, Yijun Zou, Weiliang Xu, Jianda Han, and Yiwen Zhao. A decoding scheme for incomplete motor imagery EEG with deep belief network. *Frontiers in Neuroscience*, 12:680, 2018.

[25] Denis A Engemann, Federico Raimondo, Jean-Rémi King, Benjamin Rohaut, Gilles Louppe, Frédéric Faugeras, Jitka Annen, Helena Cassol, Olivia Gosseries, Diego Fernandez-Slezak, et al. Robust EEG-based cross-site and cross-protocol classification of states of consciousness. *Brain*, 141(11):3179–3192, 2018.

[26] Tzyy-Ping Jung, Colin Humphries, Te-Won Lee, Scott Makeig, Martin J McKeown, Vicente Iragui, and Terrence J Sejnowski. Extended ICA removes artifacts from electroencephalographic recordings. In *Advances in Neural Information Processing Systems*, pages 894–900, 1998.

[27] Nadia Mammone, Fabio La Foresta, and Francesco Carlo Morabito. Automatic artifact rejection from multichannel scalp EEG by wavelet ICA. *IEEE Sensors Journal*, 12(3):533–542, 2011.

[28] Irene Winkler, Stefan Haufe, and Michael Tangermann. Automatic classification of artifactual ICA-components for artifact removal in EEG signals. *Behavioral and Brain Functions*, 7(1):30, 2011.

[29] Mikko A. Uusitalo and Risto J. Ilmoniemi. Signal-space projection method for separating MEG or EEG into components. *Medical & Biological Engineering & Computing*, 35(2):135–140, 1997.

[30] Christian Andreas Edgar Kothe and Tzyy-Ping Jung. Artifact removal techniques with signal reconstruction, April 28 2016. US Patent App. 14/895,440.

[31] Hugh Nolan, Robert Whelan, and Richard B Reilly. FASTER: fully automated statistical thresholding for EEG artifact rejection. *Journal of Neuroscience Methods*, 192(1):152–162, 2010.

[32] Nima Bigdely-Shamlo, Tim Mullen, Christian Kothe, Kyung-Min Su, and Kay A Robbins. The PREP pipeline: standardized preprocessing for large-scale EEG analysis. *Frontiers in Neuroinformatics*, 9:16, 2015.

[33] Mainak Jas, Denis A Engemann, Yousra Bekhti, Federico Raimondo, and Alexandre Gramfort. Autoreject: Automated artifact rejection for MEG and EEG data. *NeuroImage*, 159:417–429, 2017.

[34] Richard B Berry, Rita Brooks, Charlene E Gamaldo, Susan M Harding, Carole L Marcus, Bradley V Vaughn, et al. The AASM manual for the scoring of sleep and associated events. *Rules, Terminology and Technical Specifications, American Academy of Sleep Medicine*, 176, 2012.

[35] Ivana Rosenzweig, András Fogarasi, Birger Johnsen, Jørgen Alving, Martin Ejler Fabricius, Michael Scherg, Miri Y Neufeld, Ronit Pressler, Troels W Kjaer, Walter van Emde Boas, et al. Beyond the double banana: improved recognition of temporal lobe seizures in long-term EEG. *Journal of Clinical Neurophysiology*, 31(1):1–9, 2014.

[36] Isaac A Corley and Yufei Huang. Deep EEG super-resolution: Upsampling EEG spatial resolution with generative adversarial networks. In *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pages 100–103. IEEE, 2018.

[37] Mats Svantesson, Hakan Olausson, Anders Eklund, and Magnus Thordstein. Virtual EEG-electrodes: Convolutional neural networks as a method for upsampling or restoring channels. *bioRxiv*, 2020.

[38] Avijit Paul. Prediction of missing EEG channel waveform using LSTM. In *2020 4th International Conference on Computational Intelligence and Networks (CINE)*, pages 1–6. IEEE, 2020.

[39] Heba El-Fiqi, Kathryn Kasmarik, Anastasios Bezerianos, Kay Chen Tan, and Hussein A Abbass. Gate-layer autoencoders with application to incomplete EEG signal recovery. In *2019 International Joint Conference on Neural Networks (IJCNN)*, pages 1–8. IEEE, 2019.

[40] AG Ramakrishnan and JV Satyanarayana. Reconstruction of EEG from limited channel acquisition using estimated signal correlation. *Biomedical Signal Processing and Control*, 27:164–173, 2016.

[41] Jordi Sole-Casals, Cesar Federico Caiafa, Qibin Zhao, and Adrzej Cichocki. Brain-computer interface with corrupted EEG data: a tensor completion approach. *Cognitive Computation*, 10(6):1062–1074, 2018.

[42] Yang Li, Xian-Rui Zhang, Bin Zhang, Meng-Ying Lei, Wei-Gang Cui, and Yu-Zhu Guo. A channel-projection mixed-scale convolutional neural network for motor imagery EEG decoding. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(6):1170–1180, 2019.

[43] R. Schirrmeister, L. Gemein, K. Eggensperger, F. Hutter, and T. Ball. Deep learning with convolutional neural networks for decoding and visualization of EEG pathology. In *2017 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–7, 2017.

[44] Lukas AW Gemein, Robin T Schirrmeister, Patryk Chrabaszcz, Daniel Wilson, Joschka Boedecker, Andreas Schulze-Bonhage, Frank Hutter, and Tonio Ball. Machine-learning-based diagnostics of EEG pathology. *NeuroImage*, page 117021, 2020.

[45] Akara Supratak, Hao Dong, Chao Wu, and Yike Guo. DeepSleepNet: A model for automatic sleep stage scoring based on raw single-channel EEG. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 25(11):1998–2008, 2017.

[46] Antoine Guillot, Fabien Sauvet, Emmanuel H During, and Valentin Thorey. Dreem open datasets: Multi-scored sleep datasets to compare human and automated sleep staging. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 28(9):1955–1965, 2020.

[47] Huy Phan, Fernando Andreotti, Navin Cooray, Oliver Y Chén, and Maarten De Vos. SeqSleepNet: end-to-end hierarchical recurrent neural network for sequence-to-sequence automatic sleep staging. *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 27(3):400–410, 2019.

[48] Huy Phan, Oliver Y Chén, Philipp Koch, Alfred Mertins, and Maarten De Vos. XSleepNet: Multi-view sequential model for automatic sleep staging. *arXiv preprint arXiv:2007.05492*, 2020.

[49] Samu Taulu, Matti Kajola, and Juha Simola. Suppression of interference and artifacts by the signal space separation method. *Brain Topography*, 16(4):269–275, 2004.

[50] Sangjun Han, Moonyoung Kwon, Sunghan Lee, and Sung Chan Jun. Feasibility study of EEG super-resolution using deep convolutional networks. In *2018 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, pages 1033–1038. IEEE, 2018.

[51] Moonyoung Kwon, Sangjun Han, Kiwoong Kim, and Sung Chan Jun. Super-resolution for improving EEG spatial resolution using deep convolutional neural network—feasibility study. *Sensors*, 19(23):5317, 2019.

[52] Scott Makeig, Anthony J Bell, Tzyy-Ping Jung, Terrence J Sejnowski, et al. Independent component analysis of electroencephalographic data. *Advances in Neural Information Processing Systems*, pages 145–151, 1996.

[53] Dennis J McFarland, Lynn M McCane, Stephen V David, and Jonathan R Wolpaw. Spatial filter selection for EEG-based communication. *Electroencephalography and Clinical Neurophysiology*, 103(3):386–394, 1997.

[54] Lucas C Parra, Clay D Spence, Adam D Gerson, and Paul Sajda. Recipes for the linear analysis of EEG. *NeuroImage*, 28(2):326–341, 2005.

[55] Benjamin Blankertz, Ryota Tomioka, Steven Lemm, Motoaki Kawanabe, and Klaus-Robert Muller. Optimizing spatial filters for robust EEG single-trial analysis. *IEEE Signal Processing Magazine*, 25(1):41–56, 2007.

[56] Alain de Cheveigné and Jonathan Z Simon. Denoising based on spatial filtering. *Journal of Neuroscience Methods*, 171(2):331–339, 2008.

[57] Fabien Lotte and Cuntai Guan. Regularizing common spatial patterns to improve BCI designs: unified theory and new algorithms. *IEEE Transactions on Biomedical Engineering*, 58(2):355–362, 2010.

[58] Vadim V Nikulin, Guido Nolte, and Gabriel Curio. A novel method for reliable and fast extraction of neuronal EEG/MEG oscillations on the basis of spatio-spectral decomposition. *NeuroImage*, 55(4):1528–1535, 2011.

[59] François Perrin, J Pernier, O Bertrand, and JF Echallier. Spherical splines for scalp potential and current density mapping. *Electroencephalography and Clinical Neurophysiology*, 72(2):184–187, 1989.

[60] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 5998–6008, 2017.

[61] Leo Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct 2001.

[62] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

[63] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *ICCV*, pages 1026–1034, 2015.

[64] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: a simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958, 2014.

[65] Alexandre Gramfort, Martin Luessi, Eric Larson, Denis A Engemann, Daniel Strohmeier, Christian Brodbeck, Lauri Parkkonen, and Matti S Hämäläinen. MNE software for processing MEG and EEG data. *NeuroImage*, 86:446–460, 2014.

[66] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *arXiv preprint arXiv:1912.01703*, 2019.

[67] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Classification of covariance matrices using a Riemannian-based kernel for BCI applications. *Neurocomputing*, 112:172 – 178, 2013. Advances in artificial neural networks, machine learning, and computational intelligence.

[68] J-B Schiratti, Jean-Eudes Le Douget, Michel Le van Quyen, Slim Essid, and Alexandre Gramfort. An ensemble learning approach to detect epileptic seizures from long intracranial EEG recordings. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 856–860. IEEE, 2018.

[69] Fabian Pedregosa, Gaël Varoquaux, Alexandre Gramfort, Vincent Michel, Bertrand Thirion, Olivier Grisel, Mathieu Blondel, Peter Prettenhofer, Ron Weiss, Vincent Dubourg, et al. Scikit-learn: Machine learning in python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[70] Christina Jayne Bathgate and Jack D Edinger. Diagnostic criteria and assessment of sleep disorders. In *Handbook of Sleep Disorders in Medical Conditions*, pages 3–25. Elsevier, 2019.

[71] Shayan Motamedi-Fakhr, Mohamed Moshrefi-Torbati, Martyn Hill, Catherine M. Hill, and Paul R. White. Signal processing techniques applied to human sleep EEG signals—a review. *Biomedical Signal Processing and Control*, 10:21 – 33, 2014.

[72] S Æ Jónsson, E Gunnlaugsson, E Finssonn, DL Loftsdóttir, GH Ólafsdóttir, H Helgadóttir, and JS Ágústsson. 0447 ResTNet: A robust end-to-end deep learning approach to sleep staging of self applied somnography studies. *Sleep*, 43(Supplement_1):A171–A171, 2020.

[73] SJM Smith. EEG in the diagnosis, classification, and management of patients with epilepsy. *Journal of Neurology, Neurosurgery & Psychiatry*, 76(suppl 2):ii2–ii7, 2005.

[74] Christina Micanovic and Suvankar Pal. The diagnostic utility of EEG in early-onset dementia: a systematic review of the literature with narrative analysis. *Journal of Neural Transmission*, 121(1):59–69, 2014.

[75] Iyad Obeid and Joseph Picone. The temple university hospital EEG data corpus. *Frontiers in Neuroscience*, 10:196, 2016.

[76] S Lopez, G Suarez, D Jungreis, I Obeid, and J Picone. Automated identification of abnormal adult EEGs. In *2015 IEEE Signal Processing in Medicine and Biology Symposium (SPMB)*, pages 1–5. IEEE, 2015.

[77] Marco Congedo, Alexandre Barachant, and Rajendra Bhatia. Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review. *Brain-Computer Interfaces*, 4(3):155–174, 2017.

[78] Fabien Lotte, Laurent Bougrain, Andrzej Cichocki, Maureen Clerc, Marco Congedo, Alain Rakotomamonjy, and Florian Yger. A review of classification algorithms for EEG-based brain–computer interfaces: a 10 year update. *Journal of Neural Engineering*, 15(3):031005, 2018.

[79] Hubert Banville, Omar Chehab, Aapo Hyvärinen, Denis-Alexander Engemann, and Alexandre Gramfort. Uncovering the structure of clinical EEG signals with self-supervised learning. *Journal of Neural Engineering*, 18(4):046020, 2021.

[80] Yilun Chen, Ami Wiesel, Yonina C Eldar, and Alfred O Hero. Shrinkage algorithms for MMSE covariance estimation. *IEEE Transactions on Signal Processing*, 58(10):5016–5029, 2010.

[81] David Sabbagh, Pierre Ablin, Gaël Varoquaux, Alexandre Gramfort, and Denis A Engemann. Manifold-regression to predict from MEG/EEG brain signals without source modeling. In *Advances in Neural Information Processing Systems*, pages 7323–7334, 2019.

[82] Rajendra Bhatia, Tanvi Jain, and Yongdo Lim. On the Bures–Wasserstein distance between positive definite matrices. *Expositiones Mathematicae*, 2018.

[83] Mohammad M Ghassemi, Benjamin E Moody, Li-Wei H Lehman, Christopher Song, Qiao Li, Haoqi Sun, Roger G Mark, M Brandon Westover, and Gari D Clifford. You snooze, you win: the physionet/computing in cardiology challenge 2018. In *2018 Computing in Cardiology Conference (CinC)*, volume 45, pages 1–4. IEEE, 2018.

[84] 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.

[85] Silvia López, I Obeid, and J Picone. Automated interpretation of abnormal adult electroencephalograms. *MS Thesis, Temple University*, 2017.

[86] Khald Aboalayon, Miad Faezipour, Wafaa Almuhammadi, and Saeid Moslehpour. Sleep stage classification using EEG signal analysis: a comprehensive survey and new investigation. *Entropy*, 18(9):272, 2016.

[87] Olave E Krigolson, Chad C Williams, Angela Norton, Cameron D Hassall, and Francisco L Colino. Choosing MUSE: Validation of a low-cost, portable EEG system for ERP research. *Frontiers in Neuroscience*, 11:109, 2017.

[88] Ali Hashemi, Lou J Pino, Graeme Moffat, Karen J Mathewson, Chris Aimone, Patrick J Bennett, Louis A Schmidt, and Allison B Sekuler. Characterizing population EEG dynamics throughout adulthood. *ENeuro*, 3(6), 2016.

[89] Abhay Koushik, Judith Amores, and Pattie Maes. Real-time sleep staging using deep learning on a smartphone for a wearable EEG. *arXiv preprint arXiv:1811.10111*, 2018.

[90] Cassandra M Wilkinson, Jennifer I Burrell, Jonathan WP Kuziek, Sibi Thirunavukkarasu, Brian H Buck, and Kyle E Mathewson. Predicting stroke severity with a 3-min recording from the muse portable EEG system for rapid diagnosis of stroke. *Scientific Reports*, 10(1):1–11, 2020.

[91] Sheng-Fu Liang, Chin-En Kuo, Yu-Han Hu, Yu-Hsiang Pan, and Yung-Hung Wang. Automatic stage scoring of single-channel sleep EEG by using multiscale entropy and autoregressive models. *IEEE Transactions on Instrumentation and Measurement*, 61(6):1649–1657, 2012.

[92] Joseph Y Cheng, Hanlin Goh, Kaan Dogrusoz, Oncel Tuzel, and Erdrin Azemi. Subject-aware contrastive learning for biosignals. *arXiv preprint arXiv:2007.04871*, 2020.

[93] Mostafa Neo Mohsenvand, Mohammad Rasool Izadi, and Pattie Maes. Contrastive representation learning for electroencephalogram classification. In *Machine Learning for Health*, pages 238–253. PMLR, 2020.

[94] Aaqib Saeed, David Grangier, Olivier Pietquin, and Neil Zeghidour. Learning from heterogeneous EEG signals with differentiable channel reordering. *arXiv preprint arXiv:2010.13694*, 2020.

[95] Stefan Haufe, Frank Meinecke, Kai Görgen, Sven Dähne, John-Dylan Haynes, Benjamin Blankertz, and Felix Bießmann. On the interpretation of weight vectors of linear models in multivariate neuroimaging. *NeuroImage*, 87:96–110, 2014.

[96] Michael L Seltzer, Dong Yu, and Yongqiang Wang. An investigation of deep neural networks for noise robust speech recognition. In *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 7398–7402. IEEE, 2013.

[97] Dmitriy Serdyuk, Kartik Audhkhasi, Philémon Brakel, Bhuvana Ramabhadran, Samuel Thomas, and Yoshua Bengio. Invariant representations for noisy speech recognition. *arXiv preprint arXiv:1612.01928*, 2016.

[98] Davis Liang, Zhiheng Huang, and Zachary C Lipton. Learning noise-invariant representations for robust speech recognition. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, pages 56–63. IEEE, 2018.

[99] Julian Salazar, Davis Liang, Zhiheng Huang, and Zachary C Lipton. Invariant representation learning for robust deep networks. In *Workshop on Integration of Deep Learning Theories, NeurIPS*, 2018.

[100] Bo Li, Tara N Sainath, Ron J Weiss, Kevin W Wilson, and Michiel Bacchiani. Neural network adaptive beamforming for robust multichannel speech recognition. *Interspeech 2016*, pages 1976–1980, 2016.

[101] Xiong Xiao, Shinji Watanabe, Hakan Erdogan, Liang Lu, John Hershey, Michael L Seltzer, Guoguo Chen, Yu Zhang, Michael Mandel, and Dong Yu. Deep beamforming networks for multi-channel speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5745–5749. IEEE, 2016.

[102] Xiong Xiao, Shinji Watanabe, Eng Siong Chng, and Haizhou Li. Beamforming networks using spatial covariance features for far-field speech recognition. In *2016 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, pages 1–6. IEEE, 2016.

[103] Weipeng He, Lu Lu, Biqiao Zhang, Jay Mahadeokar, Kaustubh Kalgaonkar, and Christian Fuegen. Spatial attention for far-field speech recognition with deep beamforming neural networks. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7499–7503. IEEE, 2020.

[104] Ye Yuan, Kebin Jia, Fenglong Ma, Guangxu Xun, Yaqing Wang, Lu Su, and Aidong Zhang. A hybrid self-attention deep learning framework for multivariate sleep stage classification. *BMC Bioinformatics*, 20(16):586, 2019.

[105] Antoine Guillot and Valentin Thorey. RobustSleepNet: Transfer learning for automated sleep staging at scale. *arXiv preprint arXiv:2101.02452*, 2021.

[106] Ye Yuan, Guangxu Xun, Fenglong Ma, Qiuling Suo, Hongfei Xue, Kebin Jia, and Aidong Zhang. A novel channel-aware attention framework for multi-channel EEG seizure detection via multi-view deep learning. In *2018 IEEE EMBS International Conference on Biomedical & Health Informatics (BHI)*, pages 206–209. IEEE, 2018.

[107] Ye Yuan and Kebin Jia. FusionAtt: Deep fusional attention networks for multi-channel biomedical signals. *Sensors*, 19(11):2429, 2019.

[108] Yen-Cheng Huang, Jia-Ren Chang, Li-Fen Chen, and Yong-Sheng Chen. Deep neural network with attention mechanism for classification of motor imagery EEG. In *2019 9th International IEEE/EMBS Conference on Neural Engineering (NER)*, pages 1130–1133. IEEE, 2019.

[109] Samaneh Nasiri and Gari D Clifford. Attentive adversarial network for large-scale sleep staging. In *Machine Learning for Healthcare Conference*, pages 457–478. PMLR, 2020.

[110] Denis A Engemann, Oleh Kozynets, David Sabbagh, Guillaume Lemaître, Gael Varoquaux, Franziskus Liem, and Alexandre Gramfort. Combining magnetoencephalography with magnetic resonance imaging enhances learning of surrogate-biomarkers. *Elife*, 9:e54055, 2020.

[111] Guido Van Rossum and Fred L Drake Jr. *Python reference manual*. Centrum voor Wiskunde en Informatica Amsterdam, 1995.

[112] Travis E Oliphant. Python for scientific computing. *Computing in Science & Engineering*, 9(3):10–20, 2007.

[113] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, 2020.

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

[115] John D Hunter. Matplotlib: A 2D graphics environment. *Computing in Science & Engineering*, 9(3):90–95, 2007.

[116] Michael Waskom, Olga Botvinnik, Maoz Gelbart, Joel Ostblom, Paul Hobson, Saulius Lukauskas, David C Gemperline, Tom Augspurger, Yaroslav Halchenko, Jordi Warmenhoven, John B. Cole, Julian de Ruiter, Jake Vanderplas, Stephan Hoyer, Cameron Pye, Alistair Miles, Corban Swain, Kyle Meyer, Marcel Martin, Pete Bachant, Eric Quintero, Gero Kunter, Santi Villalba, Brian, Clark Fitzgerald, C.G. Evans, Mike Lee Williams, Drew O'Kane, Tal Yarkoni, and Thomas Brunner. mwaskom/seaborn: v0.11.0 (sepetmber 2020), September 2020.

[117] Wes McKinney et al. Data structures for statistical computing in python. In *Proceedings of the 9th Python in Science Conference*, volume 445, pages 51–56. Austin, TX, 2010.

[118] Omry Yadan. Hydra - a framework for elegantly configuring complex applications. Github, 2019.

[119] Alexandre Barachant, Stéphane Bonnet, Marco Congedo, and Christian Jutten. Classification of covariance matrices using a Riemannian-based kernel for BCI applications. *Neurocomputing*, 112:172 – 178, 2013. Advances in artificial neural networks, machine learning, and computational intelligence.

Appendices for "Robust learning from corrupted EEG with dynamic spatial filtering"

Hubert Banville*[1,2], Sean U.N. Wood[2], Chris Aimone[2], Denis-Alexander Engemann†[1,3], and Alexandre Gramfort†[1]

[1]Université Paris-Saclay, Inria, CEA, Palaiseau, France
[2]InteraXon Inc., Toronto, Canada
[3]Max Planck Institute for Human Cognitive and Brain Sciences, Department of Neurology, Leipzig, Germany

Appendix A Representation of spatial information in the DSF module

In this section, we discuss different spatial representations of EEG that can be used as input to a spatial attention block such as the DSF module. Specifically, we consider the spatial covariance matrix along with different vectorization schemes.

Given some EEG signals $X \in \mathbb{R}^{C \times T}$, where T is the number of time samples in X, and which we assume to be zero-mean, an unbiased estimate of their covariance reads:

$$\Sigma(X) = \frac{XX^\top}{T} \in \mathbb{R}^{C \times C} \quad . \tag{7}$$

The zero-mean assumption is justified after some high-pass filtering or simple baseline correction of the signals. To assess whether one channel is noisy or not, a human expert annotator will typically rely on the power of a signal and its similarity with the neighboring channels. This information is encoded in the covariance matrix.

Multiple well-established signal processing techniques rely on some estimate of Σ. For instance, common spatial patterns (CSP) performs generalized eigenvalue decomposition of covariance matrices to identify optimal spatial filters for maximizing the difference between two classes [1]. Riemannian geometry approaches to EEG classification and regression instead leverage the geometry of the space of symmetric positive definite (SPD) matrices to develop geometry-aware metrics. They are used to average and compare covariance matrices, which has been shown to outperform other classical approaches [2, 3]. Artifact handling pipelines such as the Riemannian potato [4] and Artifact Subspace Reconstruction [5] further rely on covariance matrices to identify bad epochs or attenuate noise.

The values in a covariance matrix often follow a heavy-tailed distribution. Therefore, knowing that neural networks are typically easier to train when the distribution of input values is fairly

*correspondence: hubert.jacob-banville@inria.fr
†joint senior authors

concentrated, it is helpful to standardize the covariance values before feeding them to the network. While scalar non-linear transformations (*e.g.,* logarithms) could help reduce the range of values and facilitate a neural network's task, the geometry of SPD matrices actually calls for metrics that respect the Riemannian structure of the SPD matrices' manifold [6]. For instance, this means using the matrix logarithm instead of naively flattening the upper triangle and diagonal of the matrix [3]. For an SPD matrix S, whose orthogonal eigendecomposition reads $S = U\Lambda U^\top$, where $\Lambda = \text{diag}(\lambda_1, ..., \lambda_n)$ contains its eigenvalues, the matrix logarithm $\log(S)$ is given by:

$$\log(S) = U \,\text{diag}(\log(\lambda_1), ..., \log(\lambda_n))U^\top \quad . \tag{8}$$

The diagonal and upper-triangular part of $\log(S)$ can then be flattened into a vector with $C(C+1)/2$ values, which is then typically used with linear models, *e.g.,* support vector machines (SVM) or logistic regression.

Other options to provide input values in a restricted range exist. For instance, one could simply use the element-wise logarithm of the diagonal of the covariance matrix, *i.e.,* the log-variance of the input signals. This is appropriate if pairwise inter-channel covariance information is deemed not critical down the line. Alternatively, Pearson's correlation matrix, which can be seen as the covariance matrix of the z-score normalized signals, could be used. It has the advantage that its values are already in a well-defined range (-1, 1), yet it is blind to channel variances. In our experiments, we focused on two spatial representations: the channel-wise variance obtained from the diagonal of Σ, and the matrix logarithm of Σ. Both helped improve robustness on the pathology detection and sleep staging tasks.

Appendix B Deep learning architectures

The ConvNets f_Θ used in our experiments are described in more detail in Fig. S1. In MSD experiments, the input sampling rate was of 128 Hz instead of 100 Hz as for PC18. Therefore, we adapted the temporal convolution and max pooling hyperparameters so that they would cover approximately the same duration: filter size of 64 samples, padding size of 13 and max pooling size of 16 (vs. 50, 10 and 13, respectively). This yielded a total of 21,369 parameters.

Appendix C Hyperparameter optimization of baseline models

A grid-search over hyperparameters of the random forest (RF) and logistic regression classifiers was performed with 3-fold cross-validation on combined training and validation sets. This search was performed for each reported experimental configuration: for each number of channels (for experiments in Section 4.1), each denoising strategy (no denoising, Autoreject and data augmentation) and each dataset (TUAB, PC18 and MSD).

For all RF models, we used 300 trees. This turned out to be a good trade-off between model performance and computational costs. For each experiment, we selected by cross-validation the depth of the trees among {13,15,17,19,21,23,25}, the split criterion between Gini and entropy, and the fraction of selected features used in each tree among 'sqrt' (the square-root of the number of features is used) , 'log2' (the logarithm in base 2 of the number of features is used), and using all features. For logistic regression models, the regularization parameter C was chosen among $\{10^{-4}, 10^{-3}, \ldots, 10\}$. We expanded the search on MSD as performance did not peak in the ranges considered above by adding the following values to the search space: depth in {1,3,5,7,9,11} and C in $\{10^2, 10^3, 10^4, 10^5\}$.

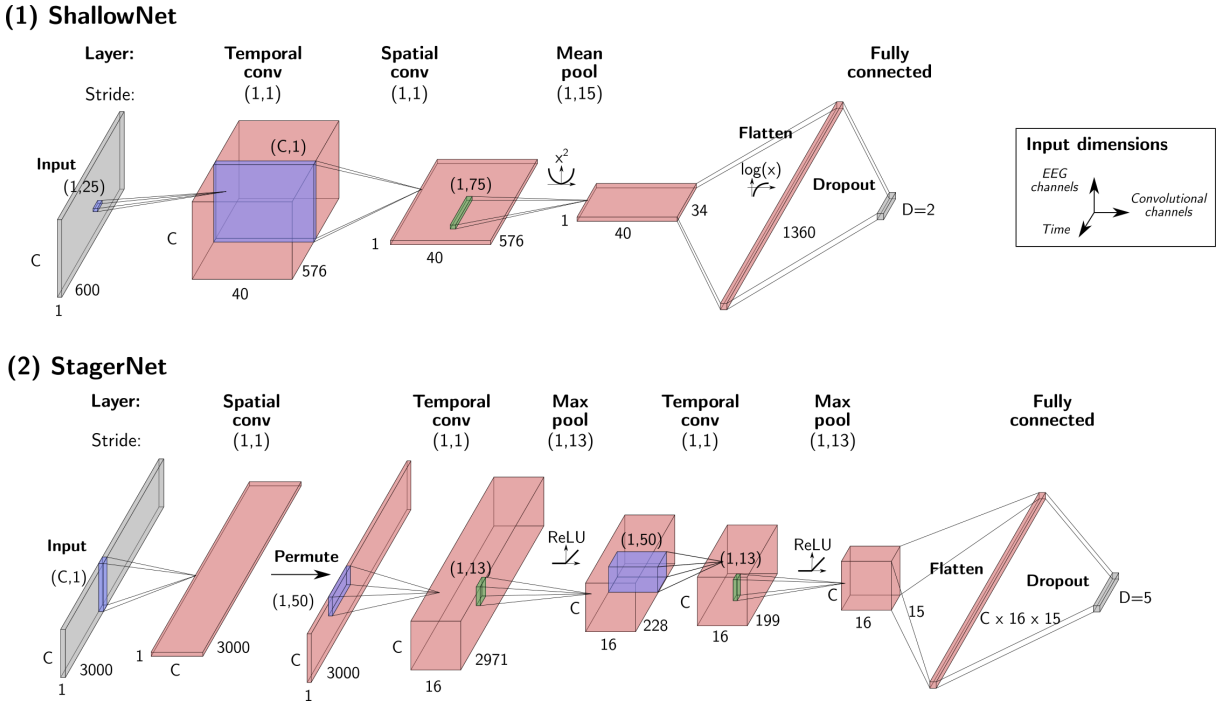

Figure S1: Neural network architectures f_Θ used in (1) pathology detection and (2) sleep staging experiments.

The selected hyperparameter configurations are listed in Tables S1 and S2 for the experiments in Sections 4.1and 4.2, respectively. Once the best hyperparameters for an experimental configuration were identified, the training and validation sets were combined into a single set on which the model with the best hyperparameters was finally trained.

Appendix D Analysis of channel corruption in the Muse Sleep Dataset

The Muse Sleep Dataset (MSD) is a collection of at-home overnight recordings. These recordings were purposefully selected to evaluate sleep staging algorithms in challenging mobile EEG conditions and therefore include recordings with highly corrupted channels. Overall, noise is

Table S1: Selected hyperparameters for experiments on number of channels (Section 4.1).

		Number of channels		
Model	**Hyperparameter**	**2**	**6**	**21**
Random Forest (RF)	Number of trees	300	300	300
	Tree depth	17	21	19
	Criterion	entropy	Gini	entropy
	Features	all	all	all
Logistic regression (LR)	C	0.1	0.1	0.001

Table S2: Selected hyperparameters for experiments on denoising strategies (Section 4.2).

			Denoising strategy		
Dataset	**Model**	**Hyperparameter**	**No denoising**	**Autoreject**	**Data augmentation**
TUAB	RF	Number of trees	300	300	300
		Tree depth	21	13	17
		Criterion	Gini	entropy	entropy
		Features	all	all	all
	LR	C	0.1	0.1	0.01
PC18	RF	Number of trees	300	300	300
		Tree depth	15	15	17
		Criterion	entropy	Gini	entropy
		Features	sqrt	sqrt	sqrt
	LR	C	1	1	10
MSD	RF	Number of trees	300	300	300
		Tree depth	9	9	11
		Criterion	entropy	entropy	entropy
		Features	all	sqrt	sqrt
	LR	C	0.1	0.1	10^5

stronger and more prevalent in these recordings than in typical sleep datasets collected under controlled laboratory conditions (*e.g.,* PC18).

To characterize the prevalence of channel corruption in MSD recordings, we inspected the variance and the slope of the power spectral density (PSD) of each EEG channel across 30-s windows. Variance is a good measure of signal quality (for instance, the DSFd variant received log-variance as input in our experiments), while the spectral slope is a global descriptor of the frequency content of a signal and allows distinguishing between channel corruption (which yields flatter spectra) and artifacts (often displaying strong low frequencies, *e.g.,* eye movements). Simple thresholds set empirically on these two markers allowed approximate detection of channel corruption events. Specifically, we flagged a channel in a window as "corrupted" if its $\log_{10}$-$\log_{10}$ spectral slope [7] between 0.1 and 30 Hz was above -0.5 (unitless) and its variance was above 1,000 μV^2. We then computed a recording-wise channel corruption metric by taking the percentage of bad windows for the most corrupted channel of each recording.

About two-thirds of the recordings had no channel corruption according to this metric, while the remaining had a value of up to 96.4% (Fig. S2). In those recordings with channel corruption, half of the corruption events (defined as a continuous block of epochs flagged as corrupted) lasted for 1.5 minutes or less, suggesting a large portion of the corruption happened intermittently, e.g. due to the temporary displacement of the electrodes relative to the head. Some corruption events however lasted much longer, for instance up to 88 minutes in one case. These longer corruption events are likely due to bad connection between the skin and the electrode or to problems with the instrumentation.

For our experiments on MSD, we therefore selected the 81 cleanest recordings (*i.e.*, with the lowest corruption fraction) for training and validation and kept the 17 noisiest recordings for testing. This procedure allowed testing whether a model trained on relatively clean data could perform well even when random channel corruption was introduced at inference time.

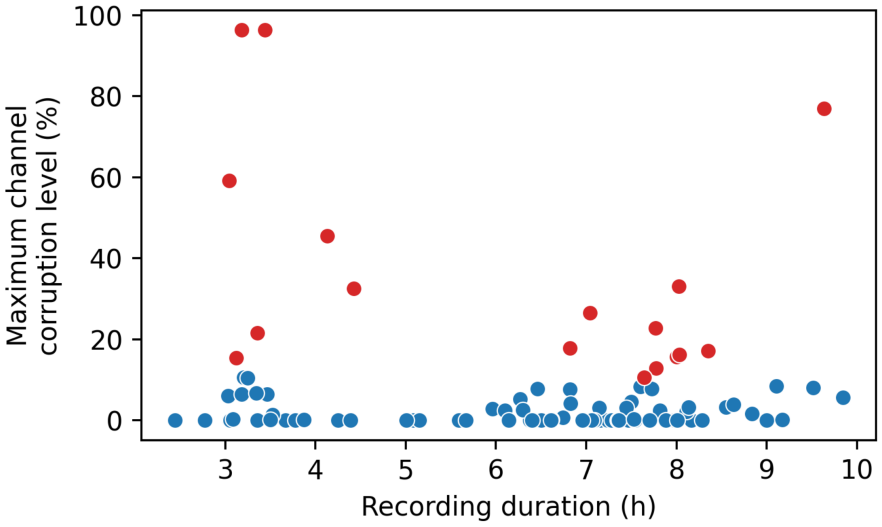

Figure S2: Corruption percentage of the most corrupted channel of each of the 98 recordings of MSD. Each point represents a single recording. The 17 most corrupted recordings (red) were used as test set in our experiments of Section 4.2.

Appendix E Baseline model performance on real-world data

The performance of the baseline models combined with the different noise handling methodologies is shown in Fig. S3.

Appendix F From simple interpolation to Dynamic Spatial Filtering

In this section, we establish a conceptual link between DSF and noise handling pipelines such as Autoreject (Section 2.1) which rely on an interpolation step to reconstruct channels that have been identified as bad. Specifically, these pipelines use head geometry-informed interpolation methods (based on the 3D coordinates of EEG electrodes and spline interpolation) to compute the weights necessary to interpolate each channel using a linear combination of the $C-1$ other channels [8]. From this perspective, a naive method of handling corrupted channels might be to always replace each input EEG channel by its interpolated version based on the other $C-1$ channels. An "interpolation-only" module m_{interp} could be written as:

$$m_{\text{interp}}(X) = W_{\text{interp}} X \ , \tag{9}$$

where W_{interp} is a $C \times C$ real-valued matrix with a 0-diagonal[1]. The limitation of this approach is that given at least one corrupted channel in the input X, the interpolated version of all non-corrupted channels will be reconstructed in part from corrupted channels. This means noise will still be present, however given enough clean channels, its impact might be mitigated.

Improving upon the naive interpolation-only approach, we might add the ability for the model to decide whether (and to what extent) channels should be replaced by their interpolated version.

[1] W_{interp} can be set or initialized using head geometry information [8] or can be learned from the data end-to-end.

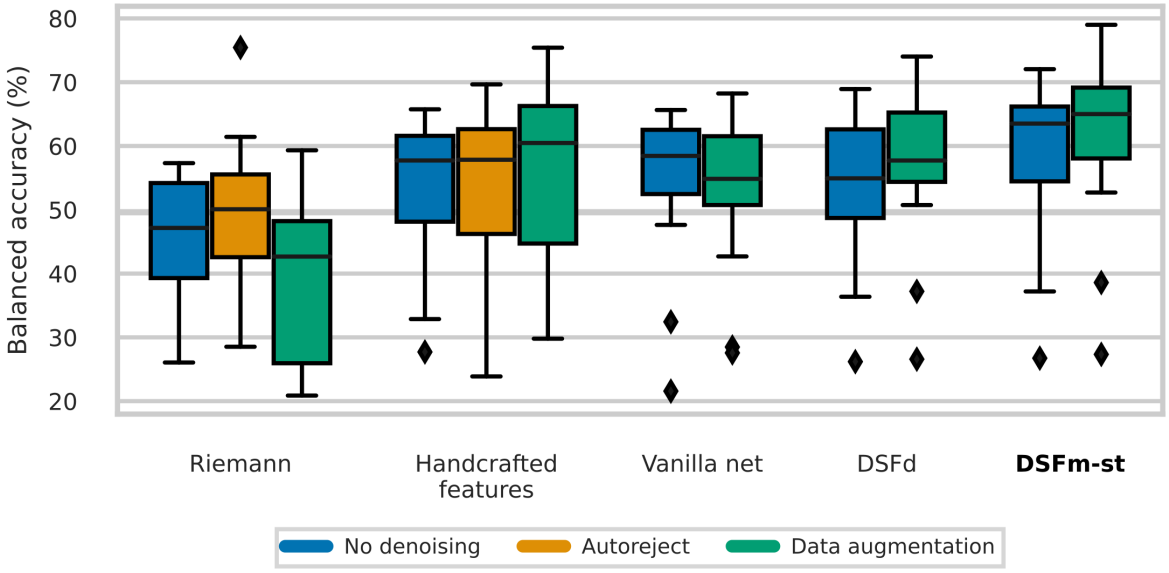

Figure S3: Performance of the different sleep staging models on MSD. As in Fig. 5, we show the distributions of performance obtained by models with different random initializations (1, 3 and 9 initializations for Riemann, handcrafted features and deep learning models, respectively) on the test recordings of MSD. Noise handling with Autoreject had no clear impact on the performance of the handcrafted features, while data augmentation was detrimental to the Riemann model. The DSFm-st models reached the highest test performance when combined with data augmentation.

For instance, if the channels in a given window are mostly clean, it might be desirable to keep the initial channels; however, if the window is overall corrupted, it might instead be better to replace channels with their interpolated version. This leads to a "scalar-attention" module m_{scalar}:

$$m_{\text{scalar}}(X) = \alpha_X X + (1 - \alpha_X) W X \ , \tag{10}$$

where $\alpha_X \in [0, 1]$ is the attention weight predicted by an MLP conditioned on X (*e.g.,* on its covariance matrix) and W is the same as for the interpolation-only module. While this approach is more flexible, it still suffers from the same limitation as before: there is a chance interpolated channels will be reconstructed from noisy channels. Moreover, the fact that the attention weight is applied globally, *i.e.*, a single weight applies to all C channels, limits the ability of the module to focus on reconstructing corrupted channels only.

Instead, the "vector attention" module m_{vector} introduces channel-wise attention weights, so that the interpolation can be independently controlled for each channel:

$$m_{\text{vector}}(X) = \text{diag}(\boldsymbol{\alpha}_X) X + (I - \text{diag}(\boldsymbol{\alpha}_X)) W X \ , \tag{11}$$

where $\boldsymbol{\alpha}_X \in [0, 1]^C$ is again obtained with an MLP and W is as above. Although more flexible, this version of the attention module still faces the same problem caused by static interpolation weights.

To solve this issue, we build on the previous approach by both predicting an attention vector $\boldsymbol{\alpha}_X$ as before and dynamically interpolating with a matrix $W_X \in \mathbb{R}^{C \times C}$ (with a 0-diagonal)

predicted by another MLP:

$$m_{\text{dynamic}}(X) = \text{diag}(\boldsymbol{\alpha}_X)X + (I - \text{diag}(\boldsymbol{\alpha}_X))W_X X \ . \tag{12}$$

In practice, a single MLP can output $C \times C$ real values, which are then reorganized into a 0-diagonal interpolation matrix W and a C-length vector whose values are passed through a sigmoid nonlinearity to obtain the attention weights $\boldsymbol{\alpha}_X$. An interesting property of this formulation which holds for m_{vector} too is that $\boldsymbol{\alpha}_X$ can be directly interpreted as the level to which each channel is replaced by its interpolated version. However, in contrast to m_{vector} the interpolation filters can dynamically adapt to focus on the most informative channels.

Finally, we observe that Eq. (12) can be rewritten as a single matrix product:

$$m_{\text{dynamic}}(X) = \left(\text{diag}(\boldsymbol{\alpha}_X) + (I - \text{diag}(\boldsymbol{\alpha}_X))W_X\right) X = \Omega_X X \ , \tag{13}$$

where, denoting the element i, j of matrix W_X as W_{ij},

$$\Omega_X = \begin{bmatrix} \alpha_1 & (1-\alpha_1)W_{12} & \dots & (1-\alpha_1)W_{1C} \\ (1-\alpha_2)W_{21} & \alpha_2 & \dots & (1-\alpha_2)W_{2C} \\ \vdots & \vdots & \ddots & \vdots \\ (1-\alpha_C)W_{C1} & (1-\alpha_C)W_{C2} & \dots & \alpha_C \end{bmatrix} \ . \tag{14}$$

The matrix Ω_X contains C^2 free variables, that are all conditioned on X through an MLP. We can then relax the constraints on Ω_X to obtain a simple matrix W_{DSF} where there are no dependencies between the parameters of a row and the diagonal elements are allowed to be real-valued. This new unconstrained formulation can be interpreted as a set of spatial filters that perform linear combinations of the input EEG channels. We can further introduce an additional bias term to recover the DSF formulation introduced in Section 2.2:

$$m_{\text{DSF}}(X) = W_{\text{DSF}}(X)X + b_{\text{DSF}}(X) \ . \tag{15}$$

This bias term can be interpreted as a dynamic re-referencing of the virtual channels. In contrast to the interpolation-based formulations, DSF allows controlling the number of "virtual channels" C' to be used in the downstream neural network in a straightforward manner (*e.g.,* enabling the use of montage-specific DSF heads that could all be plugged into the same f_Θ with fixed input shape). As shown in Section 4.4, DSF also outperformed interpolation-based formulations in our experiments.

References

[1] Zoltan J Koles, Michael S Lazar, and Steven Z Zhou. Spatial patterns underlying population differences in the background EEG. *Brain Topography*, 2(4):275–284, 1990.

[2] Marco Congedo, Alexandre Barachant, and Rajendra Bhatia. Riemannian geometry for EEG-based brain-computer interfaces; a primer and a review. *Brain-Computer Interfaces*, 4(3):155–174, 2017.

[3] David Sabbagh, Pierre Ablin, Gaël Varoquaux, Alexandre Gramfort, and Denis A Engemann. Predictive regression modeling with MEG/EEG: from source power to signals and cognitive states. *NeuroImage*, page 116893, 2020.

[4] Alexandre Barachant, Anton Andreev, and Marco Congedo. The riemannian potato: an automatic and adaptive artifact detection method for online experiments using riemannian geometry. In *TOBI Workshop lV*, pages 19–20, 2013.

[5] Tim R Mullen, Christian AE Kothe, Yu Mike Chi, Alejandro Ojeda, Trevor Kerth, Scott Makeig, Tzyy-Ping Jung, and Gert Cauwenberghs. Real-time neuroimaging and cognitive monitoring using wearable dry EEG. *IEEE Transactions on Biomedical Engineering*, 62(11):2553–2567, 2015.

[6] Zhenhua Lin. Riemannian geometry of symmetric positive definite matrices via Cholesky decomposition. *SIAM Journal on Matrix Analysis and Applications*, 40(4):1353–1370, 2019.

[7] J-B Schiratti, Jean-Eudes Le Douget, Michel Le van Quyen, Slim Essid, and Alexandre Gramfort. An ensemble learning approach to detect epileptic seizures from long intracranial EEG recordings. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 856–860. IEEE, 2018.

[8] François Perrin, J Pernier, O Bertrand, and JF Echallier. Spherical splines for scalp potential and current density mapping. *Electroencephalography and Clinical Neurophysiology*, 72(2):184–187, 1989.