# Localized component filtering for electroencephalogram artifact rejection

MARCOS DELPOZO-BANÕS
Department of Psychology, Swansea University, Swansea, Wales, UK
Swansea University Medical School, Swansea, Wales, UK

CHRISTOPH T. WEIDEMANN
Department of Psychology, Swansea University, Swansea, Wales, UK
Department of Psychology, University of Pennsylvania, Philadelphia, PA, USA

Blind source separation (BSS) based artifact rejection systems have been extensively studied in the electroencephalogram (EEG) literature. Although there have been advances in the development of techniques capable of dissociating neural and artifactual activity, these are still not perfect. As a result, a compromise between reduction of noise and leakage of neural activity has to be found. Here, we propose a new methodology to enhance the performance of existing BSS systems: Localized component filtering (LCF). In essence, LCF identifies the artifactual time segments within each component extracted by BSS and restricts the processing of components to these segments, therefore reducing neural leakage. We show that LCF can substantially reduce the neural leakage, increasing the true acceptance rate by 22 percentage points while worsening the false acceptance rate by less than 2 percentage points in a dataset consisting of simulated EEG data (4% improvement of the correlation between original and cleaned signals). Evaluated on real EEG data, we observed a significant increase of the signal-to-noise ratio of up to 9%.

Measurements of natural processes are inevitably contaminated by extraneous signals (henceforth "noise") from various sources. Such noise can pose serious problems for the interpretability of the signal, especially when its magnitude rivals or exceeds that of the signal. For almost 100 years, researchers and clinicians have been able to noninvasively record brain activity through EEG. These recordings are especially vulnerable to contamination by noise, because the neural signals recorded at the scalp are considerably smaller than other electrical activity that is regularly picked up by the sensors (e.g., due to muscle activity or interference from electric activity in the vicinity of the recordings). Advances in recording technology have increased the signal-to-noise ratio (SNR) of EEG recordings, but contamination by noise is still a major concern (Muthukumaraswamy, 2013; Fatourechi, Bashashati, Ward, & Birch, 2007).

Because noise in EEG recordings is typically considerably larger than the neural signal, it can be quite obvious when a particular epoch is contaminated by noise. It is often desirable to remove the noise from the signal rather than simply to discard contaminated epochs. Common approaches to separate signal from noise in EEG recordings involve the application of blind source separation (BSS) techniques, and in particular independent component analysis (ICA; Hyvärinen et al., 2004). These outperform other methods in rejecting high amplitude noise, such as contamination from eye movements (Daly, Nicolaou, Nasuto, & Warwick, 2013).

Activity recorded at each EEG sensor represents a combination of multiple sources, some of which are based on brain activity (signals) and some of which are not (noise). BSS algorithms transform the EEG recordings with the aim to have each dimension (component) of the data correspond to an individual source. To the extent that this separation of sources is successful and that artifactual sources can be identified, eliminating the corresponding dimensions and projecting the remaining components back into EEG-sensor space will produce a clean signal (Makeig, Bell, Jung, & Sejnowski, 1996).

The steps involved in this approach are summarized in the left panel of Figure 1. The data are initially pre-processed, for example by applying filters and by rejecting sensors and/or epochs with exceptionally high levels of noise that could interfere with the following BSS step. Often the next step is to classify each extracted component as either neural or artifactual, and to process artifactual components (usually by completely rejecting them). Finally a cleaned EEG signal is reconstructed by inverting the projection from the BSS step using the clean(ed) components, followed by any processing procedures that work better on the cleaned signal.

The literature contains a great variety of architectures following the described procedure. For example, the classification of components into clean and artifactual can be manual or automatic. Automatic classifiers can be trained on (usually manually) labeled subsets of the data (supervised techniques), or set up to achieve the classification without training examples of contaminated and uncontaminated data (unsupervised techniques). Due to the

This work was supported by a BIAL Foundation grant (#48/12) to CTW. We would like to thank Sivatharshini Sangaralingham for her assistance with data collection. We would like to thank Dr. Hugh Nolan for providing the full synthetic data set used in this research.

Python implementations of LCF, ADJUST, and FASTER are available online at https://github.com/mdelpozobanos/eeglcf, https://github.com/mdelpozobanos/eegadjust, and https://github.com/mdelpozobanos/eegfaster, respectively.

Address correspondence to: Christoph T. Weidemann, Swansea University, Department of Psychology, Singleton Park, Swansea, SA2 8PP, Wales, UK. E-mail: ctw@cogsci.info.

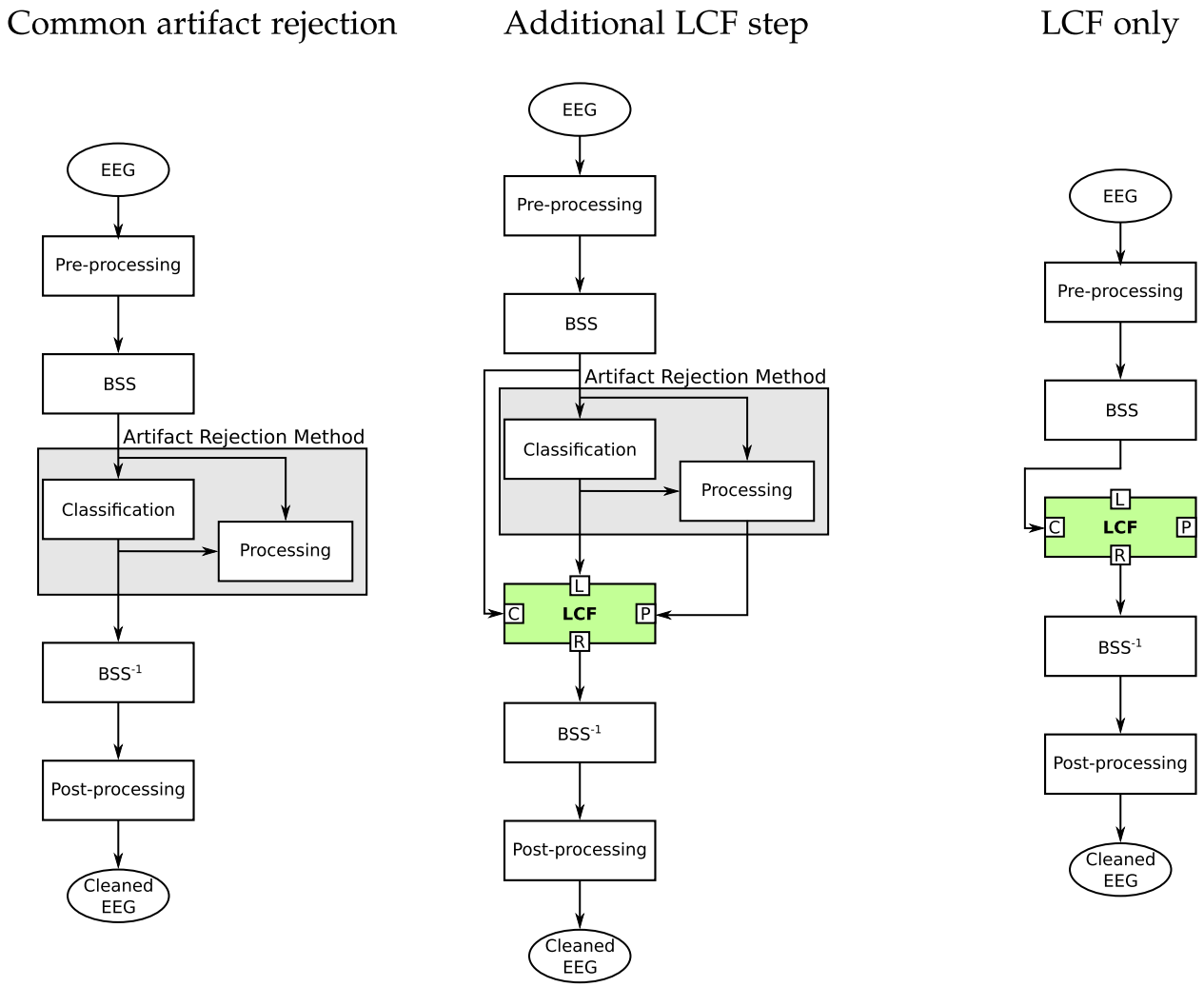


**Figure 1.** Diagrams of feature rejection systems based on blind source separation (BSS). Left: the common architecture of existing systems. The actual BSS component can be seamlessly interchanged and therefore it has been left out of the "Artifact Rejection Method" box, which defines how the output of BSS is processed. Center: A combination of the common architecture and the proposed LCF method. Right: How LCF can be used without any artifact rejection method. The labels of the inputs and output of the LCF step are explained in the caption of Figure 2.

inconvenience of labeling the data, unsupervised algorithms are more common in the EEG literature, but instances of systems integrating supervised classifiers can also be found (Shao, Shen, Ong, Wilder-Smith, & Li, 2009). Some systems lack a classification stage entirely, treating each of the components equally (Castellanos & Makarov, 2006) and hybrid approaches differentially process both clean and artifactual components (Vorobyov & Cichocki, 2002).

Unsupervised systems use a variety of rules to classify individual components as noisy. To this end, topological templates of artifacts (Li, Ma, Lu, & Li, 2006, Viola et al., 2009) and statistical properties of their temporal and frequency representations (Delorme, Sejnowski, & Makeig, 2007; Greco, Mammone, Morabito, & Versaci, 2007) have been extensively used, and the most successful approaches use combinations of some or all of such features (Nolan et al. 2010; Mognon et al., 2010; Winkler, Haufe, & Tangermann, 2011).

The BSS step is particularly crucial for the system's success, and a comparative study by Romero, Mañanas, & Barbanoj (2008) shows that, for all surveyed systems, some neural activity is rejected along with artifacts (i.e., "neural leakage"; Castellanos and Makarov, 2006; Joyce, Gorodnitsky, & Kutas, 2004). Increasing the threshold for identifying a component as artifactual would reduce neural leakage at the cost of increasing the level of remaining noise.

We propose a novel methodology to improve the balance between artifact rejection and retention of neural activity by focusing the processing of BSS components: localized component filtering (LCF). The presented algorithm localizes time segments within components contaminated by artifacts, and directs the processing to these segments, keeping the remaining parts of the component in their original forms. This removes the need for a conservative threshold on the identification of artifactual components, because components identified as containing noise undergo further scrutiny and are not generally removed entirely, reducing the probability of neural leakage (i.e., the removal of neural signal).

Furthermore, we have designed LCF to be easily integrated within existing BSS-based artifact rejection systems. The LCF component can be directly embedded before the $\mathrm{BSS}^{-1}$ step, without any modification to the other steps (Figure 1, central panel). Alternatively, LCF can also be used by itself, without a separate artifact rejection method (Figure 1, right panel). LCF is a general approach to the problem of neural leakage. In this paper, we

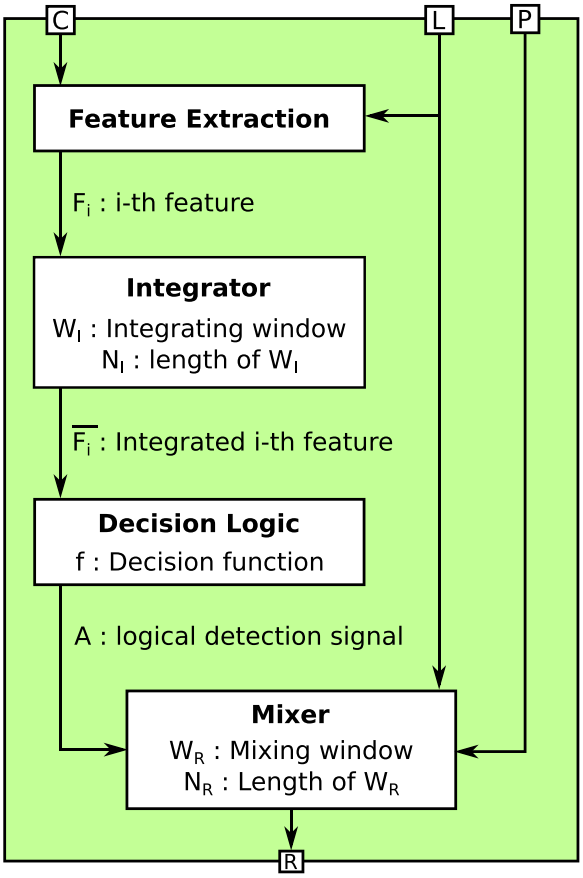


**Figure 2.** Diagram of the LCF step. The inputs and output are (C) the original BSS components, (L) the control signal pointing to the components that will be mixed, (P) the processed or alternative components, and (R) the resulting mixed components.

develop a deliberately simple implementation in an effort to assess the method, but the same framework can be used with more sophisticated approaches for identifying and eliminating artifacts.

### Localized Component Filtering

Even though the different components identified by BSS techniques generally do not perfectly separate signal and noise, separating the signal into different components does facilitate identification of artifacts. We therefore propose to apply LCF as an additional step after BSS to optimize both detection of noise and reduction of leakage of neural activity. However, LCF is an independent processing step that can be integrated with a wide range of artifact rejection systems as illustrated below. LCF consists of the following steps (Figure 2):

#### Feature Extraction

Instantaneous measurements (i.e., measures that are defined for each time instant $n$), which are characteristic of noisy activity, are extracted from the original component. Because blink, muscle, and pop-off artifacts are localized in time, components carrying them are characterized by bursts of activity. In this case, the component's voltage can be used to locate these artifacts. Similarly, high frequency noise, when localized in time, translates in a sudden increase of the amplitude of the component's time derivative. Non-instantaneous measurements could also be used by windowing the components. Windows with abnormal values of variance, Hurst exponent, and voltage range could indicate the presence of noise. When noise is localized in frequency or has a well-defined power distribution, it may be easier to detect within the frequency domain. The short-time Fourier transform or the discrete wavelet transform are two examples of tools that could be used to obtain instantaneous-like frequency representations of the BSS components. The result of the feature extraction step is a list of features, with $F_i[c,n]$ representing the $i$-th feature obtained from component $c$ at time instant $n$.

#### Integrator

Because instantaneous features are noisier than those based on the whole signal, the application of an integration window around each time instant is useful to stabilize the extracted features (Figure 3). The list of integrated features $\overline{F_i[c,n]}$ is defined as follows:

$$\overline{F_i[c,n]} = \frac{F_i[c,n] * W_I}{\sum_{k\in K} W_I(k)}, \qquad (1)$$

where $W_I$ is a window of length $N_I$, * is the convolution operator, and $K$ is the integration range defined as

$$K = [max(0, N_I/2 - n), min(N_I - 1, N_I/2 + N_C - n)], \qquad (2)$$

with $N_C$ the length of the component. This range covers the entire integrating window except when it reaches the beginning or end of the component. The denominator of Equation 1 is a normalizing factor that effectively transforms the integration to a weighted

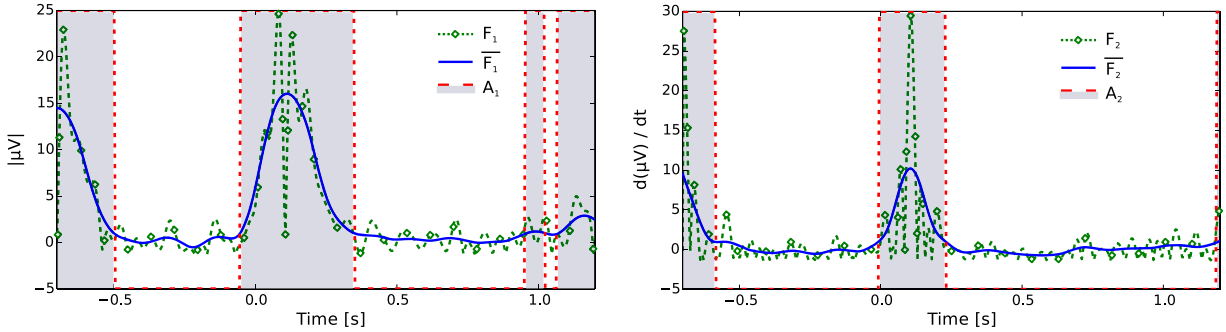


**Figure 3.** Examples of $F_1$ and $F_2$ defined in the implemented (LCF) block. Their integrated version $\overline{F_i}$ is smoother, allowing for a more robust detection of $A_i$ (in the figure we expanded the amplitude of $A_i$ to the length of the ordinate). For illustrative purposes, we set $b = 0$ for the calculation of $A_i$ (see Equation 8) in this figure.

average around each time instant. It also counteracts the boundary effect of the convolution, so that artifacts at the beginning and end of components can be correctly detected.

### Decision Logic

Once features are extracted and integrated, we need to create a vector that signals the presence of noise for each component $c$ at each time instant $n$: $A[c,n]=f(\{\overline{F_i[c,n]}|\forall i\})$, where $f$ is the decision logic function and $\overline{F_i[c,n]}$ is the integrated feature vector described above. Similar to the classification block of a traditional artifact rejection tool, the decision logic function can take many forms. The simplest implementation would be a fixed threshold. More advanced approaches include supervised and unsupervised learning algorithms, such as support vector machines or expectation maximization. Overall, these are generally more accurate options at the expense of higher computational costs and complexity. In the decision logic, we default to a simple binary logical vector that indicates the presence or absence of artifacts at each time instance.

### Mixer

The final step requires the output of the artifact rejection system, the processed components $P[c,n]$, to be mixed with the original unprocessed components $C[c,n]$ (in the absence of a separate artifact rejection system, $P[c,n]$ can be simply set to zeros). The mixing is governed by a mixing signal $M$ defined as follows:

$$M[c,n]=A[c,n]*W_M, \tag{3}$$

where $W_M$ is a window of length $N_M$ and $\sum W_M[n]=1$, used to round the edges of detected areas, which, in turn, smooths the transitions in the mix. $A[c,n]$ refers to the decision vector defined above, and * is the convolution operator. The resulting (cleaned) component is defined as

$$Q[c,n]=P[c,n]\cdot M[c,n]+C[c,n]\cdot(1-M[c,n]). \tag{4}$$

$Q$ thus is a mixed vector that consists of either $C$ (where the decision logic indicates an absence of artifacts), $P$ (where the decision logic indicates the presence of artifacts), or a mix of both (around transitions between samples with and without artifacts) that smooths the transition to avoid discontinuities (Figure 4).

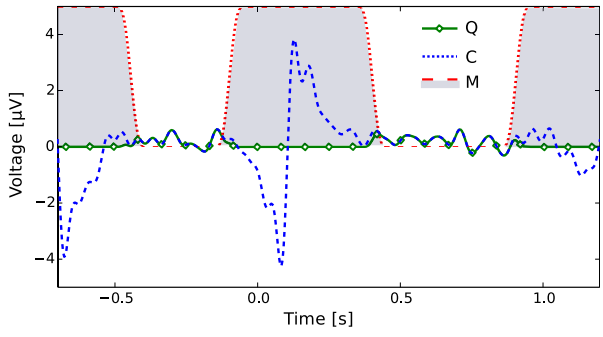


**Figure 4.** Mixing of the component $C$ whose features are depicted in Figure 3 when its processed version $P$ is all zeros. Transitions between $C$ and $P$ are smoothed to avoid discontinuities (to facilitate the figure's representation, we modified the amplitude of $M$ to match that of the ordinate such that the shaded proportion of the ordinate below $M$ denotes the proportion of $P$ in $Q$).

The resulting components can then be back-projected to the original signal space. For each component, only those samples that have been identified to either contain artifacts or to lie near artifacts are affected by these processing steps. This should reduce the leakage of neural activity and thus produce superior results to methods that eliminate entire components whenever components do not separate signal and noise sufficiently well.

Although we suggest here that LCF could be used without a separate artifact rejection method, its integration within a tool that classifies each component with respect to the noise it contains has some important benefits. Specifically, cleaning only those components that have been identified as containing artifacts allows for more aggressive LCF detection of artifacts by reducing the risk of false positives. Moreover, if the classification tool differentiates between types of artifacts (blinks, pop-off, white noise, electrocardiogram, etc.), LCF could be adapted to exploit this information to more accurately detect the different kinds of artifacts.

### Limitations of LCF

When applied to components identified as containing artifactual activity, LCF serves to focus the artifact removal on specific instances of artifactual activity within these components. Thus, LCF can only decrease the amount of reduced noise relative to the complete removal of the affected component. To the extent that LCF is successful, any reduction in the amount of removed noise should be small relative to the preserved neural activity that would otherwise have been removed along with the noise.

It is important to note that LCF is aimed at improving the cleaning of components with localized artifacts, such as blinks, muscle, or pop-off artifacts. Components with continuous contamination, such as continuous white noise, or line noise cannot be effectively tackled by LCF. Such components should be either removed entirely or appropriately filtered.

### Example Implementation of LCF

We now present applications of LCF to both simulated and real EEG data sets. In an effort to illustrate the principles of LCF, we kept the implementation deliberately simple, but note that the choices at each processing step can and should be adapted to best fit the aims of the particular application.

**Feature extraction.** Because, within (mainly) artifactual components, absolute voltage of artifactual EEG activity often exceeds the amplitude of activity from neural sources, a threshold on absolute voltage can provide a simple instantaneous indicator for the presence of noise. Likewise, because artifactual activity is often associated with sudden changes, large absolute values of the first time derivative of the voltage can also be a simple and instantaneous indicator for the presence of noise.

We normalized both of these features as follows:

$$\Theta(X)=\frac{X-m_{\hat{X}}}{s_{\hat{X}}}, \tag{5}$$

where $m$ and $s$ respectively refer to the mean and standard deviation of the variable denoted in the subscript. $\hat{X}$ is a trimmed version of the vector $X$ that excludes those samples that deviate more than three $s_X$ from $m_X$.

Thus, features are formally defined as

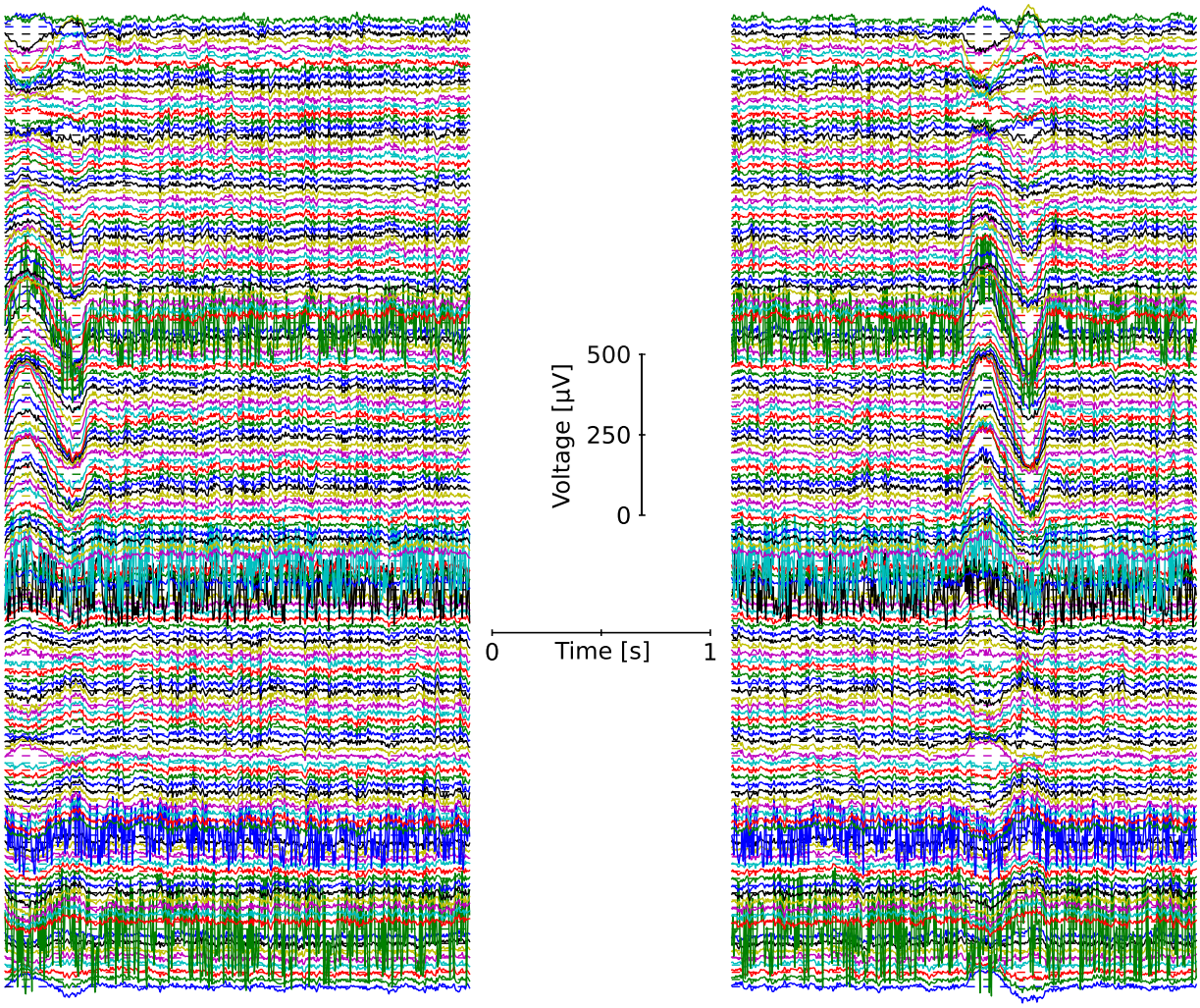


**Figure 5.** Examples of the simulated data set created by Nolan et al. (2010). Contaminated channels and blink artifacts are particularly prominent.

$$F_1 = \Theta(|C|) \tag{6}$$

and

$$F_2 = \Theta(|dC/dt|) \tag{7}$$

for each component $C$ determined in the BSS step[1].

**Integrator.** We used a Hamming window of 0.2 s as the integration window $W_I$ (Figure 3).

**Decision Logic.** We used a simple threshold ($\tau$) to determine whether a given feature indicated the presence of noise (which corresponds to a value of 1 in the decision vector $A$), so that

$$A_i[c,n] = \begin{cases} 1 & \text{if } \max([F_i[c,n-b],\ldots,F_i[c,n+b]]) > \tau \\ 0 & \text{otherwise,} \end{cases} \tag{8}$$

where $b$ corresponds to a buffer around each time point to allow samples directly adjacent to samples identified as noise to also be classified as noise. We found that a buffer of 0.1 s produced continuous noise regions that were well separated from adjacent noise regions and we used this value throughout unless otherwise noted. For the sake of simplicity, and because features were normalized to $z$-scores, we used a single threshold (manually set to $\tau = 1$) for all features.

To produce an overall decision vector that combines the information from all features, we simply combined the individual decision vectors for each feature with the logical "or" operator $\vee$:

$$A[c,n] = \bigvee_i A_i[c,n].$$

Furthermore, whenever more than 75% of a trial was labeled as noise, we rejected the entire trial. Likewise, any component where more than 75% of the samples were classified as noise was rejected entirely.

**Mixer.** We used a Hamming window of 0.1 s as the mixer window $W_M$ (Figure 4).

## Materials

We evaluated (LCF) using an artificial data set created by Nolan et al. (2010), as well as an unpublished data set recorded by us.

1. If components are classified as clean or noisy by the artifact rejection method, only artifactual components need to be processed by the LCF block.

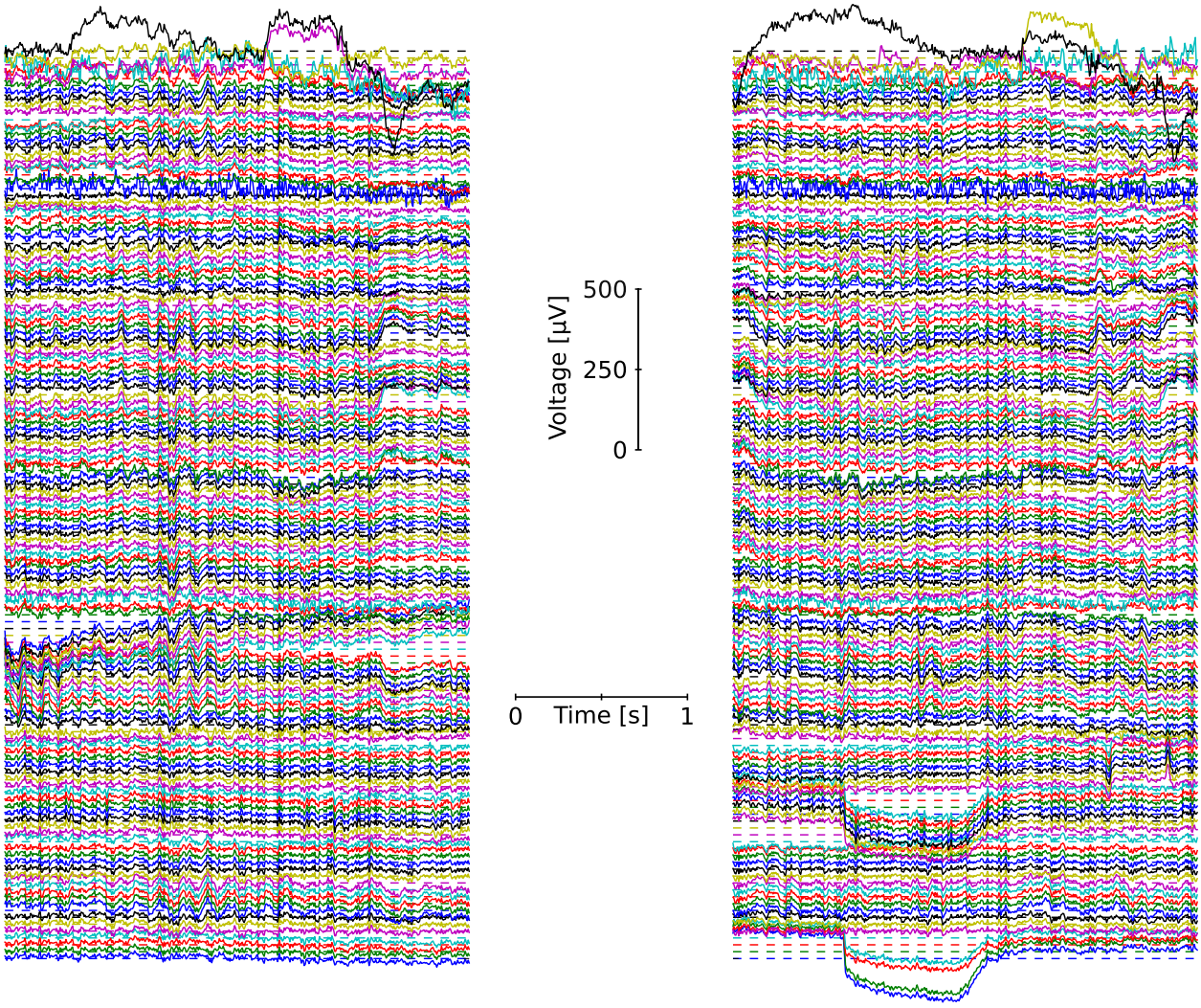


**Figure 6.** Examples of the real data set recorded from a memory task. Artifacts are substantially more complex than in the synthesized (EEG).

The artificial data set consists of 100 epochs of simulated EEG data to which artifacts were randomly added following the procedure described in Delorme et al. (2007). In addition, a random number of channels were contaminated with white noise, and some of the epochs were also corrupted by a high-amplitude (30–150 $\mu$V) low frequency (1–3 Hz) signal. A total of 47 files were created following the described procedure. Figure 5 shows two example epochs from this simulated data set.

We also evaluated LCF using a real data set recorded from 18 participants while they were engaged in a recognition memory task. We recorded a total of 133 channels (128 scalp channels, 2 mastoid channels, and 3 EOG channels) using a BioSemi Active Two system and a sampling rate of 500 Hz. We partitioned the data from each participant into 576 epochs starting 0.5 s before the onset of a test item and extending to 1 s after stimulus onset. Figure 6 shows two example epochs from this data set.

## Method

We assessed the proposed LCF methodology, using three different BSS-based algorithms. All systems shared the structure shown in Figure 1 and only differed in the artifact rejection step.

We first high-pass filtered the EEG data at 0.5 Hz using a finite impulse response (FIR) filter of order 99 and Hamming window. We chose a FIR filter to avoid distorting the phase of the signal or introducing any ripple in the pass-band. We then removed the baseline from the signal and rejected highly artifactual channels and epochs based on the $z$-scores of several statistics (variance, correlation, and Hurst exponent) following the same pre-processing procedures described in Mognon et al. (2010). Next, we used the INFOMAX ICA algorithm as the BSS step (Bell & Sejnowski, 1995) and separately applied each of the following artifact rejection methods:

ADJUST: The automatic EEG artifact detector based on the joint use of spatial and temporal features (ADJUST; Mognon et al., 2010) characterizes artifactual independent components (ICs) by both temporal and spatial features, specifically: kurtosis, variance, and the spatial distribution of IC activation. These are then automatically classified as clean or artifactual ICs by an expectation maximization algorithm (Dempster, Laird, & Rubin, 1977). The ICs identified as artifactual are then set to zero.

FASTER: The fully automated statistical thresholding for EEG artifact rejection (FASTER; Nolan et al., 2010) uses mainly temporal measures to detect artifactual ICs. In particular, this

method relies on temporal correlations with electrooculogram (EOG) channels, spectral and voltage gradients, Hurst exponent and spatial kurtosis.

ICAW: (ICAW; Castellanos & Makarov, 2006) processes each IC by thresholding its discrete wavelet transform (DWT) coefficients. We manually set a threshold equal to the 99.5 percentile of the absolute wavelet coefficients.

Finally, we post-processed the cleaned reconstructed EEG by correcting the baseline and interpolating the rejected channels using the spherical spline technique (Perrin, Pernier, Bertrand, & Echallier, 1989). In addition, we then re-referenced the signal to common average and interpolated those parts of the signal identified as artifactual as described in Mognon et al. (2010).

We applied each of the above artifact rejection techniques to the data sets twice: once with and once without the LCF step. This allowed us to assess the effectiveness of LCF under different scenarios. We also tested LCF in isolation (i.e., without using a separate artifact rejection method). For this case, we set the processed signal that is fed to the LCF step to zeros and processed all ICs.

### Analysis of Simulated EEG Data

The simulated data set has the advantage of offering fully controlled testing conditions. Specifically, the artifactual activity, $Z$, can be extracted from the artificial EEG by subtracting the simulated signal, $X$, from the simulated signal that is contaminated by noise ($Y$): $Z=Y-X$. Similarly, the rejected signal, $R$, is defined as the difference between the contaminated signal and the output of the particular artifact rejection method.

A simple difference between original and cleaned signals is insufficient to properly assess the performance of LCF. Because LCF aims specifically at reducing neural leakage, we need to be able to differentiate between the amount of noise rejected, the true rejection rate (TRR), and the amount of neural activity lost, the false rejection rate (FRR). In order to do so, we define the following measurements to be used with the artificial data set:

TRR: The TRR is the proportion of the artifactual activity that has been successfully removed. It is defined as

$$TRR=\frac{\sum_{n\in\Omega}\min(Z(n),R(n))}{\sum_{\forall n}Z(n)}, \tag{9}$$

where $\Omega=\{n\,|\,\forall\mathrm{sgn}(Z(n))=\mathrm{sgn}(R(n))\}$ and sgn is the sign function (cf. Figure 7).

FRR: The FRR quantifies the amount of neural activity that is removed in the artifact rejection effort and is defined as

$$FRR=\frac{\sum_{n\in(\Omega\cap\Psi)}(R(n)-Z(n))+\sum_{n\notin\Omega}R(n)}{\sum_{\forall n}X(n)}, \tag{10}$$

with $\Psi=\{n\,|\,|Z(n)|<|R(n)|\}$ (cf. Figure 7).

It is sometimes more convenient to reframe the performance of the artifact rejection system in terms of the true acceptance rate (TAR), which corresponds to $1-\mathrm{FRR}$, and the false acceptance rate (FAR), which corresponds to $1-\mathrm{TRR}$. Note that the TAR relates to the amount of neural signal retained, in other words, the inverse of the neural leakage (i.e., higher TARs reflect lower levels of neural leakage). Similarly, the FAR relates to the amount of noise retained, the inverse of the rejection of noise (i.e., lower FARs reflect higher levels of noise rejection). If necessary, an overall measurement of the error of the method can be computed as $(FRR+FAR)/2$.

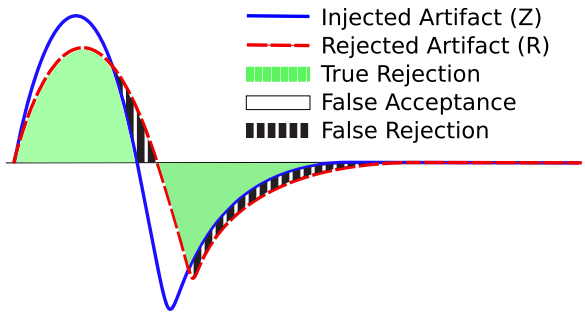


**Figure 7.** Graphic representation of the true and false rejection and false acceptance concepts.

To isolate the performance of LCF, we extracted the simulated signal, $X$, and the simulated signal contaminated with noise, $Y$, after the pre-processing stage and calculated the rejected activity $R$ from the output of the $\mathrm{BSS}^{-1}$ step (cf. Figure 1).

### ERP Analysis of EEG Data

A common way to analyze EEG data is to average EEG activity across events to produce event related potentials (ERPs). To reduce the complexity of the data set, we partitioned the 128 EEG channels into nine regions of interest (ROI; Figure 8): a central ROI (RC) surrounded by eight ROIs labeled R0, R45, R90, R135, R180, R225, R270, and R315 starting with the midfrontal region, centered around 0°, and going clockwise in 45° increments. We averaged the sensors within each region, computed the ERP for each region, and averaged those ERPs across participants.

In real data sets, without labeled artifacts, the quantities we used to assess performance on the simulated data set are not readily available. Because the baseline period of each event in the real data set lacked external stimuli that were synchronized across events, we expect the average ERP to be fairly constant and close to 0 $\mu$V for clean EEG data. However, a small proportion of artifacts (such as those produced by motion) can cause significant deviations from that mean due to their large amplitude.

Events were locked to a stimulus presentation, which should produce ERP deflections reflecting its processing. Noise, however, can obscure these synchronous effects of stimulus processing in the EEG and attenuate the resulting ERP. On the other hand, removal of neural activity will also reduce the SNR and attenuate the ERP.

We quantify the SNR as the ratio between the mean signal amplitude ($\mu$) and the standard error of the mean ($SE$) across events (Gonzalez-Moreno et al., 2014). Note that this results in a normalized mean as a function of time:

$$\mathrm{SNR}(t)=\frac{\mu(t)}{SE(t)}, \tag{11}$$

We averaged this measure over an 800 ms time window starting at stimulus onset. We directly compared SNR for artifact rejection

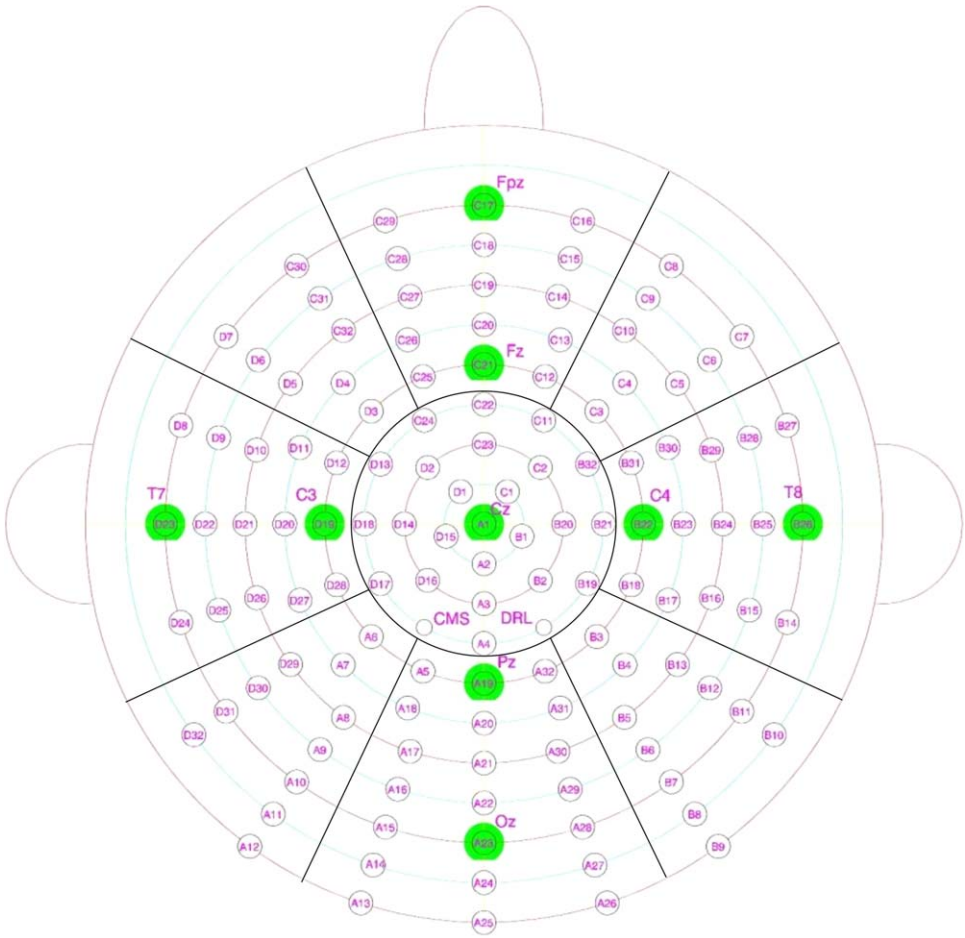


**Figure 8.** Regions of interest for the ERP analysis. We refer to the central ROI as RC and to the eight surrounding regions starting with the mid-frontal region (centered around 0°) and going clockwise in 45° increments as R0, R45, R90, R135, R180, R225, R270, and R315.

pipelines with and without LCF and also calculated the difference between these measures

$$\Delta_\zeta = \mathrm{SNR}_{\zeta+LCF} - \mathrm{SNR}_\zeta \tag{12}$$

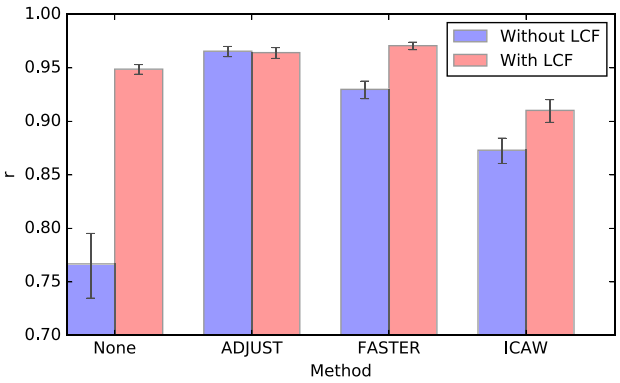


**Figure 9.** Mean percentage and *SE* of the Pearson's correlation (*r*) between the original and cleaned signal for the different methods applied to the artificial data set with and without LCF. Fisher's transform was applied before computing mean and *SE* and back-transformed to obtained the presented results.

**Table 1.** *Mean Percentage (SE) of Processed Independent Components (ICs) and Pearson's Correlation (r) Between the Original and Cleaned Signal for the Different Methods Applied to the Artificial Data Set.*

| | ICs (%) | *r* | |
|---|---|---|---|
| None | | 0.77 (0.07) | $t(46)=10.530$ |
| None+LCF | 100.00 (0.00) | 0.95 (0.07) | $SE = 0.08$ |
| Difference (%) | | +23.73% | $p < .001$ |
| ADJUST | | 0.97 (0.07) | $t(46)=-1.744$ |
| ADJUST+LCF | 2.05 (0.20) | 0.96 (0.07) | $SE = 0.01$ |
| Difference (%) | | -0.14% | $p = .088$ |
| FASTER | | 0.93 (0.06) | $t(46) = 8.639$ |
| FASTER+LCF | 10.57 (0.36) | 0.97 (0.06) | $SE = 0.05$ |
| Difference (%) | | +4.39% | $p < .001$ |
| ICAW | | 0.87 (0.05) | $t(46)=14.401$ |
| ICAW+LCF | 100.00 (0.00) | 0.91 (0.05) | $SE = 0.01$ |
| Difference (%) | | +4.26% | $p < .001$ |

*Note*. Fisher's transform was applied before computing mean and SE and back-transformed to obtained the presented results. The right most column lists the results of dependent *t*-tests between the correlations of the methods with and without LCF.

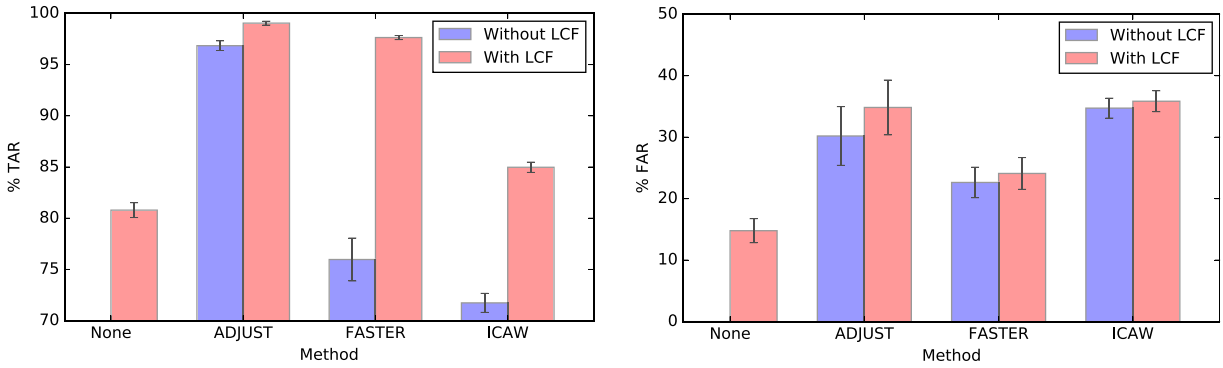


**Figure 10.** Mean percentage and *SE* of true acceptance rate (TAR), and false acceptance rate (FAR) for the different methods applied to the artificial data set with and without LCF. Higher TARs reflect lower levels of neural leakage, while lower FARs reflect higher rejection of noise.

and the improvement percentage driven by LCF

$$\Delta_{\zeta}^{\%} = 100 \times \frac{\Delta_{\zeta}}{\mathrm{SNR}_{\zeta}}\%, \tag{13}$$

with $\zeta$ indicating the artifact rejection method.

These measures, however, can only provide rough estimates of the real SNR. A visual examination of the ERPs complements the assessment of the effects of LCF. To facilitate this assessment, we examined the differences between ERPs corresponding to each of the artifact rejection methods with and without the LCF step. To the extent that LCF successfully reduces only neural leakage, LCF should increase the amplitude for these differential ERPs in the period after stimulus onset.

## Results and Discussion

### Results for Simulated EEG Data

Figure 9 and Table 1 show the correlations between the original and the cleaned synthetic signal, while Figure 10 and Table 2 show the percentages of TARs and FARs. In general, the addition of LCF resulted in an improved correlation between the original and cleaned signal, as well as in an increase in TAR without corresponding increases in FAR (differences in FAR were small relative to the associated standard deviations). In other words, it reduced the neural leakage (increase in TAR) without heavily penalizing the rejection of artifacts (small increase in FAR).

The different methods varied greatly with respect to the proportion of ICs they processed, ranging from just over 2% on average for ADJUST to 100% for ICAW. Because LCF is only applied to tagged ICs, the conservative classification of ADJUST limited the possible impact of LCF.

While the neural leakage of FASTER and ICAW was similar (76% and 72% of TAR, respectively), the introduction of LCF had a smaller (yet still substantial) effect on the latter. ICAW cleans tagged components through DWT thresholding and provides an alternative signal *P*, instead of fully rejecting them as ADJUST and FASTER do. The effectiveness of this cleaning process constraints the potential improvements on neural leakage from LCF. For both FASTER and ICAW, however, the addition of LCF resulted in a substantial increase of the TAR (reduction of neural leakage) with comparatively small increases in FAR (noise rejection penalization). Surprisingly, the application of LCF in isolation resulted in relatively good performance with a TAR 4 percentage points lower than that of ICAW+LCF and the lowest FAR of all techniques.

In some cases, we observed that the addition of LCF increased the proportion of removed artifactual activity (i.e., it reduced the FAR). This may seem counter-intuitive given that LCF limits how much of an IC is removed. However, removing non-artifactual parts of an IC, produces noise, which can result in an increase of either FRR or FAR.

These results suggest that the proposed approach works as expected: it reduces the neural leakage while preserving the ability

**Table 2.** *Mean Percentage (SE) of True and False Acceptance Rates for the Different Methods Applied to the Artificial Data Set*

| | TAR (%) | | FAR (%) | |
|---|---|---|---|---|
| None | 100.00 (0.00) | $t(46) = -26.293$ | 100.00 (0.00) | $t(46) = -43.113$ |
| None+LCF | 80.79 (0.00) | $SE = 0.73$ | 14.79 (0.00) | $SE = 1.98$ |
| | | $p < .001$ | | $p < .001$ |
| ADJUST | 96.84 (0.47) | $t(46) = 6.652$ | 30.19 (4.77) | $t(46) = 11.661$ |
| ADJUST+LCF | 99.04 (0.47) | $SE = 0.33$ | 34.83 (4.77) | $SE = 0.40$ |
| | | $p < .001$ | | $p < .001$ |
| FASTER | 75.97 (2.10) | $t(46) = 10.745$ | 22.63 (2.48) | $t(46) = 1.897$ |
| FASTER+LCF | 97.63 (2.10) | $SE = 2.02$ | 24.10 (2.48) | $SE = 0.77$ |
| | | $p < .001$ | | $p = .064$ |
| ICAW | 71.74 (0.93) | $t(46) = 29.168$ | 34.72 (1.62) | $t(46) = 8.186$ |
| ICAW+LCF | 84.97 (0.93) | $SE = 0.45$ | 35.85 (1.62) | $SE = 0.14$ |
| | | $p < .001$ | | $p < .001$ |

*Note*. The third and last columns lists the results of dependent t-tests comparing the acceptance rates of the methods with and without LCF shown in the respective previous column.

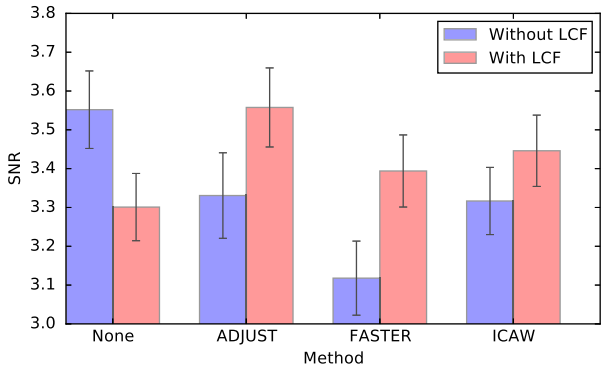


**Figure 11.** Mean and *SE* of the SNR for all channel clusters combined obtained with the different methods applied to the real data set with and without LCF.

to reject artifacts. When the system performs conservatively, tagging very few components in a bid to reduce the probability of neural leakage, the scope for improvement by LCF is reduced. With LCF, the system can be tuned to be more aggressive in the tagging of components, rejecting more noise, without corresponding increases in neural leakage.

### Results for EEG Data

The results for the real EEG data set are comparatively noisier, but they tend to mirror those from the simulated data. On average, ADJUST only tagged 5.57% ICs as artifactual, limiting the scope for improvement that could be gained by LCF. In fact, this average is substantially inflated by three outliers rejecting 22, 30, and 33 ICs. Without these outliers, the average drops to only 2%. In five out of 18 cases, ADJUST did not tag any IC for rejection. FASTER, on the other hand, rejected 15.45% of ICs on average, resulting in a higher potential to reject noise, but also increased propensity for neural leakage and hence bigger scope for improvement by LCF. ICAW processes all ICs, which maximizes the potential effectiveness of LCF, but at the same time uses a processed ("cleaned") version of the ICs as alternative signal.

Figure 11 shows the SNR for each method with and without LCF, while Figure 12 and Table 3 show the comparative results in the form of $\Delta$ and $\Delta^{\%}$. The addition of LCF to a given system resulted in improvements of the SNR for all cases, with an average increase of 7% for ADJUST, 9% for FASTER, and 4% for ICAW.

The increase in SNR should be interpreted together with two factors: First, the LCF prototype used here as a proof of concept is a rather simple one. It applies the same analysis to all tagged components, including those with continuous noise, such as white-noise. These have no salient time instants, and therefore LCF retains the component in its original form. Second, the heterogeneity of the signal quality within the real data set introduces great variability in the results. In cases where the signal quality is particularly good, BSS performs better, reducing the scope for improvement by LCF, while the opposite is true for poor signal quality instances. This can also be observed in the range of the results. LCF improves SNR by a maximum of 4.00 points, 2.62 points, and 0.85 points for ADJUST, FASTER, and ICAW, respectively, while in the worst cases it only decreases it by 0.51, 1.88, and 0.50, respectively.

When looking at the differential ERPs, comparing methods with and without LCF (Figure 13), the differences are largely confined within the ERP in response to stimulus onset—no big differences in amplitude are discernible in the baseline periods or more than about 500 ms past stimulus onset. This effectively means that LCF increased the amplitude of the stimulus locked activity (neural activity) while maintaining the rejection of noise. This figure also illustrates the total amount of noise and neural activity removed by LCF when it is used without a separate artifact rejection method as well as the distortion introduced by ICAW's alternative signal. Overall, LCF by itself (*None+LCF*), performed better than we anticipated, despite the simplicity of the current implementation. The LCF-only implementation rejected a good amount of noise while retaining most of the ERP.

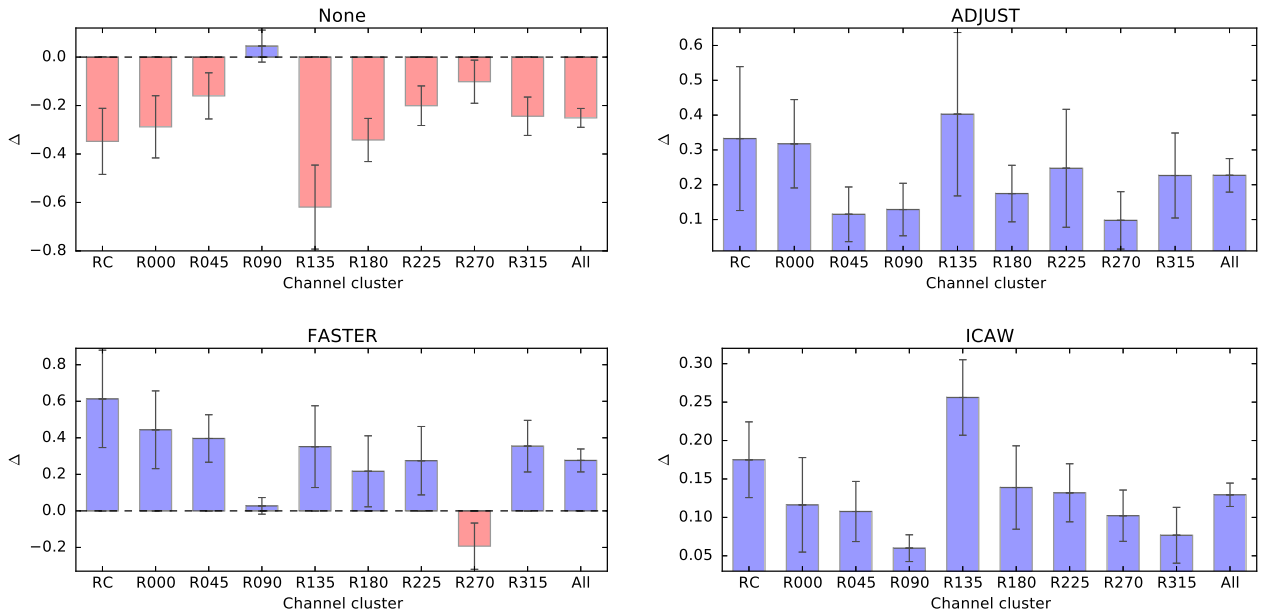


**Figure 12.** Mean and SE of the difference $\Delta$ for each channel cluster and method. Highlighted in red and blue, cases where the nominal difference is lower and higher than 0 respectively.

**Table 3.** *Mean, SE, and Range of the Difference* **Δ** *for all Channel Clusters Combined and Each Method, Together with the Corresponding LCF Improvement* $\Delta^{\%}$

| | None | ADJUST | FASTER | ICAW |
|---|---|---|---|---|
| Mean (SE) | -0.25 (0.04) | 0.23 (0.05) | 0.28 (0.06) | 0.13 (0.02) |
| [min, max] | [-2.73, 0.75] | [-0.51, 4.00] | [-1.88, 2.62] | [-0.50, 0.85] |
| Δ% | -7.06% | 6.81% | 8.86% | 3.90% |
| *t*-test | $t$ (161) = -6.428 | $t$ (161) = 4.700 | $t$ (161) = 4.383 | $t$ (161) = 8.474 |
| | $SE = 0.04$ | $SE = 0.05$ | $SE = 0.06$ | $SE = 0.02$ |
| | $p < .001$ | $p < .001$ | $p < .001$ | $p < .001$ |
| | $q < .001$ | $q < .001$ | $q < .001$ | $q < .001$ |

*Note*. Refer to Table A1 for results for individual channel clusters. The last four rows show results for dependent t-tests comparing the SNR for each method with and without LCF along with associated (uncorrected) *p*-values and False Discovery Rate *q*-values.

### Noise Rejection versus. Preservation of Neural Activity

The right balance between the rejection of noise and preservation of neural activity depends on the particular application. Considerations such as the prominence of the targeted signal, the quality of the recordings, as well as the prevalence and nature of noise, determine both the effectiveness of the BSS artifact removal and, in turn, the benefit (or lack) of an additional LCF step. Additionally, the trade-offs between failing to reject artifactual activity and removing signal vary. With LCF, we provide a tool that can reduce neural leakage at a given level of noise rejection.

## Conclusion

We propose LCF as a novel methodology to boost the performance of existing EEG artifact rejection systems that are based on BSS techniques. The method takes the original and processed components as inputs, and mixes them so that the processed (cleaned) components replace the original ones only when an artifact is detected. Quantitative and qualitative analyses on simulated and real data sets demonstrated benefits of LCF, especially in the reduction of neural activity leakage.

Benefits of LCF were more pronounced in cases where the BSS algorithm found a particularly bad solution to the problem of neural activity and noise dissociation (see Appendix). LCF can thus function to guard against inadequate levels of noise rejection and/or excessive neural leakage in cases where signal and noise are not well separated by the BSS algorithm. Indeed, we demonstrated that the application of LCF by itself (i.e., without a previous BSS step) can substantially reduce the noise while limiting neural leakage.

We designed LCF to be compatible with any of the existing systems based on BSS. Moreover, its integration is straight forward and requires no substantial changes to the system itself, facilitating its adoption. LCF's ability to retain neural activity allows for more stringent classification components (i.e., for more liberally labeling components as artifactual), while at the same time limiting the danger of signal loss. The implementation of LCF used here as a proof of concept is a very simplistic one, based on the voltage amplitude and its speed of change. More sophisticated artifact detection methods are likely to yield better results and, in turn, would allow for even more sensitive artifact rejection systems.

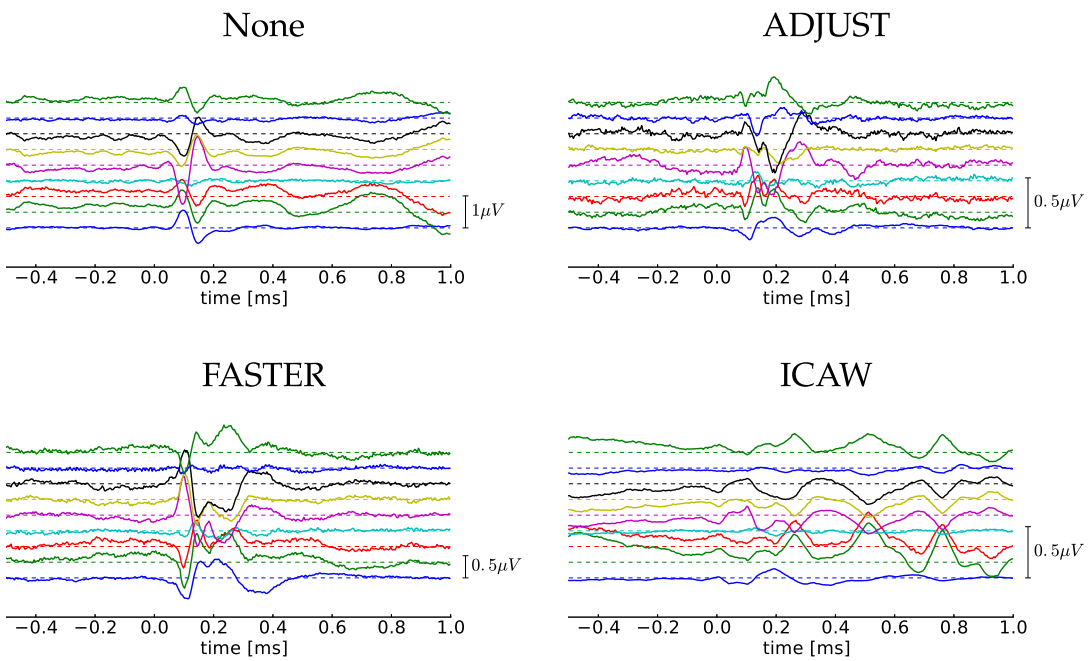


**Figure 13.** Difference between the ERPs, averaged across subjects, for multiple artifact rejection methods without and with LCF. ERPs for methods without LCF are subtracted from ERPs for methods with LCF. Lines correspond to the ERP difference of each brain region specified in Figure 8, so that the top line is RC and subsequent ones are R0, R45, R90, . . ., R315. The scale of the y-axes differs between panels as indicated.

## References


Bell, A. J. & Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. *Neural Computation*, *7*(6), 1129–1159. doi: 10.1162/neco.1995.7.6.1129

Castellanos, N. P. & Makarov, V. A. (2006). Recovering EEG brain signals: Artifact suppression with wavelet enhanced independent component analysis. *Journal of Neuroscience Methods*, *158*, 300–312. doi: 10.1016/j.jneumeth.2006.05.033

Daly, I., Nicolaou, N., Nasuto, S., & Warwick, K. (2013). Automated artifact removal from the electroencephalogram: A comparative study. *Clinical EEG and Neuroscience*, *44*, 291–306. doi: 10.1177/1550059413476485

Delorme, A., Sejnowski, T., & Makeig, S. (2007). Enhanced detection of artifacts in EEG data using higher-order statistics and independent component analysis. *Neuroimage*, *34*(4), 1443–1449. doi: 10.1016/j.neuroimage.2006.11.004

Dempster, A. P., Laird, N. M., & Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (methodological)*, 1–38.

Fatourechi, M., Bashashati, A., Ward, R. K., & Birch, G. E. (2007). EMG and EOG artifacts in brain computer interface systems: A survey. *Clinical Neurophysiology*, *118*, 480–494. doi: 10.1016/j.clinph.2006.10.019

Gonzalez-Moreno, A., Aurtenetxe, S., Lopez-Garcia, M.-E., del Pozo, F., Maestu, F., & Nevado, A. (2014). Signal-to-noise ratio of the MEG signal after preprocessing. *Journal of Neuroscience Methods*, *222*, 56–61. doi: 10.1016/j.jneumeth.2013.10.019

Greco, A., Mammone, N., Morabito, F., & Versaci, M. (2007). Semi-automatic artifact rejection procedure based on kurtosis, Renyi's entropy and independent component scalp maps. *International Journal of Medical, Health, Biomedical and Pharmaceutical Engeneering*, *1*, 466–470.

Hyvärinen, A., Karhunen, J., & Oja, E. (2004). *Independent component analysis*, volume 46. John Wiley & Sons. doi: 10.1002/0471221317

Joyce, C. A., Gorodnitsky, I. F., & Kutas, M. (2004). Automatic removal of eye movement and blink artifacts from EEG data using blind component separation. *Psychophysiology*, *41*, 313–325. doi: 10.1111/j.1469-8986.2003.00141.x

Li, Y., Ma, Z., Lu, W., & Li, Y. (2006). Automatic removal of the eye blink artifact from EEG using an ICA-based template matching approach. *Physiological Measurement*, *27*, 425. doi: 10.1088/0967-3334/27/4/008

Makeig, S., Bell, A., Jung, T., & Sejnowski, T. (1996). Independent component analysis of electroencephalographic data. In *Advances in Neural Information Processing Systems*, pages 145–151. Cambridge, MA: MIT Press.

Mognon, A., Jovicich, J., Bruzzone, L., & Buiatti, M. (2010). ADJUST: An automatic EEG artifact detector based on the joint use of spatial and temporal features. *Psychophysiology*, pages 229–240. doi: 10.1111/j.1469-8986.2010.01061.x

Muthukumaraswamy, S. D. (2013). High-frequency brain activity and muscle artifacts in MEG/EEG: A review and recommendations. *Frontiers in Human Neuroscience*, 7. doi: 10.3389/fnhum.2013.00138

Nolan, H., Whelan, R., & Reilly, R. (2010). FASTER: Fully automated statistical thresholding for EEG artifact rejection. *Journal of Neuroscience Methods*, *192*, 152–162. doi: 10.1016/j.jneumeth.2010.07.015

Perrin, F., Pernier, J., Bertrand, O., & Echallier, J. (1989). Spherical splines for scalp potential and current density mapping. *Electroencephalography and Clinical Neurophysiology*, *72*, 184–187. doi: 10.1016/0013-4694(89)90180-6

Romero, S., Mañanas, M. A., & Barbanoj, M. J. (2008). A comparative study of automatic techniques for ocular artifact reduction in spontaneous EEG signals based on clinical target variables: A simulation case. *Computers in Biology and Medicine*, *38*, 348–360. doi: 10.1016/j.compbiomed.2007.12.001

Shao, S.-Y., Shen, K.-Q., Ong, C. J., Wilder-Smith, E., & Li, X.-P. (2009). Automatic EEG artifact removal: A weighted support vector machine approach with error correction. *IEEE Transactions on Biomedical Engineering*, *56*, 336–344. doi: 10.1109/tbme.2008.2005969

Tallon-Baudry, C., Bertrand, O., Delpuech, C., & Pernier, J. (1997). Oscillatory $\gamma$-band 30–70 Hz) activity induced by a visual search task in humans. *The Journal of Neuroscience*, *17*. 722–734.

Viola, F. C., Thorne, J., Edmonds, B., Schneider, T., Eichele, T., & Debener, S. (2009). Semi-automatic identification of independent components representing EEG artifact. *Clinical Neurophysiology*, *120*, 868–877. doi: 10.1016/j.clinph.2009.01.015

Vorobyov, S. & Cichocki, A. (2002). Blind noise reduction for multisensory signals using ICA and subspace filtering, with application to EEG analysis. *Biological Cybernetics*, *86*(4), 293–303. doi: 10.1007/s00422-001-0298-6

Winkler, I., Haufe, S., & Tangermann, M. (2011). Automatic classification of artifactual ica-components for artifact removal in EEG signals. *Behavioral and Brain Functions*, *7*(1). doi: 10.1186/1744-9081-7-30