

Deep learning-based motion artifact removal in functional near-infrared spectroscopy

Yuanyuan Gao^a, Hanqing Chao^b, Lora Cavuoto^c, Pingkun Yan^{a,b},
Uwe Kruger^{a,b}, Jack E. Norfleet^{d,e,f}, Basiel A. Makled^{d,e,f},
Steven Schwaitzberg^g, Suvranu De^{a,b,*} and Xavier Intes^{a,b,*}

^aRensselaer Polytechnic Institute, Center for Modeling, Simulation and Imaging in Medicine, Troy, New York, United States

^bRensselaer Polytechnic Institute, Department of Biomedical Engineering, Troy, New York, United States

^cUniversity at Buffalo, Department of Industrial and Systems Engineering, Buffalo, New York, United States

^dU.S. Army Combat Capabilities Development Command–Soldier Center, Orlando, Florida, United States

^eSFC Paul Ray Smith Simulation and Training Technology Center, Orlando, Florida, United States

^fMedical Simulation Research Branch, Orlando, Florida, United States

^gUniversity at Buffalo, Department of Surgery, Buffalo, New York, United States

Abstract

Significance: Functional near-infrared spectroscopy (fNIRS), a well-established neuroimaging technique, enables monitoring cortical activation while subjects are unconstrained. However, motion artifact is a common type of noise that can hamper the interpretation of fNIRS data. Current methods that have been proposed to mitigate motion artifacts in fNIRS data are still dependent on expert-based knowledge and the post hoc tuning of parameters.

Aim: Here, we report a deep learning method that aims at motion artifact removal from fNIRS data while being assumption free. To the best of our knowledge, this is the first investigation to report on the use of a denoising autoencoder (DAE) architecture for motion artifact removal.

Approach: To facilitate the training of this deep learning architecture, we (i) designed a specific loss function and (ii) generated data to mimic the properties of recorded fNIRS sequences.

Results: The DAE model outperformed conventional methods in lowering residual motion artifacts, decreasing mean squared error, and increasing computational efficiency.

Conclusion: Overall, this work demonstrates the potential of deep learning models for accurate and fast motion artifact removal in fNIRS data.

© The Authors. Published by SPIE under a Creative Commons Attribution 4.0 International License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.NPh.9.4.041406](https://doi.org/10.1117/1.NPh.9.4.041406)]

Keywords: functional near-infrared spectroscopy; motion artifact; deep learning; denoising autoencoder.

Paper 21031SSRR received Jul. 8, 2021; accepted for publication Mar. 10, 2022; published online Apr. 23, 2022.

1 Introduction

Functional near-infrared spectroscopy (fNIRS) is a noninvasive neuroimaging technique to monitor brain activity indirectly. It measures the intensity of near-infrared light diffusely

*Address all correspondence to Suvranu De, des@rpi.edu; Xavier Intes, intesx@rpi.edu

scattered through cortical tissues to quantify changes in oxyhemoglobin concentration (HbO) and deoxyhemoglobin concentration (HbR) with respect to cortical regions.¹⁻⁴ HbO and HbR time series can reflect changes in neurovascular coupling and hence, neuronal activity. fNIRS has been widely used in cognitive,⁵⁻⁹ motor skill studies,¹⁰⁻¹⁵ and brain-computer interface technique.¹⁶

Although the fNIRS technology has been improved over the years, the processing of fNIRS data set can still be a challenging task. Especially, it is still difficult and time-consuming to identify and correct for motion artifacts caused by the changes in the coupling between optodes and scalp. Such artifacts are expressed as peaks or shifts in the time-series signals. Since the magnitudes of the peaks or shifts are usually much higher than the hemodynamic response function (HRF), the fNIRS signals are significantly contaminated and do not reflect the cortical activities. The phenomenon is more noticeable when the motion of the head and limbs are inevitable or even required in the experiment protocols, such as speeches,¹⁷ walking,¹⁸ and surgical tasks.^{11,12} The problem has been exacerbated by the recent rise of wearable or wireless fNIRS devices,^{19,20} which extend mobile ranges of these devices for tasks such as running or team working, that are more susceptible to motion artifacts. Thus, an efficient methodology to remove motion artifacts is essential to utilize fNIRS in those scenarios.

A few strategies that have been developed over the years include keeping any trials with motion artifacts during the data processing. This may be used only when large datasets are collected and is not the current predominant practice. Another strategy is to identify trials/channels with motion artifacts by visual inspection or to use functions such as *hmrMotionArtifact* function in the prevalent fNIRS data processing toolbox HomER2 and then discard them from further analysis. Though, the most appropriate methodology is to process these trials/channels using advanced time-series data processing methods. These include spline interpolation,²¹ wavelet filtering,²² principal component analysis (PCA),²³ Kalman filtering,²⁴ and correlation-based signal improvement (Cbsi).²⁵ The performance of these methods, however, largely depends on a set of assumptions to describe motion artifacts and the subjective selection of associated tuning of parameters (Table 1). As an example, Ref. 29 demonstrated that selecting the PCA parameter, i.e., the percentage of variance in the data that PCA removed²⁷ to be 0.80 and 0.97 produced significantly different results. Thus, a method that does not require the subjective fine-tuning of the parameters, or does not rely on stringent assumptions, is highly desirable. Here, we propose a deep learning method to learn the noise features automatically.

Over the last decade, deep neural networks have emerged as a powerful tool to suppress noise in image datasets in a fast and efficient manner. Deep learning models have been shown to produce competitive denoising results while retain more texture details when compared to conventional methods.³⁰⁻³³ Deep learning networks also showed superior performance when applied to medical imaging problems. For example, denoising autoencoder (DAE) model could denoise mammograms [structural similarity index measure (SSIM) from 0.45 to 0.73] and dental x-ray data (SSIM from 0.62 to 0.86).³⁴ A DAE model achieved 10% higher peak signal-to-noise ratio (PSNR) and SSIM than the conventional algorithm in chest radiograms.³⁵ A recent study³⁶ showed that a long short-term memory (LSTM) network increased the accuracy of voxels classification in fMRI data by more than 20%. Another study³⁷ showed that the deep learning model could completely remove the Gibbs phenomena in diffusion MRI data. DAE model has also been applied to denoising EEG data and increased PSNR values in EEG channels.³⁸ A previous study³⁹ employed artificial neural network model to optimize the weights of wavelet basis to denoise fNIRS data and achieved higher contrast-to-noise ratio (CNR) than conventional wavelet denoising methods. Another study⁴⁰ detected motion artifact types using machine learning models on broadband fNIRS data.

Herein, we report on the use of a DAE model associated with a dedicated loss function purposely designed to remove the motion artifacts. To train such a deep learning network, we implemented an autoregression (AR) model to generate a large synthetic fNIRS dataset. The performance of our DAE methodology was established using this synthetic data set and benchmarked against the current conventional methods used by the fNIRS community. Moreover, the performances of the DAE were successfully validated on an open-access experimental dataset.

Table 1 The motion artifact removal models for fNIRS.

Model name and references	Assumptions	Implementation steps	Drawbacks
Spline interpolation ²¹	The shape of the motion artifact is captured by spline interpolation.	<ol style="list-style-type: none"> 1. Identify the noise. 2. Model the noise by cubic spline interpolation. 3. Subtract the interpolation from the original signal. 4. Reconstruction. 	Denoise performance depends on the noise detection method. The interpolation degree needs to be tuned.
Wavelet filtering ^{22,26}	The wavelet coefficients are assumed to be normally distributed, and the outliers are accounted as motion artifacts.	<ol style="list-style-type: none"> 1. Wavelet discrete decomposition. 2. Identify the outliers in the coefficients larger than the probability threshold alpha. 3. Set the outliers to zeros. 	Probability threshold alpha needs to be tuned.
PCA ^{23,27}	The first several components of the PCA represent the variance caused by motion artifacts. Motion artifacts are likely to occur in most channels at the same time.	<ol style="list-style-type: none"> 1. Apply PCA to produce uncorrelated components. 2. Remove the components that have the highest contribution to the variance of the original data. 	The number/portion of the components to be removed needs to be tuned. It is limited by the total number of channels available. As a spatial filtering method, PCA depends on the geometry of the probes.
Kalman filtering ²⁴	The state is assumed to be motion-free.	<ol style="list-style-type: none"> 1. Predict the state of the next time step. 2. Correct the prediction based on the measured signal. 3. Repeat steps 1 and 2. 	Build-up errors may happen as prediction time increases. ²⁸
Cbsi ²⁵	HbO and HbR are negatively correlated. Motion artifacts are independent of Hb. The ratio between HbO and HbR is the same, irrespective of the presence of artifacts.	<ol style="list-style-type: none"> 1. $HbO' = \frac{HbO - \alpha \cdot HbR}{2}$, 2. $HbR' = -(\frac{1}{\alpha} \cdot HbO')$, where $\alpha = \text{std}(HbO) / \text{std}(HbR)$.	The impacts of motion artifacts may differ between HbO and HbR.

2 Methods

2.1 Data Collection for the Experimental Dataset

The data used in this study were adopted from a public available dataset.⁴¹ This fNIRS dataset contains motion artifacts (reading aloud, nodding their head up and down, nodding sideways, twisting right, twisting left, shaking head rapidly from side to side and raising their eyebrows) under resting state of eight participants. The details of this dataset can be found in Ref. 42. We also used a finger tapping dataset,⁴³ which had not been seen by the model before to further test the ability of the model on a real task dataset.

2.2 fNIRS Data Simulation

The fNIRS data simulation procedure is presented in Fig. 1(a). The simulated noisy HRF data ($F'(t)$) consist of the superposition of three components: the clean HRF ($F(t)$), the motion artifact ($\Phi_{MA}(t)$) and the resting-state fNIRS ($\Phi_{rs}(t)$) components:

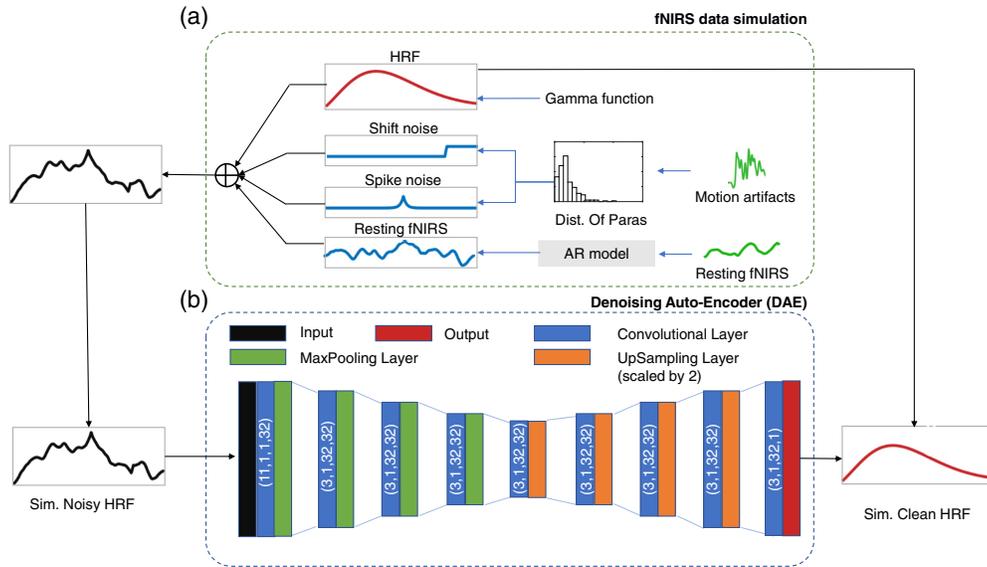


Fig. 1 Illustration of the fNIRS data simulation process and the designed DAE model. (a) The green lines are the experimental fNIRS data, including noisy HRF and resting fNIRS data, while the blue and red lines are simulated ones. The AR models are fitted to the experimental resting-state fNIRS time series data, based on whose parameters the simulated resting fNIRS data are generated. The HRFs are simulated from gamma functions. The shift and spike noise are simulated based on the same distribution of the parameters from the experimental HRF. The simulated noisy HRF data (black line) is the sum of the simulated HRF, the shift noise, the spike noise, and the resting-state fNIRS. (b) DAE model: the input data of the DAE model are the simulated noisy HRF, and the output is the corresponding clean HRF without noise. The DAE model incorporates nine convolutional layers, followed by max-pooling layers in the first four layers and upsampling layers in the next four layers, with one convolutional layer before the output. The parameters are labeled in parentheses for each convolutional layer, in the order of kernel size, stride, input channel size, and kernel number.

$$\underbrace{F'(t)}_{\text{noisy HRF}} = \underbrace{F(t)}_{\text{clean HRF}} + \underbrace{\Phi_{\text{MA}}(t)}_{\text{motion artifacts}} + \underbrace{\Phi_{\text{rs}}(t)}_{\text{resting-state fNIRS}}. \quad (1)$$

We simulated the clean HRF component ($F(t)$) by gamma functions as suggested in steps in Ref. 44. Instead of using $54 \mu\text{M} \cdot \text{mm}$ in all the simulated HRF, we were randomly selecting values from a uniform distribution between 30 and $80 \mu\text{M} \cdot \text{mm}$, such that the DAE could learn from a variety of learning samples. The HbR amplitude is always half of the HbO amplitude. The motion artifacts ($\Phi_{\text{MA}}(t)$) consisted of two types: spike noise and shift noise. Spike artifacts were simulated as Laplace distribution function⁴⁵ given as

$$f(t) = A \cdot \exp\left(-\frac{|t - t_0|}{b}\right), \quad (2)$$

where A represents the peak amplitude, b represents the scale parameter, and t_0 represents the time point of the “peak.” Shift noise was simulated as a positive or negative change in DC value. All the parameters to determine noise functions were first derived from the experimental fNIRS dataset. The parameters for each artifact were then drawn from the parameter values in the experimental data. As described in Ref. 45, the resting-state fNIRS ($\Phi_{\text{rs}}(t)$) was simulated using an AR model that included five lagged terms. We first fitted an AR model to the experimental resting-state data to determine the model parameters. Then, these parameters were used to simulate the resting-state fNIRS.

2.3 Denoising Autoencoder Model Design

The DAE concept was proposed by Vincent et al.,⁴⁶ and it has numerous applications. For our purpose, two essential design criteria needed to be established: architectural hyperparameters and specific loss function. Based on prior knowledge and empirical evidence, our final DAE model consisted of the nine stacked convolutional layers shown in Fig. 1. The convolutional layers were followed by max-pooling layers (the first four layers), followed by the next four upsampling layers, and one more convolutional layer before the output. A smaller network topology containing only four layers produced a significant reduction in performance (see our previous work⁴⁷). The network was trained using backpropagation to minimize the loss function. The time-series data of the simulated noisy HRF ($F'(t)$) was the input data, and the output data were the simulated clean HRF ($F(t)$).

The loss function was specially designed for this problem. First, the loss function minimized the discrepancy between the predicted fNIRS data with the ground truth data. For this purpose, we adopted the mean squared error (MSE) loss function (L_{mse}) here:

$$L_{\text{mse}} = \frac{1}{n} \sum (\hat{y}_i - y_i)^2, \quad (3)$$

where y_i represents ground-truth value and \hat{y}_i represents the predicted value.

Next, the loss function minimized the total variation of the predicted signal:

$$L_{\text{var}} = \frac{1}{n} \sum (\hat{y}_i - \hat{u})^2, \quad (4)$$

where \hat{u} is the mean value of \hat{y}_i .

Finally, the loss function minimized the number of motion artifacts in the output data, which we evaluated using the *hmrMotionArtifact* function in HomER2. The motion artifact detection is twofold: (i) the standard deviation (std) exceeds the standard deviation threshold or (ii) the amplitude change (amp) exceeds the amplitude threshold. Here, we designed the standard deviation loss function (L_{std}) for (i) and the amplitude loss function (L_{amp}) for (ii), as described in Eqs. (5)–(8):

$$L_{\text{std}} = \frac{1}{N_{\text{std}}} \sum_{i=1}^T \sum_{j \in | \Delta_i dc_j > m } \Delta_i dc_j, \quad (5)$$

$$L_{\text{amp}} = \frac{1}{N_{\text{amp}}} \sum_{i=1}^T \sum_{j \in | \Delta_i dc_j > C_{\text{amp}} } \Delta_i dc_j, \quad (6)$$

where

$$m = \sigma_{\Delta_1 dc} \cdot C_{\text{std}}, \quad (7)$$

$$\Delta_i dc_j = dc_j - dc(j+1). \quad (8)$$

In the above equations, dc is the projected optical intensity value derived from the predicted hemoglobin value \hat{y}_i via the Beer–Lambert Law $dc_i = (\hat{y}_i / \epsilon d)$, where ϵ is the attenuation coefficient, and d is the photo path length. N_{std} represents the number of all the time points in $1 \sim T$ and $\Delta_i dc_j > m$; while N_{amp} represents the number of all the time points in $1 \sim T$ and $\Delta_i dc_j > C_{\text{amp}}$. $\sigma_{\Delta_1 dc}$ is the standard deviation of changes in dc in one time step. C_{amp} and C_{std} were motion detection thresholds in amplitude and standard deviation changes. Those values work as the same role as $\text{AMP}_{\text{thresh}}$ and $\text{STDEV}_{\text{thresh}}$ in the HomER2 function *hmrMotionArtifact*. If the signal changes by more than C_{amp} over the time interval, then this time point is marked as a motion artifact. If the signal changes by more than $\text{STDEV}_{\text{thresh}} * \text{stdev}(d)$ over the time interval, then this time point is marked as a motion artifact. The goal of L_{std} and L_{amp} is to decrease the amplitudes of the “jump/peak” signal that are defined

as motion artifacts by the threshold values of C_{amp} and C_{std} . The users can change those values to change their criteria to define and detect motion artifacts as they would do in *homer2*.

Finally, the loss function to train the deep learning network (Loss) is a linear combination of the four individual loss functions above [Eqs. (3)–(6)] with hyperparameters, $\theta_1, \theta_2, \theta_3$ as shown as

$$\text{Loss} = L_{\text{mse}} + \theta_1 \times L_{\text{var}} + \theta_2 \times L_{\text{std}} + \theta_3 \times L_{\text{amp}}, \quad (9)$$

where $\theta_1, \theta_2, \theta_3$ were set to be 1, 1, and 10, respectively, to balance the magnitude of each term.

2.4 Model Training

We trained our DAE model using the PyTorch package in Python 3. We adopted a leave-one-subject-out cross validation scheme by simulating data from experimental data from all subjects but one and then tested the model on the one subject that has been left out. We repeated this procedure until we tested all the subjects. We randomly split the simulated dataset into training, validation, and testing datasets in the ratio of 8:1:1. The model was trained on the training dataset and validated on the validation dataset. After completing the model training, its performance was evaluated using the testing dataset. The learning rate was set at 0.0001 and divided by 10 every 25 epochs. The model was trained for 100 epochs, and the parameters were saved at the epoch corresponding to the least validation loss. The optimization method was ‘‘ADAM.’’⁴⁸ After successful training and validation on the synthetic datasets, the DAE model was applied to the experimental dataset.

2.5 Conventional Denoising Methods

We compared the performance of our deep learning denoising network to models that we obtained by applying the following competitive methods that have recently been proposed in the literature: (i) spline interpolation, (ii) wavelet filtering, (iii) PCA, (iv) Kalman filtering, and (v) Cbsi. These methods have been widely used in the fNIRS community, and we implemented these functions using *HomER2*.⁴⁹ Each method, summarized in Table 1, has been thoroughly discussed, analyzed, and compared in two review papers.^{29,44} Table 1 also summarizes the assumptions imposed on each method and briefly discusses their known drawbacks. The data flow of the data processing of these conventional denoising methods and DAE is presented in Fig. S1 in the [Supplemental Material](#). All methods are performed before block average. The motion artifacts were detected prior to spline by *hmrMotionArtifactByChannel* function from *homer2*. We detected the motion artifacts in the OD signals by *hmrMotionArtifact* for all the methods. For CBSI and DAE, since they work on concentration values, we first converted it to OD value by *hmrConc2OD* in *homer2* and use *hmrMotionArtifact* to detect motion artifacts. The standard deviation threshold ($\text{STDEV}_{\text{thresh}}$ in *homer2*) is set to 20 and the amplitude threshold ($\text{AMP}_{\text{thresh}}$ in *homer2*) is set to 0.3 to detect the motion artifacts in this particular work. But users can change these values according to suggestions in *homer2* or their experience. For PCA, we were using target recursive PCA.²⁷ After the sensitivity analysis, we ended up selecting 0.97 for PCA threshold values and 0.75 for wavelet method (see more details in Sec. 3.2). For Kalman filters, the initial state vector x_0 were set as the first values of the signals, and the initial error covariance matrix P_0 is set as the square of the first values of the signals.

2.6 Metrics for Model Evaluation

To evaluate the denoising performance for each model when applied to the simulated and the experimental dataset, we used the number of motion artifacts remaining after applying the models measured by the *hmrMotionArtifact* function in *HomER2*. The fewer the motion artifacts remaining, the better the model performance. The number of motion artifacts could not reflect how well the methods could reserve the HRF in it, so we introduce another metrics as MSE to access the ability to reserve the true HRF of each model. To further assess the processing speed, we also employed computation time. The experimental data we adopted⁴² are the resting state

data with various purposefully added motion artifacts. There is no brain activation in this dataset. We added stim markers every 73.142 s and then we labeled this dataset as “No act.” in this paper. So in this “No act.” dataset, the HRF should be zeros because there is no brain activation involved. Then we added synthetic HRF (modified gamma function of amplitude of $54 \mu\text{M} \cdot \text{mm}$), to the stim marker locations and labeled this dataset as “Act.” dataset. In this “Act.” dataset, the ground truth HRF should be a modified gamma function with amplitude of $54 \mu\text{M} \cdot \text{mm}$. We used the software G*Power to do power analysis to control type-II error. The power value of the testing dataset is 1.00; the power of No act. experimental dataset is 1.00; the power of Act. experimental dataset is 0.99.

2.7 Sensitivity Analysis

To have a fair comparison, we determined the parameters for the competitive methods to establish models that achieve the best performance. In practice, the parameters could be adjusted by a visual check of each trial data. However, for the large dataset we had in this study, a more automatic method was required. Here, we adopted the sensitivity analysis method⁴⁴ to identify the best set of parameters. For spline interpolation, the interpolation parameter (p_{Spline}) was varied from 0 to 1; for wavelet filtering, the probability threshold (iqr_{Wav}) was varied from 0.1 to 1.5.⁵⁰ The evaluation metrics for these models were MSE and the number of motion artifacts left after applying the models.

3 Results

3.1 Data Simulation

In the dataset we used to train, there were seven subjects, with one run for each subject. We generated 2000 runs of resting data from each subject’s experimental data. We added five trials of HRFs onto each run and added no HRFs to 200 runs out of the 2000 runs to form the training dataset. In other words, we generated 2200 runs from each subject and each run had five trials. Since our model denoised channel by channel, the number of channels was one here. For each round of training of the leave-one-subject-out crossover scheme, we left one subject’s data out. Then, we had 77,000 training trials in total. Figure 2 shows an example of how we extracted motion artifact period and resting period from experimental data, how we determined the shape of the motion artifacts (e.g., height and duration) then simulated motion artifacts based on those parameters, and how we simulated resting fNIRS data based on resting state period data and then synthesize the simulated data. The DAE was applied on each trial before the trial average. The SNRs of the experimental data and the simulated data are presented in Fig. S3 in the [Supplemental Material](#); the SNR values of simulated data are comparable to the experimental data.

3.2 Sensitivity Analysis

The results of the sensitivity analysis are shown in Fig. S2 in the [Supplemental Material](#). For PCA, $\sigma_{\text{PCA}} = 0.99$ yielded the lowest MSE and number of motion artifacts. In the literature, $\sigma_{\text{PCA}} = 0.97$ was selected in Refs. 29 and 44. In this work, we kept 0.97 for σ_{PCA} . For spline interpolation, $p_{\text{Spline}} = 0.99$ yielded the lowest MSE and number of motion artifacts and was also selected in Refs. 21, 29, and 44. Thus, $p_{\text{Spline}} = 0.99$ was selected in this work. For wavelet filtering, we selected $iqr_{\text{Wav}} = 0.75$ to keep both MSE and number of motion artifacts at a low level.

3.3 Comparison of Models on Denoising Performance

We applied each model except PCA on the simulated testing dataset and derived the number of residual motion artifacts. All the motion artifact removal models reduced the number of motion artifacts. Using our DAE model resulted in the minimum number of motion artifacts, with 100%

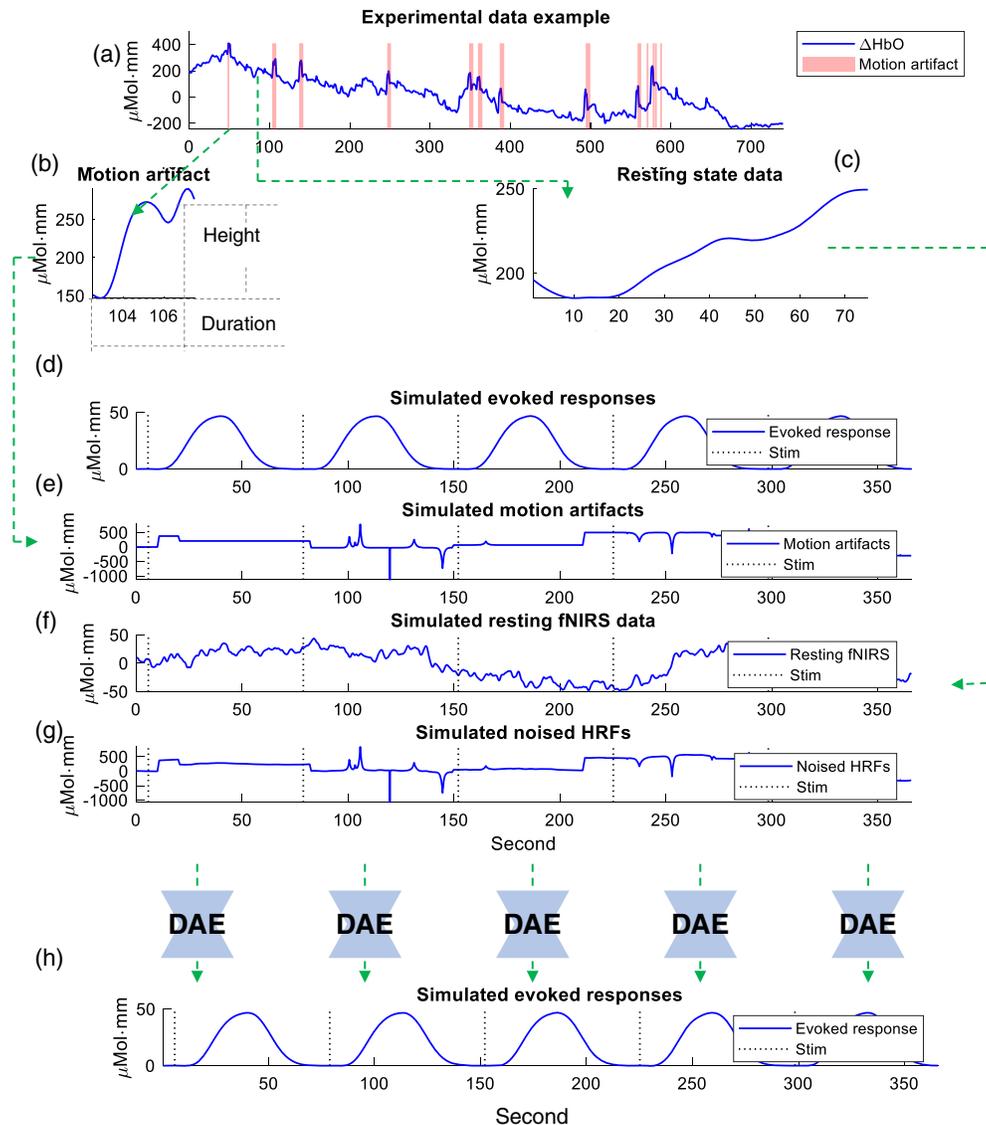


Fig. 2 An example of the fNIRS data simulation process and the designed DAE model. (a) An example of experimental data. (b) The model artifact extracted from the experimental data in (a). (c) The resting state period extracted from the experimental data in (a). (d) Simulated evoked responses. (e) Motion artifact data simulated based on the parameters extracted from the motion artifact in (b). (f) Resting state data simulated based on the data in (c). (g) Synthetic noisy HRFs, which is the sum of data in (d)–(f). (h) The expected output of DAE model, which is the same with the data in (d).

of the motion artifacts removed [Fig. 3(a)]. We then applied each model on experimental data [Fig. 3(b)]. Here, on experimental data, the DAE model also derived the second minimum residual motion artifacts next to the wavelet.

We further visualized examples from simulated and experimental data in Figs. 3(c)–3(h). From the examples, the DAE model smoothed the signal while keep the evoked feature of the HRFs. We further tested our model on a new, publicly available finger-tapping dataset that has never been seen by the model. The testing results are shown in Fig. 4. We can see the residual motion artifact of DAE model is lower than other models [Fig. 4(a)]. The recovered HRF shape in the motor region was also acceptable by visual check [Figs. 4(b)–4(e)].

The MSE values based on HbO for simulation testing dataset, experimental dataset are presented in Table 2 (the results based on HbR are in Table S1 in the [Supplemental Material](#)). The DAE model achieved the lowest MSE in all the datasets. The MSE values were statistically

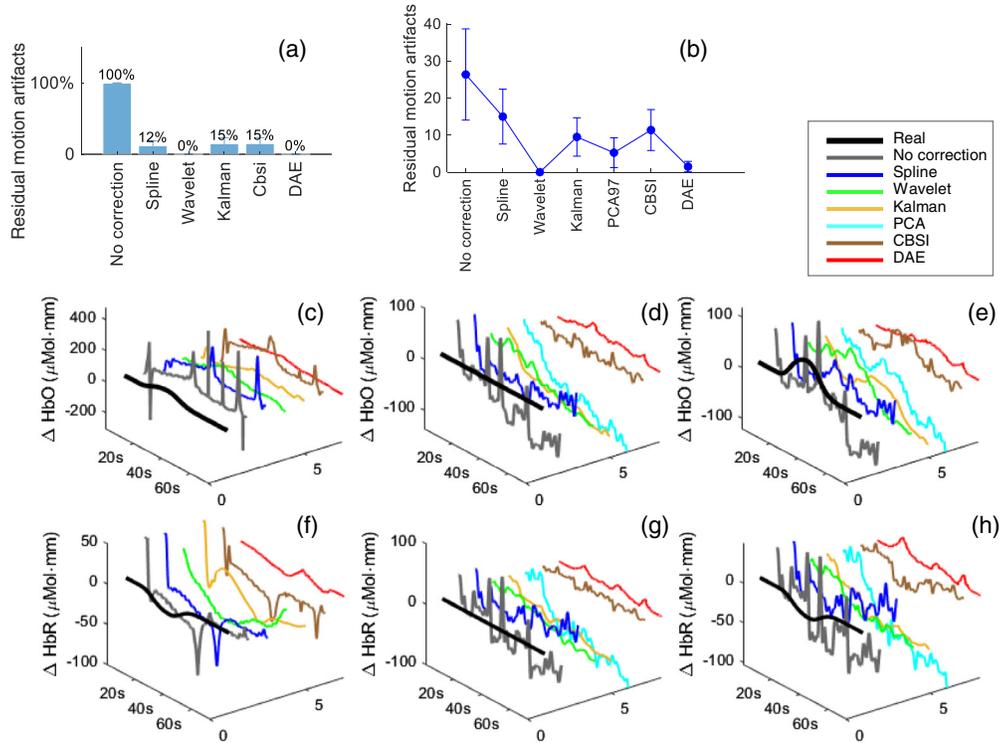


Fig. 3 The denoising results. (a) The number of residual motion artifacts for the simulated testing dataset. (b) The number of residual motion artifacts for experimental data. (c), (f) An example of processed data by different models in the simulated dataset, (d), (g) in experimental data under “No act.” condition, and (e), (h) under “Act.” condition. “No correction” indicates that no motion artifact correction model was used. An enlarged two-dimensional (2D) view of (c)–(h) is in Fig. S4 in the [Supplemental Material](#).

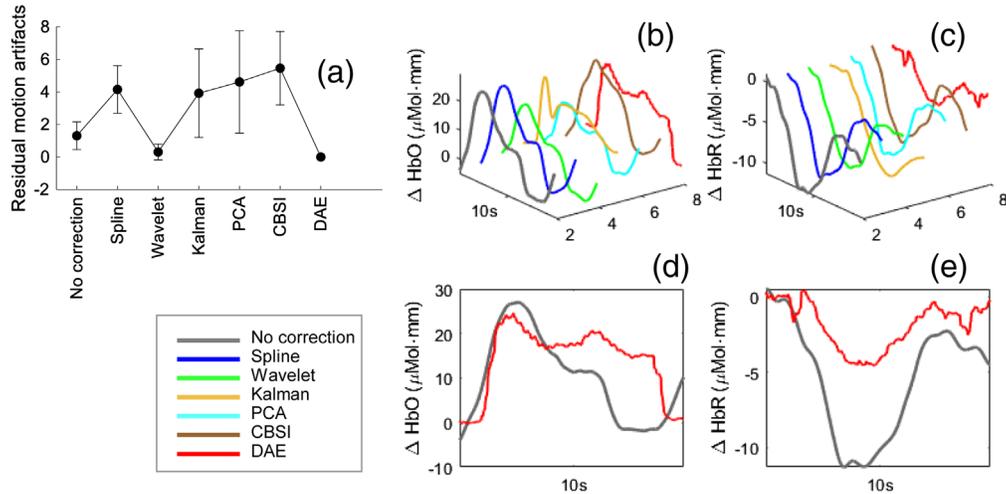


Fig. 4 The denoising results in the new dataset. (a) The number of residual motion artifacts for experimental data. (b), (c) An example of processed data by different models. (d), (e) An enlarged 2D view of (b) and (c).

analyzed comparing between DAE model and each other model. We used paired *t*-test if the samples were normal distributed (tested by Kolmogorov–Smirnov test) and paired sign test if the samples were not normal distributed. The significance level was set at 0.05.

The computation time was also recorded and is presented in Table 3 (CPU: Intel® Core™ i9-9900K CPU @ 3.60 FHz). The DAE model achieved the shortest computational times, with

Table 2 The MSE based on HbO. The median value and the IQR value of MSE for each model and the p -values in the comparison between each model and DAE.

Median (IQR) ((μ Mol \cdot mm) ²)	Sim. testing	Sig. test	Real testing (w/o act.)	Sig. test	Real testing (act.)	Sig. test
No correction	9086.23 (29798.72)	$p = 0.000$	10,663.50 (37,293.42)	$p = 0.000$	10,658.95 (37,320.75)	$p = 0.000$
Spline	3699.08 (9695.42)	$p = 0.000$	11,483.56 (37,471.50)	$p = 0.000$	11,238.91 (35,722.56)	$p = 0.000$
Wavelet	4023.11 (18,566.72)	$p = 0.000$	4438.63 (12524.89)	$p = 0.000$	4500.36 (13281.81)	$p = 0.000$
Kalman	7630.74 (31,175.33)	$p = 0.000$	6309.14 (22,436.39)	$p = 0.000$	6648.38 (23,806.81)	$p = 0.000$
PCA	—	—	11,362.13 (47,714.05)	$p = 0.000$	10,717.19 (48,408.62)	$p = 0.000$
Cbsi	4273.20 (15,380.82)	$p = 0.000$	3899.61 (22613.30)	$p = 0.000$	4198.70 (22,892.76)	$p = 0.000$
DAE	144.19(214.42)	—	226.81(121.91)	—	296.55(58.60)	—

Table 3 The computation time for each model on testing data.

Model	Computation time (s)
Spline	8.7
Wavelet	1202.1
Kalman	76.3
PCA	6.9
Cbsi	4.9
DAE	2.4

Cbsi as the second and PCA as the third. This result is expected as DAE, Cbsi, and PCA only perform matrix multiplication. The spline method is no more than cubic interpolation, but it requires a prior motion artifact detection step, which makes it slower than expected. Notably, wavelet methods are computationally costly.

4 Discussion and Conclusion

We introduced a deep learning DAE network to suppress motion artifacts in recorded fNIRS data. To train a DAE network, we used simulated data from an AR model. We validated the efficiency of the trained DAE network by comparing its performance to models obtained from other conventional modeling methods when applied to both simulated and experimental fNIRS data. The data results showed that the DAE model yielded the lowest residual motion artifacts and MSE compared with competitive models. The DAE model was also the fastest in computation.

The limitation of labeled training data often jeopardizes the effectiveness of deep neural networks.⁵¹ Self-supervised learning^{52,53} has been widely used to overcome this limitation. Specifically, various pretext tasks have been used to train the deep network,⁵⁴ such as colorizing

grayscale images,⁵⁵ image inpainting,⁵⁶ and image jigsaw puzzle.⁵⁷ In these tasks, inputs and labels were synthesized from the unlabeled training set. Then, these synthetic samples were used for training. In the work of Ref. 44, an fNIRS synthesis method was proposed as adding known HRFs to experimental resting-state data. However, this method was limited by the experimental resting-state fNIRS data available. As in our problem, we introduced a similar strategy to train the proposed deep network, but by simulating the resting-state fNIRS data by the AR model. Eighty thousand samples were simulated in this work to be able to cover the large range of HRF and noise parameters obtained from the experimental dataset. The trained DAE model was demonstrated to be efficient in fNIRS denoising. In the future, this fNIRS data synthesis scheme could be harnessed to develop more powerful deep learning models of fNIRS data processing. When applied to other real datasets, the range of HRF amplitude in the training data could be adjusted to the specific range. For example, in this paper, the range was predefined as 30 to 80 $\mu\text{M} \cdot \text{mm}$ and the ground truth (simulated HRF amplitude) in real data is 54 $\mu\text{M} \cdot \text{mm}$. So the true HRF is covered by the predefined HRF range. However, in most cases when we do not know what the true HRF amplitude is, we suggest to use a larger predefined HRF range. Other than that, larger training sample size would also give optimized performance.

We simulated our dataset aiming to train our DAE model in this work. However, the synthetic dataset might be biased unfavorably to other models here by violating their assumptions. For example, PCA depends heavily on the covariance of motion artifacts across the channels. Our simulated dataset was not designed to have multiple motion artifacts across the channels at a similar time point. Thus, we did not apply PCA onto our simulation dataset.

We would also like to mention that we did not simulate our data in a way that the motion artifacts were locked with the evoked signal, which is a feature of motion artifacts in some study paradigms. Our simulated motion artifacts were randomly distributed along the time regardless of the stimulus. In the case that motion artifacts are locked with the stimulus, general linear model (GLM) could be a better option, since it is intrinsically robust to motion artifact locked by the evoked signal. Also, GLM is capable of including variety of regressors, such as short separator and accelerometer data to directly regress out motion artifacts. Even though we did not consider this specific scenario herein, it could be the focus of future work, including the use of DAE model incorporating short separator and accelerometer.

Multiple models have been suggested in the past to remove motion artifacts in fNIRS data. However, the effectiveness of each model depends on the underlying assumptions and the experimental protocol. For example, PCA has been used to remove systemic physiology and motion artifacts that were common across the channels in infant fNIRS data.⁵⁸ Still, the effects changed with the extent of principal component filtering.⁵⁸ In another study,²⁷ targeted PCA performed better than wavelet filtering and spline interpolation when the motion artifacts were temporally independent of the stimuli. Since our data simulation method did not guarantee the covariance of motion artifacts across the channels, we did not expect PCA would work well on our simulated data. PCA is also sensitive to parameter changes. Cooper et al.⁴⁴ compared PCA, wavelet, and spline on a synthetic HRF dataset. Their results demonstrated that spline produced the most significant improvement for MSE and R^2 , while wavelet analysis produced the highest increase in CNR. Our simulation results also showed that the spline and wavelet models decreased MSE. In another study,²⁹ wavelet and Cbsi were shown to be most effective in a real speech study where motion artifacts coincided with stimuli. However, Cbsi did not show superior performance here.

It is also worth mentioning that motion artifact correction is also possible using hardware design changes. For example, an acceleration sensor could be used in experiments to capture the motion data.^{59,60} Short separation detectors were proposed to remove the physiological artifacts⁶¹⁻⁶⁵ and could also be used to remove motion artifacts.⁶⁶ It could also be combined with other filter models to better remove motion artifacts.⁶⁷ However, our work did not consider such hardware designs. Coupling of DAE with hardware solutions may be of interest in future studies.

Though real-time motion artifact removal was not the goal of this study, such requirements have been considered in Refs. 68 and 69. Our research indicates the superiority of DAE compared to other models with regard to computational speed. However, the design and software implementation of our current DAE model were not focusing on computational efficiency. It is

well appraised that increased network complexity leads to increased inference time. Hence, future studies could focus on optimizing network architecture toward increased computational efficiency. Moreover, the custom GPU implementation of the DAE model is expected to greatly improve inference speed. Last, deep learning models, such as LSTM, could be used for real-time application.

In conclusion, we demonstrated that our DAE model has promising performances in lowering the number of motion artifacts and decreasing MSE metrics. To train the DAE model, we suggested an fNIRS synthesis method to generate a large amount of fNIRS data. The results showed the potential suitability and superiority of DAE for fNIRS motion artifact removal.

Disclosures

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Acknowledgments

We are thankful for funding provided by the National Institutes of Health, National Institute of Biomedical Imaging and Bioengineering, Grant Nos. 2R01EB005807, 5R01EB010037, 1R01EB009362, 1R01EB014305, and R01EB019443; the Medical Technology Enterprise Consortium, Grant No. 20-05-IMPROVE-004; and the US Army Combat Capabilities Development Command, Grant No. W912CG2120001. We gratefully acknowledge the support of NVIDIA Corporation with the donation of the Titan Xp GPU and Quadro P6000 used in this research.

Code Availability

The code associated with this manuscript is available at https://github.com/YuanyuanGao216/fNIRS_denoise_by_DL

References

1. F. F. Jobsis, "Noninvasive, infrared monitoring of cerebral and myocardial oxygen sufficiency and circulatory parameters," *Science* **198**(4323), 1264–1267 (1977).
2. S. Lloyd-Fox, A. Blasi, and C. Elwell, "Illuminating the developing brain: the past, present and future of functional near infrared spectroscopy," *Neurosci. Biobehav. Rev.* **34**(3), 269–284 (2010).
3. H. Obrig et al., "Near-infrared spectroscopy: does it function in functional activation studies of the adult brain?" *Int. J. Psychophysiol.* **35**(2–3), 125–142 (2000).
4. H. Obrig and A. Villringer, "Beyond the visible—imaging the human brain with light," *J. Cereb. Blood Flow Metab.* **23**(1), 1–18 (2003).
5. S. Cutini et al., "Selective activation of the superior frontal gyrus in task-switching: an event-related fNIRS study," *Neuroimage* **42**(2), 945–955 (2008).
6. F. Fishburn et al., "Sensitivity of fNIRS to cognitive state and load," *Front. Hum. Neurosci.* **8**, 76 (2014).
7. A. Köchel et al., "Affective perception and imagery: a NIRS study," *Int. J. Psychophysiol.* **80**(3), 192–197 (2011).
8. S. Skau et al., "Mental fatigue and functional near-infrared spectroscopy (fNIRS) – based assessment of cognitive performance after mild traumatic brain injury," *Front. Hum. Neurosci.* **13**, 145 (2019).
9. S. V. Tupak et al., "Differential prefrontal and frontotemporal oxygenation patterns during phonemic and semantic verbal fluency," *Neuropsychologia* **50**(7), 1565–1569 (2012).
10. S. Brigadoi et al., "Exploring the role of primary and supplementary motor areas in simple motor tasks with fNIRS," *Cognit. Process.* **13**(1), 97–101 (2012).

11. A. Nemani et al., “Assessing bimanual motor skills with optical neuroimaging,” *Sci. Adv.* **4**(10), eaat3807 (2018).
12. A. Nemani et al., “Objective assessment of surgical skill transfer using non-invasive brain imaging,” *Surg. Endosc.* **33**, 2485–2494 (2018).
13. Y. Gao et al., “Functional brain imaging reliably predicts bimanual motor skill performance in a standardized surgical task,” *IEEE Trans. Biomed. Eng.* **68**(7), 2058–2066 (2021).
14. Y. Gao et al., “Decreasing the surgical errors by neurostimulation of primary motor cortex and the associated brain activation via neuroimaging,” *Front. Neurosci.* **15**, 651192 (2021).
15. S. Perrey, “Non-invasive NIR spectroscopy of human brain function during exercise,” *Methods* **45**(4), 289–299 (2008).
16. N. Naseer and K.-S. Hong, “fNIRS-based brain-computer interfaces: a review,” *Front. Hum. Neurosci.* **9**, 3 (2015).
17. S. L. Novi et al., “Functional near-infrared spectroscopy for speech protocols: characterization of motion artifacts and guidelines for improving data analysis,” *Neurophotonics* **7**(1), 015001 (2020).
18. M. Suzuki et al., “Activities in the frontal cortex and gait performance are modulated by preparation. An fNIRS study,” *Neuroimage* **39**(2), 600–607 (2008).
19. P. Pinti et al., “A review on the use of wearable functional near-infrared spectroscopy in naturalistic environments,” *Jpn. Psychol. Res.* **60**(4), 347–373 (2018).
20. A. von Lüthmann et al., “Towards neuroscience in the everyday world: progress in wearable fNIRS instrumentation and applications,” in *Opt. and the Brain*, Optical Society of America, p. BM3C-2 (2020).
21. F. Scholkmann et al., “How to detect and reduce movement artifacts in near-infrared imaging using moving standard deviation and spline interpolation,” *Physiol. Meas.* **31**, 649–662 (2010).
22. B. Molavi and G. A. Dumont, “Wavelet-based motion artifact removal for functional near-infrared spectroscopy,” *Physiol. Meas.* **33**, 259–270 (2012).
23. Y. Zhang et al., “Eigenvector-based spatial filtering for reduction of physiological interference in diffuse optical imaging,” *J. Biomed. Opt.* **10**(1), 011014 (2005).
24. M. Izzetoglu et al., “Motion artifact cancellation in NIR spectroscopy using discrete Kalman filtering,” *Biomed. Eng. Online* **9**, 16 (2010).
25. X. Cui, S. Bray, and A. L. Reiss, “Functional near infrared spectroscopy (NIRS) signal improvement based on negative correlation between oxygenated and deoxygenated hemoglobin dynamics,” *Neuroimage* **49**(4), 3039–3046 (2010).
26. K.-E. Jang et al., “Wavelet minimum description length detrending for near-infrared spectroscopy,” *J. Biomed. Opt.* **14**(3), 034004 (2009).
27. M. A. Yücel et al., “Targeted principle component analysis: a new motion artifact correction approach for near-infrared spectroscopy,” *J. Innov. Opt. Health Sci.* **7**(02), 1350066 (2014).
28. B. Yazıcı et al., “Kalman filtering for self-similar processes,” *Signal Process.* **86**(4), 760–775 (2006).
29. S. Brigadoi et al., “Motion artifacts in functional near-infrared spectroscopy: a comparison of motion correction techniques applied to real cognitive data,” *Neuroimage* **85**, 181–191 (2014).
30. H. C. Burger, C. J. Schuler, and S. Harmeling, “Image denoising: can plain neural networks compete with BM3D?” in *IEEE Conf. Comput. Vision and Pattern Recognit.*, IEEE, pp. 2392–2399 (2012).
31. K. Cho, “Boltzmann machines and denoising autoencoders for image denoising,” arXiv:1301.3468 (2013).
32. V. Jain and S. Seung, “Natural image denoising with convolutional networks,” in *Adv. Neural Inf. Process. Syst.*, Vol. **21**, pp. 769–776 (2008).
33. J. Xie, L. Xu, and E. Chen, “Image denoising and inpainting with deep neural networks,” in *Adv. Neural Inf. Process. Syst.*, Vol. **25**, pp. 341–349 (2012).
34. L. Gondara, “Medical image denoising using convolutional denoising autoencoders,” in *IEEE 16th Int. Conf. Data Mining Workshops (ICDMW)*, IEEE, pp. 241–246 (2016).
35. D. Lee, S. Choi, and H.-J. Kim, “Performance evaluation of image denoising developed using convolutional denoising autoencoders in chest radiography,” *Nucl. Instrum.*

- Methods in Phys. Res. Sect. A: Accel., Spectrom, Detect. and Assoc. Equip.* **884**, 97–104 (2018).
36. Z. Yang et al., “A robust deep neural network for denoising task-based fMRI data: an application to working memory and episodic memory,” *Med. Image Anal.* **60**, 101622 (2020).
 37. M. J. Muckley et al., “Training a neural network for Gibbs and noise removal in diffusion MRI,” *Magn. Reson. Med.* **85**(1), 413–428 (2021).
 38. N. M. N. Leite et al., “Deep convolutional autoencoder for EEG noise filtering,” in *IEEE Int. Conf. Bioinf. and Biomed. (BIBM)*, IEEE, pp. 2605–2612 (2018).
 39. G. Lee, S. H. Jin, and J. An, “Motion artifact correction of multi-measured functional near-infrared spectroscopy signals based on signal reconstruction using an artificial neural network,” *Sensors* **18**(9), 2957 (2018).
 40. J. Russell-Buckland et al., “Abroad: a machine learning based approach to detect broadband NIRS artefacts,” in *Oxygen Transport to Tissue XL*, O. Thews, J. C. LaManna, and D. K. Harrison, Eds., pp. 319–324, Springer International Publishing, Cham (2018).
 41. M. A. Yücel et al., “Reducing motion artifacts for long-term clinical NIRS monitoring using collodion-fixed prism-based optical fibers,” <https://www.nitrc.org/frs/downloadlink.php/11882> (2020).
 42. M. A. Yücel et al., “Reducing motion artifacts for long-term clinical NIRS monitoring using collodion-fixed prism-based optical fibers,” *Neuroimage* **85**, 192–201 (2014).
 43. M. A. Yücel et al., “Specificity of hemodynamic brain responses to painful stimuli: a functional near-infrared spectroscopy study,” https://www.nitrc.org/frs/?group_id=1456 (2020).
 44. R. Cooper et al., “A systematic comparison of motion artifact correction techniques for functional near-infrared spectroscopy,” *Front. Neurosci.* **6**, 147 (2012).
 45. J. W. Barker, A. Aarabi, and T. J. Huppert, “Autoregressive model based algorithm for correcting motion and serially correlated errors in fNIRS,” *Biomed. Opt. Express* **4**(8), 1366–1379 (2013).
 46. P. Vincent et al., “Extracting and composing robust features with denoising autoencoders,” in *Proc. 25th Int. Conf. Mach. Learn.*, pp. 1096–1103 (2008).
 47. Y. Gao et al., “A deep learning approach to remove motion artifacts in fNIRS data analysis,” in *Opt. and the Brain*, Optical Society of America, p. BM2C-7 (2020).
 48. D. P. Kingma and J. Ba, “Adam: a method for stochastic optimization,” arXiv:1412.6980 (2014).
 49. T. J. Huppert et al., “Homer: a review of time-series analysis methods for near-infrared spectroscopy of the brain,” *Appl. Opt.* **48**(10), D280–D298 (2009).
 50. R. Di Lorenzo et al., “Recommendations for motion correction of infant fNIRS data applicable to multiple data sets and acquisition systems,” *NeuroImage* **200**, 511–527 (2019).
 51. K. S. Button et al., “Power failure: why small sample size undermines the reliability of neuroscience,” *Nat. Rev. Neurosci.* **14**(5), 365–376 (2013).
 52. A. Kolesnikov, X. Zhai, and L. Beyer, “Revisiting self-supervised visual representation learning,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 1920–1929 (2019).
 53. T. Nathan Mundhenk, D. Ho, and B. Y. Chen, “Improvements to context based self-supervised learning,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 9339–9348 (2018).
 54. L. Jing and Y. Tian, “Self-supervised visual feature learning with deep neural networks: a survey,” *IEEE Trans. Pattern Anal. Mach. Intell.* **43**(11), 4037–4058 (2020).
 55. R. Zhang, P. Isola, and A. A. Efros, “Colorful image colorization,” in *Eur. Conf. Comput. Vision*, Springer, pp. 649–666 (2016).
 56. D. Pathak et al., “Context encoders: feature learning by inpainting,” in *Proc. IEEE Conf. Comput. Vision and Pattern Recognit.*, pp. 2536–2544 (2016).
 57. M. Noroozi and P. Favaro, “Unsupervised learning of visual representations by solving jigsaw puzzles,” in *Eur. Conf. Comput. Vision*, Springer, pp. 69–84 (2016).
 58. T. Wilcox et al., “Using near-infrared spectroscopy to assess neural activation during object processing in infants,” *J. Biomed. Opt.* **10**(1), 011010 (2005).
 59. T. Hiroyasu, Y. Nakamura, and H. Yokouchi, “Method for removing motion artifacts from fNIRS data using ICA and an acceleration sensor,” in *35th Annu. Int. Conf. IEEE Eng. in Med. and Biol. Soc. (EMBC)*, IEEE, pp. 6800–6803 (2013).

60. C.-K. Kim et al., “Development of wireless NIRS system with dynamic removal of motion artifacts,” *Biomed. Eng. Lett.* **1**(4), 254–259 (2011).
61. L. Gagnon et al., “Short separation channel location impacts the performance of short channel regression in NIRS,” *Neuroimage* **59**(3), 2518–2528 (2012).
62. L. Gagnon et al., “Improved recovery of the hemodynamic response in diffuse optical imaging using short optode separations and state-space modeling,” *Neuroimage* **56**(3), 1362–1371 (2011).
63. R. B. Saager and A. J. Berger, “Measurement of layer-like hemodynamic trends in scalp and cortex: implications for physiological baseline suppression in functional near-infrared spectroscopy,” *J. Biomed. Opt.* **13**(3), 034017 (2008).
64. R. B. Saager, N. L. Telleri, and A. J. Berger, “Two-detector corrected near infrared spectroscopy (C-NIRS) detects hemodynamic activation responses more robustly than single-detector NIRS,” *Neuroimage* **55**(4), 1679–1685 (2011).
65. Q. Zhang, G. E. Strangman, and G. Ganis, “Adaptive filtering to reduce global interference in non-invasive NIRS measures of brain activation: how well and when does it work?” *Neuroimage* **45**(3), 788–794 (2009).
66. F. C. Robertson, T. S. Douglas, and E. M. Meintjes, “Motion artifact removal for functional near infrared spectroscopy: a comparison of methods,” *IEEE Trans. Biomed. Eng.* **57**(6), 1377–1387 (2010).
67. S. Dong and J. Jeong, “Noise reduction in fNIRS data using extended Kalman filter combined with short separation measurement,” in *6th Int. Conf. Brain-Comput. Interface (BCI)*, IEEE, pp. 1–3 (2018).
68. J. W. Barker et al., “Correction of motion artifacts and serial correlations for real-time functional near-infrared spectroscopy,” *Neurophotonics* **3**(3), 031410 (2016).
69. T. Nozawa and T. Kondo, “A comparison of artifact reduction methods for real-time analysis of fNIRS data,” in *Symp. Hum. Interface*, Springer, pp. 413–422 (2009).

Biographies of the other authors are not available.