Deep Learning Approaches for Detection and Removal of Ghosting Artifacts in MR spectroscopy

Sreenath P Kyathanahally[1,2], André Döring[1,2], and Roland Kreis[1]

[1]Depts. Radiology and Biomedical Research, University of Bern, Bern, Switzerland,

[2] Graduate School for Cellular and Biomedical Sciences, University of Bern, Bern, Switzerland

Word count: 5193

Corresponding author:

Prof. Dr. sc. nat. Roland Kreis,

AMSM, University Bern,

Erlachstrasse 9a,

CH-3012 Bern, Switzerland

Tel: +41-31-632 8174

Email: roland.kreis@insel.ch

Abstract:

Purpose: To make use of “Deep Learning” (DL) methods to detect and remove ghosting artifacts in clinical magnetic resonance spectra of human brain.

Methods: DL algorithms including “Fully Connected Neural Networks”, “Deep-Convolutional Neural Networks”, and “Stacked What-Where Auto Encoders” were implemented to detect and correct MR spectra containing spurious echo ghost signals. DL methods were trained on a huge database of simulated spectra with and without ghosting artifacts that represent complex variations of ghost-ridden spectra, transformed to time-frequency spectrograms. The trained model was tested on simulated and in-vivo spectra.

Results: The preliminary results for ghost detection are very promising; reaching almost 100% accuracy and also the DL ghost removal methods show potential in simulated and in-vivo spectra, but need further refinement and quantitative testing.

Conclusions: Ghosting artifacts in spectroscopy are problematic since they superimpose with metabolites and lead to inaccurate quantification. Detection and removal of ghosting artifacts using traditional machine learning approaches with feature extraction/selection is difficult since ghosts appear at different frequencies. Here, we show that DL methods perform extremely well for ghost detection if the spectra are treated as images in the form of time-frequency representations. Further optimization for in-vivo spectra will hopefully confirm their “ghostbusting” capacity.

Keywords: artifacts, magnetic resonance spectroscopy, quality control, deep learning, machine learning, time-frequency representation, human brain

Introduction:

One of the major problems in clinical magnetic resonance spectroscopy (MRS), which has hindered its widespread use, is the need for local human expertise for quality assessment - given that artefacts are often not eye-catching for the inexperienced user (1). There has been recent progress for automated quality filtering based on machine learning techniques (2–5), but their general use is by far not established and thus automatic detection of specific artifacts in spectra - and even more so the restoration of the affected spectra - would be extremely valuable to enhance the routine use of clinical MRS. One particular type of artifact that is targeted in this work is the so-called ghost or spurious echo. In spectroscopy, spurious echoes result from insufficient spoiling gradient power in combination with local susceptibility variations. Ghosting artifacts are problematic because they superimpose with metabolite peaks at varying frequencies and may thus preclude reliable area estimation. Ghost artifact detection by human experts is possible but takes time, is subjective, and obviously relies on local expertise. So, automatic ghost detection using machine learning could be a useful alternative method.

Machine Learning (ML) methods are increasingly used in countless applications like face recognition or spam detection (4,6,7). In the context of MRS, ML has primarily been used for optimized automated classification of tumor spectra (8–12) and for quality assessment (2–5). For the latter, it has been shown that the performance of ML can reach the efficiency of humans. Traditional ML techniques for classification/regression involve first extraction/selection of the relevant features and - based on them - subsequently the training of the classifier/regression models for prediction.

Feature engineering is considered as one of the important tasks in traditional ML methods since the features that are selected/extracted from the data directly influence the results that can be achieved. Feature extraction/selection allows to reduce the amount of information necessary to represent the data by using less complex models but covering most of the variance of the data. The selected features have specific functions and play an important role in the performance of the ML method. For example: the signal to noise ratio (SNR) and skewness have turned out to be some of most important features for the detection of bad quality spectra (3,5). Obtaining good ML results with less than optimal models is feasible, but getting good results with ill-chosen features is hardly possible.

Feature engineering gets difficult sometimes, for example in object detection, if the same object is viewed from different angles or different positions or with different illuminations. Though it is the same object, it is not easy for a machine to detect this accurately if using traditional features. Also, if the objects are not completely visible, for instance a dog partly hidden in a bush, the machine would often fail to detect the targeted object. If the machine must be taught to detect correctly even with different positions, angles, illumination etc., then the features should be extracted for different combinations of positions, illuminations etc., which is tedious.

Similar to the problem of object detection, extracting/selecting features for ghosting artifacts in a spectrum is challenging, since ghosts can appear at different frequencies. It would be difficult to catch ghosting artifacts as specific features. One way to circumvent this problem is to use a subset of ML approaches called “Deep learning” (DL). DL extracts features automatically without human intervention. It consists of several hidden layers of artificial neurons with each layer trying to exploit the underlying structure of the input data. The initial layers extract the lower level features and the subsequent layers extract the hierarchy of high level features often defined in terms of low level features. This kind of feature extraction in multiple layers allows the system to model complex variations of the input data as weighted sum of all the features in each layer and predict accurately. The key aspect of DL is that the features in each layer are learned from the data itself and not designed by humans.

There are several types of DL approaches that can be used for different purposes, such as image reconstruction, image recognition, image segmentation, speech recognition, tumor detection etc. Some of the well-known DL algorithms are Fully Connected Neural Networks (FCNNs), Deep-Convolutional Neural Networks (D-CNNs) (13), Stacked What-Where Auto Encoders (SWWAEs) (14) built on Residual Block networks (15) and recurrent networks.

There has been a lot of progress in recent years in the field of image processing and analysis as a consequence of the amazing increase in computing power on one side, but also the boom of DL models (13,16). Especially, D-CNNs have shown a great impact in many fields in medical imaging and often they have shown better performance than most of the-state-of-art techniques (17). In a breakthrough paper, Krizhevsky et al. (13) showed excellent results on the ImageNet LSVRC-2010 contest where they trained 1.2 million high resolution images and classified them into 1000 different classes. Extremely large data sets are necessary to obtain such outstanding results.

In medical imaging, such databases do not yet exist or are not openly available due to various reasons including ethical issues. Especially in MR spectroscopy, getting thousands of spectra with labeled artifacts is essentially impossible. Hence, for exploring DL in MRS, we turned to simulation of spectra and artifacts. Brain metabolite spectra with and without ghosting artifacts were simulated such that they best mimic in-vivo spectra. We then explored detection of ghosting artifacts using FCNNs & D-CNNs. Simulated spectra were used for training, while performance evaluation was based on both unseen simulated and in-vivo spectra. Finally, we also suggest ways to remove ghosting artifacts from affected spectra using SWWAE built on Residual Blocks networks.

Overall, deep learning has been introduced as a novel way to deal with artifacts in MRS. Ghosting artifacts were chosen as a well defined example, rather than for its predominance, which is very hard to judge. It is hoped that the surprising initial success of this approach will lead to innovative uses of deep learning in the field of MRS.

Methods:

MR spectra and processing

Simulated spectra: Brain metabolite spectra were simulated in VESPA (18) for ideal PRESS localization (echo time of 30 ms) at 3T and scaled to yield normal brain spectra based on concentrations and T_2 values from the literature (19,20). A macromolecular baseline (MMBL) spectrum was also added at short TE (fitted MMBL extracted from experimental averaged metabolite-nulled gray matter spectra of 10 subjects). These in silico spectra were then used as basis to generate spectra with different linewidths (LW = 5, 7, 9, 11, 13 Hz) and signal to noise ratios (SNRs) (10, 20, 40, 60, 100 – as defined in time domain (TD)). Ghosting artifacts with varying linewidths and amplitudes were simulated and added randomly as echo signals with different time and frequency shifts. Each simulated spectrum consisted of 2048 complex points for a spectral width of 4000 Hz.

Two cohorts of spectra were created. Group-1 consisted of 30000 spectra with and the identical 30000 spectra without ghosts while Group-2 consisted of 15000 spectra with and the same 15000 spectra without ghosts. Group-1 contained spectra with the same brain metabolite content, but varied in LW and SNR (**Figure 1**). Group-2 enclosed spectra with variations in

SNR, LW, metabolite concentrations, macromolecular baseline (MMBL) intensity, and also contributions from lipids (**Figure 2a**).

In-vivo spectra: Thirteen healthy subjects gave informed consent according to the procedure approved by the local ethics committee. The scans were performed on a Siemens 3T scanner (Prisma, Siemens, Erlangen, Germany). A non-water suppressed diffusion-weighted sequence based on metabolite-cycled STEAM (21) with CSF nulling by water-selective inversion recovery was applied in occipital gray matter (inversion time/echo time/mixing time/repetition time = 1500/37/150/3500 ms; diffusion gradient length 11 ms, diffusion time 168ms, maximum diffusion gradient amplitude 38 mT/m, maximal b-value 5236 s/mm²). The residual water signal was eliminated using Hankel-Lankosz singular value decomposition (HLSVD)(22) in postprocessing. Thirty-two percent of the in-vivo data had ghosting artifacts (illustrated with sample spectra in **Figure 2b**).

Spectrograms: Since D-CNNs have been shown to work very well for 2D images, the 1D spectra were converted to 2D images. Given that spurious echoes appear in the free induction decay (FID) with variable time shifts, it appeared promising to use a 2D time-frequency representation of the spectra (23,24). In analogy to a Gabor transform (25), but with inherent reduction in data-size, the simulated time domain signals were transformed into spectrograms (26), which are well-known data representations in the field of audio signal processing. A spectrogram is a representation of the spectrum of frequencies that vary with time. If a Fourier transform is applied to the whole TD signal, it converts the TD signal into a frequency domain (FD) signal, which we call spectrum. If a short-time Fourier transform (STFT) is applied for subsequent short parts of the FID a 2D representation of the data is produced where each column represents the frequency content of a particular piece of the FID. Depending on the size of the Fourier analysis window, different levels of frequency and time resolution are achieved. If the size of the window is long the transform favors frequency at the expense of time resolution and one arrives at a narrow-band spectrogram. If a short analysis window is chosen, time resolution is better, but frequencies are smeared. The result is a wide-band spectrogram. We used STFT with a window size of 128 samples (≡32 ms), overlap interval of 30 samples and a Hanning window to get complex spectrograms with 70 frequency bins and 20 time stamps (70x20 matrix) (**Figure 3**) and also as a variant we used STFT with a window size of 56 samples (≡14 ms), overlap interval of 48 samples and a Hanning window to get complex spectrograms with 56 frequency bins and 250 time stamps (56x250 matrix).

Deep learning (DL)

Simulated spectra were used to train DL models that were then used for predicting previously unused simulated and in-vivo spectra.

Scripts were written in Python using the Keras library (27) on top of the Theano (28) backend to build the DL models. Creating any DL neural network consists of
1) defining, 2) compiling, 3) training, 4) evaluating, and 5) testing the model.

Models are defined in Keras as a sequence of layers. The layers are added one at a time until a satisfying network topology is found. First, the number of neurons in each layer is specified, and then the network weights are initialized to a small random number generated from a Gaussian distribution. The hidden layers are set with activation function and the output layer is set with a sigmoid activation function if the model is used for classification. The sigmoid function on the output layer ensures that the network output is between 0 and 1 for a binary classification task.

During compilation, additional properties have to be specified: a) the loss function to evaluate a set of weights, b) the optimizer to search through different weights for the network, and c) optional metrics that should be collected and reported during training.

During the training process, the weights in the network get updated after a certain number of instances, specified as batch size. The training process runs for a fixed number of iterations through the dataset and is governed by an argument called epochs.

After training the network using a part of the whole dataset, it is evaluated on the test dataset that was not seen by the model during training. The final model is then used for new predictions, including in-vivo data in our case.

DL for ghost identification

For the identification of ghosting artifacts, we defined DL models using FCNNs and D-CNNs.

A FCNN receives the inputs as vectors and transforms them through a series of hidden layers, in which each hidden layer consists of a set of neurons. Each neuron is fully connected to all neurons in the previous layer, but the neurons of the same layer do not share any connections with each other. The final layer is called the “output layer” and represents the class scores in classification settings. The FCNN was modeled by adding different hidden layers one at a time. The network topology used and the neurons in each hidden layer are shown in in the Supporting Figure S1). The network weights were initialized from random numbers generated from a Gaussian distribution. The Rectified Linear Unit (ReLU) function for hidden layers and the sigmoid function for the output layer were set. The Adam gradient descent algorithm (29) was chosen as an optimizer during compilation and the binary cross entropy as the loss function. The batch size was set to 50 and the number of epochs to 5.

Approximately two thirds of the Group-1 spectra, i.e. 20000 simulated spectra, were used as input for training of the FCNN and then tested on one third of the data (10000 spectra). Classification accuracy was evaluated based on the predicted and the true class labels.

D-CNNs have shown an outstanding performance and reliable results for image recognition and detection (13). D-CNNs work well with images since - unlike the FCNNs - they take advantage of spatial properties of images. FCNNs treat input pixels that are far apart and close together with the same priority since the inputs are depicted as a vertical line of neurons, whereas D-CNNs take advantage of these spatial structures to build a more sensible architecture. DL models using D-CNNs usually comprise four operations:
1) convolution, 2) max-pooling, 3) non-linear activation, and 4) fully connected layer classification.

In the convolution step, the feature maps are extracted by convolving a kernel across the input image. In the max-pooling step, the dimensionality of each feature map is reduced, retaining only the most important information by using the maximum of the defined neighborhood region. In the non-linear activation step, the ReLU function is used that replaces all negative pixel values in the feature maps by zeroes and preserves the positive pixel values. Finally, in the fully connected layer, the combination of feature maps from convolution and pooling steps are used for classification of the input data into various classes.

In the presented application, the D-CNN was modeled by adding different hidden layers one at a time. The network architecture is shown in **Figure 4**. Each hidden layer consisted of two 2D convolutional layers and a max-pooling layer (30). Beyond the convolution layers, the model had 2 layers of FCNNs with 512 neurons and the dropout layers (31). The final layer consisted of a sigmoid function for binary classification.

Like for the FCNN, the convolutional network weights were initialized from a random number generated from a Gaussian distribution. The ReLU function for hidden layers and the sigmoid function for the output layer were set. The adaptive gradient descent algorithm (Adadelta) was selected as optimizer and the binary cross entropy as loss function. During training, the batch size was set to 50 and the number of epochs to 5.

Different layer (n=2, 3, 4) and filter sizes for each layer (containing two convolutional layers and a pooling layer on top of FCNNs) were tried. To speed up training, the optimized architecture (batch size=50, epochs=5) was trained for our study on the Amazon Web Service infrastructure (32), using the described complex spectrograms as input (**Figure 4**).

The trained model was then evaluated and the classification accuracy determined. The training of D-CNNs was done for three different datasets:

First, it was trained on 20000 spectra of Group-1 and tested on 10000 spectra of Group-1, 10000 spectra of Group-2 and 65 in-vivo spectra.

Second, it was trained on 20000 spectra of Group-2 and tested on 10000 spectra of Group-1, 10000 spectra of Group-2 and the 65 in-vivo spectra

Finally, it was trained with a mix of Group-1 & Group-2 spectra (40000 spectra for training in total, balanced in terms of spectra with and without ghosting artifacts), then tested on 20000 spectra again from Group-1 and Group-2 as well as the 65 in-vivo spectra.

DL for ghost removal

Stacked-autoencoders have previously been used to remove noise from images (33–35). They represent a type of deep neural network that is taught to use noisy input images and reconstruct images without the noise as output. The standard stacked-autoencoder architecture consists of encoding and decoding. The encoder layer consists of standard 2D convolutional filters and activation functions followed by a max-pooling filter. The decoder network has a reverse process compared to the encoder and consists of up-sampling layers instead of max-pooling to recover the input images. Autoencoders constructed using convolutional layers have shown to perform well for noise removal. However, recently it has been shown that residual networks have an even better performance than D-CNNs when the depth of the model increases. Here we exploit this advantage and build residual networks based on two ideas:

- *SWWAE:* The first idea behind SWWAE is that the exact location of the maximal value in the pooled receptive field is lost during max-pooling. So, if the location of the maximal value can be handed over from the encoder to the corresponding decoding layer, the reliability of the reconstruction increases, which has been shown recently (14,36).
- *Residual networks:* The second idea is to use residual blocks. Residual networks have skip connections and it has been shown in Refs. (15,37,38) that residual blocks are easier to optimize and that they gain accuracy with deeper networks unlike other networks (with linearly stacked convolution layers) where the performance gets saturated and degrades rapidly due to higher training errors.

Hence, SWWAE built on residual blocks (27) were implemented and tested for removing the ghosting artifacts from spectra. Since the model takes the spectral input as images, we created spectrograms as before for ghost detection, but with increased resolution (56x250), since the inverse of the spectrogram transformation to get back the original spectrum cannot perform reliably if the overlapping window is chosen too small when creating the spectrogram.

Spectrograms with ghosting artifacts (27500) randomly selected from both Group-1 and Group-2 were used as input and corresponding spectrograms without artifacts were given as ground truth data. They were padded with the edge value of the array to have a size of 64x256. The encoder layer consisted of 5 layers of residual blocks and the Exponential Linear Units (ELU) function (39) followed by a max-pooling filter. The decoder network was the reverse process of the encoder and consisted of up-sampling layers instead of max-pooling to recover the spectrograms without the artifacts. The overall architecture for artifact removal is illustrated in **Figure 5**.

The parameters that were used to build this network featured 5 layers, ELU activation function, 3x3 kernels and 2 layers for the residual function. The first layer consisted of 8 convolutional filters and the subsequent layers included 16, 32, 64, and 128 convolutional filters, respectively.

As usual, the network weights were initialized from a random number. The sigmoid function was set for the output layer. The Adadelta (40) was set during compilation as optimizer and the binary cross entropy as loss function. For training, the batch size was set to 25 and the number of epochs to 30.

The trained SWWAE was tested on 5000 independent spectrograms with ghosting artifacts to predict the artifact free spectrogram. The inverse transform was then applied to the predicted spectrogram to arrive at the artifact-free spectrum.

For plotting and quantitative judgment of the similarity between ground truth and the reconstructed spectra output from the DL model the spectra were normalized by dividing the complex spectra by the square root of the sum of the squares of the amplitudes of all spectral points (i.e. the L-2 norm). The root-mean-squared error (RMSE) between the normalized predicted and the normalized artifact free spectrum was calculated for all test cases. The mean RMSE over all spectra and separately for the best and worst SNR cases used (SNR 10 and SNR 100) were determined as output measures.

Results:

Ghost identification

Ghost detection by DL methods was found to be feasible, though with strongly varying performance for the tested methods. FCNNs with 1D spectra as input was found to be inferior to D-CNNs with spectral input as 2D spectrograms. Detailed results are listed in Tables 1 and 2 and described in the following.

FCNNs: FCNNs with different numbers of layers were tested using 1D spectra from Group-1. When FCNNs with 7 layers were used the performance of the model was a mere 50%. With more layers it increased to almost 75%, still clearly insufficient for any practical use. (**Table 1**)

D-CNNs: DL based on 2D spectrogram input was evaluated on different datasets and with different numbers of layers (**Table 2**). The average performance of the D-CNNs was best with two-layers with a mean accuracy of 96% over all tested cases. The accuracy was extremely good when the network was trained with Group-1 spectrograms and also tested on Group-1. However, for testing on Group-2 the accuracy dropped to ~92%, and was poor for in-vivo spectra. The D-CNNs that were trained on Group-2 performed well when tested in-vivo, with the accuracy reaching 100% for 2 layers, however for Group-1, they did not perform too well. In order to cover as much as possible of the variance expected in pathology in-vivo, D-CNNs were trained combining Group-1 and Group-2 spectra, which yielded close to 100% accuracy both for testing on simulated and in-vivo spectra. **Figure 6** illustrates how sensitive the automatic ghost detection was with D-CNNs. It contains selected cases where the ghost is easily seen by eye, but also spectra where even an expert could not readily tell the artifact from random noise.

Ghost removal

Based on the proposed SWWAE method, it was possible to restore ghost-distorted spectra on simulated cases. **Figure 7** shows sample spectra with ghosting artifacts in comparison to the restored spectra and the ground truth for different SNRs and LWs. **Figure 8** illustrates the restoration of spectra for in-vivo cases. The restoration is not perfect – partly due to inaccuracies in the prediction, partly due to inaccuracies in the reconstruction of the 1D spectra from the 2D spectrograms.

The average of the RMSE for the difference between ground truth and restored spectra was 0.026 if determined for all test spectra, and ranging between 0.052 for those with SNR 10, to 0.017 for those with SNR 100, respectively. As can be judged form **Figure 7**, these mean deviations are smaller than the noise level.

Discussion:

Ghost detection using convolutional neural networks works almost perfectly for the tested simulated and in-vivo cases with accuracy reaching close to 100%. Ghost removal also looks promising as documented for the simulated cases; however, the in-vivo spectra show that this needs further improvements.

This initial success with DL in MRS came somewhat as a surprise. While standard ML techniques based on extracted or selected features can be tuned and used with databases of modest size, DL approaches are known to require very large databases for successful application. Even medical imaging datasets are relatively small compared to, for instance, 1.2 million images in the ImageNet LSVRC-2010 contest, which limits the ability of DL models to tap its full potential in MRI. Hence, their use for in-vivo MRS at first seems illusory, no matter whether addressing tumor classification or quality assessment tasks. However, if training can be performed with simulated instead of measured spectra, there are no strict limits to the number of spectra to be used. We therefore tried to simulate spectra covering as much as possible of the variance to be expected in real in-vivo spectra and to test DL for artifact detection as a first application for DL in MRS. In addition in this first trial, we restricted attention to one type of artifact, the spurious echo. Even though this artifact has characteristic time/frequency properties that make it ideal for this type of detection process, there is no reason why the developed technology could not be extended and tested for other tasks in unsupervised quality assessment in clinical MRS.

For training and in-silico testing, spectra were simulated to cover most of the variations of experimental clinical spectra. In particular, group-1 spectra emulated the case of healthy volunteers with spectra of varying LWs and SNRs representing variation in magnetic field homogeneity and voxel size and other particularities of the measurement. In contrast, group-2 spectra also embody varying metabolite concentrations expected for different pathologies. Ghost signals were varied in both groups both in frequency and time making them unpredictable in terms of where and how they occur.

DL was tested in two main flavors, FCNNs based on 1D input, and D-CNNs optimized for multidimensional input data. For 1D MRS, 1D input appears more suited, in principle. However, in our hands, conversion of 1D spectra to 2D data and use of DL tools optimized for images was much more successful.

DL with 1D input: DL models defined using FCNNs with different layers did not perform well for ghost detection, suggesting that the full connectivity with a huge number of parameters leads to overfitting in the current case. With 7 layers, the performance was around 50%, which means that half of the time the model categorized with wrong labels. The prediction accuracy improved to 72% and 74.5% when one or two additional layers were added, however the performance clearly remained unsatisfactory. FCNNs receive the input as vectors and since each spectrum consisted of 2048 complex points, a single neuron in a first hidden layer of a FCNN had 2048 weights even if only considering the real part. Considering FCNN’s first layer with 2048 input neurons and 1024 hidden neurons, it comprised a total of 2048×1024 weights, plus an extra 1024 biases; which amounts to 2’098’176 parameters in total. Adding different numbers of hidden layers would create huge numbers of parameters. To counteract, FCNNs were also tried with reduced numbers of points, restricted to the spectral region of interest, to avoid an excessive number of parameters (results not shown) but the performance was not convincing. In addition, FCNNs were also tried using 50 features extracted using principal component analysis (results not shown) but again the performance was poor.

DL with 2D input: Upon finding rather poor performances with FCNNs, we started building a model using the most popular kind of deep learning models, called D-CNNs, which have succeeded in image recognition competitions (41,42). One of the advantages of D-CNNs over FCNNs is the training speed. D-CNNs make use of shared weights and bias thus vastly reducing the number of parameters in the network. Considering the first layer in a D-CNN with 70 features mapped with a 2x2 filter, it amounts to 4 parameters plus a single shared bias, i.e. in total 70x5 = 350 parameters - ~6000 times less than with the FCNN described above.

There are many options, how to transfer a 1D spectrum into 2D arrays. It seemed well adapted to our task to convert spectra to spectrograms that provide an image-like 2D time-frequency representation, where the ghost signals can indeed appear at almost arbitrary locations since they represent ill-timed spectral contributions.

D-CNNs are known to be well-suited for images with RGB channels as input; here we used the real and imaginary parts of spectrograms as a two-channel input to D-CNNs yielding excellent artifact recognition performance in the tested cases. Overall, the performance using two layers of D-CNNs performed well with mean accuracy of 96.3% compared to 92.9 % and 89.9% accuracy of 3 and 4 layers. Note that unusually small numbers of epochs (5) were found to be sufficient for the ghost detection. Larger numbers have not been explored because the validation accuracy was neither fluctuating nor increasing when increasing the number of epochs.

The second part of our investigations dealt with attempts to not only detect ghost contributions but also to remove them and clean up the input spectra. Clearly, artifact removal by DL techniques in spectroscopy is a wide and open field and the current contribution can only scratch at the surface. Still, ghost removal using SWWAE turned out to be very promising on simulated and in-vivo cases. Even though the simulations for the input spectra were run to cover as much of the expected variance as possible, the DL model did not perform perfectly in removing the artifacts, in particular for in-vivo cases. Increasing the number of epochs may be beneficial, but has not been fully explored, given that more fundamental problems will have to be tackled first. For example, judgment of success for ghost removal has so far only been based on visual appearance. This will have to be extended to quantitative evaluations, particularly since the DL restoration step may easily distort parts of the spectrum that are not affected by the overlaid artifact and apparent success in ghost removal may come at a cost in terms of accuracy for the evaluation of small metabolite contributions throughout the spectrum.

One alternative 2D representation to mention is the Gabor transform, which has been used for denoising (43). Hence, instead of our method of constructing a spectrogram one could use the GABOR transform and base DL methods on the thus created 2D matrix. (Though, data size arguments show that this is not straightforward). Alternatively, with only ghost removal in mind, one could try to do without ML and optimize a GABOR filter to eliminate the spurious echoes – though probably also limited by the danger to filter out real metabolite signals as well.

Of course, this initial application of deep learning in MRS has multiple limitations and open issues that may be addressed in further research:

1) In this study, the residual water signal had been filtered out of the in-vivo spectra with HLSVD, because the DL models had been trained with spectra without residual water signals. The sensitivity to residual water signals should be explored and improved if needed.

2) The in-vivo spectra were all of the same type and from healthy brain tissue, hence this initial trial has to be extended to larger testing sets of in-vivo spectra including other acquisition parameters (echo time, localization methods, brain area, etc.), and pathological spectral alterations.

3) Artifact removal, but also artifact detection should be evaluated in a more quantitative fashion, where ghost detection should only flag artifacts truly affecting metabolite content estimations and ghost removal should be optimized such that metabolite content estimation is not biased.

4) Further extension of the methodology may address the detection and restoration of other artifacts, e.g. effects from residual water or out-of-phase lipid signals.

5) Other transforms into the time-frequency representation could be tried – in particular for detection of other artifacts and for better back-transformation properties.

6) It should be investigated how many spectra are indeed needed for training the networks and whether identical spectra with and without artifacts are mandatory. If hundreds or a few

thousand are sufficient and if they don't have to include identical pairs, it may be feasible to use intentionally distorted in-vivo data instead of simulations.

Conclusions:

In this study, we show that it is possible to use deep learning to detect and remove ghosting artifacts for in-vivo MRS. Initial results show promising performance, in particular for ghost detection and motivate further extensions of the methods and exploration into clinical applications.

References:

1. Kreis R. Issues of spectral quality in clinical 1H-magnetic resonance spectroscopy and a gallery of artifacts. NMR Biomed. 2004;17:361–381. doi: 10.1002/nbm.891.

2. Menze BH, Kelm BM, Weber MA, Bachert P, Hamprecht FA. Mimicking the human expert: Pattern recognition for an automated assessment of data quality in MR spectroscopic images. Magn. Reson. Med. 2008;59:1457–1466. doi: 10.1002/mrm.21519.

3. Pedrosa de Barros N, McKinley R, Knecht U, Wiest R, Slotboom J. Automatic quality control in clinical 1 H MRSI of brain cancer. NMR Biomed. 2016;29:563–575. doi: 10.1002/nbm.3470.

4. Wright AJ, Arús C, Wijnen JP, Moreno-Torres A, Griffiths JR, Celda B, Howe FA. Automated quality control protocol for MR spectra of brain tumors. Magn. Reson. Med. 2008;59:1274–1281. doi: 10.1002/mrm.21533.

5. Kyathanahally SP, Mocioiu V, Pedrosa De Barros NM, Slotboom J, Wright AJ, Julià-Sapé M, Arús C, Kreis R. Quality of clinical brain tumor MR spectra judged by humans and machine learning tools. Magn. Reson. Med. 2017:(accepted).

6. Garcia Amaro E, Nuno-Maganda M a., Morales-Sandoval M. Evaluation of machine learning techniques for face detection and recognition. In: CONIELECOMP 2012, 22nd International Conference on Electrical Communications and Computers. ; 2012. pp. 213–218. doi: 10.1109/CONIELECOMP.2012.6189911.

7. Renuka D., Hamsapriya T, Chakkaravarthi M., Surya P. Spam classification based on supervised learning using machine learning techniques. In: International Conference on Process Automation, Control and Computing (PACC). Coimbatore; 2011. pp. 1–7. doi: 10.1109/PACC.2011.5979035.

8. Preul MC, Caramanos Z, Leblanc R, Villemure JG, Arnold DL. Using pattern analysis of in vivo proton MRSI data to improve the diagnosis and surgical management of patients with brain tumors. NMR Biomed. 1998;11:192–200.

9. Tate AR, Majós C, Moreno A, Howe FA, Griffiths JR, Arús C. Automated classification of short echo time in in vivo 1H brain tumor spectra: A multicenter study. Magn. Reson. Med. 2003;49:29–36. doi: 10.1002/mrm.10315.

10. Ye CZ, Yang J, Geng DY, Zhou Y, Chen NY. Fuzzy rules to predict degree of malignancy in brain glioma. Med. Biol. Eng. Comput. 2002;40:145–52.

11. Devos A, Lukas L, Suykens JAK, et al. Classification of brain tumours using short echo time 1H MR spectra. J. Magn. Reson. 2004;170:164–175. doi: 10.1016/j.jmr.2004.06.010.

12. Luts J, Poullet J-B, Garcia-Gomez JM, Heerschap A, Robles M, Suykens JAK, Van Huffel S. Effect of feature extraction for brain tumor classification based on short echo time 1H MR spectra. Magn. Reson. Med. 2008;60:288–298. doi: 10.1002/mrm.21626.

13. Krizhevsky A, Sutskever I, Hinton GE. ImageNet classification with deep convolutional neural networks. In: Pereira F, Burges CJC, Bottou L, Weinberger KQ, editors. Advances in Neural Information Processing Systems. Curran Associates, Inc.; 2012. pp. 1097–1105.

14. Zhao J, Mathieu M, Goroshin R, LeCun Y. Stacked What-Where Auto-encoders. arXiv:1506.02351 2015;1:1–12.

15. He K, Zhang X, Ren S, Sun J. Identity mappings in deep residual networks. CoRR 2016;abs/1603.0. doi: 10.1007/978-3-319-46493-0_38.

16. LeCun Y, Bengio Y, Hinton G. Deep learning. Nature 2015;521:436–444.

17. Greenspan H, Ginneken B van, Summers RM. Guest editorial deep learning in medical imaging: Overview and future promise of an exciting new technique. IEEE Trans. Med. Imaging 2016;35:1153–1159. doi: 10.1109/TMI.2016.2553401.

18. Soher B, Semanchuk D, Todd D, Steinberg J, Young K. Vespa: Integrated applications for RF pulse design, spectral simulation and MRS data analysis. In: Proceedings of the 19th Annual Meeting ISMRM. Montreal,Canada; 2011. p. 1410.

19. Mekle R, Mlynárik V, Gambarota G, Hergt M, Krueger G, Gruetter R. MR spectroscopy of the human brain with enhanced signal intensity at ultrashort echo times on a clinical platform at 3T and 7T. Magn Reson Med. 2009;61:1279–1285. doi: 10.1002/mrm.21961.

20. Mlynárik V, Gruber S, Moser E. Proton T(1) and T(2) relaxation times of human brain metabolites at 3 Tesla. NMR Biomed. 2001;14:325–331. doi: 10.1002/nbm.713.

21. Döring A, Adalid Lopez V, Brandejsky V, Kreis R, Chris B. Diffusion weighted MR spectroscopy without water suppression allows to use water as inherent reference signal to correct for motion-related signal drop. In: Proceedings of the 24th Annual Meeting ISMRM. Singapore; 2016. p. 2395.

22. Vanhamme L, Fierro RD, Van Huffel S, de Beer R. Fast removal of residual water in proton spectra. J. Magn. Reson. 1998;132:197–203. doi: http://dx.doi.org/10.1006/jmre.1998.1425.

23. Antoine JP, Chauvin C, Coron A. Wavelets and related time-frequency techniques in magnetic resonance spectroscopy. NMR Biomed. 2001;14:265–270. doi: 10.1002/nbm.699.

24. Leclerc JHJ. Time-frequency representation of damped sinusoids. J. Magn. Reson. 1991;95:10–31. doi: http://dx.doi.org/10.1016/0022-2364(91)90321-J.

25. Gabor D. Theory of Communication. J. Inst. Electr. Eng. - Part III Radio Commun. Eng. 1946;93:429–457. doi: 10.1049/ji-3-2.1946.0074.

26. Simpson AJR. Deep transform: Cocktail party source separation via complex convolution in a deep neural network. CoRR 2015;abs/1504.0:4–7.

27. Chollet F. Keras. 2016:GitHub, https://github.com/fchollet.

28. Al-Rfou R, Alain G, Almahairi A, et al. Theano: A Python framework for fast computation of mathematical expressions. arXiv e-prints 2016;abs/1605.0:19.

29. Kingma DP, Ba J. Adam: A method for stochastic optimization. Int. Conf. Learn. Represent. 2015:1–15. doi: http://doi.acm.org.ezproxy.lib.ucf.edu/10.1145/1830483.1830503.

30. Nagi J, Ducatelle F. Max-pooling convolutional neural networks for vision-based hand gesture recognition. 2011 IEEE Int. Conf. Signal Image Process. Appl. 2011:342–347. doi: 10.1109/ICSIPA.2011.6144164.

31. Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. 2014;15:1929–1958. doi: 10.1214/12-AOS1000.

32. Amazon Inc. Amazon Web Service. http://aws.amazon.com 2016.

33. Xie J, Xu L, Chen E. Image denoising and inpainting with deep neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems. Lake Tahoe, Nevada; 2012. pp. 341–349.

34. Gondara L. Medical image denoising using convolutional denoising autoencoders. In: 2016

IEEE 16th International Conference on Data Mining Workshops (ICDMW). Barcelona; 2016. pp. 241–246. doi: 10.1109/ICDMW.2016.0041.

35. Mao X-J, Shen C, Yang Y-B. Image restoration using convolutional auto-encoders with symmetric skip connections. CoRR 2016;abs/1606.0:1–17.

36. Zeiler MD, Fergus R. Visualizing and understanding convolutional networks. CoRR 2013;abs/1311.2:818–833. doi: 10.1007/978-3-319-10590-1_53.

37. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. CoRR 2015;abs/1512.0. doi: 10.1007/s11042-017-4440-4.

38. Veit A, Wilber M, Belongie S. Residual networks behave like ensembles of relatively shallow networks. CoRR 2016;abs/1605.0.

39. Clevert D-A, Unterthiner T, Hochreiter S. Fast and accurate deep network learning by Exponential Linear Units (ELUs). CoRR 2015;abs/1511.0.

40. Zeiler MD. ADADELTA: An adaptive learning rate method. CoRR 2012;abs/1212.5.

41. Russakovsky O, Deng J, Su H, et al. ImageNet large scale visual recognition challenge. Int. J. Comput. Vis. 2015;115:211–252.

42. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. CoRR 2014;abs/1409.1.

43. Gu J-J, Tao L, Kwan HK. NMR FID signal enhancement via the oversampled Gabor transform using the Gaussian synthesis window. In: 47th Midwest Symposium on Circuits and Systems, 2004. MWSCAS '04. ; 2004. pp. 183–186. doi: 10.1109/MWSCAS.2004.1354322.

Acknowledgement:

This research was carried out in the framework of the European Marie-Curie Initial Training Network, 'TRANSACT', PITN-GA-2012-316679, 2013-2017 and also supported by the Swiss National Science Foundation (#320030-156952, #320030-175984).

Figure captions:

Figure 1: Simulated sample data from Group-1, represented as time domain signals (FIDs), in frequency space (spectra) and in the chosen time-frequency representation (spectrograms). Cases with different linewidths, SNR and ghost characteristics are depicted in the four quadrants of the figure. For each case, the identical data (including identical noise) is presented with (right) and without (left) the added ghosting artifact. In some cases the artifact is conspicuous (case in the bottom right of the figure with a large ghost centered at ~100ms and 6 ppm) while for others even experts do not recognize the artifact immediately (e.g. at bottom left: ghost at ~320 ms, 1.8 ppm)

Figure 2: (a) Simulated sample data from Group-2, presented in the same form as in Figure 1, where the cases were simulated with varying metabolite content, linewidth, SNR and ghost characteristics. (b) In-vivo data with and without ghost artifacts that were acquired using metabolite-cycled STEAM (21) in occipital gray matter of healthy subjects. For the in-vivo cases the data with and without ghosting artifacts are not identical and originate from independent examinations, just representing four different cases.

Figure 3: Illustration of how spectrograms are constructed: Time domain signals were segmented with a window size of 128 samples with an overlap interval of 30 samples and weighted by a Hanning window. A Fourier transform was applied to each segment and thus converted the FIDs into 2D time-frequency spectrograms. This provided complex spectrograms with 70 frequency bins and 20 time stamps.

Figure 4: Architecture of the convolutional network model with 4 layers used for ghost detection.

Figure 5: Architecture of the "Stacked What-Where Auto Encoder" (SWWAE) devised for removal of ghosting artifacts. FIDs with ghost were converted into spectrograms and given as

input to the autoencoder that consisted of encoder and decoder (each consisting of 5 layers of convolutional filters, ELU function (39), 3x3 convolutional kernels and residual blocks with 2 skip connections). The exact location ('where') of the maximum value in a pooled receptive field is stored and handed over to the corresponding decoder layer. The Adadelta (40) was set during compilation as optimizer, binary cross entropy as loss function and sigmoid function was set for the output layer. For training, the batch size was set to 25 and number of epochs to 30. The network gave out artifact free spectrograms which were then inverse-transformed to arrive at artifact-free FIDs.

Figure 6: Simulated sample spectra illustrating how sensitive the automatic ghost detection with D-CNNs is. The circled areas contain the spurious echoes, which are partly hard to detect, even for a human expert.

Figure 7: Sample results illustrating the performance of ghost removal using DL for eight simulated cases with different SNRs. The reconstructed artifact-free spectra are plotted together with the ghost-ridden spectra on the left in each sub-panel, whereas the reconstructed spectra are plotted together with the ground truth spectra on the right for each case. In this panel, the difference spectra between forecast and ground truth are included to illustrate that, at the current stage, the removal of the ghosts comes at the cost of signal distortions or changes at spectral ranges far from the artifact.

Figure 8: Illustration for the performance of ghost removal using DL on in-vivo spectra. The original in-vivo spectrum with ghosting artifacts is shown in blue and the “ghost-busted” spectrum predicted by the DL algorithm is shown in red.

Supporting Information Captions

Figure S1

Example for a fully connected network. In our study, the hidden layers consisted of either 7, 8 or 9 layers.

Table 1: Overall performance for ghost detection using fully connected networks (FCNNs) with 1D spectral input and different numbers of layers.

Train on	Test on	Accuracy in %		
		7 layers	8 layers	9 layers
Group 1 (20 k)	Group 1 (10k)	50.4	72.2	74.5

Table 2: Overall performance for ghost detection with D-CNNs based on 2D spectrogram input, different numbers of layers and applied to different data for training and independent testing (also documenting similar performance for different numbers of epochs).

Train on	Validation accuracy in % (range for 5 epochs)			Test on	Test accuracy in %		
	2 layers	3 layers	4 layers		2 layers	3 layers	4 layers
Group 1 spectra (20 k)	99.3 – 99.6	99.3 - 99.7	99.6 - 99.8	Group 1 (10k)	99.6	99.7	99.8
				Group 2 (10k)	91.5	92.9	93.4
				in-vivo (65)	86.2	70.8	67.7
Group 2 spectra (20 k)	99.8-99.9	99.9 - 100.0	99.9 - 100.0	Group 1 (10k)	93.1	92.2	89.4
				Group 2 (10k)	99.9	100.0	100.0
				in-vivo (65)	100.0	92.3	69.2
Group 1 + Group 2 spectra (40 k)	99.6 - 99.8	99.7-99.8	99.7-99.8	Group 1 + Group 2 (20k)	99.8	99.8	99.8
				in-vivo (65)	100.0	95.4	100.0

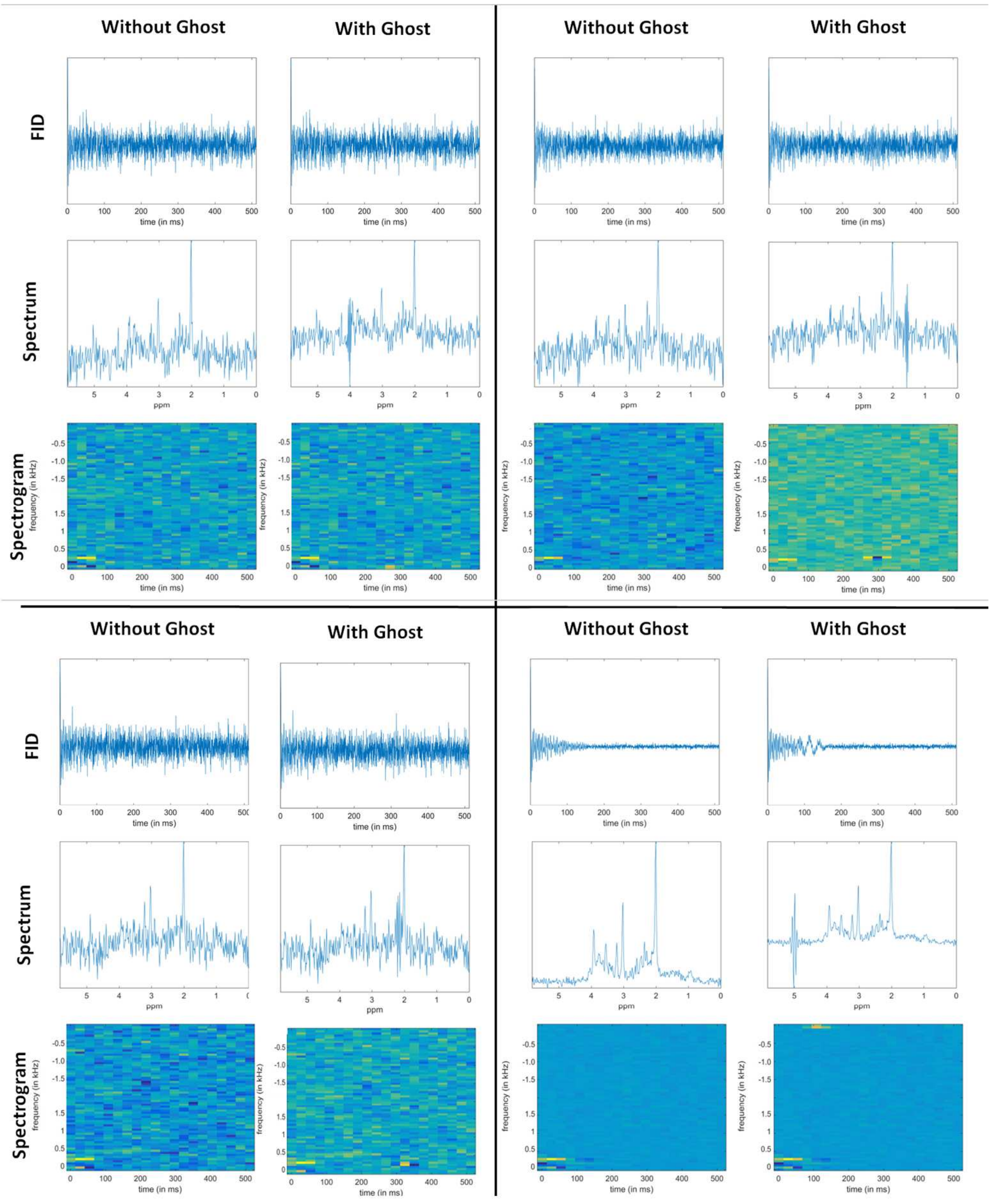


Figure 1:Simulated sample data from Group 1, represented as time domain signals (FIDs), in frequency space (spectra) and in the chosen time-frequency representation (spectrograms). Cases with different linewidths, SNR and ghost characteristics are depicted in the four quadrants of the figure. For each case, the identical data (including identical noise) is presented with (right) and without (left) the added ghosting artifact. In some cases the artifact is conspicuous (case in the bottom right of the figure with a large ghost centered at ~100ms and 6 ppm) while for others even experts do not recognize the artifact immediately (e.g. at bottom left: ghost at ~320 ms, 1.8 ppm)

181x220mm (600 x 600 DPI)

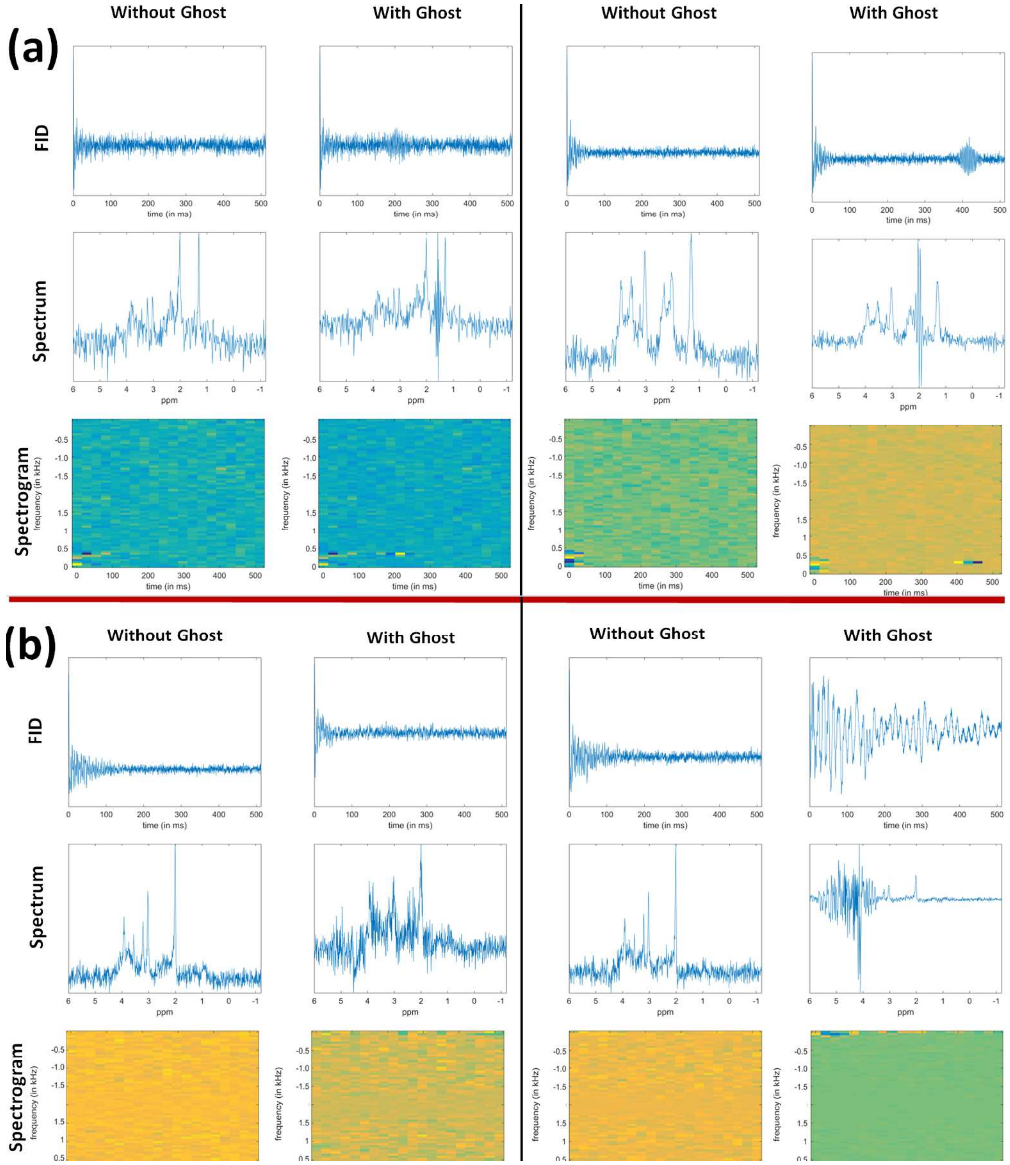


Figure 2: (a) Simulated sample data from Group 2, presented in the same form as in Figure 1, where the cases were simulated with varying metabolite content, linewidth, SNR and ghost characteristics. (b) In-vivo data with and without ghost artifacts that were acquired using metabolite-cycled STEAM (21) in occipital gray matter of healthy subjects. For the in-vivo cases the data with and without ghosting artifacts are not identical and originate from independent examinations, just representing four different cases.

173x200mm (600 x 600 DPI)

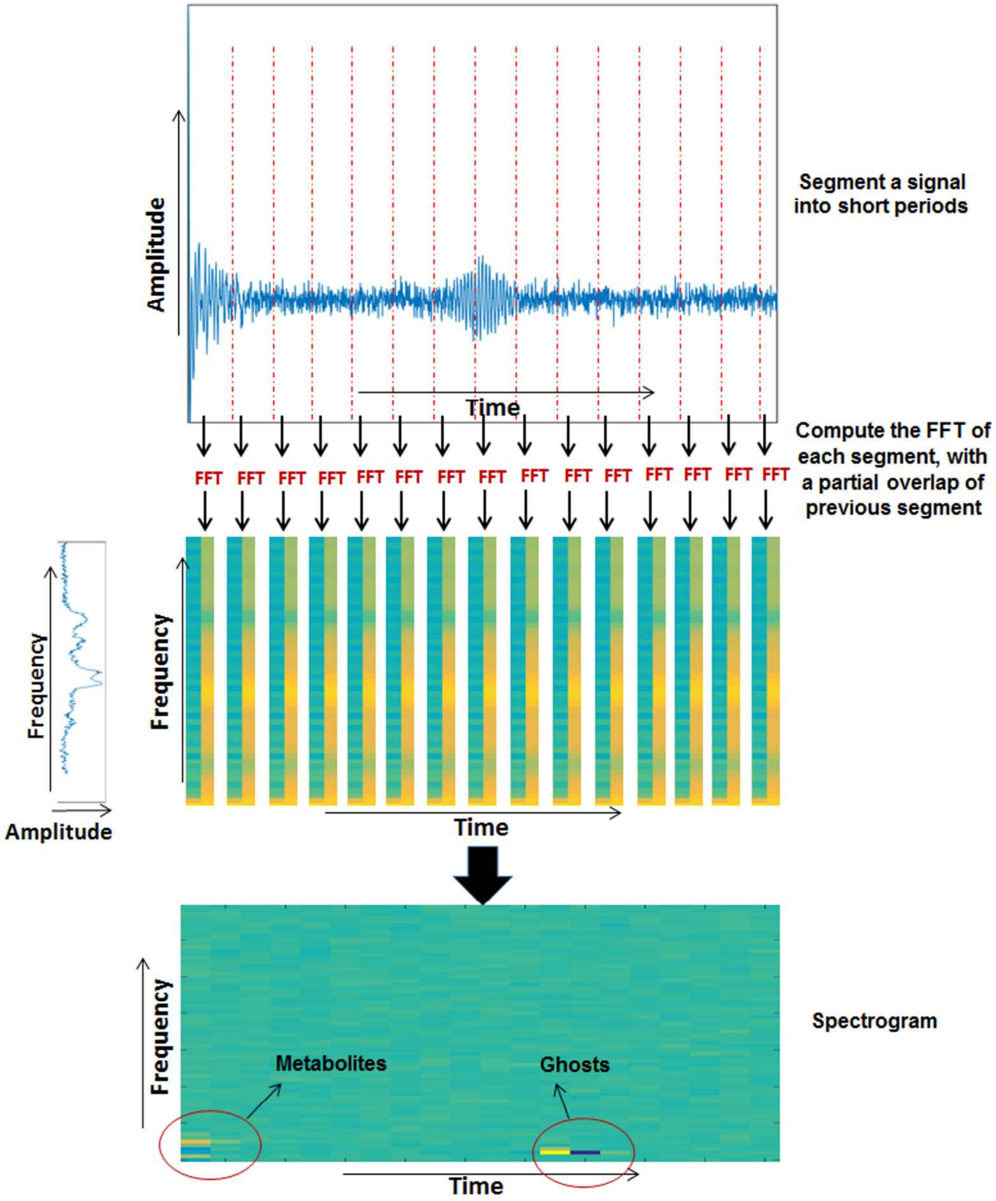

Amplitude
Time
Segment a signal into short periods
FFT
Compute the FFT of each segment, with a partial overlap of previous segment
Frequency
Amplitude
Frequency
Time
Frequency
Metabolites
Ghosts
Time
Spectrogram

Figure 3: Illustration of how spectrograms are constructed: Time domain signals were segmented with a window size of 128 samples with an overlap interval of 30 samples and weighted by a Hanning window. A Fourier transform was applied to each segment and thus converted the FIDs into 2D time-frequency spectrograms. This provided complex spectrograms with 70 frequency bins and 20-time stamps.

180x218mm (600 x 600 DPI)

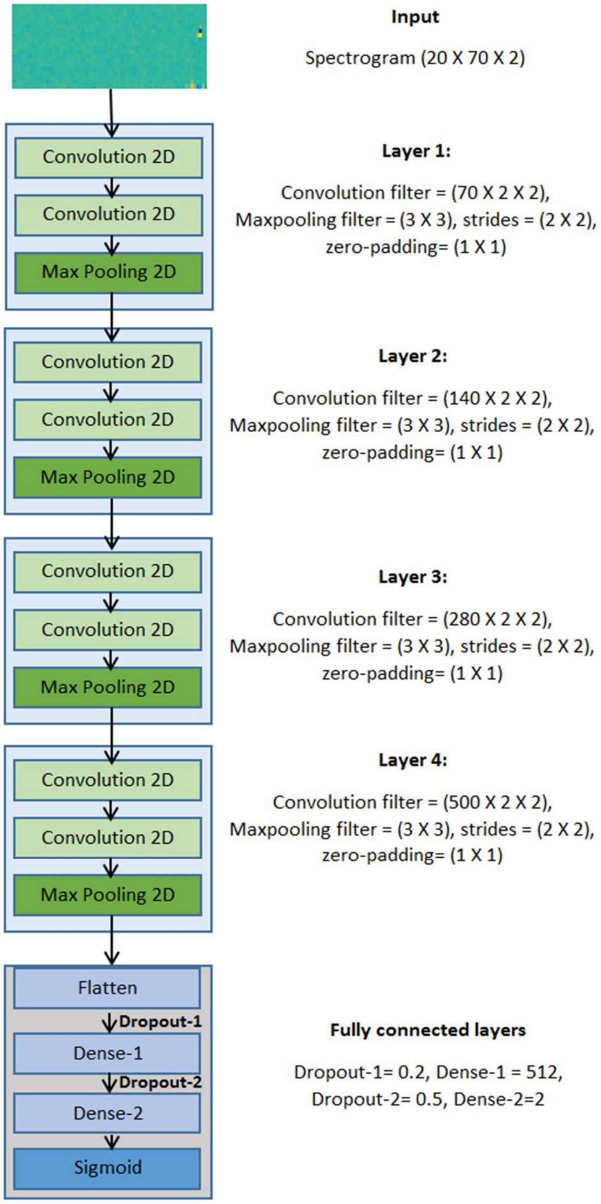

Input
Spectrogram (20 X 70 X 2)
Convolution 2D
Convolution 2D
Max Pooling 2D
Layer 1:
Convolution filter = (70 X 2 X 2),
Maxpooling filter = (3 X 3), strides = (2 X 2),
zero-padding= (1 X 1)
Convolution 2D
Convolution 2D
Max Pooling 2D
Layer 2:
Convolution filter = (140 X 2 X 2),
Maxpooling filter = (3 X 3), strides = (2 X 2),
zero-padding= (1 X 1)
Convolution 2D
Convolution 2D
Max Pooling 2D
Layer 3:
Convolution filter = (280 X 2 X 2),
Maxpooling filter = (3 X 3), strides = (2 X 2),
zero-padding= (1 X 1)
Convolution 2D
Convolution 2D
Max Pooling 2D
Layer 4:
Convolution filter = (500 X 2 X 2),
Maxpooling filter = (3 X 3), strides = (2 X 2),
zero-padding= (1 X 1)
Flatten
Dropout-1
Dense-1
Dropout-2
Dense-2
Sigmoid
Fully connected layers
Dropout-1= 0.2, Dense-1 = 512,
Dropout-2= 0.5, Dense-2=2

Figure 4: Architecture of the convolutional network model with 4 layers used for ghost detection.

300x602mm (300 x 300 DPI)

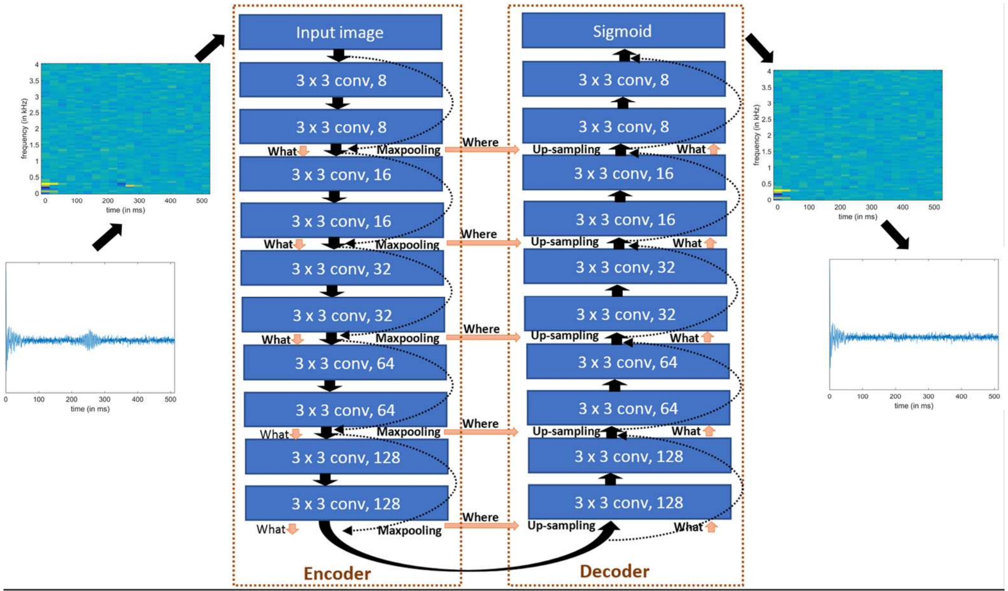


Figure 5: Architecture of the “Stacked What-Where Auto Encoder” (SWWAE) devised for removal of ghosting artifacts. FIDs with ghost were converted into spectrograms and given as input to the autoencoder that consisted of encoder and decoder (each consisting of 5 layers of convolutional filters, ELU function (39), 3x3 convolutional kernels and residual blocks with 2 skip connections). The exact location (’where’) of the maximum value in a pooled receptive field is stored and handed over to the corresponding decoder layer. The Adadelta (40) was set during compilation as optimizer, binary cross entropy as loss function and sigmoid function was set for the output layer. For training, the batch size was set to 25 and number of epochs to 30. The network gave out artifact free spectrograms which were then inverse-transformed to arrive at artifact-free FIDs

146x86mm (300 x 300 DPI)

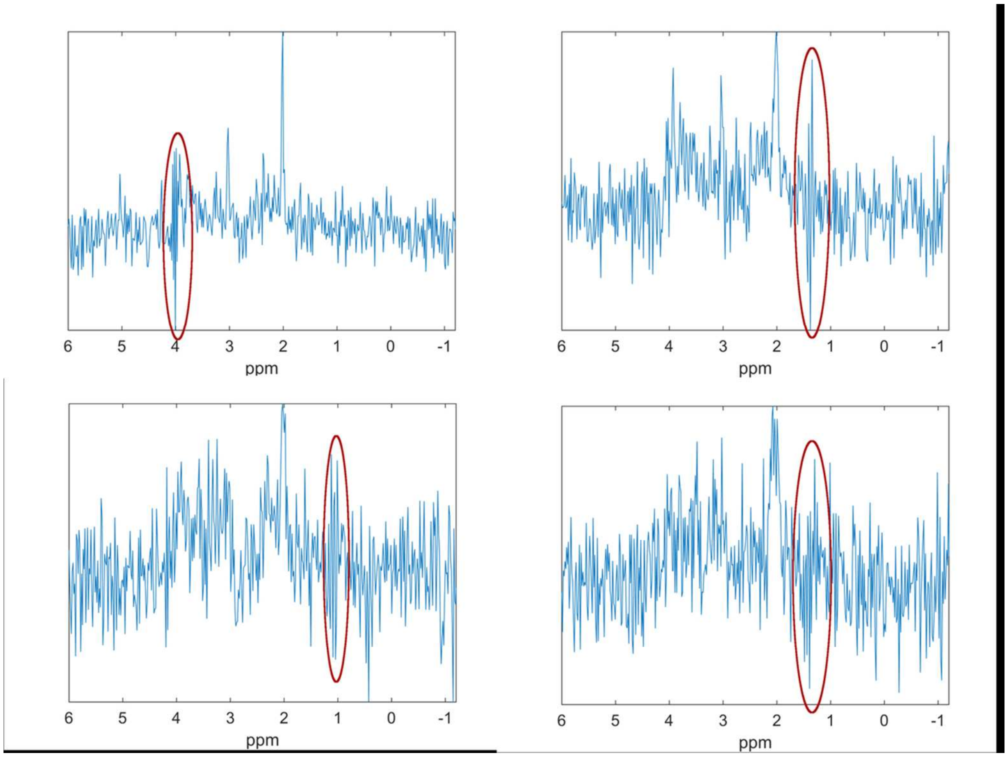


Figure 6: Simulated sample spectra illustrating how sensitive the automatic ghost detection with DCNNs is. The circled areas contain the spurious echoes, which are partly hard to detect, even for a human expert.

112x84mm (300 x 300 DPI)

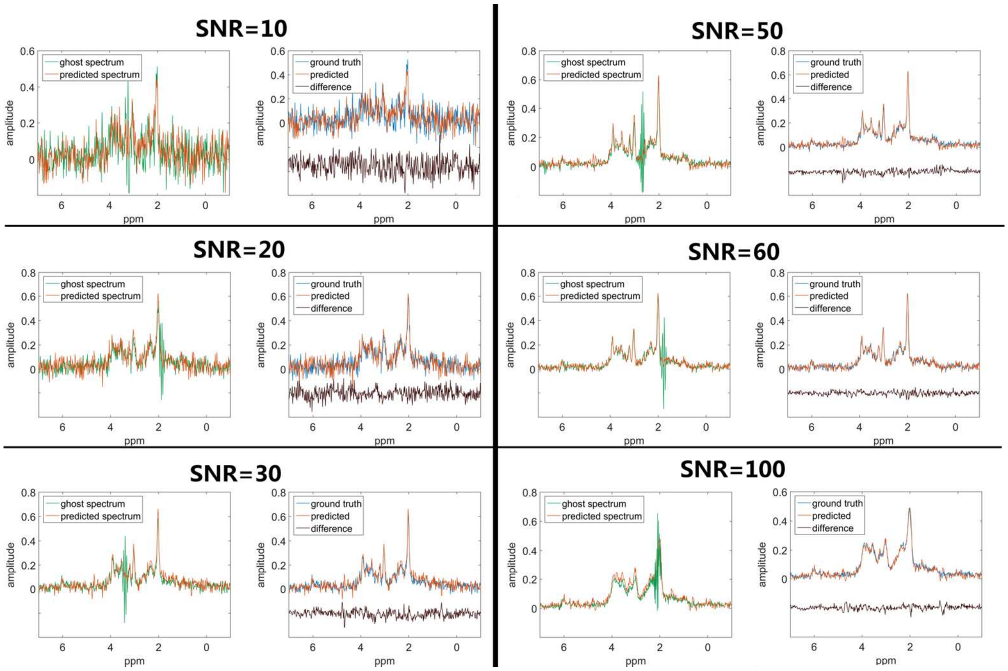


Figure 7: Sample results illustrating the performance of ghost removal using DL for eight simulated cases with different SNRs. The reconstructed artifact-free spectra are plotted together with the ghost-ridden spectra on the left in each sub-panel, whereas the reconstructed spectra are plotted together with the ground truth spectra on the right for each case. In this panel, the difference spectra between forecast and ground truth are included to illustrate that, at the current stage, the removal of the ghosts comes at the cost of signal distortions or changes at spectral ranges far from the artifact.

165x109mm (300 x 300 DPI)

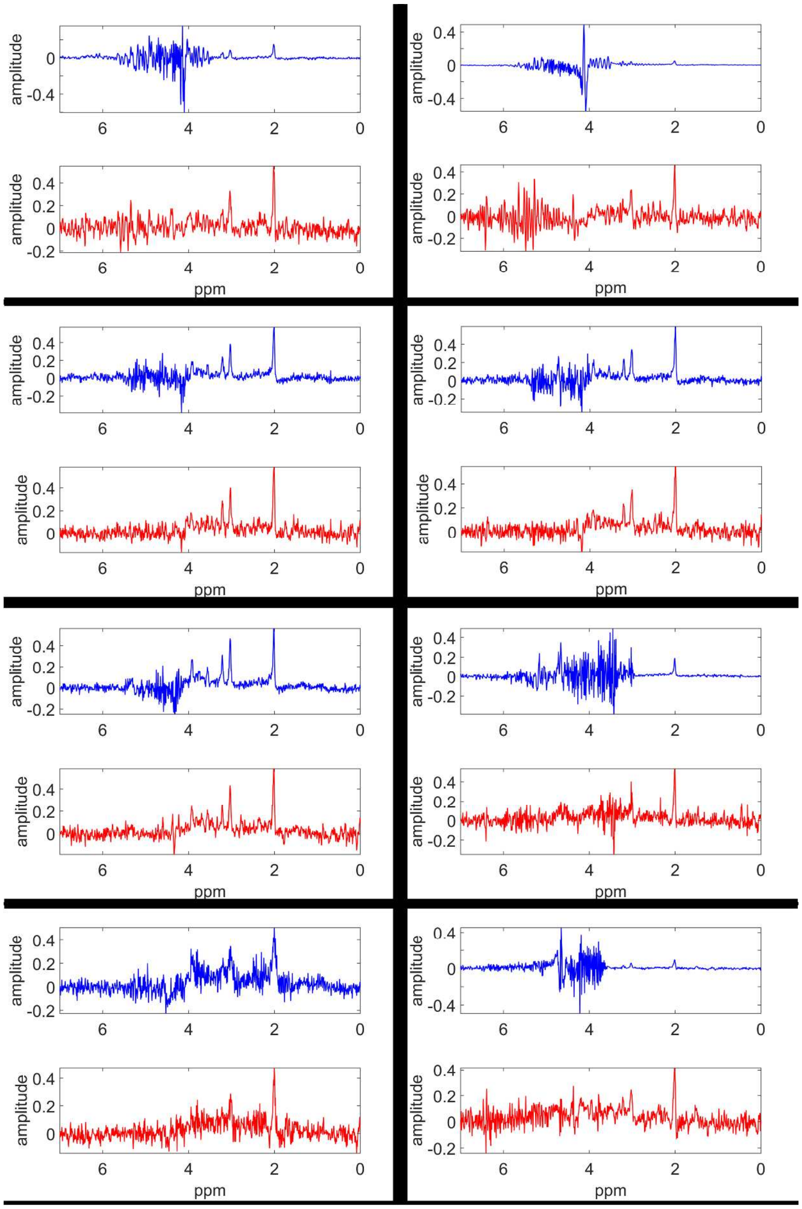
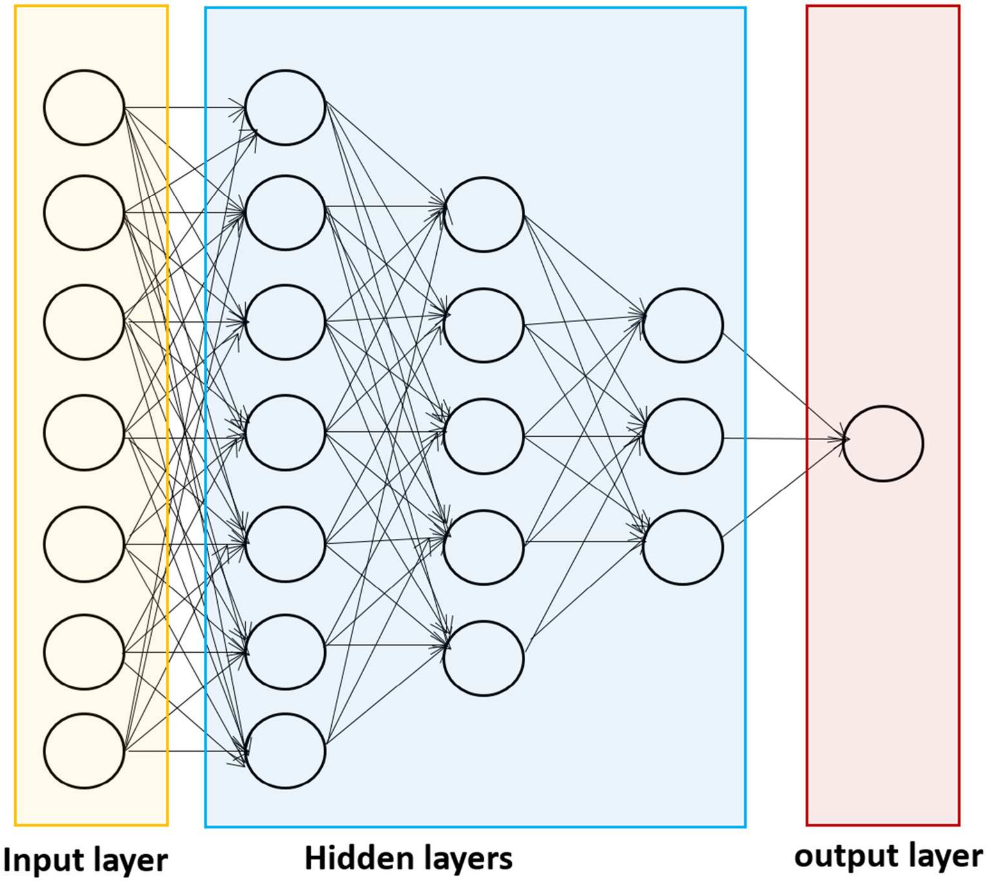


Figure 8: Illustration for the performance of ghost removal using DL on in-vivo spectra. The original in-vivo spectrum with ghosting artifacts is shown in blue and the "ghost-busted" spectrum predicted by the DL algorithm is shown in red.

226x341mm (600 x 600 DPI)



131x116mm (300 x 300 DPI)