# Real-time DCT Learning-based Reconstruction of Neural Signals

Rabeeh Karimi Mahabadi, Cosimo Aprile, Volkan Cevher
*Laboratory for Information and Inference Systems (LIONS)*
*École Polytechnique Fédérale de Lausanne (EPFL)*
{rabeeh.karimimahabadi, cosimo.aprile, volkan.cevher}@epfl.ch

***Abstract*—Wearable and implantable body sensor network systems are one of the key technologies for continuous monitoring of patient's vital health status such as temperature and blood pressure, and brain activity. Such devices are critical for early detection of emergency conditions of people at risk and offer a wide range of medical facilities and services. Despite continuous advances in the field of wearable and implantable medical devices, it still faces major challenges such as energy-efficient and low-latency reconstruction of signals. This work presents a power-efficient real-time system for recovering neural signals. Such systems are of high interest for implantable medical devices, where reconstruction of neural signals needs to be done in real-time with low energy consumption. We combine a deep network and DCT-learning based compressive sensing framework to propose a novel and efficient compression-decompression system for neural signals. We compare our approach with state-of-the-art compressive sensing methods and show that it achieves superior reconstruction performance with significantly less computing time.**

***Index Terms*—Neural signals, neural network, compressive sensing, learning-based signal processing, low-power, signal recovery.**

## I. Introduction

Implantable health monitoring devices using mobile and wireless technologies is an emerging field of research and recently has been receiving increasing attention [1]. These devices continuously monitor the patients and provide vital information about their health status. In these devices, first, the digitized neural signals are passed through a data compression module. Then, the compressed signals are transmitted to a receiver, where it reconstructs the neural signals [2].

In wearable health monitoring devices, transmitting signals by wireless requires orders of magnitude more energy than other functions of the device [2]. Therefore, it is vital to create a framework which addresses basic concerns like efficient energy consumption and low-latency reconstruction. This, in turn, requires using Compressive Sensing (CS) methods [3, 4] to reduce the cost of the wireless data transmission [2, 5]. Furthermore, low complexity of reconstruction methods in the receiver is crucial for wireless health monitoring sensors using batteries, where energy efficiency and battery life is essential.

Although the compressive sensing methods have been very successful in the past, they suffer from certain limitations. First, since they are iterative methods, they are often not fast enough to be applicable for real-time tasks. Especially, in medical monitoring devices, the real-time reconstruction is of paramount importance. For instance, when neurophysiology experts want to observe the neural signals to predict possible stroke, the reconstruction needs to be done in real-time.

Second, another challenge is limited available computational power in wireless or mobile implantable devices for monitoring activity of the brain. Solving demanding convex optimization problems requires high computation power. Moreover, the heat generated from solving these demanding problems can reduce the device's battery life.

Despite the continuous advances in implantable health monitoring devices field, it is still a challenge to find the appropriate combination of compression and reconstruction techniques to fulfill latency and power consumption requirements of such devices. We approach both of these challenges hands-on by developing a real-time reconstruction and compression method for neural signals. While our framework is suitable for general biomedical signal recovery, we focus on neural signals reconstruction, which has several applications in medical monitoring devices.

Following the recent enormous success of deep neural networks in various machine learning and computer vision applications such as classification, object detection, semantic segmentation, and natural language processing [6], we focus on building a neural network decoder in this paper. On the other hand, Baldassarre et al. [7] recently proposed a learning-based compressive subsampling approach which selects the mask indices that maximize the captured energy on average over the training set. Their method improves the recovery performance and reduces the data telemetry costs.

We leverage both of these ideas and present a DCT learning based neural network decoder for efficient reconstruction and compression of neural signals in a unified framework. By jointly using a deep network decoder for reconstruction and applying a DCT learning based compressive subsampling [7] method for compression of the neural signals, our method achieves real-time reconstruction; it substantially improves the reconstruction performances in comparison with compressive sensing methods and decreases the data transmission energy and cost.

The research leading to these results has received funding from Hasler Foundation Program: Cyber Human Systems (project number 16066), the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement n 725594 - time-data), and the Department of the Navy, Office of Naval Research (ONR) under the grant number N62909-17-1-2111.

The experiments prove that our method outperforms the solutions computed by state-of-the-art compressive sensing algorithms, while it reconstructs the signals hundreds of time faster than the conventional CS recovery methods. This real-time reconstruction comes at the expense of once offline training of the neural network, which is typical of any deep learning method. We also demonstrate empirically that for lower compression rates, our method performs as good as adaptive compression method which sets an upper limit for the achievable performance in linear decoders.

This paper is organized as follows. We provide the summary of previous related studies in Section II. We overview the main concepts of compressive sensing in Section III. In Section IV, we review the learning based compressive sensing framework, and overview the adaptive compression method. In Section V, we introduce our proposed method for joint compression and reconstruction of neural signals. We provide experimental results and comparison with state-of-the-art CS recovery methods in section VI. Finally, we conclude the paper in section VII and provide summarizing remarks.

## II. Literature review

Baldassarre et al. [7] proposed to learn the sampling mask based on the training data for linear decoders. Given a set of training signals, they learned the sampling mask which maximizes the captured energy on these signals on average. In this work, we learn the sampling mask from training data as proposed in [7]. However, in this work, we use a deep network, which is a nonlinear decoder, for signal reconstruction.

Aprile et al. [8] proposed a DCT Learning-based compressive subsampling method for neural signals. They compared their method with several recent randomized sampling approaches, i.e., Bernoulli [2], Structured Hadamard sampling [9], and Multi-Channel Sampling [5] developed for compression of neural signals. They used Hierarchical Group Lasso in conjunction with these sampling patterns to reconstruct the neural signals which were shown in [9] to result in the best performance. Their proposed method improved the reconstruction quality compared to the considered baseline approaches. The authors in [8] used linear decoder in their work. However, in this paper, by utilizing a nonlinear deep learning decoder, our method outperforms their method and obtains better reconstruction results.

Recently, Majumdar and Ward [10] proposed a neural network model based on stacked autoencoder to reconstruct the EEG signals. Their method performs slightly worse than the CS methods. However, in this work, we focus on reconstruction of neural signals. The architecture we use in this work is different from theirs. Additionally, by employing the learning-based approach to efficiently compute the mask indices, our model outperforms the CS recovery methods.

## III. Compressive sensing for signal reconstruction

Compressive sensing [3] is the problem of reconstructing a sparse vector $\boldsymbol{x} \in \mathbb{R}^p$ using a small number ($n < p$) of linear measurements. In the case of the neural signals, these measurements take the specific form of subsampled DCT measurements, described as follows:

$$\boldsymbol{b} = \boldsymbol{P}_{\boldsymbol{\Omega}} \boldsymbol{\Psi} \boldsymbol{x}, \tag{1}$$

where $\boldsymbol{\Psi}$ is the DCT transform operator applied to the signal and $\boldsymbol{P}_{\boldsymbol{\Omega}} : \mathbb{R}^p \to \mathbb{R}^n$ is a subsampling operator that retains only the rows of $\boldsymbol{\Psi}$ indexed by the set $\boldsymbol{\Omega}$, with $|\boldsymbol{\Omega}| = n$. The selected set of indices $\boldsymbol{\Omega}$ is also known as the sampling pattern or the mask. The vector $\boldsymbol{b}$ is the compressive measurement of the signal $\boldsymbol{x}$ with the compression rate of $\frac{n}{p}$. Then, the goal of the reconstruction algorithm (also known as the decoder) is to obtain an estimate $\hat{\boldsymbol{x}}$ of $\boldsymbol{x}$. The reconstruction method can be thought as a general function $g$ and is written as follows:

$$\hat{\boldsymbol{x}} = g(\boldsymbol{\Omega}, \boldsymbol{b}). \tag{2}$$

The signal can be approximately recovered using the fast linear decoder:

$$\hat{\boldsymbol{x}} = \boldsymbol{\Psi}^* \boldsymbol{P}_{\boldsymbol{\Omega}}^T \boldsymbol{b}. \tag{3}$$

A plethora of non-linear methods has also been proposed for solving compressive sensing problems. These reconstruction methods are mostly based on solving a convex optimization problem and enjoy theoretical guarantees. Here, we briefly review the most widely used CS recovery methods. One of the most well-known reconstruction methods is *Basis Pursuit* (BP) [3] which solves the following problem:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{z}:\boldsymbol{b}=\boldsymbol{P}_{\boldsymbol{\Omega}}\boldsymbol{\Psi}\boldsymbol{z}}{\arg\min} \; ||\boldsymbol{\Phi}\boldsymbol{z}||_1, \tag{4}$$

where $\boldsymbol{\Phi}$ is the sparsifying transform which converts the signal to a domain in which the signal has a sparse representation. As an example, speech signals have a sparse representation in Short Time Fourier transform domain, and images are sparse in the wavelet domain.

*Total variation* (TV) minimization formulation [11] is another widely used convex optimization based method which does not require sparsifying operators:

$$\hat{\boldsymbol{x}} = \underset{\boldsymbol{z}:\boldsymbol{b}=\boldsymbol{P}_{\boldsymbol{\Omega}}\boldsymbol{\Psi}\boldsymbol{z}}{\arg\min} \; ||\boldsymbol{z}||_{\mathrm{TV}}, \tag{5}$$

where $||\boldsymbol{z}||_{\mathrm{TV}}$ is the total variation norm.

## IV. Learning-based Compressive sensing framework

In this section, we outline the learning based compressive sensing framework [7]:

- Let $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_m$ be a set of training signals, and $\boldsymbol{x}$ be an unknown signal which has similar properties to the training signals.
- The goal is to seek a sampling pattern with the maximum empirical average performance on the training signals:

$$\hat{\boldsymbol{\Omega}} = \underset{\boldsymbol{\Omega}:|\boldsymbol{\Omega}|=n}{\arg\max} \frac{1}{m} \sum_{j=1}^{m} \eta_{\boldsymbol{\Omega}}(\boldsymbol{x}_j), \tag{6}$$

where $\eta_{\boldsymbol{\Omega}}$ represents any reconstruction performance measure (e.g., PSNR). The main idea is that since it is assumed that the training signals are close to the unknown signal $\boldsymbol{x}$, by maximizing equation (6), one can expect the obtained $\boldsymbol{\Omega}$ performs also well on $\boldsymbol{x}$.

By choosing average energy on the training signals as the performance criterion $\eta_{\boldsymbol{\Omega}}$, the sampling pattern $\boldsymbol{\Omega}$ is learned by selecting the indices which preserve most of the energy on average on the training signals:

$$\hat{\boldsymbol{\Omega}} = \underset{\boldsymbol{\Omega}, |\boldsymbol{\Omega}|=N}{\arg\max} \frac{1}{m} \sum_{j=1}^{m} \sum_{i \in \boldsymbol{\Omega}} |\langle \boldsymbol{\Psi}_i, \boldsymbol{x}_j \rangle|^2, \tag{7}$$

where $\boldsymbol{\Psi}_i$ is the transpose of the i-th row of $\boldsymbol{\Psi}$.

Solution to the equation (7) can be found efficiently via sorting and selecting the $N$ indices with largest values of $\frac{1}{m} \sum_{j=1}^{m} |\langle \boldsymbol{\Psi}_i, \boldsymbol{x}_j \rangle|^2$ [7]. The training signals are then compressed using compression operator $\boldsymbol{P}_{\boldsymbol{\Omega}}\boldsymbol{\Psi}$, where $\boldsymbol{\Psi}$ is the DCT basis and $\boldsymbol{\Omega}$ denotes the computed sampling indices. We divide the training signals into intervals of a specific length, then this method computes the indices which maximize the retained energy over all the intervals on average.

However, there is a tradeoff between energy consumption of the method and the reconstruction quality. There is another approach which computes the best indices for each window of the signal called adaptive compression method. In this method, the optimal linear encoding of neural signals requires computation of all DCT coefficients $\boldsymbol{\Psi x}$ of the full-length signals and then selecting the optimal mask indices specific to each interval of the signals. It results in a higher reconstruction quality compared to the learning-based compressive sensing framework with the cost of large energy consumption, which is critical in implantable medical devices with limited available energy.

## V. Method

In this section, we explain our proposed DCT-learning based neural network method. We take a block-based recovery approach. Such methods divide the training data into smaller blocks and reconstruct each block separately [12, 13]. We divide the signal $\boldsymbol{x}$ of length $p$ into $N = \frac{p}{l}$ blocks of length $l$. Block-based compressive sensing recovery methods have the following advantages: 1) Storing the measurement matrix $\boldsymbol{P}_{\Omega}\boldsymbol{\Psi}$ requires less memory since it needs storage of size $n \times l$ instead of the full measurement matrix of size $n \times p$. 2) The decoder recovers each block separately and the encoder does not need to transfer the entire samples of the signal to have the signal reconstructed in the receiver. Therefore, the reconstruction process can be done for each block separately, it results in a considerable speed up in reconstruction and makes the method more suitable for real-time applications.

We set the block length $l = 256$ and we divide the signals into intervals of length 256. Next, we apply the learning-based compressive sensing framework as explained in Section IV and compute the indices $\boldsymbol{\Omega}$ that maximize the preserved average energy on the training signals. Each training signal is then projected using the computed compression operator $\boldsymbol{P}_{\boldsymbol{\Omega}}\boldsymbol{\Psi}$. The compressive measurements $\boldsymbol{b}$ are then transmitted to the receiver, where the signal needs to be reconstructed.

As described in Equation (2), we need to learn a function which can map the compressive measurements $\boldsymbol{b}$ to the original signal $\boldsymbol{x}$. We first apply the transpose of the projection matrix $\boldsymbol{\Psi}^* = \boldsymbol{\Psi}^{-1}$ on the measurements to compute the poor estimate of the signal as follows:

$$\boldsymbol{x}' = \boldsymbol{\Psi}^* \boldsymbol{P}_{\boldsymbol{\Omega}}^T \boldsymbol{b}, \tag{8}$$

where $\boldsymbol{x}'$ represents the poor noisy estimate of the signal $\boldsymbol{x}$. We then train a deep network which cleans the noisy input samples, and map $\boldsymbol{x}'$ to the original signal $\boldsymbol{x}$. We explain our proposed network topology in the next section.

### A. Neural Network Topology

Our network topology is shown in Figure 1. Our proposed network consists of 1 convolutional, 1 deconvolutional, 3 dense, 2 flatten, 2 reshape and 4 Rectified Linear Unit (ReLU) layers. The detailed network topology is described as follows:

- $I_0$: The input layer with an input data size of $[1 \times 256]$, where 256 is the length of each sequence of the signal.
- $R_0$: Reshape layer which transforms the input data of size $[1 \times 256]$ to the data of size $[16 \times 16 \times 1]$.
- $CV_0$: First hidden layer, composed of 4 convolutional filters of size $[5 \times 5]$ with a ReLU layer, which introduces non-linearity in decision function of the overall network. This layer transforms the previous layer's output to the data of size $[12 \times 12 \times 4]$.
- $F_0$: This layer flattens the previous layer's activation map to the data of size $[1 \times 576]$.
- $D_0$: This layer is a fully connected layer, composed of 128 neurons, and a ReLU layer. This layer transforms the previous layer's output to the data of size $[1 \times 128]$.
- $D_1$: This layer is a fully connected layer. It is composed of 64 neurons, and a ReLU layer. This layer transforms the previous layer's output to the data of size $[12 \times 12 \times 4]$.
- $R_1$: Flattening layer which transforms the previous layer's output to the data of size $[1 \times 576]$.
- $DV_0$: This hidden layer is composed of 4 deconvolutional filters of size $[5 \times 5]$, and a ReLU layer. This layer changes the input to the layer to the data of size $[12 \times 12 \times 4]$.
- $F_1$: Flattening layer which transforms the given input to this layer to the data of size $[1 \times 256]$.
- $D_2$: Fully connected layer composed of 256 neurons and transforms the input to this layer to the output of size $[1 \times 256]$.

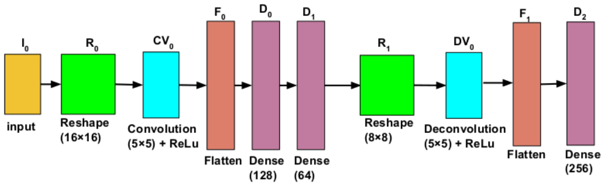

Fig. 1: Our proposed deep learning architecture for reconstruction of the neural signals.

## VI. Experimental results

In this section, we provide the numerical experiments demonstrating the performance of our proposed DCT-learning based neural network framework and compare it with the state-of-the-art CS recovery methods. We consider the decoders explained in Section III which we refer to as BP and TV, and compare them with our proposed method.

### A. Implementation Details

We implement both BP and TV minimization using NESTA (a shorthand for Nesterov's algorithm), which is a well implemented toolbox including several fast and robust first-order methods for solving basis-pursuit problems and their extentions [14]. For BP, we consider DCT transform as the sparsifying operator $\mathbf{\Phi}$.

We train our proposed deep network for a maximum of $4 \times 10^4$ epochs, using RMSprop optimizer [15] with the learning rate of $10^{-4}$ and the learning rate decay of $10^{-6}$ over each update. We use mean squared error objective. The network architecture is implemented in Python using Keras [16] and Theano [17] backend. We train the network using one Tesla K40c GPU.

### B. Datasets

The experiments are carried out on the $I001-P034-D01$ dataset from $iEEG.org$ portal. This portal contains several $EEG$ and $iEEG$ datasets which are manually annotated by expert clinicians. This data consists of approximately 1 day, 8 hours, and 10 minutes of recording at $5kHz$, which is approximately $6 \times 10^8$ samples. We used the first 4864000 samples, and we extracted signals for channel 1-31. Then, we split the samples into $60\%$ for training, $20\%$ for test, and $20\%$ for validation. The training data is used to learn both the sampling patterns with learning-based compressive sensing framework and learning the parameter of our proposed neural network.

### C. Comparison to the Baselines

We learn a fixed sampling pattern using learning-based compressive sensing framework as described in Section IV. We then employ the obtained sampling mask to compress all the intervals of the test signals. The reconstruction is then performed using the deep network described in Section V. Furthermore, we also consider the linear decoder (3) with the learned sampling mask as proposed in [8], we refer to this method as *DCT-LB Linear*. In addition, we also consider adaptive compression method explained in Section IV. In this method, we compute the specific sampling pattern for each window of the test signals. The reconstruction is then performed using the linear decoder (3), we refer to this method as *DCT-adaptive*.

### D. Performance Evaluation

We measure the reconstruction performance in terms of SNR. The SNR for each channel is computed as:

$$\text{SNR}_j = 20 \log_{10} \left( \frac{||\boldsymbol{x}_j||_2}{||\boldsymbol{x}_j - \hat{\boldsymbol{x}}_j||_2} \right), \tag{9}$$

where $\boldsymbol{x_j}$ denotes the test signal for channel $j$, and $\hat{\boldsymbol{x}}_{\boldsymbol{j}}$ represents the reconstructed test signal for channel $j$. Then, to obtain the final SNR, we compute the average SNR over all the channels:

$$\text{SNR}_{\text{avg}} = \frac{1}{D} \sum_{i=1}^{D} SNR_j, \tag{10}$$

where $D$ is the number of channels in total, and $\text{SNR}_{\text{avg}}$ represents the average obtained SNR over all the channels.

### E. Numerical Results

This section shows that our proposed method can reconstruct the neural signals in real-time while obtaining better reconstruction quality compared to the compressive sensing methods. Table I shows the reconstruction results. The results show that our proposed method outperforms the state-of-the-art CS reconstruction techniques. Additionally, for low compression rates, it even performs favorably compared to the adaptive compression method which computes the optimal mask indices for each block of the signal and sets the upper limit for achievable performance with the linear decoder.

TABLE I: The obtained averaged reconstruction SNRs with different methods.

| Method \ Compression rate | 4 | 8 | 16 | 32 |
|---|---|---|---|---|
| DCT-adaptive | 41.75 | 40.08 | 37.42 | 32.45 |
| Our method | **41.81** | **40.12** | 36.81 | 30.80 |
| DCT-LB Linear [8] | 40.63 | 38.79 | 36.01 | 30.37 |
| TV [11] | 40.30 | 38.47 | 35.75 | 30.28 |
| BP [3] | 40.30 | 38.48 | 35.75 | 30.28 |

Next, we compare the reconstruction time of our proposed method against the CS signal recovery techniques. We run the experiments on an Intel Core i7 CPU (2.80 GHz) with 16 GB RAM running on Ubuntu 17.04. Table II shows the reconstruction time for our method compared to the CS recovery methods for a signal block of length $256$. Our method is more than 100 times faster and can be used in applications with real-time reconstruction demand.

TABLE II: Required time in seconds for reconstruction of a neural signal of length $256$ with different recovery methods.

| Method | Time (seconds) |
|---|---|
| Our method | 0.00005 |
| TV [11] | 0.00594 |
| BP [3] | 0.00739 |
| DCT-adaptive | 0.00021 |
| DCT-LB Linear [8] | 0.00067 |

For visual evaluation, we present sample results in Figure 2. We observe that our method is able to reconstruct the signals with different variations.

## VII. Conclusion

In this paper, we have developed a DCT-learning based deep learning method for recovering neural signals. We showed that our framework efficiently approximates the signal while decreasing the reconstruction time drastically. By leveraging the DCT-learning based compressive sensing framework, our real-time method can be run with limited computational power, which makes it suitable for implantable monitoring medical devices used in mobile and wireless devices. We provided the comparison to the state-of-the-art compressive sensing methods and showed that our method outperformed them with substantially less running time.

## References

[1] A. Darwish and A. E. Hassanien. Wearable and implantable wireless sensor network solutions for healthcare monitoring. *Sensors*, 11(6): 5561–5595, 2011.

[2] F. Chen, A. P. Chandrakasan, and V. M. Stojanovic. Design and analysis of a hardware-efficient compressed sensing architecture for data compression in wireless sensors. *IEEE Journal of Solid-State Circuits*, 47(3):744–756, 2012.

[3] D. L. Donoho. Compressed sensing. *IEEE Transactions on information theory*, 52(4):1289–1306, 2006.

[4] E. J. Candès et al. Compressive sampling. In *Proceedings of the international congress of mathematicians*, volume 3, pages 1433–1452. Madrid, Spain, 2006.

[5] M. Shoaran, M. Kamal, C. Pollo, P. Vandergheynst, and A. Schmid. Compact low-power cortical recording architecture for compressive multichannel data acquisition. *IEEE transactions on biomedical circuits and systems*, 8(6):857–870, 2014.

[6] Y. Guo, Y. Liu, A. Oerlemans, S. Lao, S. Wu, and M. S. Lew. Deep learning for visual understanding: A review. *Neurocomputing*, 187:27–48, 2016.

[7] L. Baldassarre, Y. Li, J. Scarlett, B. Gözcü, I. Bogunovic, and V. Cevher. Learning-based compressive subsampling. *IEEE Journal of Selected Topics in Signal Processing*, 10(4):809–822, 2016.

[8] C. Aprile, J. Wüthrich, L. Baldassarre, Y. Leblebici, and V. Cevher. Dct learning-based hardware design for neural signal acquisition systems. In *Proceedings of the Computing Frontiers Conference*, pages 391–394. ACM, 2017.

[9] L. Baldassarre, C. Aprile, M. Shoaran, Y. Leblebici, and V. Cevher. Structured sampling and recovery of ieeg signals. In *Computational Advances in Multi-Sensor Adaptive Processing (CAMSAP), 2015 IEEE 6th International Workshop on*, pages 269–272. IEEE, 2015.

[10] A. Majumdar and R. Ward. Real-time reconstruction of eeg signals from compressive measurements via deep learning. In *Neural Networks (IJCNN), 2016 International Joint Conference on*, pages 2856–2863. IEEE, 2016.

[11] A. Chambolle. An algorithm for total variation minimization and applications. *Journal of Mathematical imaging and vision*, 20(1-2):89–97, 2004.

[12] Y. C. Eldar, P. Kuppinger, and H. Bolcskei. Block-sparse signals: Uncertainty relations and efficient recovery. *IEEE Transactions on Signal Processing*, 58(6):3042–3054, 2010.

[13] L. Gan. Block compressed sensing of natural images. In *Digital Signal Processing, 2007 15th International Conference on*, pages 403–406. IEEE, 2007.

[14] S. Becker, J. Bobin, and E. J. Candès. Nesta: A fast and accurate first-order method for sparse recovery. *SIAM Journal on Imaging Sciences*, 4(1):1–39, 2011.

[15] G. Hinton, N. Srivastava, and K. Swersky. Lecture 6a overview of mini–batch gradient descent.

[16] F. Chollet et al. Keras. https://github.com/fchollet/keras, 2015.

[17] R. Al-Rfou, G. Alain, A. Almahairi, C. Angermueller, D. Bahdanau, N. Ballas, F. Bastien, J. Bayer, A. Belikov, et al. Theano: A python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*, 2016.

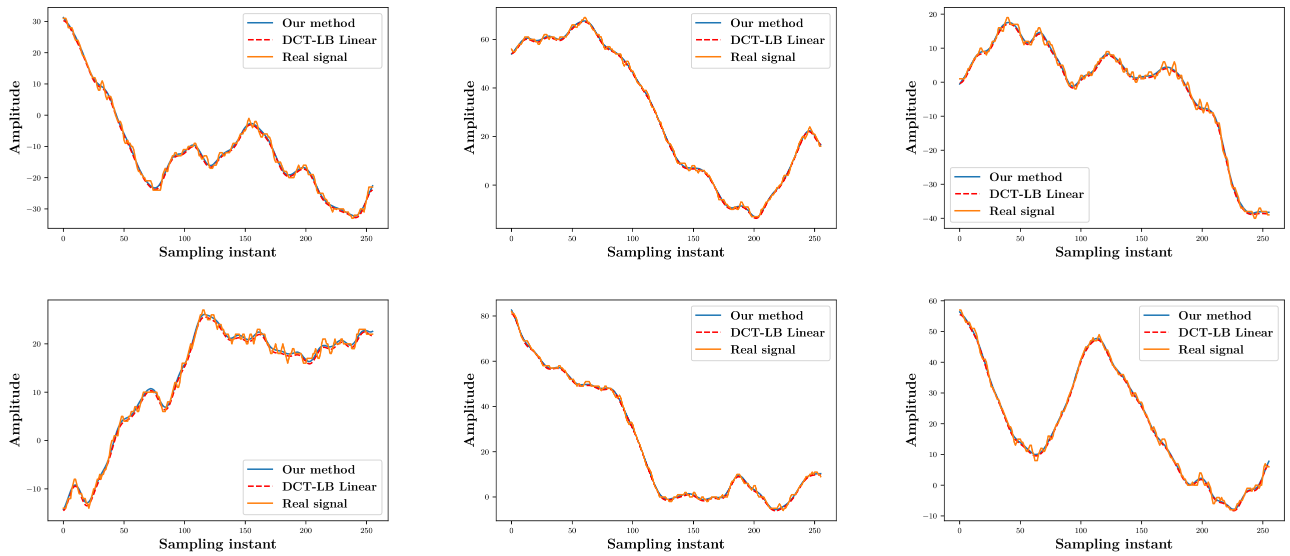

Fig. 2: The obtained sample results with our proposed method, DCT-LB linear approach, and their corresponding real signals.