An Unsupervised Compressed Sensing Algorithm for Multi-Channel Neural Recording and Spike Sorting

Tao Xiong, *Student Member, IEEE*, Jie Zhang, *Member, IEEE*, Clarissa Martinez-Rubio, Chetan S. Thakur, *Member, IEEE*, Emad N. Eskandar, Sang Peter Chin, *Member, IEEE*, Ralph Etienne-Cummings, *Fellow, IEEE*, and Trac D. Tran, *Fellow, IEEE*

***Abstract*— We propose an unsupervised compressed sensing (CS)-based framework to compress, recover, and cluster neural action potentials. This framework can be easily integrated into high-density multi-electrode neural recording VLSI systems. Embedding spectral clustering and group structures in dictionary learning, we extend the proposed framework to unsupervised spike sorting without prior label information. Additionally, we incorporate group sparsity concepts in the dictionary learning to enable the framework for multi-channel neural recordings, as in tetrodes. To further improve spike sorting success rates in the CS framework, we embed template matching in sparse coding to jointly predict clusters of spikes. Our experimental results demonstrate that the proposed CS-based framework can achieve a high compression ratio (8:1 to 20:1), with a high quality reconstruction performance (>8 dB) and a high spike sorting accuracy (>90%).**

***Index Terms*— Compressed sensing, unsupervised, dictionary learning, neural recording, spike sorting, multi-channel.**

I. Introduction

HIGH-DENSITY multi-electrode neural recording microsystems have evolved over the years to become essential tools in neural electrophysiology experiments [1]–[9]. These microsystems monitor brain activity by collecting extracellular neural action potentials (or spikes) from different areas of the brain. Using high-density multi-electrodes array (MEA) or tetrode drives, the action potential of each neuron can be recorded by multiple electrodes in its proximity. This redundancy of features can greatly improve the spike clustering accuracy. However, the drawback is that the large number of electrodes generate large amount of data. This presents itself as a challenge for the design of the implantable system in terms of chip size and power consumption. Typically, spikes are sampled at around 30 kHz at a resolution of more than 10 bits. A multi-channel neural recording system containing up to thousands of channels generates data at the rate of 300 Mbps [10]. This would cost around 50 mW to transmit wirelessly [10], which results in significant heat dissipation and impedes large-scale integration as the electronics are very close to the side of recording.

Compressed sensing (CS) [11], [12] techniques have been proposed to address the challenge of dealing with large amount of data. For example, Mamaghanian *et al.* [13] incorporated CS in a real-time energy efficient framework for electrocardiogram (ECG) compression. This CS-based ECG compression outperformed the conventional digital wavelet transform (DWT)-based approach and was able to improve power efficiency. Another CS-based ECG compression system demonstrated that the signal could be compressed and reconstructed at a compression ratio of 4:1 to 16:1 with dynamic thresholding [14]. Chen *et al.* [15] also proposed a hardware-efficient CS architecture for data compression in wireless sensors, which had a power consumption of only 1.9 μW, thus significantly improving the power efficiency of such systems. Another CS-based system that exploits the 'rakeness' approach to maximize the amount of information contained in the measurements demonstrated a superior performance with a compression ratio of 8:1 and 10:1 [16], [17]. Furthermore, the CS-based system has now been extended to multi-channel systems. Gangopadhyay *et al.* [18] designed a 64-channel CS analog front-end for biosensor applications, which could recover and preserve most of features in the signal at a compression ratio of 2:1 to 6:1. Zhang *et al.* [19] also proposed a 4-channel closed-loop CS neural recording system, which was able to achieve >10 times the compression ratio

Manuscript received August 9, 2017; revised February 14, 2018; accepted April 10, 2018. Date of publication April 26, 2018; date of current version June 6, 2018. This work was supported in part by NSF under Grant DMS-1222567 and in part by AFOSR under Grant FA9550-12-1-0136. *(Corresponding author: Tao Xiong.)*

T. Xiong, R. Etienne-Cummings, and T. D. Tran are with the Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD 21218 USA (e-mail: tao.xiong@jhu.edu).

J. Zhang is with the Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139 USA.

C. Martinez-Rubio is with National Parkinson Foundation, Miami, FL 33131 USA.

C. S. Thakur is with the Department of Electronic Systems Engineering, Indian Institute of Science, Bengaluru 560012, India.

E. N. Eskandar is with the Harvard Medical School, Boston, MA 02115 USA, and also with the Department of Neurosurgery, Massachusetts General Hospital, Boston, MA 02114 USA.

S. P. Chin is with the Department of Electrical and Computer Engineering, Johns Hopkins University, Baltimore, MD 21218 USA, also with the Department of Brain and Cognitive Sciences, Massachusetts Institute of Technology, Cambridge, MA 02139 USA, and also with the Department of Computer Science, Boston University, Boston, MA 02215 USA.

Digital Object Identifier 10.1109/TNSRE.2018.2830354

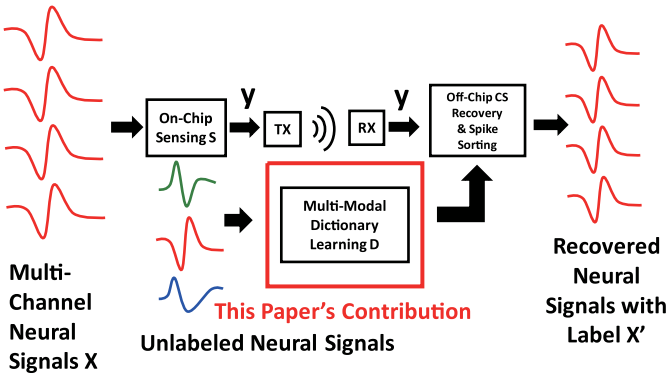

Fig. 1. Basic block diagram of the proposed CS neural recording system. In the CS approach, multi-channel signals are randomly sampled by an on-chip sensing matrix S and then wirelessly transmitted to an off-chip terminal for reconstruction and sorting. The design of the multi-modal dictionary learning D for sparsifying signals is the major contribution of our proposed CS approach.

while consuming only 0.83 μW per channel. Li *et al.* [20] presented a 256-channel digital signal processing system using the CS technique, which has achieved a power consumption of 12.5 μW per channel at a data reduction of around 90%. Liu *et al.* [21] designed a highly configurable 16-channel CS module for chronic recording and brain machine interface, featuring a compression ratio of 8:1.

Despite their advantages, the CS-based systems developed previously for signal compression suffer from several limitations. Most of the CS-based systems are non-adaptive, and use a signal-agnostic dictionary such as the identity matrix or the wavelet matrix to sparsify the signal. It has been shown that the use of a signal-dependent dictionary improves the reconstruction quality and compression ratio (>10:1) compared to the signal-agnostic dictionary [19], [22]–[24]. The signal-dependent dictionary helps increase the sparsity of the signal significantly in the signal-oriented basis. From the perspective of multi-channel neural recordings, the previous CS-based systems compress the time varying neural signal on single electrode. They do not consider the signal characteristics and correlation at adjacent recording electrodes in the system design. As a result, the model does not take these useful spatial information into account for signal compression and reconstruction. Additionally, the previous CS-based systems do not incorporate the feature of online analysis, such as spike sorting, in real-time experiments. In order to overcome these limitations in CS-based multi-channel neural recordings and to enable online analysis, we propose a CS-based approach that is more suitable for multi-channel recordings, and combine the post-processing such as spike sorting during the reconstruction process. As shown in Figure 1, the off-chip design of the multi-channel dictionary learning serves as the most important component of our CS framework with the following contributions:

A. Multi-Channel Dictionary Learning Using Joint-Group Sparsity

In multi-channel neural recording systems, such as the tetrodes [2], several close-by electrodes around neurons collect spikes simultaneously. Spikes recorded by these electrodes

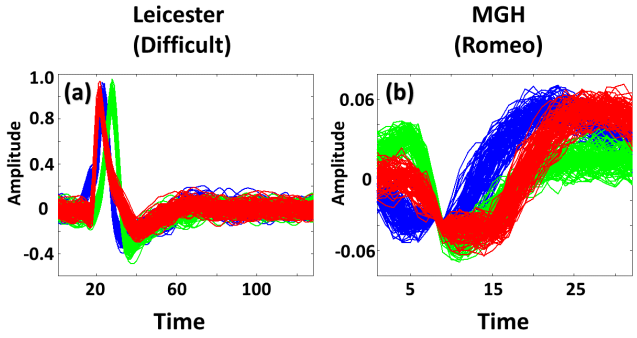

Fig. 2. An illustration of the different clusters (color-coded spikes) of neural signals from the Leicester [27] and MGH databases [26]. In spike sorting, spikes are grouped into different clusters based on their distinct shapes. Normally, one cluster corresponds to a single neuron.

share similar patterns and features. To take advantage of the correlation among these electrodes in sensing and recovering, we introduce a joint-group sparsity constraint in dictionary learning to enforce spikes recorded at electrodes in close proximity to be recovered using similar items from the dictionary. Compared to the conventional group sparsity in [25], the proposed work is focused on marrying the joint-group sparsity to the dictionary learning in both coding and update stages. This dynamic combination greatly improves the compression ratio and reconstruction performance.

B. Online Spike Sorting Using Spectral Clustering and Group Sparsity

An electrode can detect spikes from a group of neurons in its proximity. As shown in Figure 2, these spikes have particular shapes and can be clustered, corresponding to different neurons. Conventional spike sorting, which was supervised and for offline post-processing, used prior information to train a classifier. However, the large amounts of spikes generated are not labeled in real-time experiments. Neuroscientists have to manually sort and cluster these spikes using offline sorting software (e.g., Plexon) [26]. This process is neither time-efficient, nor integrated with the neural recording systems to realize online spike sorting. In our work, we incorporate spectral clustering to initialize the group structure of the dictionary and enable unsupervised spike sorting. Therefore, spikes can be represented and sorted based on the group sparsity in an unsupervised manner. Furthermore, the spectral clustering also helps the convergence of the dictionary learning.

The rest of the paper is organized as follows: In section II, the CS theory and background are recalled. In section III, we introduce our signal model, dictionary-learning algorithm, recovery and spike sorting approach. In section IV, we compare the proposed approach with other CS-based approaches using several neural databases, including the synthetic database, and real spikes recorded from animals. In section V, we present the conclusion and discussion.

II. Background

A. Compressed Sensing and Sparse Representation

The CS theory [11], [12] demonstrates that an S-sparse signal $\mathbf{x}$ of length N is able to be compressed into a measurement vector $\mathbf{y}$ of length M by a matrix $\mathbf{S}$ of dimension

M by N satisfying the Restricted Isometry Property and $M \sim S\log(\frac{N}{S})$, where normally $S \gg M < N$. Specifically, the S-sparse signal is defined as a signal of which only S coefficients are non-zero elements in the entire length of N or can be approximately represented by its largest S coefficients.

By solving the ℓ_1 norm optimization problem below, the sparse signal $\mathbf{x}$ can be recovered with high probability.

$$\min_{\mathbf{x}} \ ||\mathbf{x}||_1 \quad s.t. \ \mathbf{y} = \mathbf{S}\mathbf{x}.$$

Normally, biomedical signals such as action potentials and electroencephalogram signals are not sparse in time or frequency domains. Each neuron generates spikes with a characteristic shape and amplitude based on its morphology and proximity to electrodes. Spikes collected during neural recordings are generally stable over time. As a result, it is possible to construct a signal-dependent dictionary matrix $\mathbf{D}$ of dimension N by L to represent spikes sparsely, which transforms the non-sparse signal $\mathbf{x}$ of length N into a S-sparse vector $\mathbf{a}$ of length L and normally $N \gg L$. Therefore, the non-sparse signal $\mathbf{x}$ can be represented as $\mathbf{x} = \mathbf{D}\mathbf{a}$, which is defined as the linear combination of a few atoms from the dictionary. Now the original ℓ_1 optimization problem becomes:

$$\min_{\mathbf{a}} \ ||\mathbf{a}||_1 \quad s.t. \ \mathbf{y} = \mathbf{S}\mathbf{D}\mathbf{a}.$$

By solving the above optimization problem, we have the sparse vector $\mathbf{a}$ and the recovered non-sparse signal $\hat{\mathbf{x}} = \mathbf{D}\mathbf{a}$. Intuitively, the CS approach reconstructs the original spike $\mathbf{x}$ of length N from the measurement $\mathbf{y}$ of length M, achieving a compression ratio of $\frac{N}{M}$, which provides a promising way to compress the neural signal during data transmission.

B. Dictionary Learning

In order to further reduce the length M of the measurement $\mathbf{y}$, a dictionary $\mathbf{D}$ should be designed to sparsely represent the signal $\mathbf{x}$ as much as possible, according to the CS theory. For neural recordings, there are two approaches of choosing a sparsifying dictionary in the CS framework. The first approach incorporates signal-agnostic dictionaries such as the identity or the wavelet dictionary, which can represent spikes in the time-frequency domain. The second approach trains a signal-dependent dictionary using prior information of spikes, since neural recording electrodes collect unique and repetitive spikes from neurons. Our previous works [22], [28] have demonstrated that the signal-dependent dictionary is superior to the signal-agnostic dictionary in terms of compression ratio, reconstruction quality, and spike sorting accuracy. Therefore, the ability to design a robust dictionary is key in determining the efficiency of a CS neural recording system.

The task of "dictionary learning" involves training a dictionary $\mathbf{D}$ representing the training data samples $\mathbf{X}$ as sparse compositions by optimizing the problem:

$$\min_{\mathbf{D},\mathbf{A}} \ ||\mathbf{X} - \mathbf{D}\mathbf{A}||_2^2 \quad s.t. \ \forall i, \ ||\mathbf{a}_i||_0 \leq S.$$

$\mathbf{a}_i$, which is a column item of $\mathbf{A}$, indicates the S-sparse vector for i-th sample of training database and $\mathbf{A}$ indicates the sparse coefficients matrix.

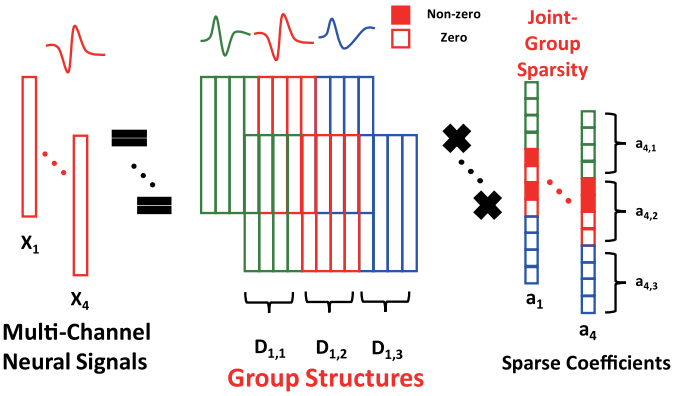

Fig. 3. Intuitive illustration of the proposed signal model in tetrode setup with discriminative group structures (color-coded blocks) and joint-group sparsity (red filled) for multi-mode structured dictionary learning.

C. Compressed Sensing Neural Recording Systems

Many compression systems have incorporated the CS technique for processing biosignals [13]–[15], [17]–[19], [24], [29]. These CS-based neural recording systems are able to achieve a high power efficiency as well as a high density integration owing to the implementation of a simple circuit. The sensing matrix $\mathbf{S}$ can be implemented on chip to compress the signal in the front-end [14], [18], [19]. Furthermore, the CS-based systems provide the flexibility of choosing a suitable dictionary $\mathbf{D}$ as the on-chip random sensing mechanism independent from the sparse representation basis. Currently, there are two different methods in the design of the sparse representation basis. One method is to use a signal-agnostic dictionary, such as the identity and wavelet basis [13]–[15], [18], which is independent from the signal itself. Another approach is to use a signal-dependent dictionary as the representation basis, which is adaptive and learns from the training samples. Previous works [19], [22], [25], [28] have demonstrated that the signal-dependent CS dictionaries have superior performance over other compression neural recording methods including spike detection, wavelet and other CS-based approaches in terms of compression ratio, reconstruction quality, spike sorting success rate, and chip power consumption.

III. Method

A. Joint-Group Sparsity

We assume neural spikes $\mathbf{X} \in \mathbb{R}^{C\times T}$, recorded from C channels, belongs to G clusters or groups. T is the discrete length of the waveforms. Now, our goal is to use an unsupervised method to train a dictionary $\mathbf{D}$ to sparsely represent signal $\mathbf{x}$ (a column of $\mathbf{X}$). We also desire that the dictionary, $\mathbf{D}$, to have discriminative properties. As shown in Figure 3, $\mathbf{D}$ can be seen as a concatenation of sub-dictionaries $\mathbf{D}_{c,g}$. This organized construction would allow spikes from channel c belonging to group g to only have only non-zero sparse coefficients $\mathbf{a}_{c,g}$ in sub-dictionary $\mathbf{D}_{c,g}$, while having zero coefficients $\mathbf{a}_{c,g'}$ in other sub-dictionaries $\mathbf{D}_{c,g'}$ where $g' \neq g$.

Furthermore, multi-electrode recording techniques ofter rely on signal correlation between different adjacent electrodes to

perform spike clustering and sorting. Similarly, we can also take this property into sparse dictionary learning. For example, in the case of tetrodes, four closely bundled electrodes capture the neural activities of the surrounding cells. Due to the close proximity of the electrodes, the spike waveforms $\mathbf{x}_{c=1}$, $\mathbf{x}_{c=2}$, $\mathbf{x}_{c=3}$ and $\mathbf{x}_{c=4}$ recorded on them at time stamp t are highly correlated. Therefore, each sparse coefficient $\mathbf{a}_c$ has the similar non-zero support in the sub-coefficient $\mathbf{a}_{c,g}$ with the same group g. Given these definitions, the joint-group sparsity is defined as:

$$||\mathbf{A}||_{group,0} = \sum_{l=1}^{G} I(||\mathbf{A}_g||_F > 0) = 1,$$
$$\mathbf{A} = [\mathbf{a}_1, \mathbf{a}_2 \ \ldots \mathbf{a}_C], \quad \mathbf{A}_g = [\mathbf{a}_{1,g}, \ \mathbf{a}_{2,g} \ \ldots \mathbf{a}_{C,g}]$$
$$\mathbf{a}_c = [\mathbf{a}_{c,1}, \quad \mathbf{a}_{c,2} \ldots \mathbf{a}_{c,G}], \ ||\mathbf{a}_c||_0 \leq S, \ \forall c.$$

In our formulation, I is the indicator function and S denotes the sparsity. $||\mathbf{A}_g||_F$ denotes the Frobenius norm. $||\mathbf{A}||_{group,0}$ is constrained to one to enforce that only one $\mathbf{A}_g$ contains non-zero coefficients while other $\mathbf{A}_{g'}$ contains zero coefficients.. Therefore, the mathematical definition of the proposed signal model is:

$$\mathbf{x}_c = [\mathbf{D}_{c,1}\mathbf{D}_{c,2} \ldots \mathbf{D}_{c,G}][\mathbf{a}_{c,1}^\top \ \mathbf{a}_{c,2}^\top \ \ldots \ \mathbf{a}_{c,G}^\top]^\top,$$
$$||\mathbf{A}||_{group,0} = 1, \quad ||\mathbf{a}_c||_0 \leq S, \ \forall c.$$

Intuitively, a spike should be represented by atoms from the corresponding group, and also be constrained by the information given by neighboring electrodes. Taking neighboring spikes into account, the compression ratio can be further improved, which also promotes the performance of the neural recording systems in terms of power efficiency.

In the following sections, we outline the details of dictionary learning using joint-group sparsity.

B. Dictionary Initialization

To begin learning the dictionary, we must first initialize the dictionary to enable fast convergence to an optimal solution. Previously, we have used the k-means to initialize the dictionary and successfully improve the spike sorting accuracy in unsupervised CS-based neural recording systems [28]. A non-random dictionary with initialized group structures would help speed up the dictionary learning. Given the preliminary group structure in the dictionary, the learning algorithm could converge to the optimal solution faster compared to the random initialization. To initialized the dictionary, we used the spectral clustering [30] method. The motivation of spectral clustering is to find a satisfactory clustering representation among the spikes and enable the initialization of group structures in the dictionary.

The initialization is divided into two stages: (i) similarity matrix initilization, and (ii) spectral clustering. As shown in Algorithm 1, the similarity matrix $\mathbf{E}$ represents the assessment of similarity (Euclidean distance) between spikes. $\mathbf{E}$ is generated based on the nearest-neighbour method and then the similarity of two spikes in the multi-channel is defined as:

$$e(t, t') = \sum_{c=1}^{C} ||\mathbf{x}_{c,t} - \mathbf{x}_{c,t'}||_2,$$
$$t, \ t' \in \{1, 2, \ldots, T\}, \ t \neq t'.$$

Intuitively, we build the similarity matrix $\mathbf{E}$ like a graph, where the spikes are vertexes. The smaller Euclidean distance $e(t, t')$ indicates the high correlation between two spikes. If the Euclidean distance is smaller than a pre-defined $error$, then we build the edge between the two spikes in $\mathbf{E}$. The details are shown in Algorithm 1.

Algorithm 1 Similarity Matrix Initialization

Require: Training data $\mathbf{X}_c = [\mathbf{x}_{c,1} \ \mathbf{x}_{c,2} \ \ldots \ \mathbf{x}_{c,T}]$, where $c = 1, 2, \ldots, C$ ($C = 1$ indicates the single channel). $K = 10$ is defined for k nearest neighbour (k-NN) classification. Pre-defined $error$.

1: Determine the set $\mathcal{V}_t$ for t-th spike $\mathbf{x}_{c,t}$ using k-NN and threshold $error$. Among the most K similar spikes, we add the index t' into the set $\mathcal{V}_t$ if $e(t, t') \leq error$.
2: Initialize similarity matrix $\mathbf{E} \in \mathbb{R}^{T \times T}$, where $\mathbf{E}(t, t') = 0$, $\forall t, t'$.
3: Set $\mathbf{E}(t, v) = 1, \forall t, v \in \mathcal{V}_t$.
4: Symmetrize the similarity matrix $\mathbf{E} = \mathbf{E} + \mathbf{E}^\top$.
5: Set $\mathbf{E}(t, t) = 1, \forall t$.
6: **Return** similarity matrix $\mathbf{E}$.

Given $\mathbf{E}$ from Algorithm 1, we pre-define the group number G and then adopt spectral clustering to group neural signals into G different clusters. Generally, as shown in Algorithm 2 the spectral clustering transforms the original clustering to another domain that forms tight clusters. From the graph cut point of view, the intuition is to find a partition of the graph such that the number of edges between clusters is minimal. The details of spectral clustering can be found in [30].

Given the clustering information $\mathbf{g}$ from Algorithm 2, the dictionary $\mathbf{D}_c$ of c-th channel is built as:

$$\mathbf{D}_c = [\mathbf{D}_{c,1}\mathbf{D}_{c,2} \ldots \mathbf{D}_{c,G}].$$

$\mathbf{D}_{c,g}$ indicates the sub-dictionary of $\mathbf{D}_c$, in which its atoms are randomly picked up from the group of cluster g. We also obtain the mean shape, defined as centroids $\mathbf{c}_{c,g}$ associated with a distinct cluster, which is used for template matching in the sparse coding stage. Centroids $\mathbf{c}_{c,g}$, representing the template and a particular pattern of groups, g are found by:

$$\mathbf{c}_{c,g} = \frac{1}{|\mathcal{S}_g|} \sum_{t \in \mathcal{S}_g} \mathbf{x}_{c,g}, \mathcal{S}_g = \{t | g_t = g\}.$$

C. Dictionary Learning

After initializing the dictionary $\mathbf{D}_c$ ($c = 1$ indicates single channel case, $c > 1$ indicates multi-channel case), as shown in Algorithm 3, the unsupervised multi-mode structured dictionary learning is divided into two stages in each iteration: the sparse coding stage and the dictionary update stage.

Algorithm 2 Spectral Clustering [30] Based on the Similarity Matrix

Require: Similarity matrix $\mathbf{E} \in \mathbb{R}^{T\times T}$ from Algorithm 1 and the number of clusters G.

1: Construct diagonal matrix $\mathbf{W}$, where $\mathbf{W}(t,t)$ is defined as the sum of t-th row of similarity matrix $\mathbf{E}$.
2: Construct the matrix $\mathbf{H} = \mathbf{W}^{-\frac{1}{2}}\mathbf{E}\mathbf{W}^{-\frac{1}{2}}$.
3: Calculate $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_G$, the G eigenvectors of $\mathbf{H}$ with the largest G eigenvalues.
4: Construct the matrix $\mathbf{V} = [\mathbf{v}_1\ \mathbf{v}_2\ \ldots\ \mathbf{v}_G] \in \mathbb{R}^{T\times G}$ and normalize each row of the matrix $\mathbf{V}$.
5: Apply k-means algorithm [31] to the rows of the matrix $\mathbf{V}$ and assign the cluster g_t to the original signal $x_{c,t}$.
6: **Return** Clusters vector $\mathbf{g}$

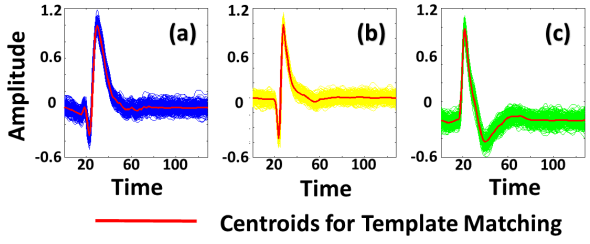

Fig. 4. Illustration of different groups of spikes with distinct shapes. The red color-coded spikes indicate the centroids (mean shape) associated with the corresponding groups. The mean shape matching provides another perspective of similarity in the sparse coding stage.

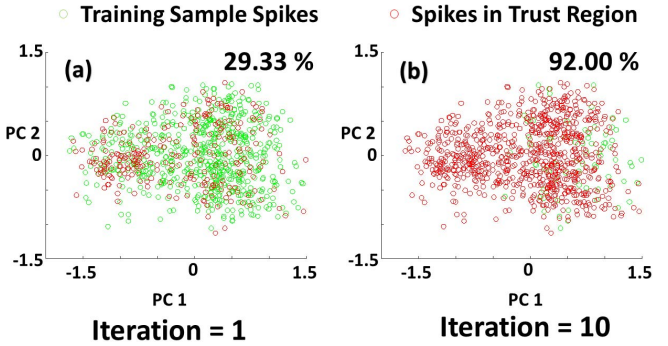

Fig. 5. An illustration of how the trust region $\mathcal{S}$ performs in principal component analysis (PCA). As Algorithm 3 iterates from 1 to 10, the percentage of spikes in the trust region increases from 29.33 % to 92.00 %, indicating that most spikes in the training samples satisfy the pre-defined reconstruction quality after 10 iterations.

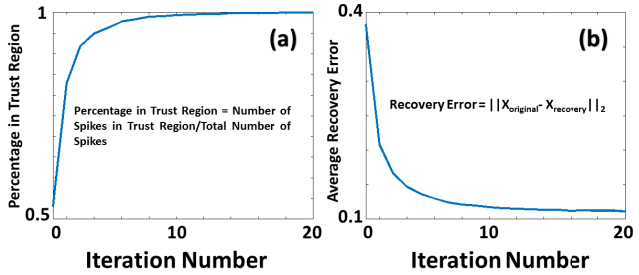

Fig. 6. Illustration of the trust region and training error during the iterative training. (a) indicates the percentage change in trust region $\mathcal{S}$ and (b) indicates average recovery error as dictionary learning iterates from 1 to 20.

In the sparse coding stage, we introduce joint-group sparsity and then solve the sparse representation problem below, using Orthogonal Matching Pursuit (OMP) [32],

$$\min_{\mathbf{a}_{c,g}} \sum_{c=1}^{C} ||\mathbf{x}_{c,t} - \mathbf{D}_{c,g}\mathbf{a}_{c,g}||_2$$
$$s.t.\ ||\mathbf{A}||_{group,0} = 1, \quad ||\mathbf{a}_{c,g}||_0 \leq S.$$

Here, we find out the best sparse representation $\mathbf{a}_{c,g}$ of each spike $\mathbf{x}_{c,t}$ in the training samples based on each sub-dictionary $\mathbf{D}_{c,g}$. Then, we use the linear combination coefficient $\lambda \in (0,1)$ to balance the residual of the sparse representation and the squared Euclidean distance between the spike and its centroids. Thereby, the cluster g of the spike is determined by solving the optimization problem below:

$$\min_{g} \sum_{c=1}^{C} \{\lambda||\mathbf{x}_{c,t} - \mathbf{D}_{c,g}\mathbf{a}_{c,g}||_2 + (1-\lambda)||\mathbf{D}_{c,g}\mathbf{a}_{c,g} - \mathbf{c}_{c,g}||_2\}.$$

As shown in Figure 4, the squared Euclidean distance for mean shape matching provides another evaluation of spikes similarity in the sparse representation stage. Previous work has demonstrated that using the centroid significantly improves the accuracy of spike sorting in the CS-based neural recordings [28]. Given group g of each spike, we define a trust region set $\mathcal{S}_g$ associated with group g. To construct the trust region set $\mathcal{S}_g$, we add the index of spike t into it if the spike is represented perfectly in the sparse coding stage, which indicates the reconstruction error is smaller than the pre-defined *error*. Intuitively, the trust region set $\mathcal{S}_g$ only contains spikes with high reconstruction quality in each learning iteration.

In the dictionary update stage, we simply fix the sparse coefficients matrix $\mathbf{A}_c$ and update each atom of the dictionary using the same approach as in the K-SVD [33]. While the K-SVD updates the dictionary based on the whole training samples, our approach only updates it based on the current trust region set $\mathcal{S}$, which is the union of set $\mathcal{S}_g$. Iteratively, the trust region covers the entire training samples. Figure 5 and 6 illustrate that the trust region set $\mathcal{S}$ approaches the entire training samples after several learning iterations. Furthermore, we dynamically update the centroid $\mathbf{c}_{c,g}$ depending on the clustering result obtained from the sparse representation stage. As shown in Figure 6, the average recovery error converges as the trust region $\mathcal{S}$ covers the entire training samples.

Taking advantage of iterative refinement in the dictionary learning, Algorithm 3 is able to correct the spike sorting error generated by the dictionary initialization, as shown in Figure 7. Figure 7(a) shows that Algorithm 1 mistakenly clusters some spikes, which are denoted as blue dots and distributed in the cluster of green dots. But as shown in Figure 7(b), after the dictionary learning, the PCA result illustrates that the spike sorting performance is refined and intuitively demonstrates the robustness of the proposed approach.

D. Reconstruction and Spike Sorting Approach

In our CS-based neural recording systems, we adopt the on-chip random Bernoulli matrix [19] $\mathbf{S} \in \mathbb{R}^{M\times N}$ to compress the

Algorithm 3 Unsupervised Multi-Mode Structured Dictionary Learning

Require: Initialized dictionary $\mathbf{D}_c$, training data $\mathbf{X}_c = [\mathbf{x}_{c,1}\ \mathbf{x}_{c,2}\ \ldots\ \mathbf{x}_{c,T}]$, where $c = 1, 2, \ldots, C$ ($C = 1$ indicates single channel). Clusters vector $\mathbf{g}$, number of clusters G, sparsity S, reconstruction $error$, linear combination coefficient $\lambda \in (0, 1)$ and number of maximum iteration $maxIter$.

1: **while** $iter \leq maxIter$ **do**

2: Set $\mathcal{S}_l = \emptyset$, $\forall l$.

3: Solve the representation problem via Orthogonal Matching Pursuit [32],

$$\min_{\mathbf{a}_{c,g}} \sum_{c=1}^{C} ||\mathbf{x}_{c,t} - \mathbf{D}_{c,g}\mathbf{a}_{c,g}||_2 \ s.t.$$
$$||\mathbf{A}||_{group,0} = 1, ||\mathbf{a}_{c,g}||_0 \leq S, \forall g, n.$$

4: Determine the cluster g for the n-th signal by solving following problem,

$$\min_g \sum_{c=1}^{C} \{\lambda||\mathbf{x}_{c,t} - \mathbf{D}_{c,g}\mathbf{a}_{c,g}||_2 + (1-\lambda)||\mathbf{x}_{c,g} - \mathbf{c}_{c,g}||_2\}.$$

5: If $\sum_{c=1}^{C} ||\mathbf{x}_{c,n} - \mathbf{D}_{c,g}\mathbf{a}_{c,g}||_2 \leq error$, then add n into $\mathcal{S}_g$.

6: Codebook update: we use the same method of approximation K-SVD [34] for updating each atom based on spikes belonging to $\mathcal{S} = \bigcup_1^G \mathcal{S}_g$.

7: Centroids update:

$$\mathbf{c}_{c,g} = \frac{1}{|\mathcal{S}_g|} \sum_{t \in \mathcal{S}_g} \mathbf{x}_{c,g}$$
$$\forall g = 1, 2, \ldots, G,\ c = 1, 2, \ldots, C,\ t = 1, 2, \ldots, T.$$

8: Set $iter = iter + 1$.

9: **end while**

10: **Return** $\mathbf{D}_c = [\mathbf{D}_{c,1}\mathbf{D}_{c,2}\ldots\mathbf{D}_{c,G}]$ and updated the centroids $\mathbf{c}_{c,1}\mathbf{c}_{c,2}\ldots\mathbf{c}_{c,G}$

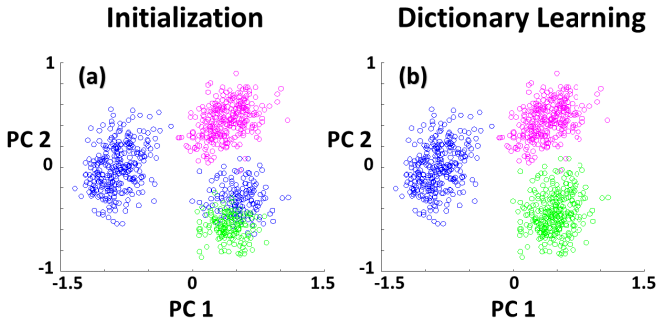

Fig. 7. An example of robustness of spike sorting from the perspective of sparse coding, visualized in the PCA domain. The iterative refinement helps correct the mistakenly sorted spikes generated from the initialization.

signal $\mathbf{x} \in \mathbb{R}^N$ into the measurement $\mathbf{y} \in \mathbb{R}^M$. Mathematically, $\mathbf{y} = \mathbf{S}\mathbf{x}$ and $M << N$. The Bernoulli matrix, of which the element is 0 or 1, is hardware friendly [19]. For each channel c, we adopt the same Bernoulli matrix $\mathbf{S}$ to sense the neural signal $\mathbf{x}_c$ into the measurements $\mathbf{y}_c$. Given the trained dictionary $\mathbf{D}_c$, the sensing matrix $\mathbf{S}$, the centroids $\mathbf{c}_{c,g}$ and the measurements $\mathbf{y}_c$, we reconstruct the signal $\hat{\mathbf{x}}_c$ and determine the cluster g as shown in Algorithm 4.

Algorithm 4 Reconstruction and Spike Sorting Approach

Require: The initialized dictionaries $\mathbf{D}_c$, the centroids $\mathbf{c}_{c,g}$, measurements $\mathbf{y}_c$, where $c = 1, 2, \ldots, C$ ($C = 1$ indicates single channel) and random Bernoulli matrix $\mathbf{S}$. Number of clusters G, sparsity S and linear combination coefficient $\lambda \in (0, 1)$.

1: Solve the representation problem via Orthogonal Matching Pursuit [32],

$$\min_{\mathbf{a}_{c,g}} \sum_{c=1}^{C} ||\mathbf{y}_c - \mathbf{S}\mathbf{D}_{c,g}\mathbf{a}_{c,g}||_2 \ s.t.\ ||\mathbf{a}_{c,g}||_0 \leq S, \forall g.$$

2: Determine the cluster g of spikes by solving following problem,

$$\min_g \sum_{c=1}^{C} \{\lambda||\mathbf{y}_c - \mathbf{S}\mathbf{D}_{c,g}\mathbf{a}_{c,g}||_2 + (1-\lambda)||\mathbf{y}_{c,g} - \mathbf{S}\mathbf{c}_{c,g}||_2\}.$$

3: **Return** The recovered signal $\hat{\mathbf{x}}_c = \mathbf{D}_{c,g}\mathbf{a}_{c,g}$ and cluster g.

E. Dictionary Update

In real-time neural recording experiments, it is impractical to observe the original signal $\mathbf{x}$ because the CS-based system only transmits the compressed information $\mathbf{y}$. Therefore, the reconstruction quality cannot be quantitatively evaluated by $\mathbf{x}$ and the recovered signal $\hat{\mathbf{x}}$. Normally, the trained dictionary is fixed during the recording. If the CS-based neural recording system encounters a new spike that is dramatically different from spikes from the training samples, it might be diffcult to reconstruct the spike sparsely using the dictionary, which significantly degrades the quality of systems. To address this challenge, the dictionary update has to be adaptive to change in spikes. Fortunately, the strong correlation between the reconstruction quality of the original signal $\mathbf{x}$ and the reconstruction quality of the measurement $\mathbf{y}$ has been found to quantify the recording performance [19]. It helps in adapting the trade-off between the reconstruction quality and compression ratio. Instead of evaluating the reconstruction quality of $\mathbf{x}$, which cannot be observed in a CS-based neural recording system, the signal-to-noise distortion ratio ($SNDR$) of $\mathbf{y}$ is adopted to efficiently quantify the online performance of the CS-based neural recording systems. When $SNDR_y$ drops below the pre-defined threshold, the CS-based system will automatically switch to the non-CS mode for collecting more samples at the full bandwidth for the dictionary update.

IV. Experiments

In this section, we compare the reconstruction and spike sorting performance of our proposed approach with the other CS-based approaches on both single-channel and multi-channel databases. In each training, the database was randomly divided into two halves: one for training, and the other for testing. The quality of reconstruction is measured in terms of the SNDR, which is defined as:

$$SNDR = 20 \times \log \frac{||\mathbf{x}||_2}{||\mathbf{x} - \hat{\mathbf{x}}||_2},$$

where $\mathbf{x}$ and $\hat{\mathbf{x}}$ indicate the original and recovered signals, respectively. The spike sorting performance is measured in terms of classification accuracy (CA), which is defined as:

$$CA = \frac{\#\ of\ Correctly\ Sorted\ Spikes}{Total\ Number\ of\ Spikes} \times 100\%.$$

All neural spikes are extracted from the raw data using a window of pre-defined length, and aligned properly before training and testing. In each experiment, the same Bernoulli matrix is adopted to compress the neural signal. We construct the K-SVD dictionary and the data dictionary with group structures based on the training samples. Then, we adopt OMP [32] and sparse representation classifier (SRC) [35] for recovery and spike sorting. For the proposed approach, we assume that the number of clusters G is pre-defined and the dictionary is learned by Algorithm 3 in an unsupervised manner. Additionally, we assign the same number of spikes to each group in the training and testing samples to eliminate the clustering bias.

Our experiments were implemented in MATLAB on a PC with Intel Core i7 and 16 GB RAM. The average computational time for the dictionary training was 20 seconds on average. The number of iterations was 10 and each training database consisted of 1000 samples of pre-defined length. It took only 6 ms to reconstruct and sort a spike.

A. Single Channel

We first compared the reconstruction performance between the proposed CS-based approach and the other dictionary learning approaches using the K-SVD, data dictionary and Wavelet dictionary on the synthetic Leicester database [27] and the Massachusetts General Hospital (MGH) database [26]. The Leicester database consists of neural signals of length 128, and the MGH database consists of neural signals of length 32, recorded from primates (monkeys), "Pogo" and "Romeo". The MGH database was collected at the MGH at a sampling rate of 40 kHz.

Furthermore, we compared the spike sorting accuracy of our CS-based approach to the other CS-based approaches using the signal-dependent dictionaries. The Leicester database consists of three classes of neural spikes grouped into two categories: "Easy" and "Difficult", which indicates the difficulty level of discriminating spikes. Generally, "Difficult" indicates a lot of noise in spikes. The MGH database contains two or three classes of spikes that have been manually sorted at the MGH.

Tables I and II, and Figure 8 demonstrate the reconstruction and spike sorting performance on the Leicester database at compression ratios of 20:1 and 10:1. The proposed approach outperforms the other CS-based approaches, and achieves an average gain of 2 dB and 4% in terms of SNDR and classification accuracy on the "Easy" database at the CR of 20:1. For the "Difficult" database, the approach attains more than 90% spike sorting success rate, while achieving a CR of 10:1 to 20:1. Tables III and IV show the reconstruction and spike sorting performance of the MGH "Pogo" and "Romeo" databases, respectively. Here too, the proposed approach outperforms the other CS-based approaches. Especially, the proposed approach shows more than 90% spike sorting success rate at the CR of 10:1, and achieves an average gain of 30% over other methods. Figure 9 intuitively illustrates the spike sorting result at the CR of 20:1 and 10:1 in the PCA domain. The pink, green and blue dots indicate distinct groups of spikes in the testing samples, while red dots indicate spikes that are

TABLE I
COMPARISON OF RECONSTRUCTION PERFORMANCE (IN SNDR) OF DIFFERENT CS METHODS ON "LEICESTER"

Database	CS Approach	CR = 20:1	CR = 10:1
EASY	Proposed Approach	**10.44**	**11.60**
	K-SVD + OMP	9.23	10.40
	Data Dictionary + OMP	7.11	7.76
	Wavelet + OMP	-1.81	-0.82
DIFFICULT	Proposed Approach	**8.64**	**10.21**
	K-SVD + OMP	6.40	8.03
	Data Dictionary + OMP	6.38	6.64
	Wavelet + OMP	-2.71	-1.78

TABLE II
COMPARISON OF CLASSIFICATION PERFORMANCE (IN CA) OF DIFFERENT CS METHODS ON "LEICESTER"

Database	CS Approach	CR = 20:1	CR = 10:1
EASY	Proposed Approach	**97.62**	**98.08**
	K-SVD + OMP	94.04	98.05
	Data Dictionary + OMP	93.88	95.77
DIFFICULT	Proposed Approach	**90.24**	**95.87**
	K-SVD + OMP	74.03	86.84
	Data Dictionary + OMP	73.24	78.22

TABLE III
COMPARISON OF RECONSTRUCTION AND CLASSIFICATION PERFORMANCE OF DIFFERENT CS METHODS ON "POGO"

Database	CS Approach	CR = 10:1	CR = 5:1
SNDR (dB)	Proposed Approach	**7.46**	**10.30**
	K-SVD + OMP	2.73	7.34
	Data Dictionary + OMP	6.96	8.18
	Wavelet + OMP	-1.51	-1.17
CA (%)	Proposed Approach	**93.63**	**95.11**
	K-SVD + OMP	51.07	64.07
	Data Dictionary + OMP	64.13	82.59

TABLE IV
COMPARISON OF RECONSTRUCTION AND CLASSIFICATION PERFORMANCE OF DIFFERENT CS METHODS ON "ROMEO"

Database	CS Approach	CR = 10:1	CR = 5:1
SNDR (dB)	Proposed Approach	**8.20**	**11.00**
	K-SVD + OMP	3.14	7.27
	Data Dictionary + OMP	6.69	8.00
	Wavelet + OMP	-1.53	-1.27
CA (%)	Proposed Approach	**94.54**	**96.88**
	K-SVD + OMP	44.32	57.98
	Data Dictionary + OMP	62.65	88.66

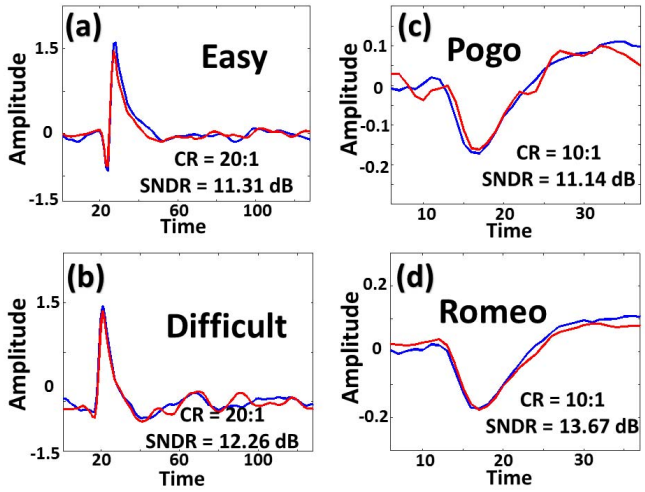

Fig. 8. Examples of reconstruction performance of single-channel neural recordings. For (a)-(d), recovered signals (red) still preserve the major features of original signals (blue) at CR of 20:1 and 10:1, respectively. (a) and (b) demonstrate synthetic spikes from the Leicester database [27], while (c) and (d) demonstrate real spikes from the MGH database [26].

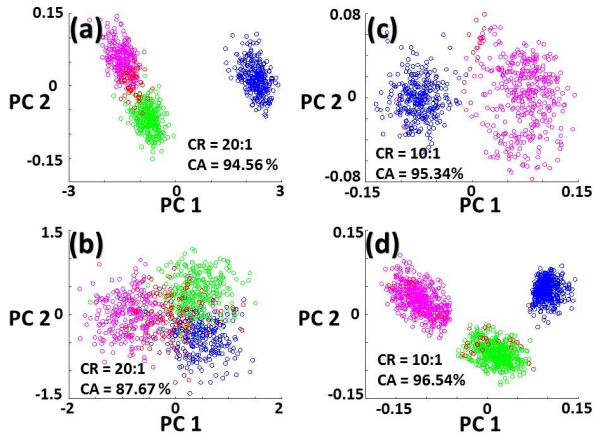

Fig. 9. Examples of spike sorting performance shown in the PCA domain. (a) and (b) illustrate the spike sorting result of Leicester's "Easy" and "Difficult" databases at a CR of 20:1. (c) and (d) illustrate the spike sorting result of MGH's "Pogo" and "Romeo" databases at a CR of 10:1.

incorrectly sorted. As shown in Figure 9, most of the spikes are correctly sorted and the spike sorting success rate can still achieve more than 90% accuracy, even after the CR increases to 20:1, which means we only use 5% information to reconstruct and sort the spike. The performance on the MGH database achieves more gains in terms of the recovery quality and spike sorting success rate compared to the performance on the Leicester database, which indicates the proposed approach is more robust to highly noisy signals.

B. Multi-Channel

In multi-channel experiments, we also compared the reconstruction quality and spike sorting success rate between the proposed approach and other methods. We evaluated the performance comprehensivley on the hc-1 (12 databases), whose neural signals were recorded by the tetrodes setup [27].

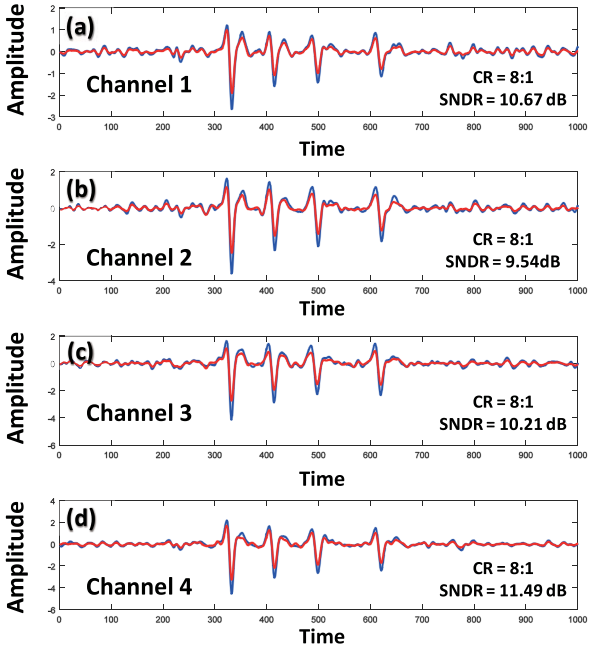

Fig. 10. An example of reconstruction performance of multi-channel neural recordings on the hc-1 database [27] at a CR of 8:1. Blue and red spikes indicate the original neural spikes and the recovered neural spikes, respectively. Spikes (not well aligned) and inter-spike intervals are reconstructed by Algorithm 4 at the window size of 128.

The hc-1 database was recorded from the hippocampus of mice in *in vivo* experiments. The tetrodes setup consists of four electrodes and one reference that indicates the firing of neurons. Based on the reference, we extracted neural spikes of length 64 from the raw data. The reconstruction quality is measured in terms of the SNDR. However, the database has no prior labels, which means there was no benchmark for us to evaluate the spike sorting success rate quantitatively. Thereby, in this session, we intuitively demonstrate the spike sorting performance using the PCA. Taking advantage of the PCA, we map the spike sorting result into the PCA domain, where different colors intuitively indicate different clusters.

Table V indicates that the proposed approach achieves an average gain of 4 to 5 dB over the other CS-based approaches in terms of the SNDR in multi-channel reconstruction. Figure 10 illustrates the multi-channel reconstruction example on the hc-1 database at a CR of 8:1. The blue signals denote the original spikes recorded from the tetrodes setup, which show similar pattern and correlation among the four channels as shown in Figure 10. The red signals denote the spikes recovered by the proposed CS-based approach. As shown in Figure 10, the recovered signals still preserve most of the features, even though only 12.5% of the information of the original signals is used for the reconstruction. The proposed approach is also able to sense and reconstruct neural signals in the continuous time domain, including the low activity region between spikes.

Figure 11 shows the multi-channel spike sorting performance at a CR of 16:1 in the PCA domain. Although only 5% of the information is collected for the spike sorting, the clustering results (color coded dots) are consistent with

TABLE V
COMPARISON OF RECONSTRUCTION PERFORMANCE (IN SNDR) OF DIFFERENT CS METHODS ON "HC-1"

CS Approach	Database	CR = 4:1	CR = 8:1	Database	CR = 4:1	CR = 8:1	Database	CR = 4:1	CR = 8:1
Proposed Approach	hc-1-11221	**14.85**	**6.69**	hc-1-11222	**16.40**	**9.93**	hc-1-12821	**13.20**	**5.41**
K-SVD+OMP		11.63	5.32		10.37	5.50		7.72	3.39
Proposed Approach	hc-1-13521	**11.23**	**5.00**	hc-1-14521	**13.93**	**8.85**	hc-1-14531	**12.18**	**9.55**
K-SVD+OMP		5.81	2.11		6.03	3.18		7.17	3.43
Proposed Approach	hc-1-14921	**12.64**	**8.54**	hc-1-15711	**10.19**	**6.37**	hc-1-15712	**8.45**	**5.06**
K-SVD+OMP		7.73	4.12		8.32	4.82		7.23	4.73
Proposed Approach	hc-1-16311	**11.32**	**7.16**	hc-1-17013	**13.52**	**7.80**	hc-1-17014	**20.49**	**13.57**
K-SVD+OMP		10.46	6.74		10.78	5.55		13.58	7.83

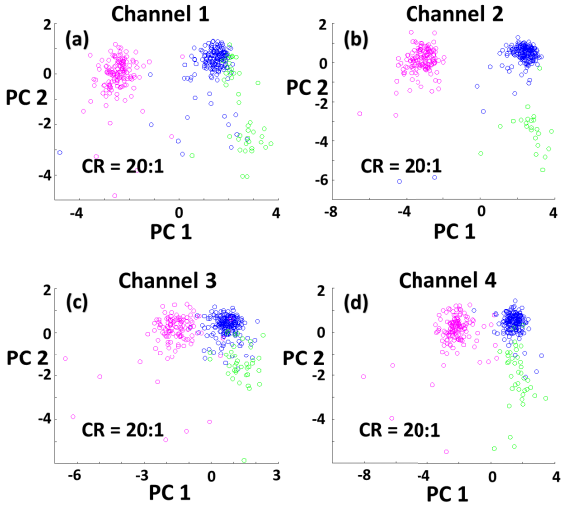

Fig. 11. An example of spike sorting performance of multi-channel neural recordings based on the hc-1 database [27]. Figure 11 (a)-(d) indicate the clustering results of channels 1-4, respectively. Different colors represent different clusters.

the distinct feature of the original spikes in the PCA domain. As shown in Figure 11, the distribution of the principal components among different channels illustrates that neural spikes share similar pattern, which implies correlation in the multi-channel neural recordings.

C. Energy Efficient CMOS Implementation

Typically, a 1000-electrode silicon probe generates data on the order of 300 Mbps in which each channel is sampled at 30 KHz while providing at least 10-bit resolution per sample [10]. For a wired system suffered from the high data rate, the overall power consumption is reduced from 50 mW to 5 mW with the 10× CR. Furthermore, from the perspective of wireless data transmission, at the CR of 10:1 the typical data rate 100 Mbps is further decreased to 10 Mbps. Given the same power budget, the transmission distance can be extended from 2m to 5m. Additionally, we can also achieve the same distance of 2m using 10 dB less transmitter power.

Based on the proposed CS framework, we have implemented the multi-channel CS neural recording systen in a 180 nm CMOS process [19], [36]. This system improves the power efficiency on the order of a few hundred nW per electrode. Given the CR of 10:1, the power consumption of our implementation is further reduced to 0.83 μW per electrode.

V. CONCLUSION

In this paper, we presented an unsupervised multi-mode CS approach for neural recording systems. We incorporate the joint-group sparsity in the dictionary learning to extend previous works to multi-channel neural recordings. Additionally, we take advantage of spectral clustering, group structure and template matching to enable spike sorting in real-time experiments in an unsupervised manner.

The approach was evaluated on both synthetic and real databases. The experimental results demonstrated that our approach significantly improved both the reconstruction quality (>8 dB) of neural signals and the spike sorting success rates (>90 %) at a high compression ratio (8:1 to 20:1). Our proposed framework, which is hardware friendly, can be integrated in CS-based implantable microsystems for *in vivo* neural recordings. From the perspective of hardware design, the proposed approach further enables energy-efficient CMOS implementations in terms of power consumption. In addition, it also enables online spike sorting in real-time neural recordings, which provides more feasibility for neuroscientists compared to conventional offline spike sorting techniques.

In order to realize a large-scale integration of neural recording systems, we plan to study the quantitative details of the correlation between spikes. By incorporating more structures in the CS framework, we will be able to further improve the performance in terms of reconstruction quality and spike sorting accuracy. Additionally, a sophisticated online dictionary update approach will also be introduced in the CS framework to enable a more adaptive real-time neural recording system in the future.

REFERENCES

[1] D. H. Hubel and T. N. Wiesel, "Receptive fields of single neurones in the cat's striate cortex," *J. Physiol.*, vol. 148, no. 3, pp. 574–591, 1959.
[2] C. M. Gray, P. E. Maldonado, M. Wilson, and B. McNaughton, "Tetrodes markedly improve the reliability and yield of multiple single-unit isolation from multi-unit recordings in cat striate cortex," *J. Neurosci. Methods*, vol. 63, nos. 1–2, pp. 43–54, 1995.
[3] E. M. Maynard, C. T. Nordhausen, and R. A. Normann, "The utah intracortical electrode array: A recording structure for potential brain-computer interfaces," *Electroencephalogr. Clin. Neurophysiol.*, vol. 102, no. 3, pp. 228–239, Mar. 1997.
[4] F. Shahrokhi, K. Abdelhalim, D. Serletis, P. L. Carlen, and R. Genov, "The 128-channel fully differential digital integrated neural recording and stimulation interface," *IEEE Trans. Biomed. Circuits Syst.*, vol. 4, no. 3, pp. 149–161, Jun. 2010.
[5] B. Gosselin, "Recent advances in neural recording microsystems," *Sensors*, vol. 11, no. 5, pp. 4572–4597, Apr. 2011.

[6] C. M. Lopez *et al.*, "An implantable 455-active-electrode 52-channel CMOS neural probe," *IEEE J. Solid-State Circuits*, vol. 49, no. 1, pp. 248–261, Jan. 2014.

[7] B. C. Raducanu *et al.*, "Time multiplexed active neural probe with 678 parallel recording sites," in *Proc. 46th Eur. Solid-State Device Res. Conf. (ESSDERC)*, Sep. 2016, pp. 385–388.

[8] A. Khalifa, J. Zhang, M. Leistner, and R. Etienne-Cummings, "A compact, low-power, fully analog implantable microstimulator," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2016, pp. 2435–2438.

[9] A. Kiourti, C. W. Lee, J. Chae, and J. L. Volakis, "A wireless fully passive neural recording device for unobtrusive neuropotential monitoring," *IEEE Trans. Biomed. Eng.*, vol. 63, no. 1, pp. 131–137, Jan. 2016.

[10] J. Zhang *et al.*, "Communication channel analysis and real time compressed sensing for high density neural recording devices," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 63, no. 5, pp. 599–608, May 2016.

[11] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.

[12] E. J. Candès, J. Romberg, and T. Tao, "Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information," *IEEE Trans. Inf. Theory*, vol. 52, no. 2, pp. 489–509, Feb. 2006.

[13] H. Mamaghanian, N. Khaled, D. Atienza, and P. Vandergheynst, "Compressed sensing for real-time energy-efficient ECG compression on wireless body sensor nodes," *IEEE Trans. Biomed. Eng.*, vol. 58, no. 9, pp. 2456–2466, Sep. 2011.

[14] A. M. R. Dixon, E. G. Allstot, D. Gangopadhyay, and D. J. Allstot, "Compressed sensing system considerations for ECG and EMG wireless biosensors," *IEEE Trans. Biomed. Circuits Syst.*, vol. 6, no. 2, pp. 156–166, Apr. 2012.

[15] 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 J. Solid-State Circuits*, vol. 47, no. 3, pp. 744–756, Mar. 2012.

[16] M. Mangia, R. Rovatti, and G. Setti, "Rakeness in the design of analog-to-information conversion of sparse and localized signals," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 59, no. 5, pp. 1001–1014, May 2012.

[17] F. Pareschi, P. Albertini, G. Frattini, M. Mangia, R. Rovatti, and G. Setti, "Hardware-algorithms co-design and implementation of an analog-to-information converter for biosignals based on compressed sensing," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 1, pp. 149–162, Feb. 2016.

[18] D. Gangopadhyay, E. G. Allstot, A. M. R. Dixon, K. Natarajan, S. Gupta, and D. J. Allstot, "Compressed sensing analog front-end for bio-sensor applications," *IEEE J. Solid-State Circuits*, vol. 49, no. 2, pp. 426–438, Feb. 2014.

[19] J. Zhang *et al.*, "A closed-loop compressive-sensing-based neural recording system," *J. Neural Eng.*, vol. 12, no. 3, p. 036005, 2015.

[20] N. Li, M. Osborn, G. Wang, and M. Sawan, "A digital multichannel neural signal processing system using compressed sensing," *Digit. Signal Process.*, vol. 55, pp. 64–77, Aug. 2016.

[21] X. Liu *et al.*, "A fully integrated wireless compressed sensing neural signal acquisition system for chronic recording and brain machine interface," *IEEE Trans. Biomed. Circuits Syst.*, vol. 10, no. 4, pp. 874–883, Aug. 2016.

[22] T. Xiong *et al.*, "A dictionary learning algorithm for multi-channel neural recordings," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2014, pp. 9–12.

[23] D. E. Carlson *et al.*, "Multichannel electrophysiological spike sorting via joint dictionary learning and mixture modeling," *IEEE Trans. Biomed. Eng.*, vol. 61, no. 1, pp. 41–54, Jan. 2014.

[24] J. Martinez-Trujillo and M. Sawan, "An adaptive recovery method in compressed sensing of extracellular neural recording," *Arch. De Med.*, vol. 6, no. 2, p. 19, 2015.

[25] Y. Suo, J. Zhang, T. Xiong, P. S. Chin, R. Etienne-Cummings, and T. D. Tran, "Energy-efficient multi-mode compressed sensing system for implantable neural recordings," *IEEE Trans. Biomed. Circuits Syst.*, vol. 8, no. 5, pp. 648–659, Oct. 2014.

[26] W. F. Asaad and E. N. Eskandar, "Encoding of both positive and negative reward prediction errors by neurons of the primate lateral prefrontal cortex and caudate nucleus," *J. Neurosci.*, vol. 31, no. 49, pp. 17772–17787, 2011.

[27] D. A. Henze, Z. Borhegyi, J. Csicsvari, A. Mamiya, K. D. Harris, and G. Buzsáki, "Intracellular features predicted by extracellular recordings in the hippocampus *in vivo*," *J. Neurophysiol.*, vol. 84, no. 1, pp. 390–400, 2000.

[28] T. Xiong *et al.*, "An unsupervised dictionary learning algorithm for neural recordings," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2015, pp. 1010–1013.

[29] H. Zamani, H. Bahrami, and P. Mohseni, "On the use of compressive sensing (CS) exploiting block sparsity for neural spike recording," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2016, pp. 228–231.

[30] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Proc. NIPS*, vol. 14. Dec. 2001, pp. 849–856.

[31] J. A. Hartigan and M. A. Wong, "Algorithm as 136: A k-means clustering algorithm," *Appl. Stat.*, vol. 28, no. 1, pp. 100–108, 1979.

[32] J. A. Tropp and A. C. Gilbert, "Signal recovery from random measurements via orthogonal matching pursuit," *IEEE Trans. Inf. Theory*, vol. 53, no. 12, pp. 4655–4666, Dec. 2007.

[33] M. Aharon, M. Elad, and A. Bruckstein, "K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation," *IEEE Trans. Signal Process.*, vol. 54, no. 11, pp. 4311–4322, Nov. 2006.

[34] R. Rubinstein, M. Zibulevsky, and M. Elad, "Efficient implementation of the K-SVD algorithm using batch orthogonal matching pursuit," *CS Technion*, vol. 40, no. 8, pp. 1–15, 2008.

[35] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[36] J. Zhang *et al.*, "Live demonstration: A closed loop compressive sensing neural recording system," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Oct. 2014, p. 170.