Diffusion Adaptation Framework for Compressive Sensing Reconstruction

Yicong He, Fei Wang, Member, IEEE, Shiyuan Wang, Member, IEEE, Badong Chen, Senior Member, IEEE

Abstract—Compressive sensing (CS) has drawn much attention in recent years due to its low sampling rate as well as high recovery accuracy. As an important procedure, reconstructing a sparse signal from few measurement data has been intensively studied. Many reconstruction algorithms have been proposed and shown good reconstruction performance. However, when dealing with large-scale sparse signal reconstruction problem, the storage requirement will be high, and many algorithms also suffer from high computational cost. In this paper, we propose a novel diffusion adaptation framework for CS reconstruction, where the reconstruction is performed in a distributed network. The data of measurement matrix are partitioned into small parts and are stored in each node, which assigns the storage load in a decentralized manner. The local information interaction provides the reconstruction ability. Then, a simple and efficient gradient-descent based diffusion algorithm has been proposed to collaboratively recover the sparse signal over network. The convergence of the proposed algorithm is analyzed. To further increase the convergence speed, a mini-batch based diffusion algorithm is also proposed. Simulation results show that the proposed algorithms can achieve good reconstruction accuracy as well as fast convergence speed.

I. INTRODUCTION

Compressive sensing (CS) is a novel sampling theory, which provides a new signal sampling (encoding) and reconstruction (decoding) approach [1]–[5]. In detail, given a compressible signal \( s = \Omega x \) where \( \Omega \) is the transform basis and \( x \) is a sparse signal, signal \( s \) can be measured by a nonadaptive linear projections (namely sensing matrix) \( \Phi \), i.e. \( y = \Phi s = \Theta x \) where \( y \) is the measurement vector and \( \Theta = \Phi \Omega \) is the measurement matrix. Then, in the decoder, \( x \) (or \( s \)) can be recovered from \( y \) using reconstruction algorithms. Since CS framework can provide far less sampling rate than Nyquist as well as high recovery accuracy, it has been widely used in many applications such as medical imaging [6] and radar imaging [7].

As an important procedure in CS, recovering a sparse signal from insufficient number of measurement data has drawn much attention in recent years. In the last decade, many algorithms have been proposed to show accurate reconstruction performance [8]–[13]. An important theoretical guarantee that behind CS reconstruction is the restricted isometry property (RIP) [14]. It has been proved that if \( \Theta \) obeys RIP, the sparse signal can be recovered from small number of measurements \( y \). It has also been shown that random matrices such as Gaussian matrix and Bernoulli matrix can satisfy the RIP condition with a high probability [15], [16].

Although CS Reconstruction problem has been intensively studied, applying CS reconstruction to large-scale data (such as image data) is still a challenging work. In [17], the authors proposed a conjugate gradient orthogonal matching pursuit (CG-OMP) algorithm. CG-OMP utilizes Structurally Random Matrix (SRM) [18] as the sensing matrix \( \Phi \), which can speed up the signal recovery process as well as reduce the storage requirement. In particular, SRM is related to operator-based approaches, and can improve all greedy algorithms and several iterative shrinkage/threshold (IST) methods such as gradient projection for sparse reconstruction algorithm (GPSR) [19] and sparse reconstruction by separable approximation (SpaRSA) [20]. However, although \( \Phi \) can be fast computed, the transform basis \( \Omega \) may not always be fast computable. Thus \( \Omega \) or \( \Theta \) may still need to be stored, which needs high requirement of storage for large scale data. To reduce the storage of \( \Theta \), a block based compressive sensing (BCS) method was proposed [21], [22]. In BCS, the input signal is separated into several small block signals, each signal is individually sensed and recovered. BCS essentially utilizes a block diagonal matrix as the sensing matrix, which, however, is lack of theoretical guarantee. Moreover, BCS needs to modify the sampling strategy at the sensing procedure, and the applicability is limited due to unclear structure.

In this paper we propose a novel diffusion adaptation framework for CS reconstruction. Unlike traditional CS reconstruction algorithms, the measurement matrix \( \Theta \) can be stored in a decentralized manner, i.e. each node of the network stores only a small part of \( \Theta \). Therefore, the whole storage load is distributed to each node in the network. Further, inspired by diffusion adaptation strategies [23]–[27], a simple yet efficient diffusion \( l_0 \)-LMS algorithm (D\( l_0 \)-LMS) is proposed. Each node utilizes the finite number of data recursively. The estimation are shared among neighbours at each iteration. Therefore, taking advantages of diffusion adaptation, the D\( l_0 \)-LMS algorithm can collaboratively recover the sparse signals across the network. Utilizing \( l_0 \)-norm as the regularization term can guarantee the sparse estimation. Information exchange within network also provide ability of fast convergence speed, thus greatly increases the computational efficiency. Moreover, a mini-batch based D\( l_0 \)-LMS (MB-D\( l_0 \)-LMS) is also proposed in this study. D\( l_0 \)-LMS utilizes the mini-batch gradient descend (MBGD) method and can further improve the convergence speed.

The proposed D\( l_0 \)-LMS is a variant of traditional sparse diffusion LMS algorithm [28], [29], which have shown abil-
algorithms. In particular, shown great performance improvement compared with other traditional sparse LMS algorithm to CS, where the network contains only 1 node. By introducing the upper bound of step size in diffusion adaptation. Algorithm. As far as we know, this is the first time to analyze algorithm, which is of significance for step size selection in CS reconstruction problem. In D
\[\text{LMS can be seen as a special case of D}_0\text{-LMS where the network contains only 1 node. By introducing traditional sparse LMS algorithm to CS, l}_0\text{-LMS algorithm has shown great performance improvement compared with other algorithms. In particular, l}_0\text{-LMS demands less requirement in memory, and achieves better reconstruction performance than other existing algorithms when dealing with large-scale CS reconstruction problem. In D}_0\text{-LMS, each node actually performs the same weight update process with l}_0\text{-LMS, thus the computational complexity of each node in each iteration are the same as l}_0\text{-LMS. Moreover, the diffusion algorithm gives ability to allow much larger step size for convergence condition, and is confirmed by experiments that the convergence speed is much faster than l}_0\text{-LMS. Further, simulations also show that D}_0\text{-LMS can achieve similar reconstruction accuracy with l}_0\text{-LMS.}

One should also distinguish our work from distributed compressive sensing (DCS) \[31\]–\[34\]. In DCS, a number of measurement data are recovered by a group of sensors. The measurement data at each node are assumed to be individually sparse in some basis and are correlated from sensor to sensor. The DCS aims to solve the jointly sparse ensemble reconstruction problem, which is not the topic of our work. Another related work is the Distributed Compressed Estimation (DCE) scheme \[35\]. The DCE incorporates compression and decompression modules into the distributed estimation procedure. The compressed estimator is estimated across the network using diffusion adaptation strategy, and then the reconstruction algorithms are employed to recover the sparse signal from compressed estimator. In DCE, each node still need to store the whole sensing matrix. Moreover, the reconstruction procedure is independent of diffusion adaptation procedure, which may still suffer from the same problem of typical reconstruction methods when dealing with large scale CS reconstruction problem.

The paper is organized as follows. In section II we briefly review the concept of compressive sensing and propose the diffusion adaptation framework for CS reconstruction. The gradient based and the mini batch based algorithms for diffusion CS reconstruction are then proposed in Section III. The stability analysis of D\_0-LMS is carried out in Section IV. In Section V, simulation results are presented to verify the reconstruction performance. Finally, the conclusion is given in Section VI.

II. DIFFUSION ADAPTATION FRAMEWORK FOR COMPRESSIVE SENSING RECONSTRUCTION

Suppose a real valued discrete signal \( s \in \mathbb{R}^{N \times 1} \) is compressible, i.e. \( s \) can be represented as \( s = \Omega x \) where \( \Omega \in \mathbb{R}^{N \times N} \) is a transform basis matrix and \( x \) is a sparse signal with sparsity \( K \ll N \). In the theory of CS, the sparse signal \( x \) can be measured by

\[
\hat{y} = \Phi \Omega s = \Theta x
\]

where \( \Theta = \Phi \Omega \) is the measurement matrix, \( \Phi \in \mathbb{R}^{M \times N} \) is the sensing matrix and \( \hat{y} \in \mathbb{R}^{M \times 1} \) is the measurement vector \( (M < N) \). In practice, the observed measurement vector \( y \) may be noisy, thus the observed measurement vector can be described as

\[
y = \Theta x + \nu
\]

where \( \nu \in \mathbb{R}^{M \times 1} \) is the additive noise vector. The CS reconstruction task is to recover the sparse signal \( x \) from the measurement matrix \( \Theta \) and the corresponding noisy measurement \( y \). To successfully recover \( x \), the measurement matrix \( \Theta \) should obey the restricted isometry property (RIP).

In practice, the CS reconstruction problem can be viewed as solving a sparse constrained least squares problem with the cost function

\[
J(x) = \|y - \Theta x\|_2^2 + \lambda \|x\|_0
\]

where \( \|x\|_0 \) is the \( l_0 \) regularized term and \( \lambda \) is regularization parameter.

To apply CS reconstruction in a decentralized manner, one can modify the above cost function. In particular, considering a connected network with \( P \) nodes (i.e. the network size is \( P \)), we can obtain the estimation of \( x \) by minimizing the following global cost function

\[
J_{\text{glob}}(x) = \sum_{k=1}^{P} \|y_k - \Theta_k x\|_2^2 + \lambda \|x\|_0
\]

where

\[
\Theta = \begin{bmatrix}
\Theta_1 \\
\vdots \\
\Theta_P
\end{bmatrix}, y = \begin{bmatrix}
y_1 \\
\vdots \\
y_P
\end{bmatrix}
\]

with \( \Theta_k \in \mathbb{R}^{L_k \times N} \), \( y_k \in \mathbb{R}^{L_k \times 1} \) and \( \sum_{k=1}^{P} L_k = M \). Since Eq.(3) and Eq.(4) are essentially the same cost function, the optimum solution will coincide.

Fig.1 shows an example of diffusion adaptation framework for CS reconstruction. The connected network includes 7 nodes. The whole \( \Theta \) is partitioned into small parts \( \{\Theta_k\}_{k=1}^{P} \).
Then, node $k$ only stores a small part of the measurement matrix $\Theta_k$ and receives the corresponding measurement data $y_k$. The information of a node can be transmitted within its neighbourhood (denoted as red links). Although each node has insufficient numbers of measurements and can only exchange information within local neighbours, the information diffusion across the whole network provides the ability to access the whole information of $\Theta$.

III. PROPOSED ALGORITHMS FOR COMPRESSIVE SENSING

A. Gradient descent $D_{l_0}$-LMS for CS

The diffusion adaptation algorithms for stream data has been intensively studied \cite{25, 29, 36, 37}. Given the temporal sparse input data sequence $\{d_k(i)\}$ and the corresponding output data sequence $\{y_k(i)\}$ at node $k$, the sparse diffusion LMS adaptation algorithm \cite{29} obtains the estimation by minimizing the following global cost function \cite{29}

$$J^{glob}(w) = \sum_{k=1}^{P} E \left( (d_k(i) - u_k^T(i)w)^2 \right) + \lambda \|w\|_0$$  \hspace{1cm} (5)

Intuitively, we can define $\{u_k(i)\}$ and $\{d_k(i)\}$ as

$$\{u_k(i)\} = \{\theta_k(1), \theta_k(2), ..., \theta_k(L_k)\}$$

$$\{d_k(i)\} = \{y_k(1), y_k(2), ..., y_k(L_k)\}$$

where $\theta_k(i)$ and $y_k(i)$ denote the transpose of the $i$th row of $\Theta_k$ and the $i$th scalar of $y_k$, respectively. Thus the solution to CS reconstruction problem in Eq.(4) can be formulated based on the the diffusion adaptation algorithm proposed for Eq.(5).

In traditional diffusion algorithm, the data size $L_k$ is always much larger than input dimension $N$. However, in CS $L_k$ is much smaller than $N$. When directly apply the sparse diffusion algorithm to CS, the adaptation process may not converge to the steady state due to insufficient number of data. To solve this problem, we follow the method described in \cite{30} and use the data recursively. In particular, the data used at $i$-th iteration in node $k$ are

$$u_k(i) = \theta_k(\text{mod}(i, L_k) + 1)$$

$$d_k(i) = y_k(\text{mod}(i, L_k) + 1)$$  \hspace{1cm} (6)

Therefore, combining diffusion adaptation strategy and modified data sequence in Eq.(6), we can derive two gradient-descent based diffusion adaptive algorithms for CS, namely, the Adapt-then-Combine(ATC) diffusion $l_0$-LMS (ATC-D$l_0$-LMS) algorithm

$$\psi_k(i + 1) = w_k(i) + \mu_k \sum_{l \in N_k} \alpha_{l,k} \hat{g}_l(w_k(i))$$

$$\hat{g}_l(y_k(i)) = \frac{\partial}{\partial y_k(i)} \|w_k(i)\|_0$$  \hspace{1cm} (7)

and the Combine-then-Adapt(CTA) diffusion $l_0$-LMS (CTA-D$l_0$-LMS) algorithm

$$\varphi_k(i) = \sum_{l \in N_k} \beta_{l,k} w_l(i)$$

$$w_k(i + 1) = \varphi_k(i) + \mu_k \sum_{l \in N_k} \alpha_{l,k} \hat{g}_l(\varphi_k(i))$$

$$\hat{g}_l(y_k(i)) = \frac{\partial}{\partial y_k(i)} \|w_k(i)\|_0$$  \hspace{1cm} (8)

where $\varphi_k(i)$ and $\psi_k(i)$ are intermediate vectors of node $k$ at time $i$, $\nabla [\|w_k(i)\|_0]$ is the derivation of $\|w_k(i)\|_0$ and

$$\hat{g}_l(y_k(i)) = (d_l(i) - \hat{x}_l^T(i)u_l(i)) u_l(i)$$  \hspace{1cm} (9)

is the instantaneous gradient vector. $N_k$ is the neighbourhood of node $k$ and $\mu_k$ is the corresponding step size. $\alpha_{l,k}$, $\beta_{l,k}$ are non-negative weights assigned to link between $l$ and $k$ for adaptation and combination step, respectively. Further, $\alpha_{l,k}$, $\beta_{l,k}$ can be seen as the $(l, k)$-th entries of matrices $S$ and $A$ respectively. Specifically, $S$ and $A$ should satisfy

$$S I = A^T 1 = 1$$  \hspace{1cm} (10)

$$S(l, k) = A(l, k) = 0 \text{ if } l \notin N_k$$  \hspace{1cm} (11)

where $1$ denotes the all one vector.

An important problem left is to calculate $\nabla [\|w(i)\|_0]$. Since $\nabla [\|w(i)\|_0]$ is non-differentiable, one should use an approximation instead. There are several approximations of $l_0$ norm \cite{38} which can work well for sparse identification \cite{25, 29}, \cite{38}. In this paper we use the zero attraction term $z_\delta(w_k(i))$ similar to $l_0$-LMS, which is defined as \cite{30}

$$-\nabla [\|w(i)\|_0] \approx z_\delta(w_k(i)) = [z_\delta(w_1(i)), z_\delta(w_2(i)), ..., z_\delta(w_N(i))].$$  \hspace{1cm} (12)

where

$$z_\delta(w_m(i)) = \begin{cases} \delta^2 w_m(i) + \delta^2 w_m(i) - 1/\delta \leq w_m(i) < 0 \\ \delta^2 w_m(i) - \delta^2 w_m(i) - 1/\delta \leq w_m(i) \leq 1/\delta \end{cases}$$  \hspace{1cm} (13)

and $\delta$ is the zero attraction parameter.

The pseudo code of ATC-D$l_0$-LMS is summarized in Algorithm 1. At iteration $i$, each node $k$ sends the data pair $\{u_k(i), d_k(i)\}$ to its neighbours. Then the adaptation is performed at each node. After adaptation, the estimation in each node is transferred to its neighbours for combination. The process of CTA-D$l_0$-LMS is similar with ATC-D$l_0$-LMS except that the order of adaptation step and combination step
Algorithm 1 ATC-D\(_0\)-LMS Algorithm

Initialization
Choose step-size \( \mu_k \) for each node \( k \), regularization parameter \( \lambda \) and zero attraction parameter \( \delta \) and maximum iteration number \( C \).

Set initial iteration number \( i = 0 \) and weight vector \( w_k(0) = 0 \) for all node \( k \).

Select \( \alpha_{l,k}, \beta_{l,k} \) according to Eq.\((10)\) and Eq.\((11)\).

Computation:
while \( i < C \) do
  for each node \( k \) do
    \( n = \text{mod}(i, L_k) + 1 \), \( u_k(i) = \theta_k(n), d_k(i) = y_k(n) \)
  \end for

  Communication 1:
  Transmit \( \{u_k(i), d_k(i)\} \) to neighbour node \( l \) in \( N_k \)

  Adaptation:
  \( \psi_k(i + 1) = w_k(i) + \mu_k \sum_{l \in N_k} \alpha_{l,k} \hat{y}_l(w_k(i)) + \mu_k \lambda z_\delta(w(i)) \)

  Communication 2:
  Transmit \( \psi_k(i + 1) \) to neighbour node \( l \) in \( N_k \)

  Combination:
  \( w_k(i + 1) = \sum_{l \in N_k} \beta_{l,k} \psi_l(i + 1) \)
end for

if the stop criterion is satisfied then
  break
end if

\%update iteration number
\( i = i + 1 \)
end while

Output: \( w = \frac{1}{T} \sum_{k=1}^{P} w_k(i) \)

are reversed.

Remark 1: Consider a special case that the network size \( P = 1 \). In this case, the reconstruction is processed in a stand-alone manner, and Eq.\((14)\) and Eq.\((15)\) will be equal to
\[
\psi(i + 1) = w(i) + \mu(d(i) - x^T(i)u(i))u(i) + \mu \lambda z_\delta(w(i))
\]
which is the typical \( l_0 \)-LMS for CS [50]. Therefore, \( l_0 \)-LMS can be viewed as a special case of ATC-D\(_0\)-LMS and CTA-D\(_0\)-LMS.

Remark 2: From Algorithm 1 one can see that there are two information exchange processes in each iteration. For large scale data, to reduce the amount of network transmission, one can put away the adaptation information exchange, i.e. \( S = I \). That is, each node only utilize its own data to perform adaptation, and then share its estimation to neighbours for combination. Simulation results in Section V will verify the feasibility of this strategy.

B. Mini-batch \( l_0 \)-LMS for CS

The proposed gradient-based \( l_0 \)-LMS is a typical extension of traditional sparse diffusion algorithm. We should notice that unlike traditional diffusion adaptation, in CS all the data is already known. Therefore, one can utilize more information during each iteration. In [30], the \( l_0 \)-EFWLMS is proposed to improve the convergence speed of \( l_0 \)-LMS. Extended from affine projection algorithm (APA) [59], \( l_0 \)-EFWLMS utilizes a sliding window approach to use more data to improve the gradient estimation. However, \( l_0 \)-EFWLMS still follows the traditional adaptive filtering method.

In data regression problem, mini-batch gradient descent (MBGD) method has been widely used. MBGD selects a small part of the sample data, computes gradient for each data, and then calculates the average gradient as the gradient estimation. For diffusion CS, the input for node \( k \) at each iteration \( i \) can be chosen as
\[
U_k(i) = [\theta(r(1)), \theta(r(2)), ..., \theta(r(Q))]^T
\]
\[
D_k(i) = [y(r(1)), y(r(2)), ..., y(r(Q))]^T
\]
where \( r = [r(1) r(2) ... r(Q)]^T \in N^Q \times 1, Q \leq \min\{L_k\} \) is the index vector whose elements are uniformly and randomly chosen from \([1, M]\). Then, according to MBGD method, the average gradient \( \hat{G}_l(x_k(i)) \) is defined as
\[
\hat{G}_l(x_k(i)) = \frac{1}{Q} U_l(i) (D_l(i) - x_k^T(i)U_l(i))
\]
Then, the weight update of corresponding ATC mini-batch diffusion \( l_0 \)-LMS algorithm (ATC-MB-D\(_0\)-LMS) can be represented as
\[
\psi_k(i + 1) = w_k(i) + \mu_k \sum_{l \in N_k} \alpha_{l,k} \hat{G}_l(w_k(i)) + \mu_k \lambda z_\delta(w(i))
\]
which is the typical \( l_0 \)-LMS for CS [50]. Therefore, \( l_0 \)-LMS can be viewed as a special case of ATC-D\(_0\)-LMS and CTA-D\(_0\)-LMS.

Remark 2: From Algorithm 1 one can see that there are two information exchange processes in each iteration. For large scale data, to reduce the amount of network transmission, one can put away the adaptation information exchange, i.e. \( S = I \). That is, each node only utilize its own data to perform adaptation, and then share its estimation to neighbours for combination. Simulation results in Section V will verify the feasibility of this strategy.

B. Mini-batch \( l_0 \)-LMS for CS

The proposed gradient-based \( l_0 \)-LMS is a typical extension of traditional sparse diffusion algorithm. We should notice that unlike traditional diffusion adaptation, in CS all the data is already known. Therefore, one can utilize more information during each iteration. In [30], the \( l_0 \)-EFWLMS is proposed to improve the convergence speed of \( l_0 \)-LMS. Extended from affine projection algorithm (APA) [59], \( l_0 \)-EFWLMS utilizes a sliding window approach to use more data to improve the gradient estimation. However, \( l_0 \)-EFWLMS still follows the traditional adaptive filtering method.

In data regression problem, mini-batch gradient descent (MBGD) method has been widely used. MBGD selects a small part of the sample data, computes gradient for each data, and then calculates the average gradient as the gradient estimation. For diffusion CS, the input data for node \( k \) at each iteration \( i \) can be chosen as
\[
U_k(i) = [\theta(r(1)), \theta(r(2)), ..., \theta(r(Q))]^T
\]
\[
D_k(i) = [y(r(1)), y(r(2)), ..., y(r(Q))]^T
\]
where \( r = [r(1) r(2) ... r(Q)]^T \in N^Q \times 1, Q \leq \min\{L_k\} \) is the index vector whose elements are uniformly and randomly chosen from \([1, M]\). Then, according to MBGD method, the average gradient \( \hat{G}_l(x_k(i)) \) is defined as
\[
\hat{G}_l(x_k(i)) = \frac{1}{Q} U_l(i) (D_l(i) - x_k^T(i)U_l(i))
\]
Then, the weight update of corresponding ATC mini-batch diffusion \( l_0 \)-LMS algorithm (ATC-MB-D\(_0\)-LMS) can be represented as
\[
\psi_k(i + 1) = w_k(i) + \mu_k \sum_{l \in N_k} \alpha_{l,k} \hat{G}_l(w_k(i)) + \mu_k \lambda z_\delta(w(i))
\]
which is the typical \( l_0 \)-LMS for CS [50]. Therefore, \( l_0 \)-LMS can be viewed as a special case of ATC-D\(_0\)-LMS and CTA-D\(_0\)-LMS.

Remark 2: From Algorithm 1 one can see that there are two information exchange processes in each iteration. For large scale data, to reduce the amount of network transmission, one can put away the adaptation information exchange, i.e. \( S = I \). That is, each node only utilize its own data to perform adaptation, and then share its estimation to neighbours for combination. Simulation results in Section V will verify the feasibility of this strategy.

B. Mini-batch \( l_0 \)-LMS for CS

The proposed gradient-based \( l_0 \)-LMS is a typical extension of traditional sparse diffusion algorithm. We should notice that unlike traditional diffusion adaptation, in CS all the data is already known. Therefore, one can utilize more information during each iteration. In [30], the \( l_0 \)-EFWLMS is proposed to improve the convergence speed of \( l_0 \)-LMS. Extended from affine projection algorithm (APA) [59], \( l_0 \)-EFWLMS utilizes a sliding window approach to use more data to improve the gradient estimation. However, \( l_0 \)-EFWLMS still follows the traditional adaptive filtering method.

In data regression problem, mini-batch gradient descent (MBGD) method has been widely used. MBGD selects a small part of the sample data, computes gradient for each data, and then calculates the average gradient as the gradient estimation. For diffusion CS, the input data for node \( k \) at each iteration \( i \) can be chosen as
\[
U_k(i) = [\theta(r(1)), \theta(r(2)), ..., \theta(r(Q))]^T
\]
\[
D_k(i) = [y(r(1)), y(r(2)), ..., y(r(Q))]^T
\]
where \( r = [r(1) r(2) ... r(Q)]^T \in N^Q \times 1, Q \leq \min\{L_k\} \) is the index vector whose elements are uniformly and randomly chosen from \([1, M]\). Then, according to MBGD method, the average gradient \( \hat{G}_l(x_k(i)) \) is defined as
\[
\hat{G}_l(x_k(i)) = \frac{1}{Q} U_l(i) (D_l(i) - x_k^T(i)U_l(i))
\]
Then, the weight update of corresponding ATC mini-batch diffusion \( l_0 \)-LMS algorithm (ATC-MB-D\(_0\)-LMS) can be represented as
\[
\psi_k(i + 1) = w_k(i) + \mu_k \sum_{l \in N_k} \alpha_{l,k} \hat{G}_l(w_k(i)) + \mu_k \lambda z_\delta(w(i))
\]
which is the typical \( l_0 \)-LMS for CS [50]. Therefore, \( l_0 \)-LMS can be viewed as a special case of ATC-D\(_0\)-LMS and CTA-D\(_0\)-LMS.

Remark 2: From Algorithm 1 one can see that there are two information exchange processes in each iteration. For large scale data, to reduce the amount of network transmission, one can put away the adaptation information exchange, i.e. \( S = I \). That is, each node only utilize its own data to perform adaptation, and then share its estimation to neighbours for combination. Simulation results in Section V will verify the feasibility of this strategy.
then sent back. Similar to ATC-DL₀-LMS, one can also set \( S = I \) to put away the adaptation step to save the amount of network transmission.

### Algorithm 2 ATC-MB-DL₀-LMS Algorithm

**Initialization**
Choose step-size \( \mu \), regularization parameter \( \lambda \) and zero attraction parameter \( \delta \), maximum iteration number \( C \), threshold \( \tau \) and mini-batch size \( Q \).

Set initial iteration number \( i = 0 \) and weight vector \( w_k(0) = 0 \) for all node \( k \).

Select \( \alpha_{l,k} \), \( \beta_{l,k} \) according to Eq.(10) and Eq.(11).

**Computation:**

while \( i < C \) do

for each node \( k \) do

Select index vector \( r \), then generate \( U_k(i) \) and \( D_k(i) \) from Eq. (15).

**Communication 1:**

Transmit \( w_k(i) \) to neighbour node \( l \) in \( \mathcal{N}_k \)

**Gradient estimation:**

Compute \( G_k(w_l(i)) \) from Eq. (16) for all \( l \in \mathcal{N}_k \)

**Communication 2:**

Transmit \( G_k(w_l(i)) \) to neighbour node \( l \)

**Adaptation:**

\[
\psi_k(i+1) = w_k(i) + \mu_k \sum_{l \in \mathcal{N}_k} \alpha_{l,k}G_l(w_k(i)) + \mu_k\lambda z_s(w(i))
\]

**Communication 3:**

Transmit \( \psi_k(i+1) \) to neighbour node \( l \) in \( \mathcal{N}_k \)

**Combination:**

\[
w_k(i+1) = \sum_{l \in \mathcal{N}_k} \beta_{l,k}\psi_l(i+1)
\]

if \( s(w_k(i+1)) - s(w_k(i)) > 1.5s(w_k(i)) \) and \( i > 0.02N \) then

\[
w_k(i+1) = w_k(i)
\]

end if

end for

if the stop criterion is satisfied then

break

end if

%update iteration number

\[ i = i + 1 \]

end while

**Output:** \( \hat{w} = \frac{1}{p} \sum_{k=1}^{p} w_k(i) \)

### C. Stop criterion

Although one can use the maximum iteration number \( C \) to stop the iteration, one would like a more practical stop criterion. In [30], the author utilizes the squared error between adjacent estimation as the index of stop condition. However, it is always not operational in real applications. Observing the fact that the sparsity of the estimation will maintain as the algorithm converges to the steady state, here we propose a new stop criterion based on the sparsity of the estimation: given the window length \( L_s \) and the threshold \( p_s \), by defining the count at iteration \( i \)

\[
S_c(i) = \sum_{j=i-L_s+1}^{i} p(s(w(j)))
\]

where

\[
p(x) = \begin{cases} 
1, & s_{\text{min}} \leq x \leq s_{\text{min}} + p_s \\
0, & x > s_{\text{min}} + p_s 
\end{cases}
\]

and \( s_{\text{min}} = \min\{s(w(j))|j=i-L_s+1\} \), if \( S_c(i) > 0.8L_s \), then the algorithm will stop.

### IV. Convergence Analysis

In this section, we carry out the convergence analysis of DL₀-LMS algorithm for CS reconstruction. We first derive the analysis of mean square stability, and then analyze the step size upper bound for convergence under general parameter settings. Finally the influence of regularization term and data reuse is also discussed.

For tractability, the analysis is based on following assumptions:

- **A1** The elements of noise vector \( v \) are i.i.d processes and independent of measurement matrix \( \Theta \).
- **A2** \( L_k \) of each node is sufficient large so that \( w_k(i) \) at arbitrary node \( k \) is independent of \( u_l(i), l \in \mathcal{N}_k \).
- **A3** The noise has finite variance.

Assumption A1 and A2 are extensions of independence assumption, which have been widely used in analysis of diffusion adaptation [23]–[27]. Specifically, assumption A2 may not be satisfied in practice.

This fact has been proved by simulation results of DL₀-LMS for CS [30]. In the analysis, without loss of generality, the variance of each element is set to \( 1/M \).

For simplicity, similar to [25], [29], we carry out the mean-square analysis on the following general diffusion framework structure

\[
\begin{align*}
\varphi_k(i) &= \sum_{l \in \mathcal{N}_k} \beta_{l,k}w_l(i) \\
\theta_k(i+1) &= \varphi_k(i) + \mu_k \sum_{l \in \mathcal{N}_k} \alpha_{l,k}g_l\left(\varphi_k(i)\right) + \mu_k\lambda z_s(w(i)) \\
w_k(i+1) &= \sum_{l \in \mathcal{N}_k} \gamma_{l,k}i(i+1) \end{align*}
\]

where \( \alpha_{l,k}, \beta_{l,k}, \gamma_{l,k} \) can be seen as the \( \{l,k\} \)-th entries of matrices \( A_1 \) and \( S \) and \( A_2 \) respectively. ATC-DL₀-LMS and CTA-DL₀-LMS can be viewed as two special cases by setting \( A_1 = I \) and \( A_2 = I \), respectively.

Subtracting both sides of Eq. (20) from the desired sparse vector \( x \), we can obtain

\[
\begin{align*}
\hat{\varphi}_k(i) &= \sum_{l \in \mathcal{N}_k} \hat{\beta}_{l,k}w_l(i) \\
\hat{\theta}_k(i+1) &= \hat{\varphi}_k(i) - \mu_k \sum_{l \in \mathcal{N}_k} \alpha_{l,k}g_l\left(\hat{\varphi}_k(i)\right) - \mu_k\lambda z_s(w(i)) \\
\hat{w}_k(i+1) &= \sum_{l \in \mathcal{N}_k} \gamma_{l,k}\hat{\theta}_l(i+1) \end{align*}
\]

(21)
where \( \varphi_l(i) = x - \varphi_l(i) \), \( \bar{w}_l(i) = x - \bar{w}_l(i) \), \( \hat{\theta}_l(i) = x - \theta_l(i) \) are error vectors. Therefore, by defining

\[
\bar{w}(i) = \text{col}\{\bar{w}_1(i)\}_{1}^{P}, \quad \bar{w}(i) = \text{col}\{\bar{w}_i(i)\}_{1}^{P}, \\
\mathcal{A}_1 = A_1 \otimes I_N, \mathcal{A}_2 = A_2 \otimes I_N, \mathcal{S} = S \otimes I_N \\
D = \text{diag}\{\mu_k\}_{1}^{P}, \mathcal{D} = D \otimes I_N \\
V(i) = \text{diag}\{v_k(i)\}_{1}^{P}, V(i) = V(i) \otimes I_N \\
\mathcal{H}(i) = \sum_{l=1}^{P} \text{diag}\{\alpha_l,k u_l(i) u_l(i)\}_{1}^{P}, s(i) = \mathcal{S}^T V^T(i) \text{col}\{u_k(i)\}_{1}^{P}
\]

one can obtain the recursion from Eq.(21)

\[
\bar{w}(i+1) = \mathcal{A}_2^T \left[ I - \mathcal{D} \mathcal{H} \right] \mathcal{A}_1^T \bar{w}(i) - \mathcal{A}_2^T \mathcal{D} s(i) - \lambda \mathcal{A}_2^T \mathcal{D} z_d(w(i)) 
\]

Postmultiplying both sides of Eq.(22) with their respective transposes, and then utilizing trace operation, we can obtain the following weighted mean square relation

\[
E\left[ \|\bar{w}(i+1)\|_\Sigma^2 \right] = E\left[ \|\bar{w}(i+1)\|_\Lambda^2 \right] + tr\{\Sigma U(i)\}
\]

where

\[
U(i) = \lambda^2 \mathcal{A}_2^T D K(i) \mathcal{D} A_2 + \mathcal{A}_2^T D S^T P(i) S \mathcal{D} A_2 \\
- \lambda \mathcal{A}_2^T \left( I - \frac{1}{M} \mathcal{G} \right) \mathcal{A}_1^T J(i) \mathcal{D} A_2 \\
- \lambda \mathcal{A}_2^T D J(i) \mathcal{A}_1 \left( I - \frac{1}{M} \mathcal{G} \right) A_2 \\
\Lambda = \mathcal{A}_1 \left( I - \frac{1}{M} \mathcal{G} \right) \mathcal{A}_2 \Sigma \mathcal{A}_2^T \left( I - \frac{1}{M} \mathcal{G} \right) \mathcal{A}_1^T \\
+ \frac{N+1}{M^2} \sum_{k=1}^{P} \mathcal{A}_1 \mathcal{T}_k \mathcal{D} A_2 \Sigma \mathcal{A}_2^T \mathcal{T}_k \mathcal{D} A_2
\]

with

\[
G = \sum_{l=1}^{P} \text{diag}\{\alpha_l,k\}_{1}^{P}, G = G \otimes I_N \\
T_k = \text{diag}\{\alpha_l,k\}_{1}^{P}, T_k = T_k \otimes I_N \\
J(i) = E[\bar{w}(i) z^T(w(i))] \\
K(i) = E[v(w(i)) z^T(w(i))] \\
P(i) = E[\mathcal{V}(i)^T \mathcal{V}(i)]
\]

Using the relationship of vectorization operator, matrix trace and Kronecker product

\[
\text{vec}\{ABC\} = (C^T \otimes A) \text{vec}\{B\} \\
tr\{A^T B\} = \text{vec}\{B\}^T \text{vec}\{A\}
\]

by defining \( \sigma = \text{vec}\{\Sigma\} \), Eq.(23) can be derived as

\[
E\left[ \|\bar{w}(i+1)\|_{\Sigma}^2 \right] = E\left[ \|\bar{w}(i+1)\|_{\Sigma}^2 \right] + \text{vec}\{U^T(i)\}^T \sigma
\]

where \( \|\bar{w}(i+1)\|_{\Sigma}^2 \) and \( \|\bar{w}(i+1)\|_{\Sigma}^2 \) denotes the same quantity, and

\[
\mathcal{F} = (\mathcal{A}_1 \otimes \mathcal{A}_1) \left( I - \frac{1}{M} \mathcal{G} \right) \otimes \left( I - \frac{1}{M} \mathcal{G} \right) \\
+ \frac{N+1}{M^2} \sum_{k=1}^{M} (\mathcal{T}_k \mathcal{D} \otimes \mathcal{T}_k \mathcal{D}) (\mathcal{A}_2 \otimes \mathcal{A}_2)
\]

It is easy to prove that \( K(i) \) and \( P(i) \) are always bounded. Moreover, for \( J(i) \) one can obtain

\[
|J(i)| = |E[\bar{w}(i) z^T(w(i))]| \leq \frac{1}{\delta} |E[\bar{w}(i) I^T]| 
\]

Therefore, to ensure the bounded property of \( J(i) \), \( E[\bar{w}(i)] \) should be always bounded. This recalls the mean recursion relation, which is obtained by taking expectation of both sides of Eq.(22)

\[
E[\bar{w}(i+1)] = QE[\bar{w}(i)] - \lambda \mathcal{A}_2^T \mathcal{D} E[z(w(i))]
\]

where

\[
Q = \mathcal{A}_2^T \left( I - \frac{1}{M} \delta \mathcal{D} \right) \mathcal{A}_1^T
\]

It is known that \( E[\bar{w}(i)] \) will be bounded for all \( i \) if \( E[\bar{w}(i)] \) converges as \( i \to \infty \). Moreover, it has been proven that the stability of \( \mathcal{F} \) and \( Q \) can ensure the convergence of Eq.(23) and Eq.(31), respectively [29]. Thus, for arbitrary \( \lambda \) the sufficient condition for both mean and mean square stability of Eq.(23) should be

\[
\rho(\mathcal{F}) < 1 \quad \text{and} \quad \rho(Q) < 1
\]

where \( \rho(X) \) denotes the spectral radius of matrix \( X \).

To further simplify the condition in Eq.(33), we propose the following theorem:

**Theorem 1:** For arbitrary real square matrices sequence \( \{B_k\}_{k=1}^{N+1} \in \mathbb{R}^{l \times l}, l \in \mathbb{N}_+ \), the following inequality will always hold

\[
\rho(B_1 \otimes B_1) \leq \rho\left( \sum_{k=1}^{t} (B_k \otimes B_k) \right)
\]

**Proof 1:** See Appendix A.

According to Theorem[1] if we set \( B_1 = Q^T \) and \( B_k = \frac{1}{M} \mathcal{T}_k \mathcal{D} A_2 \) for \( k = 2, ..., P+1 \), we can obtain

\[
\rho(G^T \otimes G^T) \leq \rho(\mathcal{F})
\]

Moreover, according to eigenvalue relationship of kronecker product, we can obtain \( \rho(Q^T \otimes Q^T) = |\rho(Q)|^2 \). Therefore, we have the following conditions

\[
\rho(\mathcal{F}) < 1 \implies \rho(Q^T \otimes Q^T) < 1 \implies \rho(Q) < 1
\]

Therefore, we can conclude that under Assumptions A1-A3, the condition \( \rho(\mathcal{F}) < 1 \) will guarantee both mean and mean square stability of the proposed algorithm.

For large scale data, computing the eigenvalue of \( \mathcal{F} \) is hard since the complexity grows significantly as \( N \) increases. To simplify the computation of \( \mathcal{F} \), we further propose the following theorem:

**Theorem 2:** Given sum of arbitrary real square matrices sequence \( \{B_k\}_{k=1}^{N+1} \in \mathbb{R}^{l \times l}, l \in \mathbb{N}_+ \)

\[
B = \sum_{k=1}^{t} (B_k \otimes B_k)
\]

By defining

\[
B \otimes I_N = \sum_{k=1}^{t} ((B_k \otimes I_N) \otimes (B_k \otimes I_N))
\]
where \( I_N \) is the arbitrary identity matrix. Then, we will have
\[
\rho(\mathcal{B} \otimes I_N) = \rho(\mathcal{B})
\]  \hspace{1cm} (39)

**Proof 2:** See Appendix B.

Rewriting Eq. (23) we can obtain
\[
\mathcal{F} = \left[ A_1 \left( I - \frac{1}{M} GD \right) A_2 \otimes I \right] \otimes \left[ A_1 \left( I - \frac{1}{M} GD \right) A_2 \otimes I \right] + \frac{N + 1}{M^2} \sum_{k=1}^{M} \left[ (A_1 T_k DA_2 \otimes I) \otimes (A_1 T_k DA_2 \otimes I) \right]
\]  \hspace{1cm} (40)

Thus, according to Theorem.2, one can compute \( \rho(\mathcal{F}) \) by simply set \( I_1 = 1 \). Specifically, by defining
\[
F = (A_1 \otimes A_1) \left[ \left( I - \frac{1}{M} GD \right) \otimes \left( I - \frac{1}{M} GD \right) \right] + \frac{N + 1}{M^2} \sum_{k=1}^{P} \left( T_k D \otimes T_k D \right) (A_2 \otimes A_2)
\]  \hspace{1cm} (41)

we will have \( \rho(F) = \rho(\mathcal{F}) \). Therefore, instead of calculating the spectrum radius of \( \mathcal{F} \) with \( P^2 N^2 \times P^2 N^2 \) dimensions, we can simplify obtain \( \rho(\mathcal{F}) \) from \( F \) which has only \( P^2 \times P^2 \) dimensions. In practical, the \( F \) is typically a sparse matrix. Thus we can use an computationally-efficient search technique \([40, 41]\) to find \( \rho(F) \), which is easy to implement.

**Remark 3:** One should notice that the formulation of Eq. (23) and Eq. (31) are similar with the analysis of sparse diffusion algorithm \([29]\) except the formulation of matrix \( \mathcal{F} \). In \([29]\), the analysis is based on the assumption that the step sizes are sufficient small so that the higher-order powers of step sizes can be ignored. While in CS, due to property of measurement matrix, this assumption can be eliminated. Therefore, we provide a more complete mean square stability analysis.

**A. Further analysis under general parameter settings**

Under Assumption A1-A3, \( \rho(F) < 1 \) gives the sufficient condition for the convergence of the proposed algorithm. Moreover, in practical diffusion adaptation, the step size of all nodes are always set to the same value, and \( S \) is always set to doubly stochastic matrix \([23, 27, 42]\). Therefore, in the following analysis we set \( D = \mu I \) where \( \mu \) is the identical step size for all nodes. The \( S \) is set to doubly stochastic such that \( G = I \). Without loss of generality, here we only analysis ATC strategy, such that \( A_1 = I \). Thus Eq. (41) can be further simplified as
\[
F = \left[ \left( 1 - \frac{\mu}{M} \right)^2 \left( I + \frac{N + 1}{M^2} \mu^2 \sum_{k=1}^{M} (T_k \otimes T_k) \right) \right] (A_2 \otimes A_2)
\]  \hspace{1cm} (42)

Under above parameter settings, one can further derive the following proposition.

**Proposition 1:** For arbitrary column stochastic matrix \( A_2 \) and doubly stochastic matrix \( S \), the upper bound of the step size \( \mu_{\text{max}} \) to guarantee \( \rho(F) < 1 \) in Eq. (41) will within the range
\[
\frac{2M}{(N + 1)\zeta + 1} \leq \mu_{\text{max}} \leq \frac{2PM}{P + N + 1}
\]  \hspace{1cm} (43)

where \( \zeta = \max\{S^T S\} \). Specifically, the maximum \( \mu_{\text{max}} \) can be obtained when the network is fully connected with \( \alpha_{l,k} = 1/P \) for all \( l \) and \( k \).

**Proof 3:** See Appendix C.

Proposition 3 reveals the step size improvement introduced by diffusion strategies. In particular, when \( P = 1 \), \( \mu_{\text{max}} \) will be \( 2M/(N + 2) \), which coincides with the upper bound of \( l_0 \)-LMS algorithm \([30]\). Moreover, when \( P > 1 \), \( \zeta < 1 \) will always hold, and \( \mu_{\text{max}} \) will be always larger than \( 2M/(N + 2) \). While when data scale \( N \) is relatively large compared with network size \( P \), \( \mu_{\text{max}} \) can achieve nearly \( P \) times of the step size upper bound of \( l_0 \)-LMS. Since the formulation of \( Dl_0 \)-LMS and \( l_0 \)-LMS are similar, it can be inferred that large step size will offer faster convergence speed. This fact will be confirmed by simulations in Section V.

Moreover, we can also search the exact \( \mu_{\text{max}} \) based on Eq. (42) under certain \( A_2 \) and \( S \). Actually, experimental results show that \( \rho(F) \) is a convex function of \( \mu \) within the range \( \mu \in \left[ \frac{2M}{(N+1)\zeta + 1}, \frac{2PM}{P + N + 1} \right] \). Thus, we can follow the numerical search algorithm proposed in \([40]\) to iteratively find the exact \( \mu_{\text{max}} \) so that \( \rho(F) = 1 \).

**B. Effect of regularization and data correlation on convergence condition**

In typical CS reconstruction task, apart form diffusion strategies, regularization parameter \( \lambda \) and data correlation also affect the \( \mu_{\text{max}} \). When \( \lambda = 0 \), \( \rho(F) < 1 \) is the necessary and sufficient condition for the convergence of Eq. (23). While when \( \lambda \neq 0 \), we have known from analysis in \([29]\) that \( \rho(F) < 1 \) is only the sufficient condition for the convergence, so that \( \lambda \) will further increase the \( \mu_{\text{max}} \). Moreover, to ensure the successful reconstruction of the sparse vector, one should select proper \( \lambda \) to constrain the sparsity of the estimation as well as achieve desirable accuracy. In practice, \( \lambda \) is always selected as small values, and the increment of \( \mu_{\text{max}} \) by \( \lambda \) will be small.

On the one hand, the independent assumption A2 is hard to satisfy in practical due to limited number of data in each node. Therefore the effect of data correlation on convergence should be taken into consideration. Specifically, the step size upper bound \( \mu_{\text{max}} \) of the adaptive filter under correlation input data has been analyzed in \([40]\), which shows that the \( \mu_{\text{max}} \) of LMS is more stringent than the bound predicted by the independence regressor assumption. Since the weight update form of each node is similar to LMS, we can deduce that the actual upper bound \( \mu_{\text{max}} \) will also be less than that estimated under assumption A2. In particular, different from adaptive system analyzed in \([40]\) where the input data is pairwise related, the input data in CS is periodic related. When \( L_k \) is large (i.e. each node has large number of data), the influence of correlation will be greatly reduced. While when \( L_k \) is small, the effect of data correlation cannot be neglected, therefore \( \mu_{\text{max}} \) will be distinctly smaller than theoretical estimation.

**V. SIMULATION**

In this section we verify the performance of the proposed algorithm in CS reconstruction task. The locations of non-zero
entries of the sparse vector $x$ are randomly selected within $[1, N]$, and the corresponding values are independently generated from uniform distribution within $[-1, -0.2] \cup [0.2, 1]$. Further, $x$ is normalized to a unit vector. The Gaussian measurement matrix is used in the simulations, i.e. each entry of $\Theta$ is generated from Gaussian distribution with zero mean and variance $1/M$. The noise $v$ is zero mean Gaussian distributed with covariance matrix $\frac{\sigma^2}{M} I_{M \times M}$. Then, the observed measurement $y$ are obtained from Eq. (2).

The reconstruction of a sparse vector $x$ is carried out by a connected network with $P$ nodes. The measurement matrix $\Theta$ and corresponding measurement $y$ are equally assigned to each node so that $L_k \in \{[M/P], [M/P] + 1\}, k = 1, 2, ..., P$. In combination step, the averaging weights are used so that $\beta_{l,k} = 1/|N_k|$ for all $l$. While in adaptation step, we use the Metropolis weights defined as

$$
\alpha_{l,k} = \begin{cases} 
\frac{1}{\max\{n_{l,n_l}\}} & l \in N_k \setminus \{k\} \\
1 - \sum_{l \in N_k} c_{l,k} & l = k \\
0 & l \notin N_k
\end{cases}
$$

Thus $S$ is the doubly stochastic matrix.

The estimation misalignment at $i$-th iteration is evaluated by mean squared deviation (MSD) in dB, which is approximated as $10 \log_{10} \left\{ \frac{1}{T} \sum_{t=1}^{T} \frac{1}{P} \sum_{k=1}^{P} \| \omega_{s}(i) - x \|^2 \right\}$ over $T$ numbers of Monte Carlo runs with different sparse signals, sensing matrices and noises. Further, for each Monte Carlo run, the reconstruction is considered successful if the average MSD value of last iteration is less than $1 \times 10^{-2}$. During the whole simulation, without mentioned, the zero attraction parameters of the proposed algorithms are set as $\alpha = 10$. The maximum iteration number $C = 10^5$. The parameters in Eq. (19) and Eq. (18) are set as $\tau = 1 \times 10^{-3}$, $p_s = 20$ and $L_s = 0.2N$.

### A. Convergence performance

In this section, we investigate the convergence performance of the proposed algorithms along with $l_0$-LMS. The simulation is carried by a connected network with 20 nodes. Each node is linked to 3 nodes in average. The CS system parameters are set as $N = 20000$, $M = 4000$ and $K = 500$. The noise variance $\sigma$ is set to $3 \times 10^{-3}$. The regularization parameter $\lambda$ for all algorithms are set to $5 \times 10^{-8}$. The mini batch size $Q$ is set to 5. The step sizes are set as $0.4, 0.4, 16$ for $l_0$-LMS, gradient descend diffusion algorithms (ATC-$l_0$-LMS and CTA-$l_0$-LMS) and mini-batch diffusion algorithms (ATC-MB-$l_0$-LMS and CTA-MB-$l_0$-LMS), respectively. Note that when $\mu = 4$ and 16, traditional $l_0$-LMS and will diverge according to the analysis in [20]. While ATC-$l_0$-LMS and CTA-$l_0$-LMS will diverge when $\mu = 16$ according to the analysis in Section IV.

The average learning curves of all algorithms are shown in Fig. 2. Note that the stop criterion is not used in this simulation. One can observe that ATC-$l_0$-LMS and D-$l_0$-LMS-CTA achieves similar reconstruction MSD with traditional $l_0$-LMS, while converge much faster than $l_0$-LMS. ATC-MB-$l_0$-LMS and MB-D-$l_0$-LMS-CTA can further converge faster than ATC-$l_0$-LMS and CTA-$l_0$-LMS with a slight loss on reconstruction accuracy. Further, one can also observe that the convergence behaviour of ATC strategy is quite similar with CTA.

To further investigate the relation between step size and convergence speed, we scale the learning curves in Fig. 2. In particular, according to the step sizes of each algorithm, the learning curves of ATC-$l_0$-LMS and D-$l_0$-LMS-CTA and shrink 4 times, while the learning curve of $l_0$-LMS shrinks 40 times. The scaled learning curves are shown in Fig. 3. One can observe that the scaled learning curves are similar. The results confirm that the convergence speed is closely related to the step size under the same regularization parameter $\lambda$.

### B. Step size upper bound

In the part, we investigate the relationship between step size upper bound $\mu_{\text{max}}$ and network size $P$. We gradually increase the network size $P$ from 2 to 40. The network grows as follows: at first the network contains only one node, then we gradually add a node which is randomly connected to $p$ nodes of current network. If $p$ is larger than the size of current network, the new node is connected to all the nodes of the

![Fig. 2. Average learning curves of different algorithms](image1)

![Fig. 3. Scaled average learning curves of different algorithms](image2)
previous network. The generated network set is represented as $T_p$. Therefore, a larger $p$ indicates more links across the network in general. In the following simulations, $\mu_{\text{max}}$ is computed as the largest step size that can reconstruct all the sparse signals under 200 Monte carlo runs.

First, we investigate the $\mu_{\text{max}}$ of ATC-Dl0-LMS under different scale of data. The dimension $N$ of the sparse vector is set to 1000 and 20000. The number of measurement $M$ is set to 0.2$N$, and the sparsity $K$ is set to 0.125$M$. The regularization parameter $\lambda$ for all algorithms are set as $5 \times 10^{-6}$ for $N = 1000$ and $5 \times 10^{-8}$ for $N = 20000$. Fig 4(a) depicts the curves of network size $P$ versus $\mu_{\text{max}}$ under different network set $T_2$ and $T_4$. As can be seen, when $N = 20000$, $\mu_{\text{max}}$ can grows linearly under $T_2$ and $T_4$. While for $N = 1000$, the growth of $\mu_{\text{max}}$ is limited when $P$ is large. Further, the curves of theoretical analysis versus simulation results are given in Fig 5. The theoretical $\mu_{\text{max}}$ corresponds to simulated $\mu_{\text{max}}$ with $\lambda = 0$. Note that the reconstruction can not succeed without $\lambda$, thus the simulated $\mu_{\text{max}}$ with $\lambda = 0$ is calculated as the largest step size that ensure the algorithms not diverge. One can observe that when $N = 20000$, simulated and theoretical $\mu_{\text{max}}$ are similar. While for $N = 1000$, due to strong correlation of data, simulated $\mu_{\text{max}}$ are much lower than theoretical analysis when the network size is large. Moreover, the curves of simulated $\mu_{\text{max}}$ with proper $\lambda$ are always above the curves without $\lambda$, which confirms the step size upper bound improvement by $\lambda$.

Second, the $\mu_{\text{max}}$ of ATC-MB-Dl0-LMS under $N = 20000$ is performed. The curves of network size $P$ versus $\mu_{\text{max}}$ under different parameter $p$ and batch size $Q$ are shown in Fig 4(b). One can see that $\mu_{\text{max}}$ is further greatly improved compared with ATC-Dl0-LMS. Specifically, a larger batch size $Q$ and link parameter $p$ allow larger $\mu_{\text{max}}$. Moreover, the results show that mini-batch method gives remarkable acceleration for small network, while the growth of $\mu_{\text{max}}$ is limited when network is large.

We also investigate the performance of ATC-Dl0-LMS and ATC-MB-Dl0-LMS without adaptation information exchange. All the parameters are the same as the first simulation in this part except $S = I$. Fig 4(c) and Fig 4(d) show the curves of network size $P$ versus $\mu_{\text{max}}$ for ATC-Dl0-LMS and ATC-MB-Dl0-LMS, respectively. One can observe that without adaptation step, $\mu_{\text{max}}$ suffers from different degrees of decline. Nevertheless, the acceleration is still significant compared with
$l_0$-LMS. In general, dropping away the adaptation information exchange reduces 50% and 66.7% of the network data transmission for D-$l_0$-LMS and MB-D-$l_0$-LMS, respectively.

Finally, the reconstruction MSD values under different network sizes are conducted. Fig. 6 shows the reconstruction MSD with $\mu_{max}$ from Fig. 4(a)-(d) under $T_2$ and $N = 20000$. One can observe that the different network sizes $P$ affect slightly on reconstruction MSD except ATC-MB-D-$l_0$-LMS without adaptation information exchange. Specifically, when $P$ is large, ATC-MB-D-$l_0$-LMS has the performance loss of about 1.5dB when dropping away adaptation information exchange. Further, ATC-D-$l_0$-LMS achieves lower reconstruction MSD compared with ATC-MB-D-$l_0$-LMS.

C. Sensitivity of regularization parameter $\lambda$ and noise variance $\sigma$

In this part we first focus on the reconstruction MSD under different noise variance $\sigma$. We select Gaussian noise with different variances $\sigma$ as the additive noise of the CS system. The simulation settings are the same as in part A. For each $\sigma$, 200 Monte Carlo runs are performed. Fig. 7 depicts the average MSD curves of three algorithms under different $\sigma$. One can see that ATC-D-$l_0$-LMS and $l_0$-LMS achieve similar reconstruction performance in different noise situations. ATC-MB-D-$l_0$-LMS achieves slightly better performance than ATC-D-$l_0$-LMS and $l_0$-LMS when noise variance is larger than -24dB, while the performance is slightly worse when noise is less than -24dB.

We further compare the reconstruction performance under different regularization parameter $\lambda$. The simulation settings are the same as in part A. The maximum iteration number $C$ is not used in the simulation, so that all the reconstruction MSD values are obtained at steady state. For each $\lambda$, 200 Monte Carlo runs are performed. The curves of average MSD versus $\lambda$ are shown in Fig. 8. One can see that all algorithms can achieve good reconstruction performance when $\lambda$ is not too large (i.e. $\lambda \leq 10^{-7}$). Further, the curve of ATC-D-$l_0$-LMS and $l_0$-LMS are quite similar, while the curve of ATC-MB-D-$l_0$-LMS is moved from the curve of $l_0$-LMS about $2 \times 10^{-8}$ to the left. The results show that the proposed algorithms can achieve similar reconstruction MSD with $l_0$-LMS. For ATC-MB-D-$l_0$-LMS, one can adjust a slightly lower $\lambda$ to obtain the similar performance with ATC-D-$l_0$-LMS and $l_0$-LMS. Moreover, although the algorithms work well for a wide range of $\lambda$, one should note that smaller $\lambda$ will cause slow convergence speed. One should select $\lambda$ in a proper range to make trade off between reconstruction accuracy and convergence speed.

VI. CONCLUSION

In this paper we propose a novel diffusion adaptation framework for CS reconstruction task. Using distributed network, the measurement matrix can be stored in a decentralized manner, thus the storage in each node can be efficiently reduced. Based on diffusion adaptation strategy, the gradient-descent diffusion CS reconstruction algorithm called D-$l_0$-LMS is proposed. D-$l_0$-LMS can collaboratively estimate the sparsity and recover the sparse signal across the network. Particularly, the convergence of D-$l_0$-LMS is analyzed. To further improve the convergence speed, a mini-batch based diffusion algorithm is also proposed. Simulation results confirm the desirable performance of the proposed algorithms.
Compared with $l_0$-LMS algorithm, the proposed algorithms can achieve much faster convergence speed while obtaining similar reconstruction accuracy.

**ACKNOWLEDGEMENTS**

This work was supported by 973 Program (No.2015CB351703), National NSF of China (No.61273366 and No.91648208) and National Science and Technology support program of China (No.2015BAH31F01).

**A. Appendix A**

We follow the proof in \[43\] and extend it to sum of arbitrary number of matrices case. Consider a linear weight update process defined as

$$
\mathbf{w}(i + 1) = \left( \sum_{k=1}^{N} \theta_k(i) \mathbf{B}_k \right) \mathbf{w}(i)
$$

(44)

where $\mathbf{w} \in \mathbb{R}^{l \times 1}$, $\theta_1(i)$ is fixed at 1, while $\theta_k(i) \in \mathbb{R}, k = 2, ..., N$ are random variables with Gaussian distribution $N(0, 1)$. By defining $\mathbf{D}(i) = E[\mathbf{w}(i)\mathbf{w}^T(i)]$ we can obtain following equation

$$
\mathbf{D}(i + 1) = E\left[ \mathbf{w}(i + 1)\mathbf{w}^T(i + 1) \right]
= E\left[ \left( \sum_{k=1}^{N} \theta_k(i) \mathbf{B}_k \right) \mathbf{w}(i)\mathbf{w}^T(i) \left( \sum_{k=1}^{N} \theta_k(i) \mathbf{B}_k^T \right) \right]
= \sum_{k=1}^{N} \mathbf{B}_k \mathbf{D}(i) \mathbf{B}_k^T
$$

(45)

Further, using vectorization operator, we obtain

$$
vec(\mathbf{D}(i + 1)) = \sum_{k=1}^{N} (\mathbf{B}_k \otimes \mathbf{B}_k) vec(\mathbf{D}(i))
$$

(46)

it is known that if $\rho \left( \sum_{k=1}^{N} (\mathbf{B}_k \otimes \mathbf{B}_k) \right) < 1$, $\mathbf{D}(i)$ will converge to 0 as $i \to \infty$ in both $l^2$ and $l$. Thus $E[\mathbf{w}(i)]$ will also converge to 0 as $i \to \infty$. Taking expectation of both sides of Eq.(44), we can get

$$
E[\mathbf{w}(i + 1)] = \mathbf{B}_1 E[\mathbf{w}(i)]
$$

(47)

To ensure the convergence, $\rho(\mathbf{B}_1) < 1$ should be always guaranteed. Therefore, we have the following relation.

$$
\rho \left( \sum_{k=1}^{N} (\mathbf{B}_k \otimes \mathbf{B}_k) \right) < 1 \Rightarrow \rho(\mathbf{B}_1) < 1 \Rightarrow \rho(\mathbf{B}_1 \otimes \mathbf{B}_1) < 1
$$

(48)

Moreover, assume that

$$
\rho \left( \sum_{k=1}^{N} (\mathbf{B}_k \otimes \mathbf{B}_k) \right) = \lambda
$$

(49)

where $\lambda$ is arbitrary positive value. By defining $\beta = \lambda + \varepsilon$ with arbitrary positive value $\varepsilon$, we will have

$$
\rho \left( \sum_{k=1}^{N} (\mathbf{B}_k \otimes \mathbf{B}_k) \right) = \frac{\lambda}{\beta} < 1
$$

(50)

Then, according to Eq.(48),

$$
\rho(\mathbf{B}_1 \otimes \mathbf{B}_1) < 1
$$

(51)

which is equivalently

$$
\rho(\mathbf{B}_1 \otimes \mathbf{B}_1) < \lambda + \varepsilon
$$

(52)

Since Eq.(52) always holds for arbitrary $\varepsilon$, we can obtain

$$
\rho(\mathbf{B}_1 \otimes \mathbf{B}_1) \leq \rho \left( \sum_{k=1}^{N} (\mathbf{B}_k \otimes \mathbf{B}_k) \right)
$$

(53)

That’s end the proof.

**B. Appendix B**

Rewrite $\mathbf{B}$ as a block matrix

$$
\mathbf{B} = \begin{bmatrix}
C_{11} & C_{12} & \cdots & C_{1l} \\
C_{21} & C_{22} & \cdots & \vdots \\
\vdots & \vdots & \ddots & C_{l-1,l} \\
C_{1l} & \cdots & C_{l,l-1} & C_{ll}
\end{bmatrix}
$$

(54)

where $C_{ij}, i, j \in \{1, 2, ..., l\}$ are $l \times l$ matrices. Then $\mathbf{B} \otimes \mathbf{I}_N$ can be derived as

$$
\mathbf{B} \otimes \mathbf{I}_N = \begin{bmatrix}
I_N \otimes C_{11} \otimes I_N & I_N \otimes C_{12} \otimes I_N & \cdots & I_N \otimes C_{1l} \otimes I_N \\
I_N \otimes C_{21} \otimes I_N & \cdots & \cdots & \cdots \\
\vdots & \vdots & \ddots & \vdots \\
I_N \otimes C_{l1} \otimes I_N & \cdots & \cdots & I_N \otimes C_{ll} \otimes I_N
\end{bmatrix}
$$

(55)

Assume $\lambda$ is an eigenvector of $\mathbf{B}$, i.e. $\mathbf{Bx} = \lambda \mathbf{x}$ where $\mathbf{x}$ is the corresponding eigenvector. Moreover, $\mathbf{x}$ can be rewrote as

$$
\mathbf{x} = \begin{bmatrix}
x_1^T & x_2^T & \cdots & x_l^T
\end{bmatrix}^T
$$

(56)

where $x_i \in \mathbb{R}^{l \times 1}$, $i = 1, 2, ..., l$. We define a new vector $\mathbf{y}_p$

$$
\mathbf{y}_p = vec\left\{ \begin{bmatrix}
x_1 & x_2 & \cdots & x_l
\end{bmatrix} \otimes \mathbf{z}_p \right\}, p = 1, ..., N^2
$$

(57)
where $Z_p$ obeys that the $p$-th element of $\text{vec}\{Z_p\}$ is 1 while other elements are set to 0. Therefore, one can get that following relation

$$B^\otimes I_N y_p = \lambda y_p, p = 1, ..., N^2$$ (58)

That is, given arbitrary eigenvalue $\lambda$ of $B$, $B^\otimes I_N$ will have $N^2$ numbers of $\lambda$ as the eigenvalues. Moreover, since the number of eigenvalues of $B$ is $I^2$ while $B^\otimes I_N$ is $I^2 \times N^2$, the eigenvalues of $B^\otimes I_N$ will be

$$\{\lambda_1, \lambda_1, ..., \lambda_1, \lambda_2, \lambda_2, ..., \lambda_2, ..., \lambda_{I^2}, \lambda_{I^2}, ..., \lambda_{I^2}\}$$ (59)

and consequently

$$\rho(B^\otimes I_N) = \rho(B)$$ (60)

That’s end the proof.

C. Appendix C

One can observe from of Eq.(42) that

$$V = (1 - \frac{\mu}{M})^2 I + \frac{N + 1}{M^2} \mu^2 \sum_{k=1}^{M} (T_k \otimes T_k)$$ (61)

$$= (1 - \frac{\mu}{M})^2 I + \frac{N + 1}{M^2} \mu^2 \text{diag}(\text{vec}\{S^T S\})$$

is an diagonal matrix and $A_2^\otimes = (A_2 \otimes A_2)$ is a column stochastic matrix where the sum of each column is equal to 1. Thus, according to the theorem that the spectral radius is not more than any norms of the matrix, we will have

$$\rho(F) = \rho(V \ A_2^\otimes)$$

$$= \rho(A_2^\otimes V)$$

$$\leq \max\{\text{diag}(V)\}$$ (62)

$$= \left(1 - \frac{\mu}{M}\right)^2 I + \frac{N + 1}{M^2} \mu^2 \max\{S^T S\}$$

Therefore, if $\max\{\text{diag}(V)\} < 1$, then $\rho(F) < 1$ will be guaranteed. Thus the sufficient condition for $\rho(F) < 1$ will be

$$\mu < \frac{2M}{(N + 1)\max\{S^T S\} + 1}$$ (63)

To proof the right inequality of Eq.(43), we first propose the following theorem

**Theorem 3**: For arbitrary real square matrices sequence $\{B_k\}_{k=1}^{N} \in \mathbb{R}^{l \times l}, l \in \mathbb{N}_+$, the following inequality will always hold

$$\rho\left(\sum_{k=1}^{N} B_k \otimes \sum_{k=1}^{N} B_k + q \sum_{k=1}^{N} (B_k \otimes B_k)\right) \geq \left(p + \frac{q}{N}\right) \rho\left(\sum_{k=1}^{N} B_k \otimes \sum_{k=1}^{N} B_k\right)$$ (64)

where the equality will always hold when $B_1 = B_2 = ... = B_N$. The proof is given in Appendix D.

Since $\sum_{k=1}^{N} T_k = I$, according to Theorem 3 we can obtain

$$\rho(F) \geq \left(1 - \frac{2\mu}{M} + \frac{\mu^2}{M^2} + \frac{(N + 1)\mu^2}{M^2 P}\right) \rho(A_2 \otimes A_2)$$ (65)

It is known that $\rho(A_2) = 1$ since $A_2$ are both column stochastic matrices. Therefore $\rho(A_2 \otimes A_2) = 1$ and

$$\rho(F) \geq 1 - \frac{2\mu}{M} + \frac{\mu^2}{M^2} + \frac{(N + 1)\mu^2}{M^2 P}$$ (66)

The equality will hold if $T_1 = T_2 = ... = T_p$. If $1 - 2\frac{\mu}{M} + \frac{\mu^2}{M^2} + \frac{(N + 1)\mu^2}{M^2 P} \geq 1$, $\rho(F)$ will always no less than 1. Thus one can obtain the necessary condition for the convergence

$$\mu \leq \frac{2PM}{P + N + 1}$$ (67)

Combining the sufficient condition in Eq.(65) and necessary condition in Eq.(67), we can obtain the range of upper bound $\mu_{\max}$

$$\frac{2M}{(N + 1)\max\{S^T S\} + 1} \leq \mu_{\max} \leq \frac{2PM}{P + N + 1}$$ (68)

That’s end the proof.

D. Appendix D

For arbitrary real square matrices sequence $\{B_k\}_{k=1}^{N} \in \mathbb{R}^{l \times l}, l \in \mathbb{N}_+$, we will have

$$p \sum_{k=1}^{N} B_k \otimes \sum_{k=1}^{N} B_k + q \sum_{k=1}^{N} (B_k \otimes B_k)$$

$$= p \sum_{k=1}^{N} B_k \otimes \sum_{k=1}^{N} B_k$$

$$+ \frac{q}{N} \left(\sum_{k=1}^{N} B_k \otimes \sum_{k=1}^{N} B_k + \sum_{i=1}^{N} \sum_{j=1}^{N} (B_i - B_j) \otimes (B_i - B_j)\right)$$

$$\geq \left(p + \frac{q}{N}\right) \sum_{k=1}^{N} B_k \otimes \sum_{k=1}^{N} B_k$$

According to Theorem 1, we can obtain

$$\rho\left(\sum_{k=1}^{N} B_k \otimes \sum_{k=1}^{N} B_k + q \sum_{k=1}^{N} (B_k \otimes B_k)\right)$$

$$\geq \left(p + \frac{q}{N}\right) \rho\left(\sum_{k=1}^{N} B_k \otimes \sum_{k=1}^{N} B_k\right)$$ (70)

Specifically, when $B_1 = B_2 = ... = B_N$ the equality will hold. That’s end the proof.

REFERENCES

[1] D. L. Donoho, “Compressed sensing,” IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289–1306, 2006.
[2] E. J. Candès et al., “Compressive sampling,” in Proceedings of the international congress of mathematicians, vol. 3. Madrid, Spain, 2006, pp. 1433–1452.
[3] E. J. Candès, J. Romberg, and T. Tao, “Robust uncertainty principles: exact signal reconstruction from highly incomplete frequency information,” IEEE Transactions on Information Theory, vol. 52, no. 2, pp. 489–509, 2006.
[4] E. J. Candès and T. Tao, “Near-optimal signal recovery from random projections: Universal encoding strategies?” IEEE Transactions on Information Theory, vol. 52, no. 12, pp. 5406–5425, 2004.

[5] E. J. Candès and M. B. Wakin, “An introduction to compressive sampling.” IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 21–30, 2008.

[6] M. Lustig and J. Donoho, Dpaula, “Sparse mri: The application of compressed sensing for rapid mr imaging.” Magnetic Resonance in Medicine, vol. 58, no. 6, p. 1182C1195, 2007.

[7] L. C. Potter, E. Ertin, J. T. Parker, and M. Cetin, “Sparsity and compressed sensing in radar imaging,” Proceedings of the IEEE, vol. 98, no. 6, pp. 1006–1020, 2010.

[8] S. G. Mallat and Z. Zhang, “Matching pursuits with time-frequency dictionaries,” IEEE Transactions on Signal Processing, vol. 41, no. 12, pp. 3397 – 3415, 1993.

[9] J. A. Tropp and A. C. Gilbert, “Signal recovery from random measurements via orthogonal matching pursuit,” Information Theory IEEE Transactions on, vol. 53, no. 12, pp. 4655–4666, 2007.

[10] T. Blumensath and M. E. Davies, “Iterative hard thresholding for compressed sensing.” Applied and Computational Harmonic Analysis, vol. 27, no. 3, pp. 265–274, 2009.

[11] S. Chen, M. A. Saunders, and D. L. Donoho, “Atomic decomposition by basis pursuit,” Siam Review, vol. 43, no. 1, pp. 129–159, 2001.

[12] R. Chartrand and W. Yin, “Iteratively reweighted algorithms for compressive sensing,” in IEEE International Conference on Acoustics, Speech and Signal Processing, 2008, pp. 3869–3872.

[13] R. Tibshirani, “Regression shrinkage and selection via the lasso: a retrospective,” Journal of the Royal Statistical Society, vol. 73, no. 3, p. 277C307, 2011.

[14] Cands and J. Emmanuel, “The restricted isometry property and its implications for compressed sensing,” Comptes Rendus Mathematique, vol. 346, no. 9, pp. 589–592, 2008.

[15] E. Cands and J. Romberg, “Sparsity and incoherence in compressive sampling,” vol. 23, no. 3, pp. 969–985(17), 2006.

[16] R. Baraniuk, M. Davenport, R. Devore, and M. Wakin, “A simple proof of the restricted isometry property for random matrices,” in CONSTR APPROX, 2008, pp. 253–263.

[17] S. H. Hsieh, C. S. Lu, and S. C. Pei, “Fast greedy approaches for compressive sensing of large-scale signals,” 2015. [Online]. Available: http://arxiv.org/abs/1509.03979

[18] T. T. Do, G. Lu, N. H. Nguyen, and T. D. Tran, “Fast and efficient compressive sensing using structurally random matrices,” IEEE Transactions on Signal Processing, vol. 60, no. 1, pp. 139–154, 2012.

[19] M. A. T. Figueiredo, R. D. Nowak, and S. J. Wright, “Gradient projection for sparse reconstruction: Application to compressed sensing and other inverse problems,” in IEEE Journal of Selected Topics in Signal Processing, 2007, pp. 586 – 597.

[20] S. J. Wright, R. D. Nowak, and M. A. T. Figueiredo, “Sparse reconstruction by separable approximation,” IEEE Transactions on Signal Processing, vol. 57, no. 7, pp. 2479–2493, 2009.

[21] L. Gan, “Block compressed sensing of natural images,” in International Conference on Digital Signal Processing, 2007, pp. 403–406.

[22] S. Mun and J. E. Fowler, “Block compressed sensing of images using directional transforms,” in 2009 16th IEEE International Conference on Image Processing (ICIP), 2009, pp. 3021–3024.

[23] C. G. Lopes and A. H. Sayed, “Diffusion least-mean squares over adaptive networks: Formulation and performance analysis.” IEEE Transactions on Signal Processing, vol. 56, no. 7, pp. 3122–3136, 2008.

[24] F. S. Cattivelli and A. H. Sayed, “Diffusion lms strategies for distributed estimation,” IEEE Transactions on Signal Processing, vol. 58, no. 3, pp. 1035–1048, 2010.

[25] J. Chen and A. H. Sayed, “Diffusion adaptation strategies for distributed optimization and learning over networks,” IEEE Transactions on Signal Processing, vol. 60, no. 8, pp. 4289–4305, 2012.

[26] A. H. Sayed, “Diffusion adaptation over networks,” Academic Press Library in Signal Processing, vol. 3, pp. 323–454, 2013.

[27] R. Abdollee and B. Champagne, “Diffusion lms strategies in sensor networks with noisy input data,” IEEE/ACM Transactions on Networking, vol. 24, no. 1, pp. 3–14, 2015.

[28] Y. Chen, Y. Gu, and A. O. Hero, “Sparse lms for system identification,” in IEEE International Conference on Acoustics, Speech and Signal Processing, 2009, pp. 3125–3128.

[29] P. D. Lorenzo and A. H. Sayed, “Sparse distributed learning based on diffusion adaptation,” IEEE Transactions on Signal Processing, vol. 61, no. 6, pp. 1419–1433, 2013.