Convolutional Sparse Coding Fast Approximation With Application to Seismic Reflectivity Estimation

Deborah Pereg, Israel Cohen, *Fellow, IEEE*, and Anthony A. Vassiliou, *Member, IEEE*

***Abstract*— In sparse coding, we attempt to extract features of input vectors, assuming that the data is inherently structured as a sparse superposition of basic building blocks. Similarly, neural networks perform a given task by learning features of the training dataset. Recently, both data- and model-driven feature extracting methods have become extremely popular and have achieved remarkable results. Nevertheless, practical implementations are often too slow to be employed in real-life scenarios, especially for real-time applications. We propose a speed-up upgraded version of the classic iterative thresholding algorithm (ITA), which produces a good approximation of the convolutional sparse code (CSC) within 2–5 iterations. The speed advantage is gained mostly from the observation that most solvers are slowed down by inefficient global thresholding. The main idea is to normalize each data point by the local receptive field energy, before applying a threshold. This way, the natural inclination toward strong feature expressions is suppressed, so that one can rely on a global threshold that can be easily approximated, or learned during training. The proposed algorithm can be employed with a known predetermined dictionary, or with a trained dictionary. The trained version is implemented as a neural net designed as the unfolding of the proposed solver. The performance of the proposed solution is demonstrated via the seismic inversion problem in both synthetic and real data scenarios. We also provide theoretical guarantees for a stable support recovery, namely we prove that under certain conditions, the true support is perfectly recovered within the first iteration.**

***Index Terms*— Convolutional neural network (CNN), convolutional sparse coding (CSC), deep learning, seismic inversion, sparse reflectivity.**

I. Introduction

IN SPARSE coding, we attempt to decompose an observation signal $\mathbf{y} \in \mathbb{R}^{N\times 1}$ into its building blocks (atoms) [1]. Namely, the sparse representations model [2] assumes a signal $\mathbf{y} \in \mathbb{R}^{N\times 1}$ that can be formulated as a sparse superposition of atoms. Mathematically speaking, we assume that the observation signal $\mathbf{y}$ obeys

$$\mathbf{y} = \mathbf{D}\mathbf{x} \tag{1}$$

where $\mathbf{D} \in \mathbb{R}^{N\times M}$ is a matrix called the dictionary that consists of the atoms $\mathbf{d}_i \in \mathbb{R}^{N\times 1}$, $i = 1, \ldots, M$, as its columns, and $\mathbf{x} \in \mathbb{R}^{M\times 1}$ is the *sparse* vector of the atom's weights. Generally, we do not impose any relation between N and M. That is, the dictionary could either be complete ($N = M$), over-complete ($N < M$), or under-complete ($N > M$).

Over the years, great efforts have been invested in finding sparse solutions to (2). Breaking down a signal into its building blocks has become a popular task in many fields related to signal processing, such as image processing [2], computer vision [3], compressed sensing [4], radar [5], ultrasound imaging [6], seismology [7]–[9], visual neurosciense [10], [11], and more. Moreover, from a certain perspective, neural networks can be viewed as an unfolding of an iterative sparse coding solver [12]. In a sense, deep neural nets (DNNs) are trained to seek the atom's weights, specifically convolutional neural networks (CNNs) [13], which are trained to find a set of filters tailored to perform a classification or a regression task at hand. However, in many real-time applications, such as pattern recognition, sparse coding is still a bottleneck in terms of inference time. Most often, a sparse code is required for every image patch. Consequently, many attempts have been made to pursue faster methods for sparse coding.

The iterative shrinkage thresholding algorithm (ISTA) [14] is one of the most popular algorithms for sparse coding. Despite its simplicity, ISTA is considered as a slow algorithm. Over time, faster extensions have been suggested, such as fast-ISTA (FISTA) [15], learned-ISTA (LISTA) [16], and ada-LISTA [17]. LISTA [16], for example, uses a learned substitute dictionary, and ada-LISTA [17] incorporates an adaptive threshold, where the first threshold corresponds to the maximal feature weight in the data. Then, the thresholds are gradually decreasing at each iteration. Since the seminal idea of LISTA—to unroll the iterative algorithm into feed-forward layers—was first proposed, many similar sparse coding model-based deep learning methods have been proposed, such as ADMM with CNN [18], ADMM-CSNet [19], and FISTA-Net [20]. These methods are designed to provide accurate and fast reconstruction compared with other deep learning methods.

Inspired by the classic iterative thresholding algorithms (ITAs), in this work, we propose a fast alternative algorithm that produces a good approximation of a convolutional sparse code (CSC). Most solvers are slowed down by the use of one

Manuscript received January 13, 2021; revised March 16, 2021, May 30, 2021, and June 29, 2021; accepted August 2, 2021. Date of publication August 26, 2021; date of current version January 31, 2022. *(Corresponding author: Deborah Pereg.)*

Deborah Pereg and Israel Cohen are with Andrew and Erna Viterby Faculty of Electrical and Computer Engineering, Technion–Israel Institute of Technology, Technion, Haifa 3200003, Israel (e-mail: deborahp@campus.technion.ac.il; icohen@ee.technion.ac.il).

Anthony A. Vassiliou is with GeoEnergy, Inc., Houston, TX 77042 USA (e-mail: anthony@geoenergycorp.com).

Digital Object Identifier 10.1109/TGRS.2021.3105300

1558-0644 © 2021 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

global threshold (bias) to detect each local feature shift along the signal, or a predetermined constant local threshold. This way, even if the signal (or the input batch) is normalized, when we apply a threshold at each iteration, if the threshold is too high, weak expressions are annihilated, and strong expressions can "cast a shadow" over low-energy regions in the signal, which can be interpreted as false-positive support locations. On the other hand, if the threshold is very small, as often is the case in ISTA, many iterations are required to compensate for false detections in early iterations, especially in the presence of noise, and in real-time applications due to model perturbations.

Alternatively, we propose to normalize each data point by a locally focused data energy measure, before applying a threshold. In other words, each receptive field of the data is scaled with respect to the local energy. This way even when the data is inherently unbalanced, we can still use a common bias for all receptive fields, without requiring many iterations that are usually required in order to globally detect the features support. The proposed algorithm can be employed with a known predetermined dictionary and global fixed bias terms for each iteration, or with a learned dictionary and learned bias terms. In practice, the trained version is implemented as a small recurrent CNN that corresponds to a few unfolded iterations of the proposed method. An approximate solution is produced within only 2–5 iterations, that is, to our knowledge, the state-of-the-art performance in terms of speed and computational complexity versus accuracy error.

We further demonstrate the applicability of the proposed solution to seismic inversion, via experimental results with real data and synthetic data, demonstrating that even if the mutual coherence of the dictionary is relatively high, the first iteration of the method accurately detects at least 70% of the features weights. The following few iterations make the required corrections. We performed extensive synthetic data and real data numerical experiments, in order to verify the robustness of the method in noisy and attenuating environment. We also prove that under sufficient separation and sufficiently low mutual coherence, the first iteration of our method is guaranteed to perfectly recover the true support. Furthermore, in our opinion, the proposed predictor can be potentially included in learning systems in many applications, such as recognition systems and biomedical-imaging super-resolution.

The main contribution of this work is a highly efficient method for fast CSC approximation. We also propose a learning-based (data-driven) variation of the proposed method. The performance is demonstrated via extensive numerical experiments conducted with seismic real data as well as with synthetic data. Furthermore, we prove that the proposed method achieves a reliable solution under sufficient separation or low mutual coherence.

The remainder of this article is organized as follows. Section II provides the necessary background for sparse representations CSC and ITAs. Section III describes the seismic inversion problem. In Section IV, we present the proposed method and its theoretical guarantees. Sections V–VI introduce synthetic and real data experimental results. Section VII proposes a learned version of the proposed algorithm and demonstrates its employment experimentally. Finally, Section VIII concludes and discusses future research directions.

II. Background and Related Work

A. Sparse Representations

The sparse representations model [2] assumes a signal $\mathbf{y} \in \mathbb{R}^{N\times 1}$ that is formulated as a sparse superposition of atoms

$$\mathbf{y} = \mathbf{D}\mathbf{x} \tag{2}$$

where $\mathbf{D} \in \mathbb{R}^{N\times M}$ is a matrix called the dictionary, built of the atoms $\mathbf{d}_i \in \mathbb{R}^{N\times 1}$, $i = 1,\ldots,M$, as its columns, and $\mathbf{x} \in \mathbb{R}^{M\times 1}$ is the sparse vector of the atoms weights.

An immense amount of work has been dedicated to sparse coding, that is, to the recovery of $\mathbf{x}$. To find the sparsest solution, the one with the smallest ℓ_0-norm, we attempt to solve

$$(P_0): \ \min_{\mathbf{x}} \|\mathbf{x}\|_0 \ \text{ s.t. } \mathbf{y} = \mathbf{D}\mathbf{x} \tag{3}$$

where $\|\mathbf{x}\|_0$ denotes the number of non-zeros in $\mathbf{x}$. Since P_0 has been proved to be, in general, NP-hard [21], we often replace the ℓ_0-norm with the ℓ_1-norm

$$(P_1): \ \min_{\mathbf{x}} \|\mathbf{x}\|_1 \ \text{ s.t. } \mathbf{y} = \mathbf{D}\mathbf{x} \tag{4}$$

where $\|\mathbf{x}\|_1 \triangleq \sum_i |x_i|$. In noisy environment or when some error is allowed, we attempt to solve

$$(P_{1,\varepsilon}): \ \min_{\mathbf{x}} \|\mathbf{x}\|_1 \ \text{ s.t. } \|\mathbf{y} - \mathbf{D}\mathbf{x}\|_2 \le \varepsilon \tag{5}$$

where $\|\mathbf{x}\|_2 \triangleq \sqrt{\sum_i x_i^2}$. Under certain conditions, the sparsest solution to P_0 and P_1 has been proved to be unique and can be retrieved using practical algorithms, such as orthonormal matching pursuit (OMP) or basis pursuit (BP), depending on the dictionary's properties and the sparsity of $\mathbf{x}$. Namely, under the assumption that $\|\mathbf{x}\|_0 < \frac{1}{2}(1 + \frac{1}{\mu(\mathbf{D})})$, where $\mu(\mathbf{D})$ is the mutual coherence defined as the maximal correlation coefficient between two dictionary atoms

$$\mu(\mathbf{D}) = \max_{i\neq j} \frac{\left|\mathbf{d}_i^T \mathbf{d}_j\right|}{\|\mathbf{d}_i\|_2 \cdot \|\mathbf{d}_j\|_2} \tag{6}$$

the true sparse code $\mathbf{x}$ can be perfectly recovered [22].

One of the most intuitive ways to recover $\mathbf{x}$ is to project $\mathbf{y}$ on the dictionary and then extract the atoms with the strongest response by taking a hard or a soft threshold. In other words, the solution is a closed-form solution, formulated as $\mathbf{x} = \mathcal{H}_\beta(\mathbf{D}^T\mathbf{y})$ or $\mathbf{x} = \mathcal{S}_\beta(\mathbf{D}^T\mathbf{y})$, where the hard threshold and the soft threshold operators are, respectively, defined as

$$\mathcal{H}_\beta(z) = \begin{cases} z, & |z| > \beta \\ 0, & |z| \le \beta \end{cases}$$

and

$$\mathcal{S}_\beta(z) = \begin{cases} z+\beta, & z < -\beta \\ 0, & |z| \le \beta \\ z-\beta, & z > \beta. \end{cases}$$

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

Note that the rectified linear units (ReLUs) activation function commonly used in DNNs satisfies

$$\text{ReLU}(z-\beta)=\max(z-\beta,0)=\mathcal{S}_\beta^+(z)\triangleq\begin{cases}0, & z\le\beta\\ z-\beta, & z>\beta.\end{cases}$$

Therefore, the soft threshold solution can also be written as

$$\begin{aligned}\mathbf{x}&=\mathcal{S}_\beta^+(\mathbf{D}^T\mathbf{y})-\mathcal{S}_\beta^+(-\mathbf{D}^T\mathbf{y})\\&=\text{ReLU}(\mathbf{D}^T\mathbf{y}-\beta)-\text{ReLU}(-\mathbf{D}^T\mathbf{y}-\beta).\end{aligned}$$

However, simple thresholding is guaranteed to recover the true support only under a more restrictive assumption that $\|\mathbf{x}\|_0<\frac{1}{2}(1+\frac{1}{\mu(\mathbf{D})}\frac{|\mathbf{x}|_{\min}}{|\mathbf{x}|_{\max}})$, where $|\mathbf{x}|_{\min}$ and $|\mathbf{x}|_{\max}$ are the minimum and maximum values of the vector $|\mathbf{x}|$ on the support, respectively, implying that when the data is unbalanced, this approach is bound to collapse.

Papyan *et al.* [12] show that the forward pass of CNNs is equivalent to the layered thresholding algorithm designed to solve the CSC problem. A convolutional layer's forward pass is guaranteed to recover a sparse estimate of the underlying representations of an input signal, only for dictionaries with very low mutual coherence, and when the ratio between the maximal and minimal spikes' amplitudes in absolute value is close to one. In addition, the $\ell_{0,\infty}$ of the true solution $\mathbf{x}$, that is, the required maximal number of non-zeros in a stripe of coefficients contributing to a data point, depends on the ratio $(|\mathbf{x}|_{\min}/|\mathbf{x}|_{\max})$. Therefore, under these very limiting conditions, when applied to real-life applications, such as the seismic reflectivity estimation problem, for which the mutual coherence of the dictionary is very high and the seismic dataset is inherently unbalanced (i.e., with peak and trough amplitudes that are not close), this course of action is inadequate.

One may also wonder whether the dictionary $\mathbf{D}$ is known, and if not, then how and under what conditions could one find the atoms or features concerning his or her specific problem. Some of these questions are addressed in Section VII and in Appendixes A and C.

B. Iterative Shrinkage Algorithms

Consider the cost function

$$f(\mathbf{x})=\frac{1}{2}\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2^2+\lambda\|\mathbf{x}\|_1.$$

Following majorization minimization (MM) strategy, we can build a surrogate function [2], [14]

$$\begin{aligned}Q(\mathbf{x},\mathbf{x}_\theta)&=f(\mathbf{x})+d(\mathbf{x},\mathbf{x}_\theta)\\&=\frac{1}{2}\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2^2+\lambda\|\mathbf{x}\|_1+\frac{c}{2}\|\mathbf{x}-\mathbf{x}_\theta\|_2^2\\&\quad-\frac{1}{2}\|\mathbf{D}\mathbf{x}-\mathbf{D}\mathbf{x}_\theta\|_2^2.\end{aligned}$$

The parameter c is chosen such that the added expression

$$d(\mathbf{x},\mathbf{x}_\theta)=Q(\mathbf{x},\mathbf{x}_\theta)-f(\mathbf{x})=\frac{c}{2}\|\mathbf{x}-\mathbf{x}_\theta\|_2^2-\frac{1}{2}\|\mathbf{D}\mathbf{x}-\mathbf{D}\mathbf{x}_\theta\|_2^2$$

is strictly convex, requiring its Hessian to be positive definite, $c\mathbf{I}-\mathbf{D}^T\mathbf{D}\prec\mathbf{0}$. Therefore $c>\|\mathbf{D}^T\mathbf{D}\|_2=\lambda_{max}(\mathbf{D}^T\mathbf{D})$, that is, greater than the largest eigenvalue of the coherence matrix $\mathbf{D}^T\mathbf{D}$. In essence, the term $d(\mathbf{x},\mathbf{x}_\theta)$ is a measure of proximity to a previous solution $\mathbf{x}_\theta$. If the vector difference $\mathbf{x}-\mathbf{x}_\theta$ is spanned by $\mathbf{D}$, the distance drops to nearly zero (we usually choose $c=\|\mathbf{D}\|_2^2$). Then, we remain with a minimization over the original cost function $f(\mathbf{x})$. Alternatively, if $\mathbf{D}$ is not full rank and the change $\mathbf{x}-\mathbf{x}_\theta$ is close to the null space of $\mathbf{D}$, the distance is simply the approximate Euclidean distance between the current solution to the previous one.

The surrogate function $Q(\mathbf{x},\mathbf{x}_\theta)$ obeys equality at $\mathbf{x}_\theta$: $Q(\mathbf{x}_\theta,\mathbf{x}_\theta)=f(\mathbf{x}_\theta)$. It is upper-bounded by the original function: $Q(\mathbf{x},\mathbf{x}_\theta)\ge f(\mathbf{x})\ \forall\mathbf{x}$, and tangent at $\mathbf{x}_\theta$: $\nabla Q(\mathbf{x},\mathbf{x}_\theta)|_{\mathbf{x}=\mathbf{x}_\theta}=\nabla f(\mathbf{x})|_{\mathbf{x}=\mathbf{x}_\theta}$. Hence, the solution sequence is guaranteed to yield decreasing values of the original cost function $f(\mathbf{x})$ because

$$\begin{aligned}f(\mathbf{x}_{\theta+1})&\le Q(\mathbf{x}_{\theta+1},\mathbf{x}_\theta)=\min_{\mathbf{x}}Q(\mathbf{x},\mathbf{x}_\theta)\\&\le Q(\mathbf{x}_\theta,\mathbf{x}_\theta)=f(\mathbf{x}_\theta).\end{aligned}$$

Following the MM strategy, that is, minimizing $Q(\mathbf{x},\mathbf{x}_\theta)$ instead of $f(\mathbf{x})$, the sequence of iterative solutions is generated by the recurrent formula

$$\mathbf{x}_{\theta+1}=\arg\min_{\mathbf{x}}Q(\mathbf{x},\mathbf{x}_\theta)$$

where $\theta\in\mathbb{N}$ is the iteration index. Therefore, we can find a closed-form solution for its global minimizer

$$\mathbf{x}_{\theta+1}=\mathcal{S}_{\frac{\lambda}{c}}\left(\frac{1}{c}\mathbf{D}^T(\mathbf{y}-\mathbf{D}\mathbf{x}_\theta)+\mathbf{x}_\theta\right).$$

Intuitively, this sequence can be interpreted as an iterative projection of the dictionary on the residual term, starting from the initial solution that is simply a thresholded projection of the dictionary on the observation signal (assuming $\mathbf{x}_0=\mathbf{0}$)

$$\mathbf{x}_{\theta+1}=\mathcal{S}_{\frac{\lambda}{c}}\left(\frac{1}{c}\overbrace{\mathbf{D}^T}^{\text{project on dictionary}}\underbrace{(\mathbf{y}-\mathbf{D}\mathbf{x}_\theta)}_{\text{residual term}}+\underbrace{\mathbf{x}_\theta}_{\text{add to current solution}}\right).$$

Under the assumption that the constant c is large enough, it was shown in [14] that the above algorithm is guaranteed to converge to its global minimum. Hence, we are guaranteed to recover a local minimum of $f(\mathbf{x})$. This approach can also be viewed as a proximal-point algorithm [23], or as a simple projected gradient descent algorithm.

It is worth mentioning that ISTA can also be viewed as a recurrent neural net (RNN) unfolded through time [12]. As previously stated, despite its simplicity, ISTA is considered as a slow algorithm. Over time, faster extensions have been suggested, such as FISTA [15], LISTA [16], and Ada-LISTA [17]. We refer the reader to the corresponding references for further details.

C. Convolutional Sparse Coding

In the special case where $\mathbf{D}$ is a convolutional dictionary, the task of extracting $\mathbf{x}$ is referred to as CSC. In this case, the dictionary $\mathbf{D}$ is a convolutional matrix constructed by shifting a local matrix of m filters in all possible positions. Equivalently, let us assume a convolutional dictionary that is structured as a concatenation of m convolution matrices

$$\mathbf{D}=[\mathcal{D}_1,\mathcal{D}_2,\ldots,\mathcal{D}_m]\tag{7}$$

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

where $\mathcal{D}_p \in \mathbb{R}^{L_y \times L_x}$, $p \in \mathbb{N}$, $1 \le p \le m$, is the convolution matrix of the p'th filter denoted by $\tilde{\mathbf{d}}_p \in \mathbb{R}^{L_d \times 1}$ shifted in all possible directions. We assume a linear convolution, such that $L_y = L_x + L_d - 1$. Accordingly, the sparse weights vector $\mathbf{x} \in \mathbb{R}^{mL_x \times 1}$ obeys

$$\mathbf{x} = \left[\mathbf{x}_{\mathcal{D}_1}^T, \mathbf{x}_{\mathcal{D}_2}^T, \ldots, \mathbf{x}_{\mathcal{D}_m}^T\right]^T \tag{8}$$

where L_x is the support size for each of the filters, and $\mathbf{x}_{\mathcal{D}_p}$, $p \in \mathbb{N}$, $1 \le p \le m$, is the sparse weights vector corresponding to the pth filter shifts along the signal. Overall, there are $M = mL_x$ atoms in the dictionary, such that the ith atom $\mathbf{d}_i$, $i = (p-1)L_x + l$, $l \in \mathbb{N}, 1 \le l \le L_x$, is the pth filter's shift to the lth support location.

In the forward pass of each layer in a CNN, the input is convolved with a set of learned filters. Then, we apply a pointwise nonlinear function to the computed feature map summed with a bias term. This process can be viewed as equivalent to the layered thresholding algorithm for the CSC model [12]. In other words, the forward pass of a CNN is inherently based on revealing an estimate of a hidden CSC of a given signal.

Since CNNs were inspired by the study of the visual brain cortex, the term receptive field is borrowed to describe a small limited local visual area that a neuron reacts to [24]. In other words, in CNNs, a neuron located in a certain layer is connected only to the output of neurons in a limited small area of the previous layer. Considering (7), since the ith atom of dictionary $\mathcal{D}_p$ is the pth filter shifted to the lth support location, having a small restricted support of L_d samples around l in the data, each of these small local support areas is referred to as a receptive field.

D. Signature Dictionary

The signature dictionary [25] is essentially a convolutional dictionary of a single filter ($m = 1$). In this case, we attempt to represent a signal solely by one small kernel. An elaborated discussion on the learning of the signature dictionary and its signal representation can be found in [25].

III. Problem Formulation

In this section, the mathematical formulation focuses on the settings of the seismic inversion task. Nonetheless, the model can be applied to a wide range of applications (such as medical imaging, computer vision, pattern recognition, and so on) and can be incorporated in any system employing CNNs and RNNs. The method is not restricted to signature dictionaries and can be applied to convolutional dictionaries with more than one filter and to non-stationary convolutional operators as described below.

A. Signal Model

Consider an unknown 2-D reflectivity signal $\mathbf{X} \in \mathbb{R}^{L_x \times J}$ of J channels representing the true reflectivity cross section. We assume a layered subsurface structure and acoustic wave propagation, where reflections are generated at acoustic impedance boundaries. Hence, each hidden column of index l in the reflectivity image is a 1-D signal $\mathbf{x}^{(l)} \in \mathbb{R}^{L_x \times 1}$ independently modeled as a *sparse* weights vector. In a discrete setting, we consider a set of two-way travel times $T = \{t_m\}$ lying on a grid kT_s, $k \in \mathbb{Z}$, where $t_m = k_m T_s$, with a sampling rate $F_s = (1/T_s)$, corresponding to a reflector's time-depth location in the ground. Accordingly, a 1-D reflectivity signal is formulated as

$$x^{(l)}[k] = \sum_m c_m \delta[k - k_m], \quad k \in \mathbb{Z}, \ c_m \in \mathbb{R}, \ l = 1, \ldots, J \tag{9}$$

where $\delta[k]$ denotes the Kronecker delta function [26], $\sum_m |c_m| < \infty$, and $K = \{k_m\}$ is the set of discrete time delays.

We assume that the support K is sufficiently separated. In other words, it obeys the minimal separation condition (see [27, Definition 2.2]), with a separation constant ν, namely

$$\Delta_k \triangleq \min_{k_m, k_n \in K, m \ne n} |k_m - k_n| \ge F_s \nu \sigma$$

where $\sigma > 0$ is a given kernel scaling. $\Delta_t \triangleq \nu\sigma$ is the smallest time interval between two reflectors, for which we are guaranteed to perfectly recover two distinct spikes in a noise-free environment. In [27], we prove that under the minimal separation condition, in a noise-free environment, $x^{(l)}[k]$ is perfectly recovered by solving a constrained ℓ_1 norm optimization problem. We also presented theoretical bounds on the seismic reflectivity recovery error and on the localization error, based on Earth Q model.

In a time-variant model, we take into account the attenuation and dispersion of the reflected pulses recorded at the geophones on the ground. In this case, $\mathbf{y}^{(l)} \in \mathbb{R}^{L_y \times 1}$, an observed seismic discrete trace of channel l, in the observed seismic 2-D data $\mathbf{Y} \in \mathbb{R}^{L_y \times J}$, is of the form

$$y^{(l)}[k] = \sum_n x^{(l)}[n] g_{\sigma,n}[k - n] + w^{(l)}[k], \quad n \in \mathbb{Z} \tag{10}$$

where $\{g_{\sigma,n}\}$ is a known set of kernels (pulses) corresponding to a possible set of time delays [27]. As stated before, $\sigma > 0$ is a known scaling parameter, and $w^{(l)}[k]$ is an additive noise signal. The shape of each pulse $g_{\sigma,n}$ depends on the time (depth) t_n it corresponds to, and the subterrain characteristics, that can be mathematically described by the Earth Q model [27]–[30]. A brief review of the time-variant pulses model-based estimation can be found in Appendix D.

Alternatively, one can assume a conventional convolution model where the wavelet is time-invariant. In other words, all kernels are identical: $g_{\sigma,n}[k] = g_\sigma[k]\ \forall n$. Namely, in the time-invariant case, each seismic observed trace can be described as

$$y^{(l)}[k] = \sum_n x^{(l)}[n] g[k - n] + w^{(l)}[k], \quad n \in \mathbb{Z} \tag{11}$$

where $g[k]$ is a seismic wavelet of length L_g, and $w^{(l)}[k]$ is an additive noise. Clearly $L_y = L_x + L_g - 1$. The wavelet is assumed to be invariant in both time and space (i.e., both in horizontal and vertical directions). We assume that the seismic signal is free of multiple reflections [31].

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

In the matrix-vector form, we can model the observed 2-D seismic data image $\mathbf{Y} \in \mathbb{R}^{L_y \times J}$ in both cases as

$$\mathbf{Y} = \mathbf{G}\mathbf{X} + \mathbf{W}. \tag{12}$$

Generally, $\mathbf{G}$ is an operator matrix such that $G_{k,n} = g_{\sigma,n}[k-n]$. In the time-invariant case, $\mathbf{G}$ is a convolution matrix of size $L_y \times L_x$ such that $G_{k,n} = g[k-n]$, and $\mathbf{W}$ is an additive i.i.d. white noise matrix independent of $\mathbf{X}$, with zero mean and variance σ_w^2. We do not impose any prior knowledge of the structure or possible patterns in the reflectivity image.

Note that practically, even in a noise-free scenario under the separation condition, $\mathbf{x}^{(l)}$ and $\mathbf{G}$ do not obey the bound guaranteeing neither a unique solution for P_1 nor a stable solution for $P_{1,\varepsilon}$. Namely, in most practical cases we have

$$\|\mathbf{x}^{(l)}\|_0 > \frac{1}{2}\left(1 + \frac{1}{\mu(\mathbf{G})}\right).$$

The mathematical analysis takes a worst case point of view. For this reason, the stability and success guarantees are known to be quite pessimistic. Tighter bounds could have probably been obtained. Yet, this bound is still widely used, perhaps due to its simplicity.

IV. Proposed Method

As described in Section II, one way to look at the seismic inversion problem is to simply solve a CSC problem, trace by trace, using one of the popular methods (such as ISTA, FISTA, LISTA, ada-LISTA, and so on [14]–[17]). Classical pursuit methods, such as ISTA, for example, are guaranteed to recover the true unique solution to the P_0 problem [2] providing that the number of non-zeros per stripe is less than $(1/2)(1 + (1/\mu(\mathbf{D})))$. In real-life scenarios, where we usually consider the $P_{1,\varepsilon}$ problem, stability of the results is also guaranteed under the assumption that $\mathbf{x}$ is sparse enough. Namely, if $\|\mathbf{x}\|_0 < (1/2)(1+(1/\mu(\mathbf{D})))$, then the deviation of the recovery $\hat{\mathbf{x}}$ from the true $\mathbf{x}$ is bounded by [2]

$$\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2 \leq \frac{4\varepsilon^2}{1 - \mu(\mathbf{D})(2\|\mathbf{x}\|_0 - 1)}.$$

Nevertheless, in many practical cases, we observe four major issues inherent to the data.

1) The mutual coherence of the dictionary is relatively high. For example, in the seismic scenario, usually $0.5 < \mu(\mathbf{G}) < 1$ (depending on the sampling rate and the wavelet's scaling σ and on the Q attenuation factor).
2) The signal [or even a signal stripe (or a patch)] is not sufficiently sparse for a successful recovery (see [12]).
3) The ratio between the global maximal and minimal $\mathbf{x}$ values in absolute value ($|\mathbf{x}|_{\max}/|\mathbf{x}|_{\min}$) is high. This in turn leads to longer convergence time and erroneous results due to difficulty to determine the required parameters, especially the thresholds (biases).
4) Recovering the sparse code takes too much time and cannot be applied to real-time applications.

Motivated by these challenges, we suggest a simple modification in the conventional approach. Besides its simplicity, the main advantage of the proposed method is a substantial speed up, without requiring any pre-training. Our approach could also be potentially incorporated in conventional neural nets forward pass.

A. Receptive Field Normalization ITA

Most thresholding algorithms are inherently limited by the challenge of setting a (global or local) threshold that is not inclined toward spikes of strong amplitudes, even when assuming sufficient separation and imposing low mutual coherence. In other words, when projecting the signal on the dictionary, in order to detect the presence of a dictionary atom in the signal (or the residual), we need to choose a threshold that would fit strong spikes as well as small spikes. A threshold that is too small would yield a smeared solution (not sparse enough), whereas a threshold that is too large results in missed spikes (too sparse). To cope with this problem, ISTA [14] repeatedly iterates over the residual term, with a constant yet relatively small threshold, which makes convergence considerably slow. In ISTA, the threshold (λ/c) controls the desired sparsity. To cope with this issue, FISTA incorporates a momentum term in the update step at each iteration [15]. On the other hand, LISTA [16] and Ada-LISTA [17] attempt to learn more suitable weight matrices and use adaptive thresholds.

We propose a different approach. That is, instead of modifying the threshold and/or the dictionary, or having learned different local thresholds for each receptive field (as can be done in CNNs), we normalize the energy of each receptive field, before projecting it on the features space.

Definition 1 (Receptive Field Normalization Kernel): A kernel $h[k]$ can be referred to as a receptive field normalization kernel.

1) If the kernel is positive: $h[k] \geq 0\ \forall k$.
2) If the kernel is symmetric: $h[k] = h[-k]\ \forall k$.
3) If the kernel's global maximum is at its center: $h[0] = 1 \geq h[k]\ \forall k \neq 0$.
4) If the kernel's energy is finite: $\sum_k h[k] < \infty$.

Definition 2 (Receptive Field Normalization): Assuming a receptive field normalization kernel $h[k]$ of odd length L_h, we define the local weighted energy of a time window centered around the kth sample of a 1-D observed data signal $\mathbf{y} \in \mathbb{R}^{L_y \times 1}$

$$\sigma_y[k] \triangleq \left(\sum_{n=-\frac{L_h-1}{2}}^{\frac{L_h-1}{2}} h[n]y^2[k-n]\right)^{\frac{1}{2}}. \tag{13}$$

When $y[k]$ is modeled in accordance to (10), or any other application where $\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{e}$, and $h[k]$ is a receptive field normalization window function of length $L_h \leq L_d$ odd number of samples. For our application, we used a truncated Gaussian-shaped window, but it is possible to use any other window function depending on the application, such as a rectangular window, Epanechnikov window, and so on. The choice of the normalization window and its length affects the choice of the thresholding parameters. If $h[k]$ is a rectangular window, then $\sigma_y[k]$ is simply the ℓ_2 norm of a data stripe centered around the kth location. Otherwise, if the chosen receptive field normalization window is attenuating, then the energy is

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

focused in the center of the receptive field, and possible events at the margins are repressed. The window size is $L_{\mathrm{h}} \leq L_{\mathrm{d}}$ so as to avoid interference of adjacent events as much as possible. Yet, it is recommended not to use a window that is too small. $L_{\mathrm{h}} \geq (L_{\mathrm{d}}/2)$ can serve as a good rule of thumb.

Receptive field normalization is employed by dividing each data point by a local energy measure, before projecting the signal on the dictionary and taking a threshold. Assuming a signal $\mathbf{y}$ and a convolutional dictionary $\mathbf{D}$, an initial solution $\mathbf{x}_0 = \mathbf{0}$ and an initial estimated support $\Delta\mathbf{q}_0 = \mathbf{0}$ at $\theta = 0$. At the first iteration $\theta = 1$, we compute the local variance of $y[k]$ as defined in (13), namely

$$\sigma_y[k] = \sqrt{h[k] * y^2[k]} \tag{14}$$

where $h[k]$ is a receptive field normalization window, and $*$ denotes the convolution operation. Then, we normalize the signal by dividing each data point by the corresponding receptive field energy. In order to avoid amplification of low-energy regions, we use a clipped version of $\sigma_y[k]$. Namely

$$\tilde{\sigma}_y[k] = \begin{cases} \sigma_y[k], & \left|\sigma_y[k]\right| \geq \tau_1 \\ 1, & \left|\sigma_y[k]\right| < \tau_1 \end{cases} \tag{15}$$

where $\tau_1 > 0$ is a predetermined threshold for the first iteration. Empirically, for our application, $0.15 \leq \tau_1 \leq 0.4$ works well. We define the signal normalization weight matrix $\mathbf{W}_0 \in \mathbb{R}^{L_{\mathrm{y}} \times L_{\mathrm{y}}}$

$$\mathbf{W}_0 = \mathrm{diag}\left(\frac{1}{\tilde{\sigma}_y[k]}\right), \quad k = 1, \ldots, L_{\mathrm{y}}. \tag{16}$$

Also, we define the dictionary normalization weight matrix as

$$\mathbf{W}_D = \mathrm{diag}\left(\frac{1}{\sigma_{d_i}}\right), \quad i = 1, \ldots, mL_{\mathrm{x}} \tag{17}$$

where $\sigma_{d_i} = \|\mathbf{d}_i\|_2$ is the ℓ_2 norm of the ith atom. Recall that since the dictionary is convolutional, the ith atom, $i = (k-1)L_{\mathrm{x}} + l$ is the kth filter shifted to the lth support location. Clearly, when the dictionary is a time-invariant convolutional dictionary, $\sigma_{d_{(k-1)L_{\mathrm{x}}+l}} = \sigma_{d_k}\ \forall k \in [1, m]$.

The detected support at the first iteration is

$$\mathbf{q}_1 = \mathcal{I}_{\beta_1}(\mathbf{W}_D\mathbf{D}^T\mathbf{W}_0\mathbf{y}) \tag{18}$$

where $\mathcal{I}$ is an element-wise thresholding indicator function

$$\mathcal{I}_{\beta_1}(x_k) = \begin{cases} 1, & |x_k| \geq \beta_1 \\ 0, & |x_k| < \beta_1. \end{cases} \tag{19}$$

If the dictionary atoms are assumed to be normalized, such that $\sigma_{d_i} = 1\ \forall i$, then of course $\mathbf{W}_D$ is simply an identity matrix, and one can simply ignore $\mathbf{W}_D$ throughout the entire formulation.

It is a well-known fact that according to Cauchy–Schwartz inequality, the correlation coefficient defined as

$$\rho_{ab} \triangleq \frac{\mathbf{a}^T\mathbf{b}}{\|\mathbf{a}\|_2\|\mathbf{b}\|_2} \tag{20}$$

is bounded by one (in absolute value): $|\rho_{\mathrm{ab}}| \leq 1$. Therefore, if we were to divide the inner product $\mathbf{D}^T\mathbf{y}$ by each receptive field's energy and by the corresponding atom's energy, we could detect an atom's sole presence and a perfect match in a time-window of the signal $\mathbf{y}$, when the result is exactly 1. Of course, when several pulses interfere in a single time window, when the mutual coherence is not small enough, the threshold needs to be adjusted accordingly, and a few iterations for corrections may be needed. Choosing an attenuating normalization kernel can repress adjacent spikes. Overall, accurate detection depends on the correlation between neighboring atoms and on neighboring spikes amplitudes, as analytically described in Theorems 1–3.

As stated below, if the support is sufficiently separated, and the mutual coherence is sufficiently small, then the true support is *perfectly* recovered at this stage, within the first iteration. Once the support is recovered there are two simple ways to determine the amplitudes. The first, and more accurate, is simply to solve an LS problem for the subsystem $\mathbf{y} = \mathbf{D}_{K_1}\mathbf{x}_{K_1}$, where $\mathbf{D}_{K_1}$ denotes the partial dictionary matrix having $|K_1|$ columns from the columns of $\mathbf{D}$ with indices in K_1—the estimated support in the first iteration. Alternatively, if one wishes to speed up the algorithm, or in real-life applications where the support is not sufficiently sparse, it is possible to approximate the amplitudes simply as $\mathbf{x}_1 = \mathbf{q}_1 \odot (\mathbf{W}_D^2\mathbf{D}^T\mathbf{y})$, where $\odot$ denotes the Hadamard product.

In the special case of a signature dictionary ($m = 1$), it is possible to approximate the support amplitudes by $x[k] = y[k + \Delta_d]/d_k^p$, where $k \in K_1$ is a support index, $\Delta_d \triangleq (L_{\mathrm{d}} - 1/2)$ is the convolution time shift, and $d_k^p = d_k[k + \Delta_d]$ denotes the corresponding atom's central value. For a signature dictionary of a Ricker wavelet, $d_k^p = \|\mathbf{d}_k\|_\infty = g[0] = 1,\ \forall k$. Empirically, we have not witnessed a significant advantage for the first accurate method over the approximate one, in the case of the signature dictionary. For the sake of brevity, in this section, we assume without loss of generality that the dictionary atoms central value is one, that is, $d_k^p = 1,\ \forall k$.

If the support is sufficiently separated, the algorithm is done at this stage, within one iteration. Otherwise, we can proceed to perform iterative stages as follows. At each iteration, we find the required change in the support by projecting the dictionary on the normalized residual. The residual at iteration θ is

$$\Delta\mathbf{r}_{\theta+1} = \mathbf{y} - \mathbf{D}\mathbf{x}_\theta. \tag{21}$$

We now compute the local weighted variance of the residual term

$$\sigma_{\Delta r}^{\theta+1}[k] = \sqrt{h[k] * \Delta r_{\theta+1}^2[k]} \tag{22}$$

and its clipped version

$$\tilde{\sigma}_{\Delta r}^{\theta+1}[k] = \begin{cases} \sigma_{\Delta r}^{\theta+1}[k], & \left|\sigma_{\Delta r}^{\theta+1}[k]\right| \geq \tau_\theta \\ 1, & \left|\sigma_{\Delta r}^{\theta+1}[k]\right| < \tau_\theta. \end{cases} \tag{23}$$

Once again, we build the corresponding normalization weight matrix

$$\mathbf{W}_\theta = \mathrm{diag}\left(\frac{1}{\tilde{\sigma}_{\Delta r}^{\theta+1}[k]}\right), \quad k = 1, \ldots, L_{\mathrm{y}}. \tag{24}$$

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

Then, we project the dictionary on the normalized residual and threshold the obtained signal

$$\begin{aligned}\Delta\mathbf{q}_{\theta+1} &= \mathcal{I}_{\beta_{\theta+1}}\left(\mathbf{W}_D\mathbf{D}^T\mathbf{W}_\theta\Delta\mathbf{r}_{\theta+1}\right)\\ &= \mathcal{I}_{\beta_{\theta+1}}\left(\mathbf{W}_D\mathbf{D}^T\mathbf{W}_\theta(\mathbf{y}-\mathbf{D}\mathbf{x}_\theta)\right).\end{aligned} \tag{25}$$

The updated solution at iteration $\theta+1$ is

$$\mathbf{x}_{\theta+1} = \Delta\mathbf{q}_{\theta+1}\odot\left(\mathbf{W}_D^2\mathbf{D}^T\Delta\mathbf{r}_{\theta+1}\right)+\mathbf{x}_\theta \tag{26}$$

where $\odot$ denotes the Hadamard product. Note that the algorithm should converge in a few iterations. About 2–5 iterations should be enough. To ensure stable convergence, incorporating a momentum is also possible

$$\mathbf{x}_{\theta+1} = \alpha_{\mathrm{r}}\left(\Delta\mathbf{q}_{\theta+1}\odot\left(\mathbf{W}_D^2\mathbf{D}^T\Delta\mathbf{r}_{\theta+1}\right)\right)+\mathbf{x}_\theta \tag{27}$$

where $0<\alpha_{\mathrm{r}}\leq 1$ is a constant step size. In practical cases, when the data is not sufficiently separated and the mutual coherence is relatively high, an exact stopping rule may be hard to determine and solutions may be unstable. In these cases, early stopping after a predetermined small number of iterations is recommended. Note that as opposed to ISTA, here the shrinkage operator is applied only to the projection on the residual, without formerly adding it to the previous solution. As previously stated, in the case of a signature dictionary, it is possible to use

$$\mathbf{x}_{\theta+1} = \alpha_{\mathrm{r}}(\Delta\mathbf{q}_{\theta+1}\odot\Delta\mathbf{r}_{\theta+1})+\mathbf{x}_\theta \tag{28}$$

instead of (27).

From a neural network perspective, receptive-field normalization (RFN) increases the sensitivity of a neuron to its receptive field. In some cases, where the energy of receptive fields in different locations is unbalanced, some atoms may be strongly expressed, whereas others are significantly weaker. RFN overcomes this obstacle, without having to set local or adaptive thresholds. This way, the activation of a neuron is independent of the scaling of other events in the signal. In other words, the neuron is able to detect a feature (reflection), even if its energy is relatively low, comparing to other events in the data.

Note that τ_θ essentially determines the minimal $|\mathbf{x}|_{\min}$ that can be detected at each of the above steps. It should be determined taking into account the noise expected level. Namely, since we usually assume the nuisance noise is uncorrelated with the signal, therefore

$$E\|\mathbf{y}\|_2^2 = E\|\mathbf{D}\mathbf{x}\|_2^2+E\|\mathbf{e}\|_2^2$$

where E denotes mathematical expectation. In order to avoid noise amplification, we would set τ_θ such that

$$\tau_\theta \geq |\mathbf{x}|_{\min}\min_i\|\mathbf{d}_i\|_2+\varepsilon_{\mathrm{d}} \tag{29}$$

where ε_{d} denotes the noise ℓ_2 norm over a data stripe of length L_{h}.

We project the dictionary on the normalized signal though it is possible to normalize the projection $\mathbf{D}^T(\mathbf{y}-\mathbf{D}\mathbf{x}_\theta)$ instead. Yet, the two options are not equivalent. Normalization of the signal prior to projection as in (24) rescales each sample and therefore might cause some distortion to the signal. It is more suitable for admissible kernels (see [27, Definition 2.1]), where most of the kernel's energy is focused at its center. On the other hand, it promotes muting of close spikes and decreases noise influence especially when dealing with relatively small environments.

Algorithm 1 Receptive Field Normalization ITA

input : signal $\mathbf{y}$, dictionary $\mathbf{D}$
Init: $\mathbf{x}_0=\mathbf{0}$, $\Delta\mathbf{q}_0=\mathbf{0}$, $\theta=0$
compute: $\mathbf{W}_{\mathrm{D}}=\mathrm{diag}\left(\sigma_{d_i}^{-1}\right), i=1,\ldots,mL_{\mathrm{x}}$.
while $\|\mathbf{x}_{\theta+1}-\mathbf{x}_\theta\|_2<\delta$ *or* $\theta\leq N_{\mathrm{it}}$ **do**
  $\Delta\mathbf{r}_{\theta+1}=\mathbf{y}-\mathbf{D}\mathbf{x}_\theta$
  compute:

$$\tilde{\sigma}_{\Delta r}^{\theta+1}[k]=\begin{cases}\sigma_{\Delta r}^{\theta+1}[k], & \left|\sigma_{\Delta r}^{\theta+1}[k]\right|\geq\tau_\theta\\ 1, & \left|\sigma_{\Delta r}^{\theta+1}[k]\right|<\tau_\theta\end{cases}$$

  $\mathbf{W}_\theta=\mathrm{diag}\left(\tilde{\sigma}_{\Delta r}^{\theta+1}[k]\right)^{-1}, k=1,\ldots,L_{\mathrm{y}}$
  $\Delta\mathbf{q}_{\theta+1}=\mathcal{I}_{\beta_{\theta+1}}(\mathbf{W}_{\mathrm{D}}\mathbf{D}^T\mathbf{W}_\theta\Delta\mathbf{r}_{\theta+1})$
  solve the subsystem $\Delta\mathbf{r}_{\theta+1}=\mathbf{D}_{K_{\theta+1}}\Delta\mathbf{x}_{\theta+1}$,
  $\mathbf{D}_{K_{\theta+1}}$ - partial dictionary matrix corresponding to the support $\Delta\mathbf{q}_{\theta+1}$.
  $\mathbf{x}_{\theta+1}=\alpha_{\mathrm{r}}\Delta\mathbf{x}_{\theta+1}+\mathbf{x}_\theta$
  $\theta\leftarrow\theta+1$
end

Algorithm 2 Approximate Receptive Field Normalization ITA

input : signal $\mathbf{y}$, dictionary $\mathbf{D}$
Init: $\mathbf{x}_0=\mathbf{0}$, $\Delta\mathbf{q}_0=\mathbf{0}$, $\theta=0$
compute: $\mathbf{W}_{\mathrm{D}}=\mathrm{diag}\left(\sigma_{d_i}^{-1}\right), i=1,\ldots,mL_{\mathrm{x}}$.
while $\|\mathbf{x}_{\theta+1}-\mathbf{x}_\theta\|_2<\delta$ *or* $\theta\leq N_{\mathrm{it}}$ **do**
  $\Delta\mathbf{r}_{\theta+1}=\mathbf{y}-\mathbf{D}\mathbf{x}_\theta$
  compute:

$$\tilde{\sigma}_{\Delta r}^{\theta+1}[k]=\begin{cases}\sigma_{\Delta r}^{\theta+1}[k], & \left|\sigma_{\Delta r}^{\theta+1}[k]\right|\geq\tau_\theta\\ 1, & \left|\sigma_{\Delta r}^{\theta+1}[k]\right|<\tau_\theta\end{cases}$$

  $\mathbf{W}_\theta=\mathrm{diag}\left(\tilde{\sigma}_{\Delta r}^{\theta+1}[k]\right)^{-1}, k=1,\ldots,L_{\mathrm{y}}$
  $\Delta\mathbf{q}_{\theta+1}=\mathcal{I}_{\beta_{\theta+1}}(\mathbf{W}_{\mathrm{D}}\mathbf{D}^T\mathbf{W}_\theta\Delta\mathbf{r}_{\theta+1})$
  $\mathbf{x}_{\theta+1}=\alpha_{\mathrm{r}}\left(\Delta\mathbf{q}_{\theta+1}\odot\left(\mathbf{W}_{\mathrm{D}}^2\mathbf{D}^T\Delta\mathbf{r}_{\theta+1}\right)\right)+\mathbf{x}_\theta$
  $\theta\leftarrow\theta+1$
end

A summary of the proposed methods is presented in Algorithms 1–3.

Algorithm 4 shows a further simplification of the proposed method. Namely, after computing the locally normalized signal $\tilde{\mathbf{y}}\triangleq\mathbf{W}_0\mathbf{y}$ at the first iteration, we propagate through the iterations without computing the spikes amplitude and without normalizing again, in order to estimate only the support. When the support is fully revealed, we calculate the weights by solving an LS problem or an approximation as described above. This variation can be used if more speed is necessary, on the expense of accuracy, or for implementing a learned version as discussed in Section VII.

B. Theoretical Analysis for the RFN-Thresholding With a Convolutional Sparse Model

Consider a stripe of length L_{h} around some index i and the corresponding atoms shifts around this location in the signal.

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

Algorithm 3 Approximate Receptive Field Normalization ITA for a Signature Dictionary

input : signal $\mathbf{y}$, dictionary $\mathbf{D}$
Init: $\mathbf{x}_0 = \mathbf{0},\ \Delta\mathbf{q}_0 = \mathbf{0},\ \theta = 0$
compute: $\mathbf{W}_{\mathrm{D}} = \mathrm{diag}\big(\sigma_{d_i}^{-1}\big), i = 1, \ldots, mL_{\mathrm{x}}$.
while $\|\mathbf{x}_{\theta+1} - \mathbf{x}_\theta\|_2 < \delta$ *or* $\theta \le N_{\mathrm{it}}$ **do**
  $\Delta\mathbf{r}_{\theta+1} = \mathbf{y} - \mathbf{D}\mathbf{x}_\theta$
  compute:

$$\tilde{\sigma}_{\Delta r}^{\theta+1}[k] = \begin{cases} \sigma_{\Delta r}^{\theta+1}[k], & \big|\sigma_{\Delta r}^{\theta+1}[k]\big| \ge \tau_\theta \\ 1, & \big|\sigma_{\Delta r}^{\theta+1}[k]\big| < \tau_\theta \end{cases}$$

  $\mathbf{W}_\theta = \mathrm{diag}\big(\tilde{\sigma}_{\Delta r}^{\theta+1}[k]\big)^{-1}, k = 1, \ldots, L_{\mathrm{y}}$
  $\Delta\mathbf{q}_{\theta+1} = \mathcal{I}_{\beta_{\theta+1}}(\mathbf{W}_{\mathrm{D}}\mathbf{D}^T\mathbf{W}_\theta\,\Delta\mathbf{r}_{\theta+1})$
  $\mathbf{x}_{\theta+1} = \alpha_{\mathrm{r}}(\Delta\mathbf{q}_{\theta+1} \odot \Delta\mathbf{r}_{\theta+1}) + \mathbf{x}_\theta$
  $\theta \leftarrow \theta + 1$
end

Algorithm 4 Support Detection Approximate Receptive Field Normalization ITA for a Signature Dictionary

input : signal $\mathbf{y}$, dictionary $\mathbf{D}$
Init: $\mathbf{x}_0 = \mathbf{0},\ \Delta\mathbf{q}_0 = \mathbf{0},\ \mathbf{q}_0 = \mathbf{0},\ \theta = 0$
compute:

$$\tilde{\sigma}_y[k] = \begin{cases} \sigma_y[k], & |\sigma_y[k]| \ge \tau_1 \\ 1, & |\sigma_y[k]| < \tau_1 \end{cases}$$

$\mathbf{W}_{\mathrm{D}} = \mathrm{diag}\big(\sigma_{d_i}^{-1}\big),\ i = 1, \ldots, mL_{\mathrm{x}}$.
$\mathbf{W}_0 = \mathrm{diag}\big(\tilde{\sigma}_y^{-1}[k]\big),\ k = 1, \ldots, L_{\mathrm{y}}$,
$\tilde{\mathbf{y}} = \mathbf{W}_0\mathbf{y}$
while $\|\Delta\tilde{\mathbf{y}}_{\theta+1} - \Delta\tilde{\mathbf{y}}_\theta\|_2 < \delta$ *or* $\theta \le N_{\mathrm{it}}$ **do**
  $\Delta\tilde{\mathbf{y}}_{\theta+1} = \tilde{\mathbf{y}} - \mathbf{D}\mathbf{q}_\theta$
  $\Delta\mathbf{q}_{\theta+1} = \mathcal{I}_{\beta_{\theta+1}}(\mathbf{W}_{\mathrm{D}}\mathbf{D}^T\,\Delta\tilde{\mathbf{y}}_{\theta+1})$
  $\mathbf{q}_{\theta+1} = \alpha_{\mathrm{r}}\,\Delta\mathbf{q}_{\theta+1} + \mathbf{q}_\theta$
  $\theta \leftarrow \theta + 1$
end
$\hat{\mathbf{x}} = \hat{\mathbf{q}} \odot \mathbf{y}$

Hereafter, the ith entry of a vector $\mathbf{v}$ is denoted by $\mathbf{v}[i]$, and a stripe cropped around the ith index is denoted by $\mathbf{v}_i$. Note that due to the convolution properties of the dictionary, a data stripe $\mathbf{y}_i \in \mathbb{R}^{L_{\mathrm{h}}\times 1}$ is affected from input spikes in $\mathbf{x}_i \in \mathbb{R}^{mL_{\mathrm{s}}\times 1}$, such that $L_{\mathrm{s}} = L_{\mathrm{h}} + L_{\mathrm{d}} - 1$. To shed some light on the theoretical aspects of the proposed method, let us define the local sparsity of a stripe of $\mathbf{x}$ as the maximum number of non-zeros weights corresponding to a data stripe of length L_{h} at location i

$$s = \|\mathbf{x}\|_{0,\infty}^{S_h^i} = \max_i \|\mathbf{x}_i\|_0 = \max_i \sum_{l\in S_h^i} \mathcal{I}_0(\mathbf{x}[l]) \tag{30}$$

where S_h^i denotes a neighborhood of mL_{s} point values corresponding to L_{h} data points symmetrically distributed around index i and redefining $\mathcal{I}_0(0) = 0$. Let us rephrase (13) in the matrix-vector form

$$\sigma_{\mathrm{y}}[i] = \|\mathbf{H}\mathbf{y}_i\|_2 \tag{31}$$

where $\mathbf{y}_i$ denotes a stripe of length L_{h} around some index $i \in [1, L_{\mathrm{y}}]$, that is, $\mathbf{y}_i = [y[i-(L_{\mathrm{h}}-1/2)], \ldots, y[i+(L_{\mathrm{h}}-1/2)]]^T$ and $\mathbf{H} = \mathrm{diag}(h^{(1/2)})$ (ignoring boundary issues). Throughout the proofs, we assume $\sigma_{\mathrm{y}}[i] > \tau\ \forall i \in [1, L_{\mathrm{y}}]$.

Theorem 1 (Support Recovery Using RFN in the Presence of Noise): Let $\mathbf{y} = \mathbf{D}\mathbf{x} + \mathbf{e}$, where $\mathbf{D}$ is a convolutional dictionary, with normalized atom's variance $\|\mathbf{d}_i\|_2 = \sigma_{\mathbf{d}} = 1\ \forall i \in [1, mL_{\mathrm{x}}]$. Assume a rectangular receptive field normalization kernel, that is, $\mathbf{h}[k] = 1$, $\mathbf{h} \in \mathbb{R}^{L_{\mathrm{h}}\times 1}$, such that $L_{\mathrm{h}} = L_{\mathrm{d}}$. In other words, $\mathbf{H}$ simply extracts a local data stripe.

1) Assuming that

$$\frac{|\mathbf{x}_i|_{\min}}{|\mathbf{x}_i|_{\max} + \frac{\varepsilon_{\mathrm{d}}}{s}} > \frac{s\mu}{1+\mu}\left(1 + \frac{\sqrt{s}}{\sqrt{1-(s-1)\mu} - \tilde{\varepsilon}_{\mathrm{d},\infty}}\right) + \frac{2s\varepsilon_{\mathrm{s}}}{1+\mu} \quad \forall i \in K \tag{32}$$

where $|\mathbf{x}_i|_{\max}$ and $|\mathbf{x}_i|_{\min}$ are the highest and the lowest entries in a stripe $\mathbf{x}_i$, respectively, $\varepsilon_{\mathrm{d}} \triangleq \|\mathbf{e}_i\|_2$, $\tilde{\varepsilon}_{\mathrm{d},\infty} \triangleq (\varepsilon_{\mathrm{d}}/|\mathbf{x}|_{\min})$ and $\varepsilon_{\mathrm{s}} \triangleq (\varepsilon_{\mathrm{d}}/\tau)$.

2) The threshold β_1 is chosen according to

$$\frac{(1+\mu)|\mathbf{x}_i|_{\min}}{s|\mathbf{x}_i|_{\max} + \varepsilon_{\mathrm{d}}} - \mu - \varepsilon_{\mathrm{s}} > \beta_1 > \frac{\sqrt{s}\mu}{\sqrt{1-(s-1)\mu} - \tilde{\varepsilon}_{\mathrm{d},\infty}} + \varepsilon_{\mathrm{s}}. \tag{33}$$

Then, the support of $\mathbf{x}$ is perfectly recovered with RFN thresholding

$$\mathrm{Supp}(\mathbf{x}_1) = \mathrm{Supp}(\mathbf{x})$$

where $\mathrm{Supp}(\cdot)$ is the support of a vector, and $\mathbf{x}_1$ is the recovered sparse vector within one step of RFN thresholding (or one iteration of Algorithms 1–4).

Proof: See Appendix A.

Consequently, in the noise-free model $\mathbf{y} = \mathbf{D}\mathbf{x}$, the true support is perfectly recovered within one iteration of Algorithms 1–4 provided as follows.

1)

$$\frac{|\mathbf{x}_i|_{\min}}{|\mathbf{x}_i|_{\max}} > \frac{s\mu}{1+\mu}\left(1 + \frac{\sqrt{s}}{\sqrt{1-(s-1)\mu}}\right) \quad \forall i \in K \tag{34}$$

where $|\mathbf{x}_i|_{\max}$ and $|\mathbf{x}_i|_{\min}$ are the highest and the lowest entries in a stripe $\mathbf{x}_i$, respectively.

2) The global threshold β_1 is chosen according to

$$\frac{1+\mu}{s}\left(\min_{i\in K}\frac{|\mathbf{x}_i|_{\min}}{|\mathbf{x}_i|_{\max}}\right) - \mu > \beta_1 > \frac{\sqrt{s}\mu}{\sqrt{1-(s-1)\mu}}. \tag{35}$$

The conditions (32) and (34) may appear too strict at the first glance, but note that we are considering the *local* ratio $(|\mathbf{x}_i|_{\min}/|\mathbf{x}_i|_{\max})$ rather than the *global* ratio $(|\mathbf{x}|_{\min}/|\mathbf{x}|_{\max})$. Also the bound is far from being tight since in practice

$$|\mathbf{a}_i^T\mathbf{d}_j| \ll \frac{\mu}{\sigma_y[i]} \quad \forall\{(i,j) : i \ne j, i \in K, j \notin K\}$$

and

$$|\mathbf{a}_i^T\mathbf{e}| \ll \varepsilon_{\mathrm{s}} \quad \forall i$$

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

where we denote $\mathbf{a}_i = \mathbf{W}_0\mathbf{d}_i$. Good support recovery can be achieved in practical cases with sufficient separation for much higher values of μ, s, and $(|\mathbf{x}_i|_{\min}/|\mathbf{x}_i|_{\max})$, as demonstrated in Section V.

Stable recovery of hard thresholding in the noiseless case is acquired as long as [2]

$$\min_{i\in K} |\mathbf{d}_i^T\mathbf{y}| > \max_{j\notin K} |\mathbf{d}_j^T\mathbf{y}|. \tag{36}$$

Leading to

$$|\mathbf{x}|_{\min} - (s-1)\mu|\mathbf{x}|_{\max} > \beta_1 > s\mu|\mathbf{x}|_{\max}. \tag{37}$$

In other words,

$$\frac{|\mathbf{x}|_{\min}}{|\mathbf{x}|_{\max}} > (2s-1)\mu. \tag{38}$$

Suppose $s = 1$, meaning that there can only be one atom in each stripe, than we have

$$\frac{|\mathbf{x}|_{\min}}{|\mathbf{x}|_{\max}} > \mu. \tag{39}$$

Practically, in many convolutional models, if the stride is not large enough, this condition is not met, leading us to the following theorem.

Theorem 2 (Support Recovery Using RFN Under Sufficient Separation): Perfect recovery in the noiseless case where $s = 1$ with RFN is guaranteed for any ratio

$$0 < \frac{|\mathbf{x}|_{\min}}{|\mathbf{x}|_{\max}} \leq 1.$$

Proof: See Appendix B.

It is important to emphasize that in most practical cases, even if the above assumptions do not hold, it does not mean the support detection will fail. It solely means that we cannot guarantee, by means of the following proofs, perfect support detection at the first iteration.

Denote $\mathbf{D}_i \in \mathbb{R}^{L_{\mathrm{h}}\times mL_{\mathrm{s}}}$ as the submatrix partial dictionary yielding point $\mathbf{y}[i]$, that is obtained by restricting the dictionary $\mathbf{D}$ to the support of mL_{s} samples equally distributed around point i.

Theorem 3 (Support Recovery With Attenuating RFN Kernel in the Noiseless Case): Let $\mathbf{y} = \mathbf{D}\mathbf{x}$, where $\mathbf{D}$ is a convolutional dictionary, with normalized atom's variance $\|\mathbf{d}_i\| = \sigma_d = 1\ \forall i$. Assume a symmetric strictly decreasing receptive field normalization kernel, that is, $h[|k|] > h[|k+1|]$ (e.g., a truncated Gaussian window), of length $L_{\mathrm{h}} = L_{\mathrm{d}}$, such that $\mu(\mathbf{H}\mathbf{D}_i) \ll \mu(\mathbf{D}),\ \forall i \in [1, L_{\mathrm{y}}]$.

1) Assuming that K obeys the minimal separation condition with a separation constant ν.
2)

$$\frac{\sqrt{s}\mu}{h_{\mathrm{d,min}}} < \beta_1 < 1 - (h_d(\nu) + \mu) \times \frac{\|\mathbf{x}_{-i}\|_1}{|\mathbf{x}[i]| + h_d(\nu)\|\mathbf{x}_{-i}\|_1} \quad \forall i \in K \tag{40}$$

where $h_d(\nu) \triangleq \max_{p\in[1,m]} \mathbf{H}_d[\Delta_k+(p-1)L_{\mathrm{s}}, \Delta_k+(p-1)L_{\mathrm{s}}]$, such that $\mathbf{H}_d$ is a diagonal matrix holding the attenuated atoms ℓ_2 norms at the support around some point data at index i. Namely, $\mathbf{H}_d[k,k] = \|\mathbf{H}\mathbf{d}_k^i\|_2,\ k \in S_i^h$, where $\mathbf{d}_k^i \in \mathbb{R}^{L_{\mathrm{h}}\times 1}$ denotes the kth atom of the submatrix $\mathbf{D}_i$. In other words, $h_d(\nu)$ is the maximal ℓ_2 norm of an atom's shift at the minimal separation distance multiplied by the RFN window. Recall that we defined Δ_k as the minimal number of samples between adjacent events. On the other hand, $h_{\mathrm{d,min}} \triangleq \min_k \mathbf{H}_d[k,k]$ is the lowest ℓ_2 norm of a shifted atom multiplied by the RFN window centered around some point i, and $\|\mathbf{x}_{-i}\|_1 \triangleq \sum_{t\in S_h^i, t\neq i} |\mathbf{x}[t]| = \|\mathbf{x}_i\|_1 - |\mathbf{x}[i]|$ is the sum of weights affecting the ith stripe, besides the spike at location i. Then, the support of $\mathbf{x}$ is perfectly recovered with RFN thresholding

$$\mathrm{Supp}(\mathbf{x}_1) = \mathrm{Supp}(\mathbf{x})$$

where $\mathrm{Supp}(\cdot)$ is the support of a vector, and $\mathbf{x}_1$ is the recovered sparse vector within one iteration of Algorithms 1–4.

Proof: See Appendix C.

The suggested approach and the above theorems shed light on the role of spatial stride and maxpooling in CNNs [24]. When using a spatial stride, we convolve an input signal is a certain layer, while skipping a fixed number of spatial locations, primarily in an attempt to reduce the computational burden. Consequently, the mutual coherence of the stride convolutional dictionary is smaller, guaranteeing more accuracy and less redundancy at the activation stage. Alternatively, when the atom's shifts are small, usually due to a high sampling rate, the mutual coherence of the convolutional dictionary is relatively high. Naturally, the projection of a shifted atom on the signal may yield strong values in a small area around its true location in the signal. In this case, thresholding can lead to a group of spikes around each true spike. In their work, Gregor and Lecun [16] refer to these issues as "mutual inhibition and explaining away." One possible way to address this issue in CNNs is by maxpooling, namely by taking into account only the maximal or average value in a small local samples neighborhood. In the above method, we propose another possible way to address this issue. That is, one can suppress the strong expressiveness of one component over the other by using an attenuating normalization window, such as a Gaussian window (as opposed to an averaging window) and by fine-tuning of the applied thresholds.

The proposed method is closely related to local response normalization (LRN) [32]. LRN normalizes each neuron's response with respect to the sum of the squared responses of adjacent kernel maps at the same spatial position. The idea, inspired by natural processes in biological neurons, is that strong neuron responses will inhibit weaker neurons responses. In terms of neural nets training, LRN reduces feature maps similarity and encourages competitiveness, which in turn can improve generalization. Practically, the main difference is that we propose to locally normalize the input signal, with respect to spatially close inputs, before applying a filter, in an attempt to scale all responses to the same range and to prevent stronger spikes from "casting" a shadow over weaker spikes, both locally and globally, that is, across all of the signal.

V. Experimental Results

In Sections V-A and V-B, we provide synthetic and real data examples demonstrating the performance of the proposed

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

TABLE I

SYNTHETIC EXAMPLE—RESULTS AND PARAMETERS: WAVELET'S SCALING ω_0, MINIMAL SEPARATION CONSTANT ν, THRESHOLDS β_1 AND β_2, NORMALIZATION WINDOW SIZE L_h, NORMALIZATION WINDOW STD σ_h, RECOVERED REFLECTIVITY SCORE AT THE FIRST ITERATION $\rho^1_{\mathbf{X},\hat{\mathbf{X}}}$, FINAL RECOVERED REFLECTIVITY SCORE $\rho_{\mathbf{X},\hat{\mathbf{X}}}$, AND AVERAGE NUMBER OF ITERATIONS M_{it}

ω_0	ν	β_1	β_2	L_{h}	σ_{h}	$\rho^1_{\mathbf{X},\hat{\mathbf{X}}}$	$\rho_{\mathbf{X},\hat{\mathbf{X}}}$	M_{it}
80π (40 Hz)	5	0.95	0.88	11	2	0.97	**0.995**	2.58
80π (40 Hz)	3	0.95	0.87	11	2	0.92	**0.97**	2.64
80π (40 Hz)	1	0.8	0.66	9	2	0.81	**0.89**	3.6
50π (25 Hz)	5	0.98	0.98	17	3	0.93	**0.985**	2.19
50π (25 Hz)	3	0.98	0.87	17	4	0.83	**0.9**	2.38

technique. Hereafter, we refer to the proposed method as RFN-ITA.

A. Synthetic Data

In order to verify the robustness of the proposed method, we conducted extensive experiments in different scenarios.

First, we generated $J = 1000$ random reflectivity independent sequences of length $L_{\text{x}} = 60$. Each reflectivity column is modeled as a zero-mean Bernoulli–Gaussian process [33], that is, reflectors times (depths) are independently distributed with Bernoulli probability p, and reflectors amplitudes are normally distributed with mean μ and variance σ^2. Mathematically, each reflectivity sequence $x^{(l)}[k]$ is described as

$$x^{(l)}[k] = r^{(l)}[k]q^{(l)}[k], \quad j = 1, \ldots, J$$

where $r^{(l)}[k] \sim \mathcal{N}(\mu, \sigma_r^2)$ and $q^{(l)}[k] \sim B(p)$. Clearly, p determines the degree of sparsity. In the following experiments, $\sigma_r = 3$, and $0.1 \leq p \leq 0.4$, depending on the chosen minimal separation Δ_k—the minimal separation between two spikes. Here, Δ_k does not necessarily obey the minimal separation condition.

We assume a source waveform $g(t)$ defined as the real-valued Ricker wavelet

$$g(t) = \left(1 - \frac{1}{2}\omega_0^2 t^2\right) \exp\left(-\frac{1}{4}\omega_0^2 t^2\right) \tag{41}$$

where ω_0 is the dominant radial frequency [28], determining the width of the pulse. The seismic traces are calculated according to (11), in a time-invariant noise-free environment, with sampling interval $T_{\text{s}} = 4$ ms. Since in this case, $\mathbf{G}$ is a signature dictionary, we implemented Algorithm 3 using a truncated zero-mean Gaussian window of length L_{h}, with variance σ_h^2. The thresholds β_1 and β_2 are explicitly determined, and for the rest of the iterations, we set $\beta_l = 0.5\beta_{l-1}$, where l is the iteration number. The algorithms is stopped when $\|\mathbf{x}_l - \mathbf{x}_{l-1}\|_2 < 10^{-4}$. We allow up to four iterations ($N_{\text{it}} = 4$). The step size $\alpha_{\text{r}} = 0.5$ is constant for all experiments.

As a figure of merit, we use the correlation coefficient between the true reflectivity and the recovered reflectivity

$$\rho_{\mathbf{X},\hat{\mathbf{X}}} = \frac{\mathbf{X}_{\text{cs}}^T \hat{\mathbf{X}}_{\text{cs}}}{\|\mathbf{X}_{\text{cs}}\|_2 \|\hat{\mathbf{X}}_{\text{cs}}\|_2}$$

where $\mathbf{X}_{\text{cs}}$ and $\hat{\mathbf{X}}_{\text{cs}}$ are column-stack vectors of the true reflectivity image and the recovered reflectivity image, respectively. Table I presents the estimated reflectivities score at the first iteration and at the final iteration, and the average number of iterations per trace with respect to the wavelets width, the minimal separation distances and the algorithm parameters $\beta_1, \beta_2, L_{\text{h}}$, and σ_h. Usually β_1 is close to 1, depending on the minimal separation. Notice that in these experiments, the mutual coherence is relatively high: $\mu(\mathbf{G}) = 0.764, 0.585$ for $\omega_0 = 50\pi, 80\pi$, respectively.

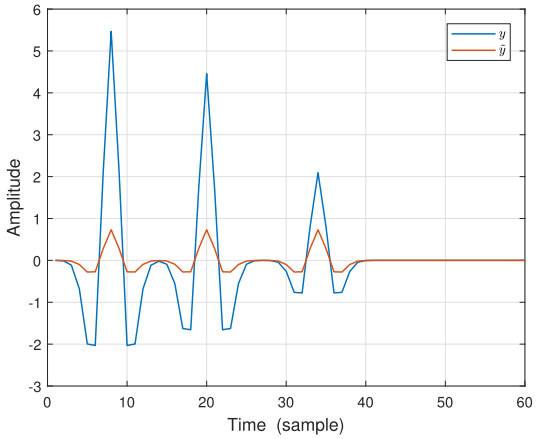

Fig. 1. Synthetic data receptive field normalization example: the seismic trace is depicted in the blue line and the seismic trace after applying RFN, with an averaging window of size $L_{\text{h}} = 15$.

Fig. 1 presents a simple example of a trace after partial RFN $\tilde{\mathbf{y}} = \mathbf{W}_0\mathbf{y}$, with $\omega_0 = 80\pi$, and $\Delta_k = 5$ samples. In this example, the pulses are intentionally completely separated. As can be clearly seen, when the pulses are sufficiently separated, the normalized signal is perfectly balanced regardless of the original local energy. Also, the pulses shape is preserved. Therefore, beyond the theoretical analysis, it can be intuitively comprehended why given sufficient separation, all spikes locations can be detected by simply projecting the dictionary on the normalized signal and applying a threshold, without considering the ratio between the minimal and the maximal coefficients in the absolute value in $\mathbf{x}^{(l)}$. Hence, all iterations beyond the first one are required only when reflectors are insufficiently separated with respect to the wavelet's length and the mutual coherence.

Fig. 2(a) presents an example of a reflectivity channel of length $L_{\text{x}} = 110$ samples ($T_{\text{s}} = 4$ ms), with $p = 0.4$, $\sigma_r = 3$, and a minimal separation distance of $\Delta_k = 3$ samples. The corresponding seismic trace with $\omega_0 = 80\pi$ is depicted

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

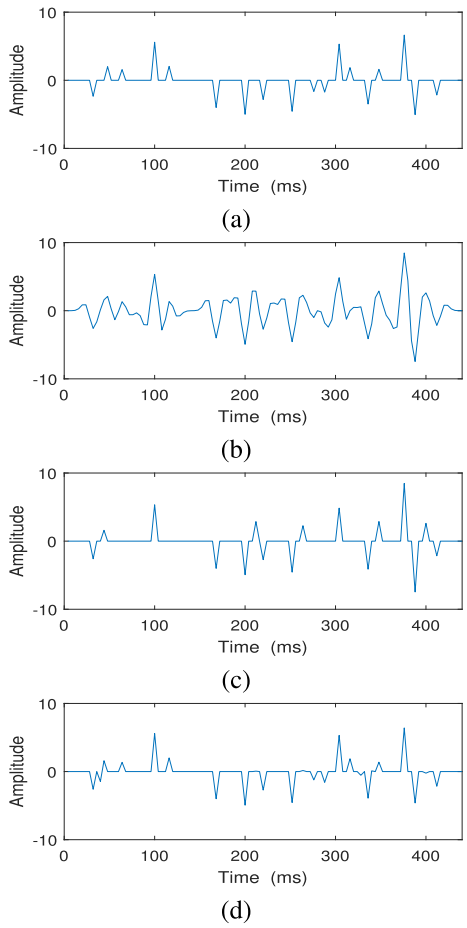

Fig. 2. One-dimensional synthetic test. (a) True reflectivity. (b) Synthetic trace with 40 Hz Ricker wavelet. (c) Recovered 1-D channel reflectivity signal, first iteration $\rho^1_{\mathbf{X},\hat{\mathbf{X}}} = 0.91$. (d) Recovered 1-D channel of reflectivity signal $\rho_{\mathbf{X},\hat{\mathbf{X}}} = 0.98$.

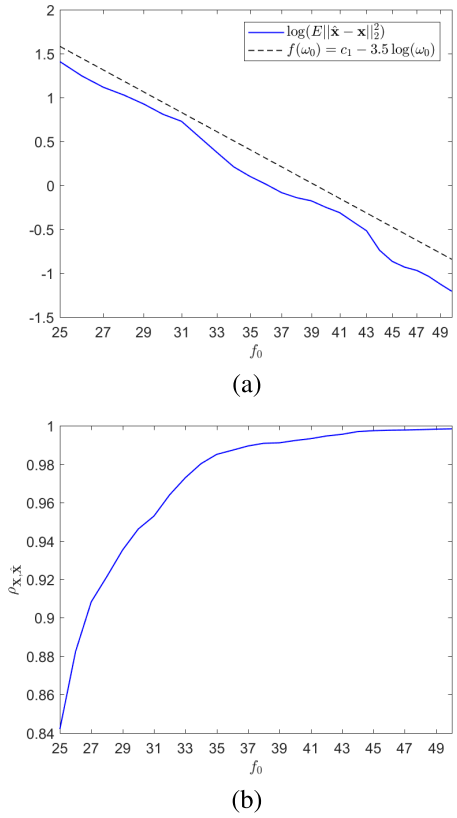

Fig. 3. Reflectivity estimation as a function of the dominant frequency of the Ricker wavelet, SNR = 40 dB. (a) The blue line corresponds to the log of the mean square error $\log(E\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2)$ as a function of $\log(\omega_0)$. The black-dashed line corresponds to $f(\omega_0) = c_1 - 3.5\log(\omega_0)$, where $c_1 = 8F_s^2(L_x)^{1/2}\sigma_n/\beta_g$ is a constant. (b) The correlation coefficient $\rho_{\mathbf{X},\hat{\mathbf{X}}}$ as a function of the wavelet dominant frequency f_0 (Hz).

in Fig. 2(b), and the corresponding estimated reflectivity with $\beta_1 = 0.95, \beta_l = 0.9, 2 \le l \le 4$ and step size $\alpha_r = 1$ is presented in Fig. 2(d). The estimated reflectivity in the first iteration is presented in Fig. 2(c).

Fig. 3 shows how well one can estimate the reflectivity depending on the dominant frequency of the Ricker wavelet $\omega_0 = 2\pi f_0$ ranging from 25 to 50 Hz. Here, we used a reflectivity of $J = 1200$ channels of length $L_x = 60$ samples ($T_s = 4$ ms), with $p = 0.4$, $\sigma_r = 3$, and a minimal separation distance of $\Delta_k = 5$ samples. The seismic data was further destructed by additive noise with SNR = 40 dB. Here, we used a Gaussian RFN window of length $L_h = 11$ and $\sigma_h = 2$. We allow up to four iterations where $\beta_1 = 1.22 - 0.01(f_0 - 25)$, $\beta_2 = \beta_1 + 0.2$, $\beta_l = 0.5\beta_{l-1}, l = 3, 4$, and $\alpha_r = 0.5$. In Fig. 3(a) the blue line corresponds to the log of the mean square error $\log(E\|\mathbf{x}-\hat{\mathbf{x}}\|_2^2)$ as a function of $\log(\omega_0)$. The black dashed line corresponds to $f(\omega_0) = c_1 - 3.5\log(\omega_0)$, where $c_1 = 8F_s^2(L_x)^{1/2}E\|\mathbf{e}\|_2^2/\beta_g$ is a constant, β_g is a parameter that characterizes the concavity of the wavelet as defined in [27, Definition 2.1]. As can be observed, the mean square error $E\|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$ decreases by a rate of $\omega_0^{3.5}$. In [34, Th. 1], we prove that under the separation condition, the mean square error is inversely proportional to ω_0^2. Hence, the estimation error produced by the proposed algorithm decreases faster than expected, with respect to the wavelet's frequency, even when the separation condition is not satisfied. Fig. 3(b) depicts the correlation coefficient $\rho_{\mathbf{X},\hat{\mathbf{X}}}$ as a function of the wavelet dominant frequency f_0 (Hz).

VI. Real Data

We applied the proposed method, to real seismic data from a small land 3-D survey (courtesy of GeoEnergy Inc., TX, USA). A small 2-D seismic image that consists of 400 traces is used for demonstration. Each trace is 1.2 s in duration (300 time samples), with 4 ms sampling rate. The seismic image is shown in Fig. 4(a). As can be observed, the data is non-stationary, reflectors are closely spaced and not sufficiently separated, and the reflector's amplitudes are unbalanced. Assuming an initial Ricker wavelet with $\omega_0 = 80\pi$ (40 Hz), we estimated $Q = 200$ as described in [36] and derived the set of time-variant pulses $\{g_{\sigma,n}\}$ (see [27], Appendix D). The estimated reflectivity image and the corresponding reconstructed seismic image using RFN-ITA, at the first iteration and at the second (and final) iteration are depicted in Figs. 4 and 5, respectively. For the first iteration, we set $\beta_1 = 1$ and $\tau_1 = 0.4$. For the second iteration, we set $\beta_2 = 0.7$, $\tau_2 = 1$, and

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

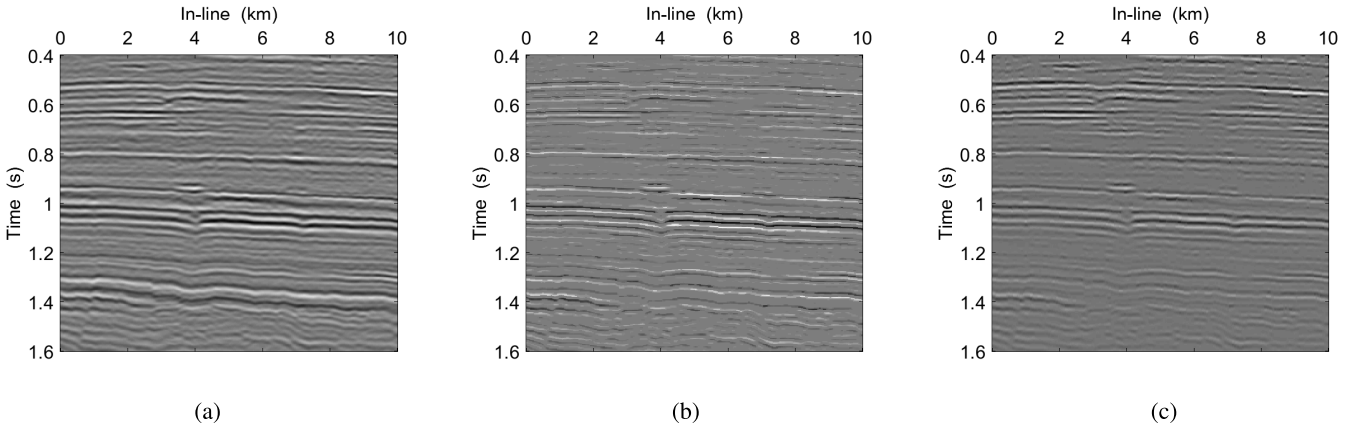

Fig. 4. Real data inversion results. (a) Seismic data. (b) Estimated reflectivity with $N_{\text{it}} = 2$ maximum number of iterations per trace. (c) Reconstructed seismic data, $\rho_{\mathbf{Y},\hat{\mathbf{Y}}} = 0.89$.

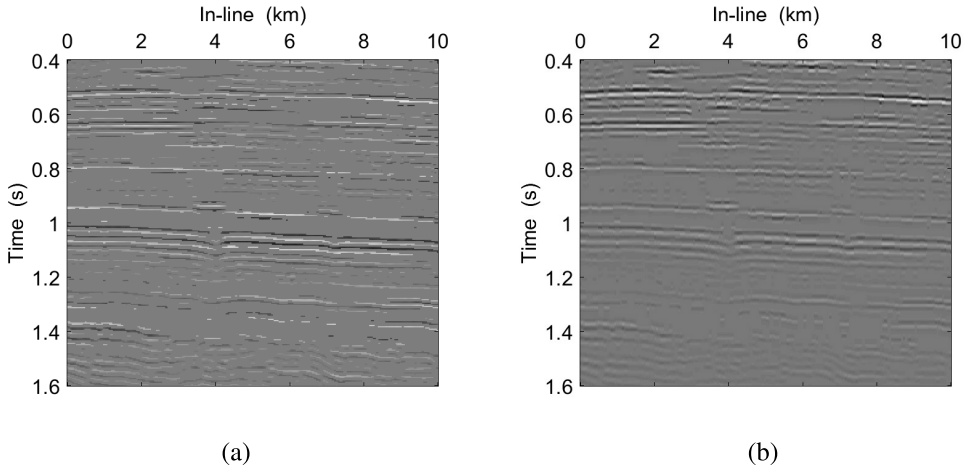

Fig. 5. Real data inversion results. (a) Estimated reflectivity in the first iteration. (b) Reconstructed seismic data in the first iteration, $\rho^1_{\mathbf{Y},\hat{\mathbf{Y}}} = 0.77$.

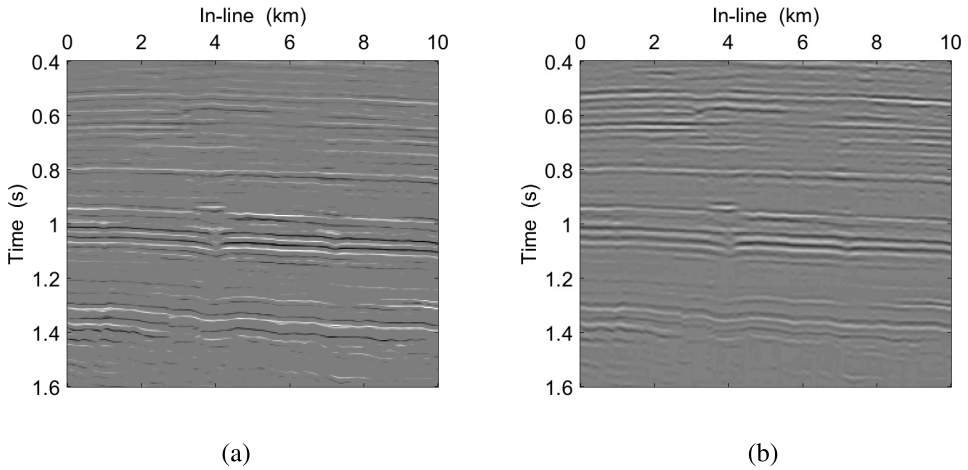

Fig. 6. Real data inversion results using ISTA. (a) Estimated reflectivity with $M_{\text{it}} = 1087$ average number of iterations per trace. (b) Reconstructed seismic data, $\rho_{\mathbf{Y},\hat{\mathbf{Y}}} = 0.91$.

$\alpha_{\text{r}} = 0.3$. The normalization window is a Gaussian window of length $L_{\text{h}} = 9$ with $\sigma_h = 2$.

Since the ground truth is unknown, in order to quantify the estimation success, we calculate the reconstructed seismic image from the recovered reflectivity, that is, $\hat{\mathbf{Y}} = \mathbf{G}\hat{\mathbf{X}}$, depicted in Fig. 4(c). We assess its correspondence with the given data by the correlation coefficient

$$\rho_{\mathbf{Y},\hat{\mathbf{Y}}} = \frac{\mathbf{Y}_{\text{cs}}^T \hat{\mathbf{Y}}_{\text{cs}}}{\|\mathbf{Y}_{\text{cs}}\|_2 \|\hat{\mathbf{Y}}_{\text{cs}}\|_2}$$

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

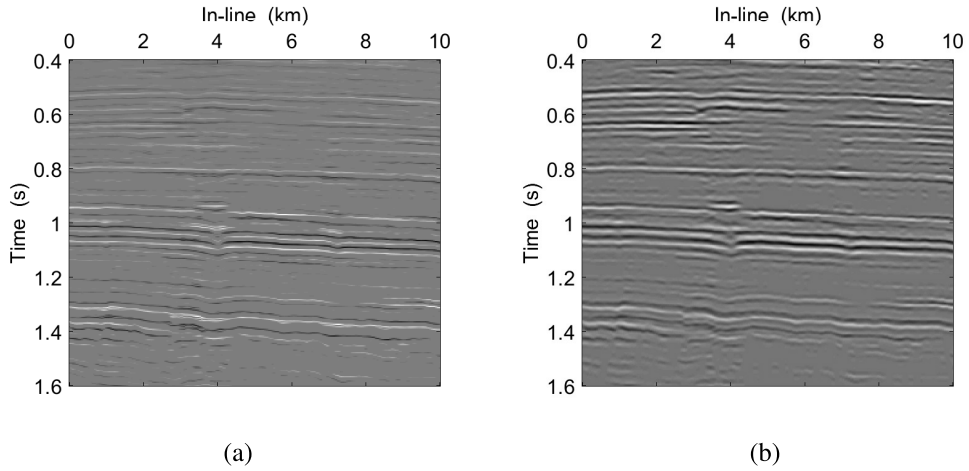

Fig. 7. Real data inversion results using multichannel time-variant deconvolution [34] with three channels taken into account for each channel estimation, employing LSE discontinuity measure [35]. (a) Estimated reflectivity. (b) Reconstructed seismic data, $\rho_{\mathbf{Y},\hat{\mathbf{Y}}} = 0.92$.

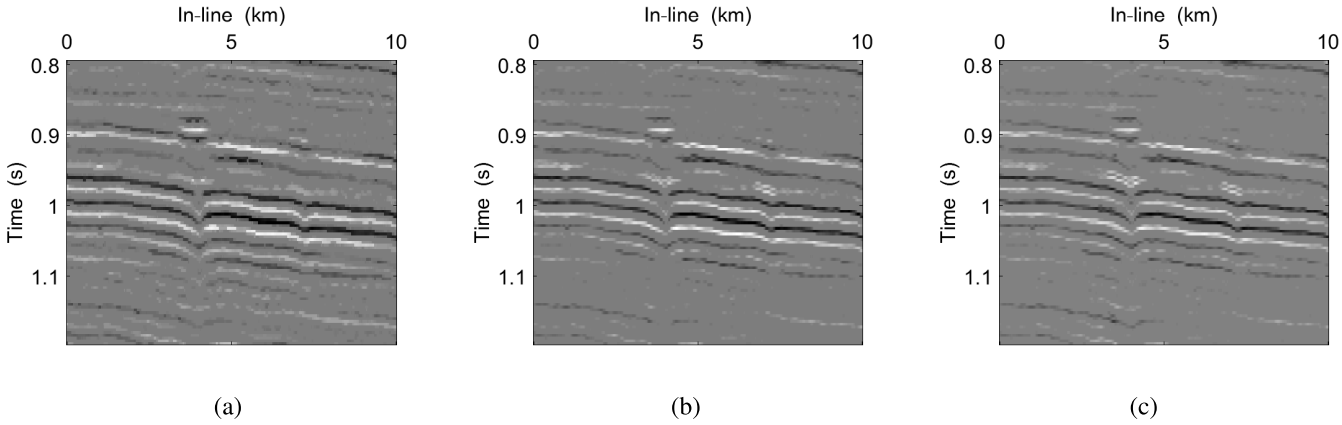

Fig. 8. Zoom into the predicted reflectivities in Figs. 4–7. (a) RFN-ITA. (b) ISTA. (c) Multichannel time-variant deconvolution.

where $\mathbf{Y}_{\text{cs}}$ and $\hat{\mathbf{Y}}_{\text{cs}}$ are column-stack vectors of the observed seismic image and the reconstructed seismic image, respectively. In terms of the correlation coefficient between the predicted data to the observed data, the first iteration achieved $\rho^{1}_{\mathbf{Y},\hat{\mathbf{Y}}} = 0.77$, while the second iteration achieved $\rho_{\mathbf{Y},\hat{\mathbf{Y}}} = 0.89$. Increasing the number of iterations and/or decreasing β_l or τ_l can increase the correlation score, on the expense of the reflector's localization resolution, that is, these parameter settings can produce smeared results of decreased sparsity, which is usually unwanted. As can be seen, reflector curves, in the reflectivity estimated in the first iteration, are a bit thick due to high mutual coherence values and insufficient spikes separation. Also, some reflectors are missing. Yet, this image is very close to the final one. Using our method, the most relevant information is recovered in the first iteration.

Figs. 6–8 compare the proposed method's results to ISTA results with $\beta = 0.14$ and a score of $\rho_{\mathbf{Y},\hat{\mathbf{Y}}} = 0.91$, and to the multichannel time-variant method in [34] with a score of $\rho_{\mathbf{Y},\hat{\mathbf{Y}}} = 0.92$. As can be seen, a slightly better score does not necessarily indicate a visually enhanced reflectivity. As can be observed in the predicted reflectivity produced by ISTA, some layers are incomplete, and it appears that some spikes are annihilated. Here, ISTA required an average of 1087 iterations per trace. In this example, RFN-ITA reduces the number of required iterations by a factor of 500. Fig. 9 presents one seismic trace at in-line offset 7.5 km, and the corresponding

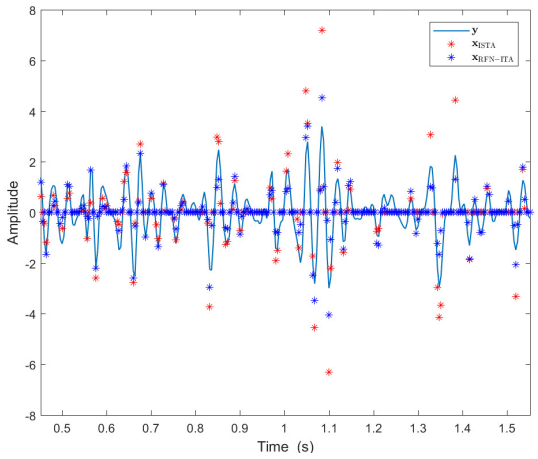

Fig. 9. Real data results of an 1-D example: seismic trace; ISTA estimated reflectivity and RFN-ITA estimated reflectivity.

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

TABLE II

SYNTHETIC EXAMPLE—RESULTS AND PARAMETERS: WAVELET'S SCALING ω_0, MINIMAL SEPARATION CONSTANT ν, THRESHOLDS β_1 AND β_2, NORMALIZATION WINDOW SIZE L_{h}, NORMALIZATION WINDOW STD σ_h, RECOVERED SUPPORT SCORE $\rho^{S}_{\mathbf{X},\hat{\mathbf{X}}}$, FINAL RECOVERED REFLECTIVITY SCORE $\rho^{A}_{\mathbf{X},\hat{\mathbf{X}}}$ FOR LEARNED ALGORITHM 4 IMPLEMENTATION, FINAL RECOVERED REFLECTIVITY SCORE $\rho^{B}_{\mathbf{X},\hat{\mathbf{X}}}$ FOR LEARNED ALGORITHM 3 IMPLEMENTATION, AND FINAL RECOVERED REFLECTIVITY SCORE FOR LEARNED ALGORITHM 3 IMPLEMENTATION $\rho^{C}_{\mathbf{X},\hat{\mathbf{X}}}$ WITH $\mathbf{D}_1 = \mathbf{D}_2$

ω_0	ν	β_1	β_2	L_{h}	σ_h	$\rho^{S}_{\mathbf{X},\hat{\mathbf{X}}}$	$\rho^{A}_{\mathbf{X},\hat{\mathbf{X}}}$	$\rho^{B}_{\mathbf{X},\hat{\mathbf{X}}}$	$\rho^{C}_{\mathbf{X},\hat{\mathbf{X}}}$
80π (40 Hz)	5	0.90	0.57	11	2	0.99	0.998	0.985	0.986
80π (40 Hz)	3	1.24	0.43	11	2	0.95	0.97	0.96	0.96
80π (40 Hz)	1	0.81	0.29	9	2	0.89	0.91	0.89	0.91
50π (25 Hz)	5	2.17	0.80	17	3	0.99	0.99	0.985	0.987
50π (25 Hz)	3	2.58	0.43	17	2	0.94	0.93	0.87	0.957

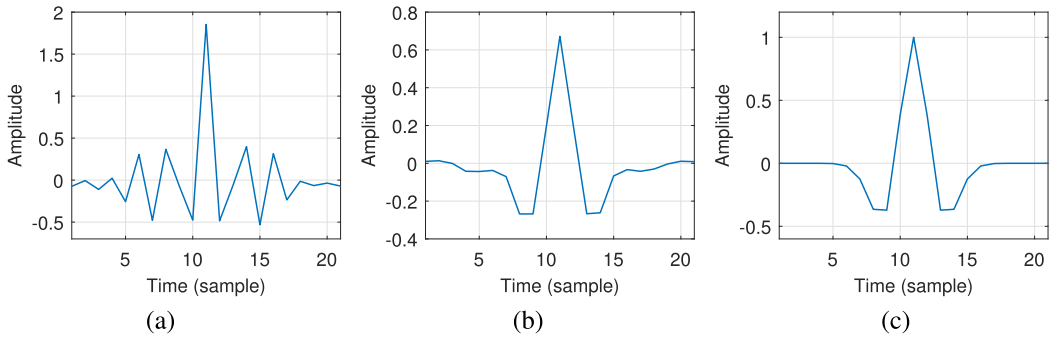

Fig. 10. Learned RFN-ITA parameters. (a) $\mathbf{D}_1$ kernel. (b) $\mathbf{D}_2$ kernel. (c) 80π ricker wavelet ($F_{\mathrm{s}} = 250$ Hz).

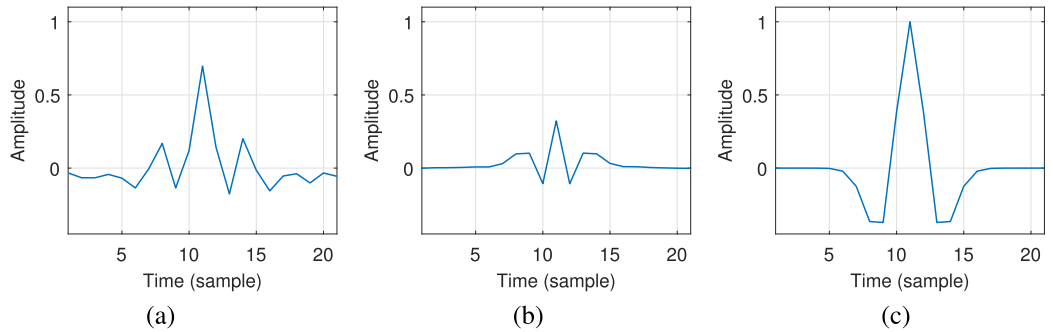

Fig. 11. Learned RFN-ITA parameters. (a) $\mathbf{D}_1$ kernel. (b) $\mathbf{D}_2$ kernel. (c) 80π ricker wavelet ($F_{\mathrm{s}} = 250$ Hz).

estimated 1-D reflectivity by ISTA and by RFN-ITA. As can be visually observed, the data is of inherent ambiguity, and the estimated results differ mostly in amplitude and share relatively close supports.

Throughout our experiments with ISTA, we observed that decreasing $\beta = (c/\lambda)$, which decreases the sparsity of the solution, comes at the expense of significantly increasing the number of iterations. Also, as can be seen in the presented examples, ISTA performance is terms of the visual quality of the image deteriorates with a time-variant dictionary, taking a Q factor into account. Moreover, in practice, ISTA number of iterations increases dramatically with the length of the signal. Another drawback is that the constant c/λ depends on the maximum eigenvalue of $\mathbf{D}^T\mathbf{D}$ that is harder to compute at larger scales [15]. RFN does not suffer from this complication because an approximate solution is reached within a limited number of iterations, where each data stripe is analyzed locally, yet the signal is updated globally without significant increase in computational complexity.

It is important to emphasize that most real data super resolution problems, and specifically seismic inversion, do not comply with the theoretical bound constraints (e.g., [2], [12]) for separation, amplitude balance, and noise. Therefore, applying classical algorithms to real data applications is not expected to result in perfect recovery. In [27] and [34], we show that under the separation condition in a noise-free environment, we can perfectly recover the reflectivity in an attenuating environment, providing that we can estimate Q. However, when working with real data, there is no guarantee that the true reflectors are sufficiently separated nor that Q is constant or estimated correctly. Also, we observe that the resolution of the estimation results is highly sensitive to user-dependent parameters. Overall, real data is of inherent uncertainty.

Since the proposed method requires only 2–4 iterations per trace, its implementation entails significantly low computational complexity. Solving using MATLAB, the processing time of the above dataset of 300 × 401 on a standard CPU Intel(R)Core(TM)i7-7820HQ@2.90 GHz is 122 ms, that is,

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

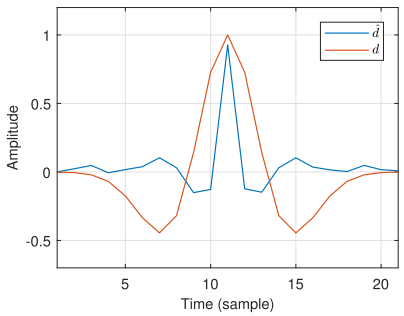

Fig. 12. Learned RFN-ITA parameters: $\mathbf{D}_1 = \mathbf{D}_2$ compared to original 50π ricker wavelet ($F_s = 250$ Hz).

about 305 μs per trace. Whereas the processing time of the same dataset for ISTA is 13.13 s, that is, 32.8 ms per trace. For the 3-D time-variant multichannel method, the processing time is 16.06 min, and 2.41 s per trace, which means that the proposed method is about 100 times faster than ISTA. This in turn assures us of a promising potential of the suggested method to be incorporated in the future in large-scale real-time data processing applications.

VII. Learned RFN-ITA

At this stage, a learned version of the proposed method, inspired by LISTA, seems almost inevitable. Namely, let us unfold Θ iterations of RFN-ITA, and set the filters building each convolutional dictionary at each iteration $\mathbf{D}_\theta$ as trainable parameters. In other words, we design a neural net, such that the output at layer θ is

$$\begin{aligned}\Delta\mathbf{r}_{\theta+1} &= \mathbf{y} - \mathbf{D}_\theta \mathbf{x}_\theta \\ \Delta\mathbf{q}_{\theta+1} &= \mathcal{S}_{\beta_{\theta+1}}\left(\mathbf{D}_{\theta+1}^T \tilde{\mathbf{W}}_\theta \Delta\mathbf{r}_{\theta+1}\right) \\ \mathbf{x}_{\theta+1} &= \alpha_{\theta+1}\left(\Delta\mathbf{q}_{\theta+1} \odot \Delta\mathbf{r}_{\theta+1}\right) + \mathbf{x}_\theta, \quad \theta = 0, 1, \ldots, \Theta - 1\end{aligned} \tag{42}$$

where $\tilde{\mathbf{W}}_\theta$ is a scaled RFN matrix as defined in (24), without dividing by $\|\mathbf{d}_i\|_2$, since at the training stage the dictionary is unknown. In other words,

$$\tilde{\mathbf{W}}_\theta = \operatorname{diag}\left(\frac{1}{\tilde{\sigma}_{\Delta r}^{\theta+1}[k]}\right), \quad k = 1, \ldots, L_y. \tag{43}$$

$\Psi = \{\alpha_\theta, \beta_\theta, \mathbf{D}_\theta\}$ are the learned parameters. Here, we have used the soft-thresholding operation because its implementation using the ReLU function facilitates convergence. Training is performed using stochastic gradient descent to minimize the loss $\mathcal{L}(\mathbf{y}; \Psi)$ between the model predicted code $\mathbf{x}_\Theta = f(\mathbf{y}; \Psi)$ to a known code over a training set of known $\{\mathbf{y}\}_{p=1}^P$ and the corresponding $\{\mathbf{x}\}_{p=1}^P$

$$\min_\Psi \frac{1}{P}\sum_{p=1}^{P} \|f(\mathbf{y}_p; \Psi) - \mathbf{x}_p\|_2^2. \tag{44}$$

Once trained, the model is expected to produce sparse codes for signals from the same probability distribution, without requiring their original true dictionary.

Fig. 10 compares the learned filters to the true wavelet $g[k]$ that was used to create the observed traces, for a two-layer model, that is, $\Theta = 2$, trained to recover only the support, as described in Algorithm 4. In this section, we used Pytorch [37] for the numerical implementation. A training set of $N = 997$ random reflectivity independent sequences of length $L_x = 60$, as described in Section V-A, with $\nu = 3$, $\omega_0 = 80\pi$, and $F_s = 250$ Hz. Table II presents the testing correlation score for this example $\rho^A_{\mathbf{X},\hat{\mathbf{X}}}$, the support correlation score $\rho^S_{\mathbf{X},\hat{\mathbf{X}}}$, and the learned parameters β_1, β_2 for a test set of $N = 997$ traces, with different minimal separation constants $\nu = 5, 3, 1$. We do not force any constraints on the weight parameters.

Fig. 11 shows the learned filters versus the true wavelet, for a two layer model, that is, $\Theta = 2$, trained to recover both the support and amplitude with $\nu = 1$, as described in Algorithm 3. Fig. 12 presents the learned filter for Algorithm 3 when $\nu = 1$, $\omega_0 = 50\pi$, and we force $\mathbf{D}_1 = \mathbf{D}_2$. As can be seen, the learned pulse is significantly narrower than the original one, and its side-lobes are shallower. The corresponding correlation scores $\rho^B_{\mathbf{X},\hat{\mathbf{X}}}$ and $\rho^C_{\mathbf{X},\hat{\mathbf{X}}}$, respectively, are also presented in Table II. As evident, there is no significant variations between the models' performance, since the prediction precision is inherently dependent on sufficient separation between spikes, and the signal bandwidth, represented by ν and ω_0.

At this stage, we shall leave further work on the learned extension of RFN-ITA, the relevant mathematical analysis, and its applications to our future research.

VIII. Conclusion

We have presented an efficient modification of the classic ITAs for CSC. We have shown that receptive field normalization leads to a substantial reduction in the number of required iterations to reach an approximate sparse code vector. We demonstrated that whether the dictionary is known or learned via neural network training, about 2–4 iterations are sufficient to produce a sparse code close to the true one, even when the dictionary's mutual coherence is relatively high. The proposed algorithms entail a significantly low computational complexity, and therefore can potentially contribute to a dramatic speed-up in many real-time applications, such as super resolution and pattern recognition systems, as well as neural nets feature extraction in other existing systems. Hopefully, future work will further investigate the proposed methods and their extension to other applications. An extension to seismic blind deconvolution is also possible. The theoretical analysis could be extended to a statistical point of view rather than a worst-case scenario approach.

Appendix A
Proof of Theorem 1

We introduce the following assumptions.

1) A 1 (Stationary Receptive-Field Energy): We assume that for every support index i, the local weighted energy as defined in (13) is approximately constant and equal to the energy at i, that is,

$$\sigma_y[k] = \sigma_y[i] \quad \forall k \in K_i, \; i \in K \tag{45}$$

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

where K_i is the neighborhood of i and k belongs to the neighborhood K_i provided that $|k-i| \leq (L_{\mathrm{d}}/2)-1$. Assume without loss of generality the dictionary columns are normalized, and denote $\mathbf{a}_i = \mathbf{W}_0\mathbf{d}_i$. According to the above assumption, we have

$$\left|\mathbf{a}_i^T\mathbf{d}_i\right| = \frac{1}{\sigma_y[i]} \quad \forall i \in K. \tag{46}$$

In other words, the corresponding atom after RFN is approximately equivalent to a scaled version of the original atom. Denote $\mathbf{v}_i$ such that $\mathbf{a}_i = \mathrm{diag}^{-1}(\sigma_y[i]) \odot (\mathbf{d}_i - \mathbf{v}_i)$, then (46) is true when either $\mathbf{v}_i^T\mathbf{d}_i = 0$ or $\mathbf{v}_i = \mathbf{0}$. Clearly, assumption **A 1** is not completely accurate, since for each data sample, the normalization window changes its location, and therefore the local energy changes. However, practically we observe that the difference $\|\sigma_y[i]\mathbf{a}_i - \mathbf{d}_i\|_2$ is relatively very small and that $\mathbf{v}_i^T\mathbf{d}_i$ can be neglected for the purposes of the following mathematical analysis and for the sake of brevity.

2) A 2 (No Atoms Mismatch):

$$\begin{aligned}\left|\mathbf{a}_i^T\mathbf{d}_j\right| &\leq \frac{1}{\sigma_y[i]}\left|\mathbf{d}_i^T\mathbf{d}_j\right| \\ &\leq \frac{\mu}{\sigma_y[i]} \quad \forall\{(i,j): i \neq j, i \in K, j \notin K\}.\end{aligned} \tag{47}$$

In other words, RFN can only decrease local stripe correlation to dictionary atoms that are not on the support. This assumption is very accurate, even in practical cases.

3) A 3 (Noise Scaling):

$$\left|\mathbf{a}_i^T\mathbf{e}\right| \leq \left|\mathbf{W}_0\mathbf{d}_i^T\mathbf{e}\right| \leq \frac{\left\|\mathbf{d}_i^T\right\|_2\|\mathbf{e}_i\|_2}{\min_i \sigma_y[i]} \leq \frac{\varepsilon_{\mathrm{d}}}{\tau} = \varepsilon_{\mathrm{s}} \quad \forall i \tag{48}$$

where $\mathbf{e}_i$ is a noise stripe of length L_{h} such that $\|\mathbf{e}_i\|_2 = \varepsilon_{\mathrm{d}}$, based on Cauchy–Schwartz and assuming atoms with unit norm $\mathbf{d}_i^T\mathbf{e} \leq \|\mathbf{d}_i^T\|_2\|\mathbf{e}_i\|_2 = \|\mathbf{e}_i\|_2$. In other words, the correlation of a local stripe noise to the dictionary atoms after RFN remains well bounded. Note that $\varepsilon_{\mathrm{d}} \ll \varepsilon$.

Let us denote $\mathbf{W}_{0,K} \in \mathbb{R}^{L_{\mathrm{x}}\times L_{\mathrm{x}}}$ as the partial signal normalization weight submatrix on the support of $\mathbf{x}$. In other words, $\mathbf{W}_{0,K} = \mathrm{diag}(\tilde{\sigma}_y^{-1}[k])$, $k = \Delta_d+1, \ldots, \Delta_d+L_{\mathrm{x}}$. One may suggest that the above assumptions simply propose that thresholding the expression $\mathbf{D}^T\mathbf{W}_0\mathbf{y}$ is equivalent to thresholding $\mathbf{W}_{0,K}\mathbf{D}^T\mathbf{y}$. In other words, scaling the projection values is equivalent (or advantageous) to normalizing the (error) signal and then projecting it on the dictionary. In practice, the two approaches are mathematically equivalent only when the atoms are at least L_{s} samples apart. Indeed, projection post-normalization can distort the observed signal. However, it is beneficial for preventing false detections. For example, in Section V, we presented examples using the Ricker wavelet. RFN helps prevent the "side wings" from being misdetected as the main lobes of the wavelet. This is achieved by the neighborhood energy reducing the normalized side lobe's correlation to atoms at these locations, implying that it cannot be a true support location.

These three assumptions are not necessarily realistic in the case of the seismic inversion. However, many assumptions made for seismic deconvolution in the field of seismic processing are not necessarily realistic either, and yet seismic deconvolution has been applied for many years with considerable success [38].

Now, we shall proceed to the proof of Theorem 1. Perfect support detection is guaranteed if the requirement

$$\min_{i\in K}\left|\mathbf{a}_i^T\mathbf{y}\right| > \max_{j\notin K}\left|\mathbf{a}_j^T\mathbf{y}\right| \tag{49}$$

is met. Assuming a rectangular receptive field normalization window of length $L_{\mathrm{h}} = L_{\mathrm{d}}$

$$\sigma_y[i] = \|\mathbf{D}_i\mathbf{x}_i + \mathbf{e}_i\|_2 \tag{50}$$

where $\mathbf{x}_i$ denotes a stripe of length mL_{s} located around a data point i, and $\mathbf{D}_i$ is the submatrix partial dictionary yielding point $\mathbf{y}[i]$, that is obtained by restricting the dictionary $\mathbf{D}$ to the support of mL_{s} weights corresponding to L_{h} data points equally distributed around point i. Denote $\mu = \mu(\mathbf{D}_i)$ (obviously $\mu = \mu(\mathbf{D})$) and assume without loss of generality the dictionary columns are normalized, that is, $\sigma_{d_i} = 1 \forall i \in [1, M]$, according to the stripe restricted isometry property (sRIP) ([39, Definition 14])

$$(1-(s-1)\mu)\ \|\mathbf{x}_i\|_2^2 \leq \|\mathbf{D}_i\mathbf{x}_i\|_2^2 \leq (1+(s-1)\mu)\ \|\mathbf{x}_i\|_2^2. \tag{51}$$

Moreover,

$$\|\mathbf{D}_i\mathbf{x}_i\|_2 \leq \sum_{j\in S_i^h}\|\mathbf{d}_j\|_2\|\mathbf{x}_i\|_2 \leq \|\mathbf{x}_i\|_2. \tag{52}$$

Then, we can lower-bound the right term in (66) by

$$\begin{aligned}\min_{i\in K}\left|\mathbf{a}_i^T\mathbf{y}\right| &\geq \min_{i\in K}\left|\mathbf{a}_i^T(\mathbf{y}-\mathbf{e})\right| - \left|\mathbf{a}_i^T\mathbf{e}\right| \\ &\geq \min_{i\in K}\frac{\left|\mathbf{d}_i^T\mathbf{D}\mathbf{x}\right|}{\|\mathbf{D}_i\mathbf{x}_i+\mathbf{e}_i\|_2} - \varepsilon_{\mathrm{s}} \\ &\geq \min_{i\in K}\frac{\left|\mathbf{d}_i^T\mathbf{D}_i\mathbf{x}_i\right|}{\|\mathbf{D}_i\mathbf{x}_i\|_2+\|\mathbf{e}_i\|_2} - \varepsilon_{\mathrm{s}} \\ &\geq \min_{i\in K}\frac{\left|\mathbf{d}_i^T\mathbf{D}_i\mathbf{x}_i\right|}{\|\mathbf{x}_i\|_2+\|\mathbf{e}_i\|_2} - \varepsilon_{\mathrm{s}} \\ &\geq \min_{i\in K}\frac{\left|\mathbf{d}_i^T\mathbf{D}_i\mathbf{x}_i\right|}{\|\mathbf{x}_i\|_1+\varepsilon_{\mathrm{d}}} - \varepsilon_{\mathrm{s}} \\ &\geq \min_{i\in K}\frac{|\mathbf{x}[i]|\mathbf{d}_i^T\mathbf{d}_i}{\|\mathbf{x}_i\|_1+\varepsilon_{\mathrm{d}}} - \left|\sum_{t\in S_h^i, t\neq i}\frac{x[t]\mathbf{d}_i^T\mathbf{d}_t}{\|\mathbf{x}_i\|_1+\varepsilon_{\mathrm{d}}}\right| - \varepsilon_{\mathrm{s}} \\ &\geq \min_{i\in K}\frac{(1+\mu)|\mathbf{x}[i]|}{\|\mathbf{x}_i\|_1+\varepsilon_{\mathrm{d}}} - \mu\left(\sum_{t\in S_h^i}\frac{|\mathbf{x}[t]|}{\|\mathbf{x}_i\|_1}\right) - \varepsilon_{\mathrm{s}} \\ &\geq \min_{i\in K}\frac{(1+\mu)|\mathbf{x}[i]|}{\|\mathbf{x}_i\|_1+\varepsilon_{\mathrm{d}}} - \mu - \varepsilon_{\mathrm{s}}\end{aligned} \tag{53}$$

where we have used the relation $\|\mathbf{x}_i\|_1 \geq \|\mathbf{x}_i\|_2$ and assumptions A 1–A 3. On the other hand,

$$\begin{aligned}\max_{j\notin K}\left|\mathbf{a}_j^T\mathbf{y}\right| &= \max_{j\notin K}\frac{\left|\mathbf{d}_j^T\mathbf{y}\right|}{\|\mathbf{D}_j\mathbf{x}_j+\mathbf{e}_j\|_2} + |\mathbf{a}_j^T\mathbf{e}| \\ &\leq \frac{\mu\|\mathbf{x}_j\|_1}{\|\mathbf{x}_j\|_2\sqrt{1-(s-1)\mu}-\varepsilon_{\mathrm{d}}} + \varepsilon_{\mathrm{s}} \\ &\leq \frac{\sqrt{s}\mu\|\mathbf{x}_j\|_2}{\|\mathbf{x}_j\|_2(\sqrt{1-(s-1)\mu}-\tilde{\varepsilon}_{\mathrm{d}})} + \varepsilon_{\mathrm{s}}\end{aligned}$$

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

$$\leq \frac{\sqrt{s}\mu}{\sqrt{1-(s-1)\mu}-\tilde{\varepsilon}_{\mathrm{d},\infty}}+\varepsilon_{\mathrm{s}} \tag{54}$$

where we denote $\tilde{\varepsilon}_{\mathrm{d}} = (\varepsilon_{\mathrm{d}}/\|\mathbf{x}_j\|_2)$, and $\tilde{\varepsilon}_{\mathrm{d},\infty} = (\varepsilon_{\mathrm{d}}/|\mathbf{x}|_{\min})$. Combining (53) and (54), we require

$$\min_{i\in K}\frac{|\mathbf{x}[i]|(1+\mu)}{\|\mathbf{x}_i\|_1+\varepsilon_{\mathrm{d}}}-\mu-\varepsilon_{\mathrm{s}}>\frac{\sqrt{s}\mu}{\sqrt{1-(s-1)\mu}-\tilde{\varepsilon}_{\mathrm{d},\infty}}+\varepsilon_{\mathrm{s}}. \tag{55}$$

It follows that

$$\frac{|\mathbf{x}[i]|}{\|\mathbf{x}_i\|_1+\varepsilon_{\mathrm{d}}}>\frac{\mu}{1+\mu}\left(1+\frac{\sqrt{s}}{\sqrt{1-(s-1)\mu}-\tilde{\varepsilon}_{\mathrm{d},\infty}}\right)+\frac{2\varepsilon_{\mathrm{s}}}{1+\mu}\quad \forall i\in K. \tag{56}$$

Taking the worst-case analysis point of view, considering that $\|\mathbf{x}_i\|_1 \leq s|\mathbf{x}_i|_{\max}$, we deduce

$$\frac{|\mathbf{x}_i|_{\min}}{|\mathbf{x}_i|_{\max}+\frac{\varepsilon_{\mathrm{d}}}{s}}>\frac{s\mu}{1+\mu}\left(1+\frac{\sqrt{s}}{\sqrt{1-(s-1)\mu}-\tilde{\varepsilon}_{\mathrm{d},\infty}}\right)+\frac{2\,s\varepsilon_{\mathrm{s}}}{1+\mu}\forall i\in K. \tag{57}$$

In a noise-free model, we have

$$\frac{|\mathbf{x}_i|_{\min}}{|\mathbf{x}_i|_{\max}}>\frac{s\mu}{1+\mu}\left(1+\frac{\sqrt{s}}{\sqrt{1-(s-1)\mu}}\right)\quad \forall i\in K. \tag{58}$$

This completes the proof. □

Appendix B
Proof of Theorem 2

Perfect support detection is guaranteed if the requirement

$$\min_{i\in K}\left|\mathbf{a}_i^T\mathbf{y}\right|>\max_{j\notin K}\left|\mathbf{a}_j^T\mathbf{y}\right| \tag{59}$$

is met. When $s=1$ assuming $\mathbf{y}=\mathbf{D}\mathbf{x}$, we have

$$\min_{i\in K}\left|\mathbf{a}_i^T\mathbf{y}\right| = \min_{i\in K}\frac{\left|\mathbf{d}_i^T\mathbf{y}\right|}{\|\mathbf{D}_i\mathbf{x}_i\|_2}\geq\min_{i\in K}\frac{\left|\mathbf{d}_i^T\mathbf{y}\right|}{\|\mathbf{x}_i\|_2} \geq \min_{i\in K}\frac{|\mathbf{x}[i]|}{|\mathbf{x}[i]|}\mathbf{d}_i^T\mathbf{d}_i=1. \tag{60}$$

On the other hand,

$$\max_{j\notin K}\left|\mathbf{a}_j^T\mathbf{y}\right| = \max_{j\notin K}\frac{\left|\mathbf{d}_j^T\mathbf{y}\right|}{\|\mathbf{D}_j\mathbf{x}_j\|_2} \leq \frac{\mu|\mathbf{x}_j[k_j]|}{|\mathbf{x}_j[k_j]|}\leq\mu,\quad k_j\in K, k_j\in S_h^j. \tag{61}$$

Since $\mu<1$, for any $0<(|\mathbf{x}|_{\min}/|\mathbf{x}|_{\max})\leq 1$, the support is perfectly recovered at the first iteration using RFN. □

Appendix C
Proof of Theorem 3

Recall that we have defined the stripe local energy in (31) as

$$\sigma_{\mathrm{y}}[k]=\|\mathbf{H}\mathbf{y}_k\|_2=\|\mathbf{H}\mathbf{D}_k\mathbf{x}_k\|_2. \tag{62}$$

Similar to the sRIP ([39, Definition 14]), we can observe that

$$\begin{aligned}\|\mathbf{H}\mathbf{D}_i\mathbf{x}_i\|_2^2 &= \mathbf{x}_i^T(\mathbf{H}\mathbf{D}_i)^T\mathbf{H}\mathbf{D}_i\mathbf{x}_i\\ &\leq |\mathbf{x}_i|^T\left[\mathbf{H}_d^2+\mu(\mathbf{H}\mathbf{D}_i)(\mathbf{1}-\mathbf{I})\right]|\mathbf{x}_i|\\ &\leq \|\mathbf{H}_d\mathbf{x}_i\|_2^2-\mu(\mathbf{H}\mathbf{D}_i)\|\mathbf{x}_i\|_2^2+\mu(\mathbf{H}\mathbf{D}_i)\|\mathbf{x}_i\|_1^2\\ &\leq \|\mathbf{H}_d\mathbf{x}_i\|_2^2+(s-1)\mu(\mathbf{H}\mathbf{D}_i)\|\mathbf{x}_i\|_2^2\end{aligned} \tag{63}$$

where $\mathbf{H}_d$ is a diagonal matrix holding the attenuated atoms ℓ_2 norms, namely $\mathbf{H}_d^2=\mathrm{diag}(\mathrm{trace}((\mathbf{H}\mathbf{D}_i)^T\mathbf{H}\mathbf{D}_i))$ and $\mathbf{H}_d[j,j]=\|\mathbf{H}\mathbf{d}_j^i\|_2$, $j\in S_i^h$, where $\mathbf{d}_j^i\in\mathbb{R}^{L_{\mathrm{h}}\times 1}$ is the jth atom of the submatrix $\mathbf{D}_i$. Clearly, we also have

$$\begin{aligned}\|\mathbf{H}\mathbf{D}_i\mathbf{x}_i\|_2^2 &= \mathbf{x}_i^T(\mathbf{H}\mathbf{D}_i)^T\mathbf{H}\mathbf{D}_i\mathbf{x}_i\\ &\geq |\mathbf{x}_i|^T\left[\mathbf{H}_d^2-\mu(\mathbf{H}\mathbf{D}_i)(\mathbf{1}-\mathbf{I})\right]|\mathbf{x}_i|\\ &\geq \|\mathbf{H}_d\mathbf{x}_i\|_2^2+\mu(\mathbf{H}\mathbf{D}_i)\|\mathbf{x}_i\|_2^2-\mu(\mathbf{H}\mathbf{D}_i)\|\mathbf{x}_i\|_1^2\\ &\geq \|\mathbf{H}_d\mathbf{x}_i\|_2^2-(s-1)\mu(\mathbf{H}\mathbf{D}_i)\|\mathbf{x}_i\|_2^2.\end{aligned} \tag{64}$$

By assumption, h is a receptive field normalization kernel that is monotonically descending away from its center, and therefore we assume $\mu(\mathbf{H}\mathbf{D}_i)\ll\mu(\mathbf{D}_i)$, and hereafter use the approximation

$$\sigma_{\mathrm{y}}[k]\approx\|\mathbf{H}_d\mathbf{x}_i\|_2. \tag{65}$$

Also, under the separation condition, let us denote $h_d(\nu)\triangleq\max_{p\in[1,m]}\mathbf{H}_d[\Delta_k+(p-1)L_{\mathrm{s}},\Delta_k+(p-1)L_{\mathrm{s}}]$. In other words, $h_d(\nu)$ is the maximal ℓ_2 norm of an atom shifted by the separation distance multiplied by the RFN window.

As stated above, to guarantee perfect support detection, we have to show that the requirement

$$\min_{i\in K}\left|\mathbf{a}_i^T\mathbf{y}\right|>\max_{j\notin K}\left|\mathbf{a}_j^T\mathbf{y}\right| \tag{66}$$

is satisfied. We shall begin with the left-hand side term

$$\begin{aligned}\min_{i\in K}\left|\mathbf{a}_i^T\mathbf{y}\right| &\geq \min_{i\in K}\frac{\left|\mathbf{d}_i^T\mathbf{D}\mathbf{x}\right|}{\|\mathbf{H}\mathbf{D}_i\mathbf{x}_i\|_2}\approx\min_{i\in K}\frac{\left|\mathbf{d}_i^T\mathbf{D}\mathbf{x}\right|}{\|\mathbf{H}_d\mathbf{x}_i\|_2}\\ &\geq \min_{i\in K}\frac{|\mathbf{x}[i]|}{\|\mathbf{H}_d\mathbf{x}_i\|_2}\mathbf{d}_i^T\mathbf{d}_i-\left|\sum_{t\in S_h^i,t\neq i}\frac{x[t]}{\|\mathbf{H}_d\mathbf{x}_i\|_2}\mathbf{d}_i^T\mathbf{d}_t\right|\\ &\geq \min_{i\in K}\frac{|\mathbf{x}[i]|-\mu\sum_{t\in S_h^i,t\neq i}|\mathbf{x}[t]|}{\|\mathbf{H}_d\mathbf{x}_i\|_1}\\ &\geq \min_{i\in K}\frac{|\mathbf{x}[i]|-\mu\sum_{t\in S_h^i,t\neq i}|\mathbf{x}[t]|}{|\mathbf{x}[i]|+h_d(\nu)\sum_{t\in S_h^i,t\neq i}|\mathbf{x}[t]|}\\ &= 1-(\mu+h_d(\nu))\max_{i\in K}\\ &\quad\times\frac{\sum_{t\in S_h^i,t\neq i}|\mathbf{x}[t]|}{|\mathbf{x}[i]|+h_d(\nu)\sum_{t\in S_h^i,t\neq i}|\mathbf{x}[t]|}\end{aligned} \tag{67}$$

where we have used $\|\mathbf{H}_d\mathbf{x}_i\|_1\leq|\mathbf{x}[i]|+h(\nu)\sum_{t\in S_h^i,t\neq i}|\mathbf{x}[t]|$ since the closest nonzero input entry is at distant Δ_k samples from i. Turning now to the right-hand side of (66), we have

$$\max_{j\notin K}\left|\mathbf{a}_j^T\mathbf{y}\right| = \max_{j\notin K}\frac{\left|\mathbf{d}_j^T\mathbf{y}\right|}{\|\mathbf{H}\mathbf{D}_j\mathbf{x}_j\|_2} \leq \frac{\mu\|\mathbf{x}_j\|_1}{\|\mathbf{H}_d\mathbf{x}_j\|_2}\leq\frac{\mu\|\mathbf{x}_j\|_1}{\frac{h_{\mathrm{d,min}}}{\sqrt{s}}\|\mathbf{x}_j\|_1}\leq\frac{\sqrt{s}\mu}{h_{\mathrm{d,min}}} \tag{68}$$

where we have lower bounded the denominator, assuming $\sigma_{\mathrm{y}}[j]>\tau$

$$\|\mathbf{H}_d\mathbf{x}_j\|_2\geq h_{\mathrm{d,min}}\|\mathbf{x}_j\|_2\geq\frac{h_{\mathrm{d,min}}}{\sqrt{s}}\|\mathbf{x}_j\|_1. \tag{69}$$

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

It is worth mentioning that a tighter bound could be obtained using

$$\max_{j\notin K}\left|\mathbf{a}_j^T\mathbf{y}\right| = \max_{j\notin K}\frac{\left|\mathbf{d}_j^T\mathbf{y}\right|}{\|\mathbf{H}\mathbf{D}_j\mathbf{x}_j\|_2} \le \frac{\sqrt{s}\sum_{k\in S_j^h,k\neq j}\left|\mathbf{d}_j^T\mathbf{d}_k\right||\mathbf{x}_j[k]|}{\sum_{k\in S_j^h,k\neq j}\mathbf{H}_d[|k-j|]|\mathbf{x}_j[k]|}. \quad (70)$$

This means that no false detection is guaranteed as long as

$$\frac{\sqrt{s}\left|\mathbf{d}_j^T\mathbf{d}_k\right|}{\mathbf{H}_d[|k-j|]} < \beta_1 \quad \forall k \in S_j^h,\ j \notin K. \quad (71)$$

Combining (67) and (69), and denote $\|\mathbf{x}_{-i}\|_1 \triangleq \sum_{t\in S_h^i,t\neq i}|\mathbf{x}[t]| = \|\mathbf{x}_i\|_1 - |\mathbf{x}[i]|$, we have that as long as

$$\frac{\sqrt{s}\mu}{h_{\mathrm{d,min}}} < \beta_1 \quad (72)$$

and

$$\frac{\|\mathbf{x}_{-i}\|_1}{|\mathbf{x}[i]| + h_d(v)\|\mathbf{x}_{-i}\|_1} < \frac{1-\beta_1}{\mu + h_d(v)} \quad \forall i \in K \quad (73)$$

the true support is fully recovered by RFN thresholding. This concludes the proof. □

APPENDIX D
EARTH Q-MODEL TIME-VARIANT WAVELETS ESTIMATION

We recall from [27], [34] that we can estimate the set of time-variant pulses $\{g_{\sigma,n}\}$ in (2.2) according to the Earth Q-model. Namely, we begin with an initial wavelet with a dominant frequency ω_0, for example, the real-valued Ricker wavelet

$$g(t) = \left(1 - \frac{1}{2}\omega_0^2 t^2\right)\exp\left(-\frac{1}{4}\omega_0^2 t^2\right). \quad (74)$$

In this setting, the scaling parameter is $\sigma = \omega_0^{-1}$. Following the Earth Q-model [29], assuming we know Q [30], a reflected pulse at travel time t_n is

$$u_n(t-t_n) = \mathrm{Re}\left\{\frac{1}{\pi}\int_0^\infty G(\omega)\exp[j(\omega t - \kappa r(\omega))]d\omega\right\} \quad (75)$$

where $G(\omega)$ is the Fourier transform of the source waveform $g(t)$

$$\kappa r(\omega) \triangleq \left(1 - \frac{j}{2Q}\right)\left|\frac{\omega}{\omega_0}\right|^{-\gamma}\omega t_n$$
$$\gamma \triangleq \frac{2}{\pi}\tan^{-1}\left(\frac{1}{2Q}\right) \approx \frac{1}{\pi Q}. \quad (76)$$

Note that in the frequency domain, the phase change exponential operator represents velocity dispersion, while the amplitude attenuation exponential operator corresponds to energy absorption of the traveling pulses

$$U_n(\omega)\exp^{-j\omega t_n} = G(\omega)\exp\left(-j\left|\frac{\omega}{\omega_0}\right|^{-\gamma}\omega t_n\right) \times \exp\left(-\left|\frac{\omega}{\omega_0}\right|^{-\gamma}\frac{\omega t_n}{2Q}\right). \quad (77)$$

The time-domain seismic pulse reflected at two-way travel time (depth) t_n is

$$u_n(t-t_n) = \frac{1}{2\pi}\int U_n(\omega)\exp[j\omega(t-t_n)]d\omega. \quad (78)$$

Therefore, the estimated set of pulses $\{g_{\sigma,n}\}$ based on the Earth Q model is defined as

$$g_{\sigma,n}(t-t_n) = u(t-t_n)|_{\sigma=\omega_0^{-1}}. \quad (79)$$

REFERENCES

[1] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.
[2] M. Elad, *Sparse Redundant Representations*. New York, NY, USA: Springer, 2010.
[3] K. Jarrett, K. Kavukcuoglu, M. A. Ranzato, and Y. LeCun, "What is the best multi-stage architecture for object recognition?" in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep. 2009, pp. 2146–2153.
[4] D. L. Donoho, "Compressed sensing," *IEEE Trans. Inf. Theory*, vol. 52, no. 4, pp. 1289–1306, Apr. 2006.
[5] R. G. Baraniuk and P. Steeghs, "Compressive radar imaging," in *Proc. IEEE Radar Conf.*, Apr. 2007, pp. 128–133.
[6] T. Bendory, A. Bar-Zion, D. Adam, S. Dekel, and A. Feuer, "Stable support recovery of stream of pulses with application to ultrasound imaging," *IEEE Trans. Signal Process.*, vol. 64, no. 14, pp. 3750–3759, 2016.
[7] H. L. Taylor, S. C. Banks, and J. F. McCoy, "Deconvolution with the ℓ_1 norm," *Geophysics*, vol. 44, no. 1, pp. 39–52, 1979.
[8] T. Nguyen and J. Castagna, "High resolution seismic reflectivity inversion," *J. Seismic Explor.*, vol. 19, no. 4, pp. 303–320, 2010.
[9] R. Zhang and J. Castagna, "Seismic sparse-layer reflectivity inversion using basis pursuit decomposition," *Geophysics*, vol. 76, no. 6, pp. 147–158, 2011.
[10] B. Olshausen and D. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, pp. 607–609, Jul. 1996.
[11] H. Lee, R. Grosse, R. Ranganath, and A. Y. Ng, *Convolutional Deep Belief Networks for Scalable Unsupervised Learning of Hierarchical Representations*. New York, NY, USA: ACM Press, 2009, pp. 609–616.
[12] V. Papyan, Y. Romano, and M. Elad, "Convolutional neural networks analyzed via convolutional sparse coding," *J. Mach. Learn. Res.*, vol. 18, no. 1, pp. 2887–2938, 2017.
[13] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2324, Nov. 1998.
[14] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.
[15] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, Mar. 2009.
[16] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proc. 27th Int. Conf. Int. Conf. Mach. Learn.* Madison, WI, USA: Omnipress, 2010, pp. 399–406.
[17] A. Aberdam, A. Golts, and M. Elad, "Ada-LISTA: Learned solvers adaptive to varying models," 2020, *arXiv:2001.08456*. [Online]. Available: http://arxiv.org/abs/2001.08456
[18] C. Zhao, J. Zhang, R. Wang, and W. Gao, "CREAM: CNN-REgularized ADMM framework for compressive-sensed image reconstruction," *IEEE Access*, vol. 6, pp. 76838–76853, 2018.
[19] Y. Yang, J. Sun, H. Li, and Z. Xu, "ADMM-CSNet: A deep learning approach for image compressive sensing," *IEEE Trans. Pattern Anal. Mach. Intell.* vol. 42, no. 3, pp. 521–538, Mar. 2020.
[20] J. Xiang, Y. Dong, and Y. Yang, "FISTA-net: Learning a fast iterative shrinkage thresholding network for inverse problems in imaging," *IEEE Trans. Med. Imag.*, vol. 40, no. 5, pp. 1329–1339, May 2021.
[21] B. K. Natarajan, "Sparse approximate solutions to linear systems," *SIAM J. Comput.*, vol. 24, no. 2, pp. 227–234, 1995.
[22] E. J. Candès, Y. C. Eldar, D. Needell, and P. Randall, "Compressed sensing with coherent and redundant dictionaries," *Appl. Comput. Harmon. Anal.*, vol. 31, no. 1, pp. 59–73, Jul. 2011.
[23] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. Simul.*, vol. 4, no. 4, pp. 1168–1200, 2005.

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.

[24] A. Géron, *Hands-on Machine Learning With Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*. Sebastopol, CA, USA: O'Reilly Media, 2017.
[25] M. Aharon and M. Elad, "Sparse and redundant modeling of image content using an image-signature-dictionary," *SIAM J. Imag. Sci.*, vol. 1, no. 3, pp. 228–247, 2008.
[26] N. Ricker, "The form and nature of seismic waves and the structure of seismogram," *Geophysics*, vol. 5, no. 4, pp. 348–366, 1940.
[27] D. Pereg and I. Cohen, "Seismic signal recovery based on Earth q model," *Signal Process.*, vol. 137, pp. 373–386, Aug. 2017.
[28] Y. Wang, "Frequencies of the Ricker wavelet," *Geophysics*, vol. 80, no. 2, pp. A31–A37, Mar. 2015.
[29] Y. Wang, "A stable and efficient approach of inverse Q filtering," *Geophysics*, vol. 67, no. 2, pp. 657–663, 2002.
[30] E. Kjartansson, "Constant Q-wave propagtion and attenuation," *J. Geophys. Res.*, no. 84, pp. 4737–4747, 1979.
[31] R. Sherif and L. Geldart, *Exploration Seismology*, 2nd ed. Cambridge, U.K.: Cambridge Univ. Press, 1983.
[32] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "ImageNet classification with deep convolutional neural networks," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1097–1105.
[33] J. Kormylo and J. Mendel, "Maximum likelihood detection and estimation of Bernoulli–Gaussian processes," *IEEE Trans. Inf. Theory*, vol. 28, no. 3, pp. 482–488, May 1982.
[34] D. Pereg, I. Cohen, and A. A. Vassiliou, "Three-dimensional sparse seismic deconvolution based on Earth Q model," *Signal Process.*, vol. 154, pp. 97–107, Jan. 2019.
[35] I. Cohen and R. R. Coifman, "Local discontinuity measures for 3-D seismic data," *Geophysics*, vol. 67, no. 6, pp. 1933–1945, 2002.
[36] C. Zhang and T. Ulrych, "Estimation of quality factors from CMP record," *Geophysics*, vol. 67, no. 5, pp. 1542–1547, 2002.
[37] A. Paszke *et al.*, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, vol. 32, H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, Eds. New York, NY, USA: Curran Associates, 2019, pp. 8024–8035.
[38] O. Yilmaz, *Seismic Data Analysis*. Houston, TX, USA: Society of Exploration Geophysicists, 2001.
[39] V. Papyan, J. Sulam, and M. Elad, "Working locally thinking globally: Theoretical guarantees for convolutional sparse coding," *IEEE Trans. Signal Process.*, vol. 65, no. 21, pp. 5687–5701, Nov. 2017.

Deborah Pereg received the B.Sc. degree *(summa cum laude)* in electrical engineering from Azrieli College of Engineering, Jerusalem, Israel, in 2012, and the M.Sc. and Ph.D. degrees from Technion-Israel Institute of Technology, Haifa, Israel, in 2017 and 2021, respectively.

Her research interests are in the areas of signal processing and machine learning.

Dr. Pereg was a recipient of the 2020 Jacobs-Qualcomm Fellowship for excellent Ph.D. students.

Israel Cohen (Fellow, IEEE) received the B.Sc. *(summa cum laude)*, M.Sc., and Ph.D. degrees in electrical engineering from Technion–Israel Institute of Technology, Haifa, Israel, in 1990, 1993, and 1998, respectively.

From 1990 to 1998, he was a Research Scientist with RAFAEL Research Laboratories, Israel Ministry of Defense, Haifa. From 1998 to 2001, he was a Post-Doctoral Research Associate with the Computer Science Department, Yale University, New Haven, CT, USA. In 2001, he joined the Electrical Engineering Department, Technion–Israel Institute of Technology, where he is the Louis and Samuel Seidan Professor in electrical and computer engineering. He is a coeditor of the Multichannel Speech Processing Section of the *Springer Handbook of Speech Processing* (Springer, 2008), and the coauthor of *Fundamentals of Signal Enhancement and Array Signal Processing* (Wiley-IEEE Press, 2018). His research interests are array processing, statistical signal processing, deep learning, analysis and modeling of acoustic signals, speech enhancement, noise estimation, microphone arrays, source localization, blind source separation, system identification, and adaptive filtering.

Dr. Cohen was a member of the IEEE Audio and Acoustic Signal Processing Technical Committee and the IEEE Speech and Language Processing Technical Committee. He was awarded the Muriel and David Jacknow Award for Excellence in Teaching in 2009, the Alexander Goldberg Prize for Excellence in Research in 2010, the SPS Signal Processing Letters Best Paper Award in 2014, and the Norman Seiden Prize for Academic Excellence in 2017. He was an Associate Editor of the IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE PROCESSING and IEEE SIGNAL PROCESSING LETTERS. He was also a Distinguished Lecturer of the IEEE Signal Processing Society.

Anthony A. Vassiliou (Member, IEEE) received the B.Sc. degree from the National Technical University of Athens, Athens, Greece, in 1981, and the M.Sc. and Ph.D. degrees from the University of Calgary, Calgary, AB, Canada, in 1984 and 1986, respectively.

He worked at Mobil Oil Research and Development from 1987 to 1992 as a Research Geophysicist and at Amoco Production Research from 1993 to 1998 as a Senior Research Geophysicist. He founded GeoEnergy, Inc., Houston, TX, USA, in early 1998 and he has been the GeoEnergy's CEO since the beginning of its operations. He has conducted research in inertial navigation, airborne gravity gradiometry, seismic tomography, seismic signal processing, and seismic imaging. His research interests are seismic depth imaging, velocity model building, seismic waveform inversion, and deep learning.

Authorized licensed use limited to: Technion Israel Institute of Technology. Downloaded on February 01,2022 at 09:00:43 UTC from IEEE Xplore. Restrictions apply.