Deep Unfolded Robust PCA with Application to Clutter Suppression in Ultrasound

Oren Solomon, *Student Member, IEEE,* Regev Cohen, *Student Member, IEEE,* Yi Zhang, Yi Yang, He Qiong, Jianwen Luo, Ruud J.G. van Sloun, *Member, IEEE*, and Yonina C. Eldar, *Fellow, IEEE*

***Abstract*—Contrast enhanced ultrasound is a radiation-free imaging modality which uses encapsulated gas microbubbles for improved visualization of the vascular bed deep within the tissue. It has recently been used to enable imaging with unprecedented subwavelength spatial resolution by relying on super-resolution techniques. A typical preprocessing step in super-resolution ultrasound is to separate the microbubble signal from the cluttering tissue signal. This step has a crucial impact on the final image quality. Here, we propose a new approach to clutter removal based on robust principle component analysis (PCA) and deep learning. We begin by modeling the acquired contrast enhanced ultrasound signal as a combination of a low rank and sparse components. This model is used in robust PCA and was previously suggested in the context of ultrasound Doppler processing and dynamic magnetic resonance imaging. We then illustrate that an iterative algorithm based on this model exhibits improved separation of microbubble signal from the tissue signal over commonly practiced methods. Next, we apply the concept of deep unfolding to suggest a deep network architecture tailored to our clutter filtering problem which exhibits improved convergence speed and accuracy with respect to its iterative counterpart. We compare the performance of the suggested deep network on both simulations and in-vivo rat brain scans, with a commonly practiced deep-network architecture and the fast iterative shrinkage algorithm, and show that our architecture exhibits better image quality and contrast.**

***Index Terms*—Ultrasound, Machine learning, Inverse methods, Neural network.**

I. Introduction

MEDICAL ultrasound (US) is a radiation-free imaging modality used extensively for diagnosis in a wide range of clinical segments such as radiology, cardiology, vascular, obstetrics and emergency medicine. Ultrasound-based imaging modalities include brightness, motion, Doppler, harmonic modes, elastography and more [1].

One important imaging modality is contrast-enhanced ultrasound (CEUS) [2] which allows the detection and visualization of blood vessels whose physical parameters such as relative blood volume (rBV), velocity, shape and density are associated with different clinical conditions [3]. CEUS uses encapsulated gas microbubbles as ultrasound contrast agents (UCAs) which are administrated intravenously and are similar in size to red blood cells and thus can flow throughout the vascular system [4]. Among its many applications, CEUS is used for imaging of perfusion at the capillary level [5, 6], for estimating blood velocity in small vessels arteriole by applying Doppler processing [7, 8] and for sub-wavelength vascular imaging [9–14].

A major challenge in ultrasonic vascular imaging such as CEUS is to suppress clutter signals stemming from stationary and slowly moving tissue as they introduce significant artifacts in blood flow imaging [15]. Over the past few decades several approaches have been suggested for clutter removal. The simplest method to remove tissue signal is to filter the ultrasonic signal along the temporal dimension using high-pass finite impulse response (FIR) or infinite impulse response (IIR) filters [16]. However, FIR filters need to have high order while IIR filters exhibit a long settling time which leads to a low number of temporal samples in each spatial location [17] when using focused transmission. The above methods rely on the assumption that tissue motion, if exists, is slow while blood flow is fast. This high-pass filtering approach is prone to failure in the presence of fast tissue motion, as in cardiology, or when imaging microvasculature in which blood velocities are low.

An alternative method for tissue suppression is second harmonic imaging [18], which separates the blood and tissue signals by exploiting the non-linear response of the UCAs to low acoustic pressures, compared with the mostly linear tissue response. This technique, however, limits the frame-rate of the US scanner, and does not remove the tissue signal completely, as tissue can also exhibit a nonlinear response.

The above techniques are based only on temporal information and neglect the high spatial coherence of the tissue, compared to the blood. To take advantage of these spatial characteristics of tissue, a method for clutter removal was presented in [19], based on the singular value decomposition (SVD) of the correlation matrix of successive temporal samples. SVD filtering operates by stacking the (typically beamformed) acquired frames as vectors in a matrix whose column index indicates frame number. Then, an SVD of the matrix is performed and the largest singular values, which correspond to the highly correlated tissue, are zeroed out. Finally, a new matrix is composed based on the remaining singular values and reshaped to produce the blood/UCA

O. Solomon and R. Cohen contributed equally to this work.

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 646804-ERC-COG-BNYQ.

O. Solomon (e-mail: orensol@campus.technion.ac.il), R. Cohen (e-mail: regev,cohen@campus.technion.ac.il) and Y. C. Eldar (e-mail: yonina@ee.technion.ac.il) are with the Department of Electrical Engineering, Technion-Israel Institute of Technology, Haifa 32000. Y. Zhang (e-mail: yizhang.ch.2015@gmail.com) is with the department of electrical engineering, Tsinghua University, Beijing 100084, China. Y. Yang, Q. He and J. Luo are with the Department of Biomedical Engineering, Tsinghua University, Beijing 100084, China. R. J. G. van Sloun (e-mail: R.J.G.v.Sloun@tue.nl) is with the Department of Electrical Engineering, Eindhoven University of Technology, Eindhoven, The Netherlands.

movie.

Several SVD-based techniques have been proposed [15, 20–23], such as down-mixing [15] for tissue motion estimation, adaptive clutter rejection for color flow proposed by Lovstakken *et al.* [24] and the principal component analysis (PCA) for blood velocity estimation presented in [25]. However, these methods are based on focused transmission schemes which limit the frame rate and the field of view. This in turn leads to a small number of temporal and spatial samples, reducing the effectiveness of SVD-based filtering. To overcome this limitation, SVD-based clutter removal was extended to ultrafast plane-wave imaging [13, 26–28], demonstrating substantially improved clutter rejection and microvascular extraction. This strategy gained a lot of interest in recent years and nowadays it is used in numerous ultrafast US imaging applications such as functional ultrasound [29, 30], super-resolution ultrasound localization microscopy [13, 14] and high-sensitivity microvessel perfusion imaging [26, 27].

A major limitation of SVD-based filtering is the requirement to determine a threshold which discriminates between tissue related and blood related singular values. The appropriate setting of this threshold is typically unclear, especially when the eigenvalue spectra of the tissue and contrast signals overlap. This threshold uncertainty motivates the use of a different model for the acquired data, one that can differentiate between tissue and contrast signals based on the spatio-temporal information, as well as additional information unique to the contrast signal - its sparse distribution in the imaging plane.

Here, we propose two main contributions. The first, is the adaptation of a new model for the tissue/contrast separation problem. We show that similar to other applications such as MRI [31] and recent US Doppler applications [32], we can decompose the acquired, beamformed US movie as a sum of a low-rank matrix (tissue) and a sparse outlier signal (UCAs). This decomposition is also known as robust principle component analysis (RPCA) [33]. We then propose to solve a convex minimization problem to retrieve the UCA signal, which leads to an iterative principal component pursuit (PCP) [33]. Second, we utilize recent ideas from the field of deep learning [34] to dramatically improve the convergence rate and image reconstruction quality of the iterative algorithm. We do so by unfolding [35] the algorithm into a fixed-length deep network which we term Convolutional rObust pRincipal cOmpoNent Analysis (CORONA). This approach harnesses the power of both deep learning and model-based frameworks, and leads to improved performance in various fields [36–40].

CORONA is trained on sets of separated tissue/UCA signals from both *in-vivo* and simulated data. Similar to [37], we utilize convolution layers instead of fully-connected (FC) layers, to exploit the shared spatial information between neighboring image pixels. Our training policy is a two stage process. We start by training the network on simulated data, and then train the resulting network on *in-vivo* data. This hybrid policy allows us to improve the network's performance and to achieve a fully-automated network, in which all the regularization parameters are also learned. We compare the performance of CORONA with the commonly practiced SVD approach, the iterative RPCA algorithm and an adaptation of the residual network (ResNet), which is considered to be one of the leading deep architectures for a wide variety of problems [41]. We show that CORONA outperforms all other approaches in terms of image quality and contrast.

Unfolding, or unrolling an iterative algorithm, was first suggested by Gregor and LeCun [35] to accelerate algorithm convergence. In the context of deep learning, an important question is what type of network architecture to use. Iterative algorithms provide a natural recurrent architecture, designed to solve a specific problem, such as sparse approximations, channel estimation [42] and more. The authors of [35] showed that by considering each iteration of an iterative algorithm as a layer in a deep network and subsequent concatenation of a few such layers it is possible to train such networks to achieve a dramatic improvement in convergence, i.e., to reduce the number of iterations significantly.

In the context of RPCA, a principled way to construct learnable pursuit architectures for structured sparse and robust low rank models was introduced in [36]. The proposed networks, derived from the iteration of proximal descent algorithms, were shown to faithfully approximate the solution of RPCA while demonstrating several orders of magnitude speed-up compared to standard optimization algorithms. However, this approach is based on a non-convex formulation in which the rank of the low-rank part (or an upper bound on it) is assumed to be known a priori. This poses a network design limitation, as the rank can vary between different applications or even different realizations of the same application, as in CEUS. Thus, for each choice of the rank upper bound, a new network needs to be trained, which can limit its applicability. In contrast, our approach does not require a-priori knowledge of the rank. Moreover, the use of convolutional layers exploits spatial invariance and facilitates our training process as it reduces the number of learnable parameters dramatically.

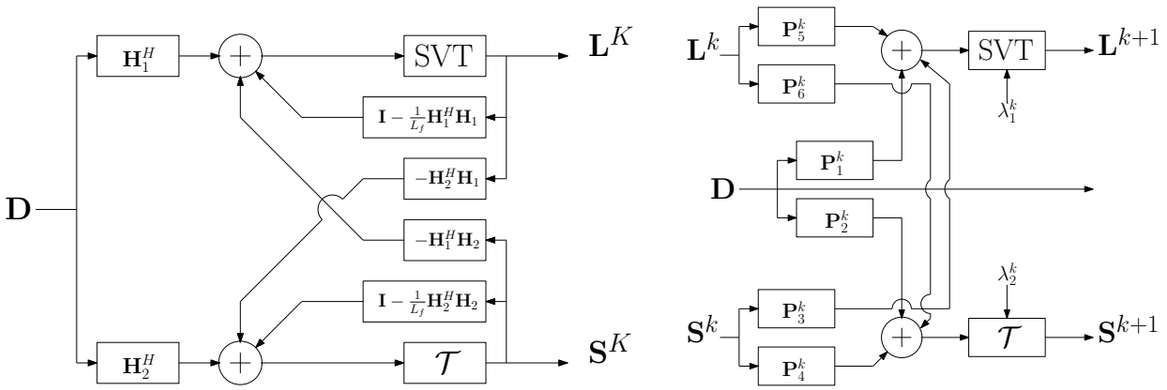
The rest of the paper is organized as follows. In Section II we introduce the mathematical formulation of the low-rank and sparse decomposition. Section III describes the protocol of the experiments and technical details regarding the realizations of CORONA and ResNet. Section IV presents *in-silico* as well as *in-vivo* results of both the iterative algorithm and the proposed deep networks. Finally, we discuss the results, limitations and further research directions in Section V.

Throughput the paper, x represents a scalar, $\mathbf{x}$ a vector, $\mathbf{X}$ a matrix and $\mathbf{I}_{N\times N}$ is the $N \times N$ identity matrix. The notation $||\cdot||_p$ represents the standard p-norm and $||\cdot||_F$ is the Frobenius norm. Subscript x_l denotes the lth element of $\mathbf{x}$ and $\mathbf{x}_l$ is the lth column of $\mathbf{X}$. Superscript $\mathbf{x}^{(p)}$ represents $\mathbf{x}$ at iteration p, T^* denotes the adjoint of T, and $\bar{\mathbf{A}}$ is the complex conjugate of $\mathbf{A}$.

II. Deep learning strategy for RPCA in US

A. Problem formulation

We start by providing a low-rank plus sparse (L+S) model for the acquired US signal. In US imaging, typically a series of pulses are transmitted to the imaged medium. The resulting echoes from the medium are received in each transducer element and then combined in a process called beamforming



(a) Iterative algorithm for sparse and low-rank separation.

(b) Single layer from the corresponding unfolded algorithm.

Fig. 1: Architecture comparison between the iterative algorithm applied for K iterations (panel (a)) and its unfolded counterpart (panel (b)). The learned network in panel (b) draws its architecture from the iterative algorithm, and is trained on examples from a given dataset. In both panels, $\mathbf{D}$ is the input measurement matrix, and $\mathbf{S}_k$ and $\mathbf{L}_k$ are the estimated sparse and low-rank matrices in each iteration/layer, respectively.

to produce a focused image. As presented in [43], after demodulation the complex analytical (IQ) signal can be represented as

$$D(x,z,t) = I(x,z,t) + iQ(x,z,t),$$

where $I(x,z,t)$ and $Q(x,z,t)$ are the in-phase and quadrature components of the demodulated signal, x, z are the vertical and axial coordinates, and t indicates frame number. The signal $D(x,z,t)$ is a sum of echoes returned from the blood/CEUS signal $S(x,z,t)$ as well as from the tissue $L(x,z,t)$, contaminated by additive noise $N(x,z,t)$

$$D(x,z,t) = L(x,z,t) + S(x,z,t) + N(x,z,t).$$

Acquiring a series of movie frames $t = 1,\ldots,T$, and stacking them as vectors in a matrix $\mathbf{D}$, leads to the following model

$$\mathbf{D} = \mathbf{L} + \mathbf{S} + \mathbf{N}. \tag{1}$$

In (1), we assume that the tissue matrix $\mathbf{L}$ can be described as a low-rank matrix, due to its high spatio-temporal coherence. The CEUS echoes matrix $\mathbf{S}$ is assumed to be sparse, as blood vessels typically sparsely populate the imaged medium. Assuming that each movie frame is of size $M \times M$ pixels, the matrices in (1) are of size $M^2 \times T$. From here on, we consider a more general model, in which the acquired matrix $\mathbf{D}$ is composed as

$$\mathbf{D} = \mathbf{H}_1\mathbf{L} + \mathbf{H}_2\mathbf{S} + \mathbf{N}, \tag{2}$$

with $\mathbf{H}_1$ and $\mathbf{H}_2$ being the measurement matrices of appropriate dimensions. The model (2) can also be applied to MR imaging, video compression and additional US applications, as we discuss in Section V. Our goal is to formalize a minimization problem to extract both $\mathbf{L}$ and $\mathbf{S}$ from $\mathbf{D}$ under the corresponding assumptions of L+S matrices.

Similar to [31], we propose solving the following minimization problem

$$\min_{\mathbf{L},\mathbf{S}} \frac{1}{2}||\mathbf{D} - (\mathbf{H}_1\mathbf{L} + \mathbf{H}_2\mathbf{S})||_F^2 + \lambda_1||\mathbf{L}||_* + \lambda_2||\mathbf{S}||_{1,2}, \tag{3}$$

where $||\cdot||_*$ stands for the nuclear norm, which sums the singular values of $\mathbf{L}$, and $||\cdot||_{1,2}$ is the mixed $l_{1,2}$ norm, which sums the l_2 norms of each row of $\mathbf{S}$. We use the mixed $l_{1,2}$ norm since the pattern of the sparse outlier (blood or CEUS signal) is the same between different frames, and ultimately corresponds to the locations of the blood vessels, which are assumed to be fixed, or change very slowly during the acquisition period. The nuclear norm is known to promote low-rank solutions, and is the convex relaxation of the non-convex rank minimization constraint [44].

By defining

$$\mathbf{X} = \begin{bmatrix} \mathbf{L} \\ \mathbf{S} \end{bmatrix}, \ \mathbf{P}_1 = \begin{bmatrix} \mathbf{I} \\ \mathbf{0} \end{bmatrix}, \ \mathbf{P}_2 = \begin{bmatrix} \mathbf{0} \\ \mathbf{I} \end{bmatrix}$$

and $\mathbf{A} = [\mathbf{H}_1, \mathbf{H}_2]$, (3) can be rewritten as

$$\min_{\mathbf{L},\mathbf{S}} \frac{1}{2}||\mathbf{D} - \mathbf{A}\mathbf{X}||_F^2 + h(\mathbf{X}), \tag{4}$$

where $h(\mathbf{X}) = \sum_{i=1}^{2} \lambda_i \rho_i(\mathbf{P}_i\mathbf{X})$ with $\rho_1 = ||\cdot||_*$ and $\rho_2 = ||\cdot||_{1,2}$. The minimization problem (4) is a regularized least-squares problem, for which numerous numerical minimization algorithms exist. Specifically, the (fast) iterative shrinkage/thresholding algorithm, (F)ISTA, [45, 46] involves finding the *Moreau's proximal* (prox) mapping [47, 48] of h, defined as

$$\text{prox}_h(\mathbf{X}) = \underset{\mathbf{U}}{\text{argmin}} \left\{ h(\mathbf{U}) + \frac{1}{2}||\mathbf{U} - \mathbf{X}||_F^2 \right\}. \tag{5}$$

Plugging the definition of $\mathbf{X}$ into (5) yields

$$\text{prox}_h(\mathbf{X}) = \underset{\mathbf{U}_1,\mathbf{U}_2}{\text{argmin}} \left\{ \lambda_1\rho_1(\mathbf{U}_1) + \frac{1}{2}||\mathbf{U}_1 - \mathbf{L}||_F^2 \right.$$

$$+\lambda_2\rho_2(\mathbf{U}_2)+\frac{1}{2}||\mathbf{U}_2-\mathbf{S}||_F^2\Big\}.$$

Since $\text{prox}_h(\mathbf{X})$ is separable in $\mathbf{L}$ and $\mathbf{S}$, it holds that

$$\text{prox}_h(\mathbf{X})=\begin{bmatrix}\text{prox}_{\rho_1}(\mathbf{L})\\ \text{prox}_{\rho_2}(\mathbf{S})\end{bmatrix}=\begin{bmatrix}\text{SVT}_{\lambda_1}(\mathbf{L})\\ \mathcal{T}_{\lambda_2}(\mathbf{S})\end{bmatrix}. \quad (6)$$

The operators

$$\mathcal{T}_\alpha(\mathbf{x})=\max(0,1-\alpha/||\mathbf{x}||_2)\mathbf{x}$$

and

$$\text{SVT}_\alpha(\mathbf{X})=\mathbf{U}\text{diag}(\max(0,\sigma_i-\alpha))\mathbf{V}^H,\ i=1,\ldots,r$$

are the mixed $l_{1/2}$ soft thresholding [45] and singular value thresholding [49] operators. Here $\mathbf{X}$ is assumed to have an SVD given by $\mathbf{X}=\mathbf{U}\mathbf{\Sigma}\mathbf{V}^H$ with $\mathbf{\Sigma}=\text{diag}(\sigma_i,\ldots,\sigma_r)$, a diagonal matrix of the eigenvalues of $\mathbf{X}$. The proximal mapping (6) is applied in each iteration to the gradient of the quadratic part of (4), given by

$$g(\mathbf{X})=\frac{d}{d\mathbf{X}}\frac{1}{2}||\mathbf{D}-\mathbf{A}\mathbf{X}||_F^2=\mathbf{A}^H(\mathbf{A}\mathbf{X}-\mathbf{D}),$$

and more specifically,

$$\begin{bmatrix}\frac{d}{d\mathbf{L}}\\ \frac{d}{d\mathbf{S}}\end{bmatrix}=\begin{bmatrix}\mathbf{H}_1^H(\mathbf{H}_1\mathbf{L}+\mathbf{H}_2\mathbf{S}-\mathbf{D})\\ \mathbf{H}_2^H(\mathbf{H}_1\mathbf{L}+\mathbf{H}_2\mathbf{S}-\mathbf{D})\end{bmatrix}.$$

The general iterative step of ISTA applied to minimizing (3) (L+S ISTA) is thus given by

$$\mathbf{X}^{k+1}=\text{prox}_h\left(\mathbf{X}^k-\frac{1}{L_f}g(\mathbf{X}^k)\right),$$

or

$$\begin{aligned}\mathbf{L}^{k+1}&=\text{SVT}_{\lambda_1/L_f}\left\{\left(\mathbf{I}-\tfrac{1}{L_f}\mathbf{H}_1^H\mathbf{H}_1\right)\mathbf{L}^k-\mathbf{H}_1^H\mathbf{H}_2\mathbf{S}^k+\mathbf{H}_1^H\mathbf{D}\right\}\\ \mathbf{S}^{k+1}&=\mathcal{T}_{\lambda_2/L_f}\left\{\left(\mathbf{I}-\tfrac{1}{L_f}\mathbf{H}_2^H\mathbf{H}_2\right)\mathbf{S}^k-\mathbf{H}_2^H\mathbf{H}_1\mathbf{L}^k+\mathbf{H}_2^H\mathbf{D}\right\}\end{aligned}, \quad (7)$$

where L_f is the Lipschitz constant of the quadratic term of (4), given by the spectral norm of $\mathbf{A}^H\mathbf{A}$.

The L+S ISTA algorithm for minimizing (3) is summarized in Algorithm 1. The diagram in Fig. 1(a) presents the iterative algorithm, which relies on knowledge of $\mathbf{H}_1,\mathbf{H}_2$ and selection of λ_1 and λ_2.

Algorithm 1 L+S ISTA for minimizing (3)

Require: $\mathbf{D}$, $\lambda_1>0,\lambda_2>0$, maximum iterations K_{max}
  Initialize $\mathbf{S}=\mathbf{L}=\mathbf{0}$ and $k=1$
  while $k\le K_{\text{max}}$ or stopping criteria not fulfilled **do**
    1: $\mathbf{G}_{1_k}=\left(\mathbf{I}-\frac{1}{L_f}\mathbf{H}_1^H\mathbf{H}_1\right)\mathbf{L}^k-\mathbf{H}_1^H\mathbf{H}_2\mathbf{S}^k+\mathbf{H}_1^H\mathbf{D}$
    2: $\mathbf{G}_{2_k}=\left(\mathbf{I}-\frac{1}{L_f}\mathbf{H}_2^H\mathbf{H}_2\right)\mathbf{S}^k-\mathbf{H}_2^H\mathbf{H}_1\mathbf{L}^k+\mathbf{H}_2^H\mathbf{D}$
    3: $\mathbf{L}^{k+1}=\text{SVT}_{\lambda_1/L_f}\{\mathbf{G}_{1_k}\}$
    4: $\mathbf{S}^{k+1}=\mathcal{T}_{\lambda_2/L_f}\{\mathbf{G}_{2_k}\}$
    5: $k\leftarrow k+1$
  end while
  return $\mathbf{S}_{K\text{max}},\mathbf{L}_{K\text{max}}$

The dynamic range between returned echoes from the tissue and UCA/blood signal can range from 10dB to 60dB. As this dynamic range expands, more iterations are required to achieve good separation of the signals. This observation motivates the pursuit of a fixed complexity algorithm. In the next section we propose CORONA which is based on unfolding Algorithm 1. Background on learning fast sparse approximations is given in Section I of the supplementary materials.

B. Unfolding the iterative algorithm

An iterative algorithm can be considered as a recurrent neural network, in which the kth iteration is regarded as the kth layer in a feedforward network [36]. To form a convolutional network, one may consider convolutional layers instead of matrix multiplications. With this philosophy, we form a network from (7) by replacing each of the matrices dependent on $\mathbf{H}_1$ and $\mathbf{H}_2$ with convolution layers (kernels) $\mathbf{P}_1^k,\ldots,\mathbf{P}_6^k$ of appropriate sizes. These will be learned from training data. Contrary to previous works in unfolding RPCA which considered training fully connected (FC) layers [36], we employ convolution kernels in our implementation which allows us to achieve spatial invariance while reducing the number of learned parameters considerably.

The kernels as well as the regularization parameters λ_1^k and λ_2^k are learned during training. By doing so, the following equations for the kth layer are obtained

$$\begin{aligned}\mathbf{L}^{k+1}&=\text{SVT}_{\lambda_1^k}\left\{\mathbf{P}_5^k*\mathbf{L}^k+\mathbf{P}_3^k*\mathbf{S}^k+\mathbf{P}_1^k*\mathbf{D}\right\},\\ \mathbf{S}^{k+1}&=\mathcal{T}_{\lambda_2^k}\left\{\mathbf{P}_6^k*\mathbf{L}^k+\mathbf{P}_4^k*\mathbf{S}^k+\mathbf{P}_2^k*\mathbf{D}\right\},\end{aligned}$$

with $*$ being a convolution operator. The latter can be considered as a single layer in a multi-layer feedforward network, which we refer to as CORONA: Convolutional rObust pRincipal cOmpoNent Analysis. A diagram of a single layer from the unfolded architecture is given in Fig. 1(b), where the supposedly known model matrices were replaced by the 2D convolution kernels $\mathbf{P}_1^k,\ldots,\mathbf{P}_6^k$, which are learned as part of the training process of the overall network.

In many applications, the recovered matrices $\mathbf{S}$ and $\mathbf{L}$ represent a 3D volume, or movie, of dynamic objects imposed on a (quasi) static background. Each column in $\mathbf{S}$ and $\mathbf{L}$ is a vectorized frame from the recovered sparse and low-rank movies. Thus, we consider in practice our data as a 3D volume and apply 2D convolutions. The SVT operation (which has similar complexity as the SVD operation) at the kth layer is performed after reshaping the input 3D volume into a 2D matrix, by vectorizing and column-wise stacking each frame.

The thresholding coefficients are learned independently for each layer. Given the kth layer, the actual thresholding values for both the SVT and soft-thresholding operations are given by $\text{thr}_L^k=\sigma(\lambda_L^k)\cdot a_L\cdot\max(L^k)$ and $\text{thr}_S^k=\sigma(\lambda_S^k)\cdot a_S\cdot\text{mean}(S^k)$ respectively, where $\sigma(x)=1/(1+\exp(-x))$ is a sigmoid function, a_L and a_S are fixed scalars (in our application we chose $a_L=0.4$ and $a_S=1.8$) and λ_L^k and λ_S^k are learned in each layer by the network.

C. Training CORONA

CORONA is trained using back-propagation in a supervised manner. Generally speaking, we obtain training examples $\mathbf{D}_i$ and corresponding sparse $\hat{\mathbf{S}}_i$ and low-rank $\hat{\mathbf{L}}_i$ decompositions. In practice, $\hat{\mathbf{S}}_i$ and $\hat{\mathbf{L}}_i$ can either be obtained from simulations

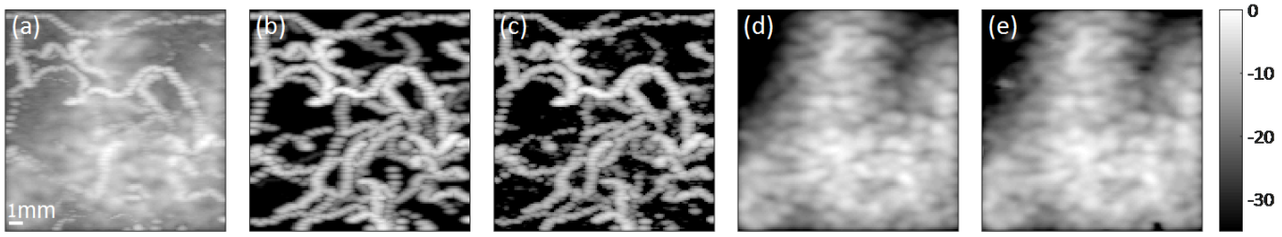


Fig. 2: Simulation results of CORONA. (a) MIP image of the input movie, composed from 50 frames of simulated UCAs cluttered by tissue. (b) Ground-truth UCA MIP image. (c) Recovered UCA MIP image via CORONA. (d) Ground-truth tissue MIP image. (d) Recovered tissue MIP image via CORONA. Color bar is in dB.

or by decomposing $\mathbf{D}_i$ using iterative algorithms such as FISTA [50]. The loss function is chosen as the sum of the mean squared errors (MSE) between the predicted $\mathbf{S}$ and $\mathbf{L}$ values of the network and $\hat{\mathbf{S}}_i$, $\hat{\mathbf{L}}_i$, respectively,

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{2N}\sum_{i=1}^{N}||f_S(\mathbf{D}_i,\boldsymbol{\theta}) - \hat{\mathbf{S}}_i||_F^2 + \frac{1}{2N}\sum_{i=1}^{N}||f_L(\mathbf{D}_i,\boldsymbol{\theta}) - \hat{\mathbf{L}}_i||_F^2. \tag{8}$$

In the latter equation, f_S/f_L is the sparse/low-rank output of CORONA with learnable parameters $\boldsymbol{\theta} = \{\mathbf{P}_1^k,\ldots,\mathbf{P}_6^k,\lambda_1^k,\lambda_2^k\}$, $k = 1,\ldots,K$, where K is the number of chosen layers.

Training a deep network typically requires a large amount of training examples, and in practice, US scans of specific organs are not available in abundance. To be able to train CORONA, we thus rely on two strategies: patch-based analysis and simulations. Instead of training the network over entire scans, we divide the US movie used for training into 3D patches (axial coordinate, lateral coordinate and frame number). Then we apply Algorithm 1 on each of these 3D patches. The SVD operations in Algorithm 1 become computationally tractable since we work on relatively small patches. The resulting separated UCA movie is then considered as the desirable outcome of the network and the network is trained over these pairs of extracted 3D patches from the acquired movie, and the resulting reconstructed UCA movies. In practice, the CEUS movie used for training is divided into 3D patches of size $32 \times 32 \times 20$ (32×32 pixels over 20 consecutive frames) with 50% overlap between neighboring patches. The regularization parameters of Algorithm 1, λ_1 and λ_2 are chosen empirically, but are chosen once for all the extracted patches.

In Section VI of the supplementary materials, we provide a detailed description of how the simulations of the UCA and tissue movies were generated. In particular, we detail how individual UCAs were modeled and propagated in the imaging plane, and describe the cluttering tissue signal model. We then demonstrate the importance of training on both simulations and *in-vivo* data in Section IV of the supplementary materials.

III. Experiments

The brains of two rats were scanned using a Vantage 256 system (Verasonics Inc., Kirkland, WA, USA). An L20-10 probe was utilized, with a central frequency of 15MHz. The rats underwent craniotomy after anesthesia to obtain an imaging window of $6 \times 2\text{mm}^2$. A bolus of 100μL SonoVue$^{\text{TM}}$ (Bracco, Milan, Italy) contrast agent, diluted with normal saline with a ratio of 1:4, was administered intravenously to the rats tail vein. Plane-wave compounding of five steering angles (from $-12°$ to $12°$, with a step of $6°$) was adopted for ultrasound imaging. For each rat, over 6000 consecutive frames were acquired with a frame rate of 100Hz. 300 frames with relatively high B-mode intensity were manually selected for data processing in this work.

In recent years, several deep learning based architectures have been proposed and applied successfully to classification problems. One such approach is the residual network, or ResNet [41]. ResNet utilizes convolution layers, along with batch normalization and skip connections, which allow the network to avoid vanishing gradients and reduce the overall number of network parameters.

To compare with CORONA, we implemented ResNet using complex convolutions for the tissue clutter suppression task. The network does not recover the tissue signal, as CORONA, but only the UCA signal. In Section IV and in the supporting materials file, we compare both architectures and assess the advantages and disadvantages of each network. In Section IV-B, we show that CORONA outperforms ResNet in terms of image quality (contrast) of the CEUS signal.

Both ResNet and CORONA were implemented in Python 3.5.2, using the PyTorch 0.4.1 package. CORONA consists of 10 layers. First three layers used convolution filters of size $5 \times 5 \times 1$ with stride $(1,1,1)$, padding $(2,2,0)$ and bias, while the last seven layers used filters of size $3 \times 3 \times 1$ with stride $(1,1,1)$, padding $(1,1,0)$ and bias. Training was performed using the ADAM optimizer with a learning rate of 0.002. For the *in-vivo* experiments in Section IV, we trained the network over 2400 simulated training pairs and additional 2400 *in-vivo* pairs taken only from the first rat. Training pairs were generated from the acquired US clips, after dividing each clip to $32 \times 32 \times 20$ patches. We then applied Algorithm 1 for each patch with $\lambda_1 = 0.02$, $\lambda_2 = 0.001$ and $D_{\text{max}} = 30000$ iterations to obtain the separated UCA signal for the training process. Algorithm 1 was implemented in MATLAB (Mathworks Inc.) and was applied to the complex-valued IQ signal. PyTorch performs automatic differentiation and back-propagation using the Autograd functionality, and version 0.4.1 also supports back-propagation through SVD, but only

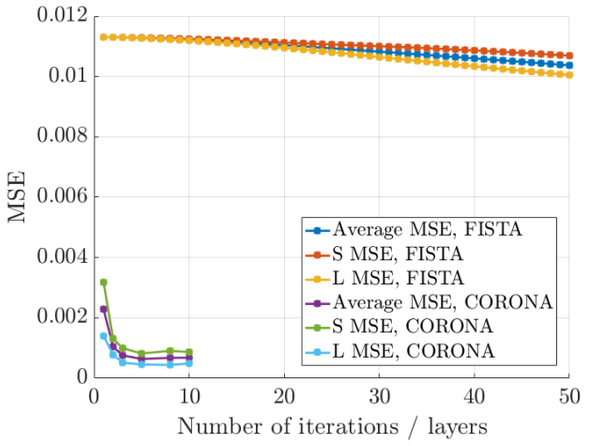


Fig. 3: MSE plot for the FISTA algorithm and CORONA as a function of the number of iterations/layers.

for real valued numbers. Thus, complex valued convolution layers and SVD operations were implemented.

IV. Results

A. Simulation results

In this section we provide reconstruction results for CORONA applied to a simulated dataset, and trained on simulations. Figure 2 presents reconstruction results of the UCA signal $\mathbf{S}$ and the low-rank tissue $\mathbf{L}$ against the ground truth images. Panel (a) shows a representative image in the form of maximum intensity projection (MIP)[1] of the input cluttered movie (50 frames). It is evident that the UCA signal, depicted as randomly twisting lines, is masked considerably by the simulated tissue signal. Panel (b) illustrates the ground truth MIP image of the UCA signal, while panel (c) presents the MIP image of the recovered UCA signal via CORONA. Panels (d) and (e) show MIP images of the ground truth and CORONA recovery, respectively.

Observing all panels, it is clear that CORONA is able to recover reliably both the UCA signal and the tissue signal. Section II in the supporting materials provides additional simulation results, showing also the recovered UCA signal by ResNet. Although qualitatively ResNet manages to recover well the UCA signal, its contrast is lower than the contrast of the CORONA recovery, which presents a clearer depiction of the random vascular structure of the simulation. Moreover, ResNet does not recover the tissue signal, while CORONA does.

As CORONA draws its architecture from the iterative ISTA algorithm, our second aim in this section is to assess the performance of both CORONA and the FISTA algorithm by calculating the MSE of each method as a function of iteration/layer number. Each layer in CORONA can be thought of as an iteration in the iterative algorithm. To that end, we next quantify the MSE over the simulated validation batch (sequence of 100 frames) as a function of layer number (CORONA) and iteration number (FISTA), as presented in Fig. 3. For both methods, the MSE for the recovered sparse part (UCA signal) $\mathbf{S}$ and the low-rank part (tissue signal) $\mathbf{L}$ were calculated as a function of iteration/layer number, as well as the average MSE of both parts, according to (8) ($\alpha = 0.5$). For each layer number, we constructed an unfolded network with that number of layers, and trained it for 50 epochs on simulated data only.

Observing Fig. 3, it is clear that even when considering CORONA with only 1 layer, its performance in terms of MSE in an order of magnitude better than FISTA applied with 50 iterations. Adding more layers improves the CORONA MSE, though after 5 layers, the performance remains roughly the same. Figure 3 also shows that a clear decreasing trend is present for the FISTA MSE, however a dramatic increase in the number of iterations is required by FISTA to achieve the same MSE values.

B. In-vivo experiments

We now proceed to demonstrate the performance of CORONA on *in-vivo* data. As was described in Section III, CORONA was trained on both simulated and experimental data. In Fig. 4, panel (a) depicts SVD based separation of the CEUS signal, panel (b) shows the FISTA based separation and panel (c) shows the result of CORONA. The lower panels of Fig. 4 also compare the performance of the trained ResNet (panel (f)) on the *in-vivo* data as well as provide additional comparison to the commonly used wall filtering. Specifically, we use a 6th order Butterworth filter with two cutoff frequencies of 0.2π (panel (d)) and 0.9π (panel (e)) radians/samples. Two frequencies were chosen which represent two scenarios. The cutoff frequency of the recovery in panel (d) was chosen to suppress as much tissue signal as possible, without rejecting slow moving UCAs. In panel (e), a higher frequency was chosen, to suppress the slow moving tissue signal even further, but as can be seen, at a cost of removing also some of the slower bubbles. The result is a less consistent vascular image. Visually judging, all panels of Fig. 4 shows that ResNet outperforms both the SVD and wall filtering approaches. However, a more careful observation shows that the ResNet output, although more similar to CORONA's output, seems more grainy and less smooth than CORONA's image. CORONA's recovery exhibits the highest contrast, and produces the best visual.

In each panel, the green and red boxes indicate selected areas, whose enlarged views are presented in the corresponding green and red boxes below each panel. Visual inspection of the panels (a)-(f) shows that FISTA, ResNet and CORONA achieve CEUS signal separation which is less noisy than the naive SVD approach and wall filtering. Considering the enlarged regions below the panels further supports this conclusion, showing better contrast of the FISTA and deep networks outputs. The enlarged panels below panels (d) and (e) show that indeed, as the cutoff frequency of the wall filter is increased, slow moving UCAs are also filtered out. Both deep networks exhibit higher contrast than the other approaches.

[1]In order to present a single representative image, we take the pixel-wise maximum from each movie. This process is also referred to as maximum intensity projection, and is a common method to visualize CEUS images.

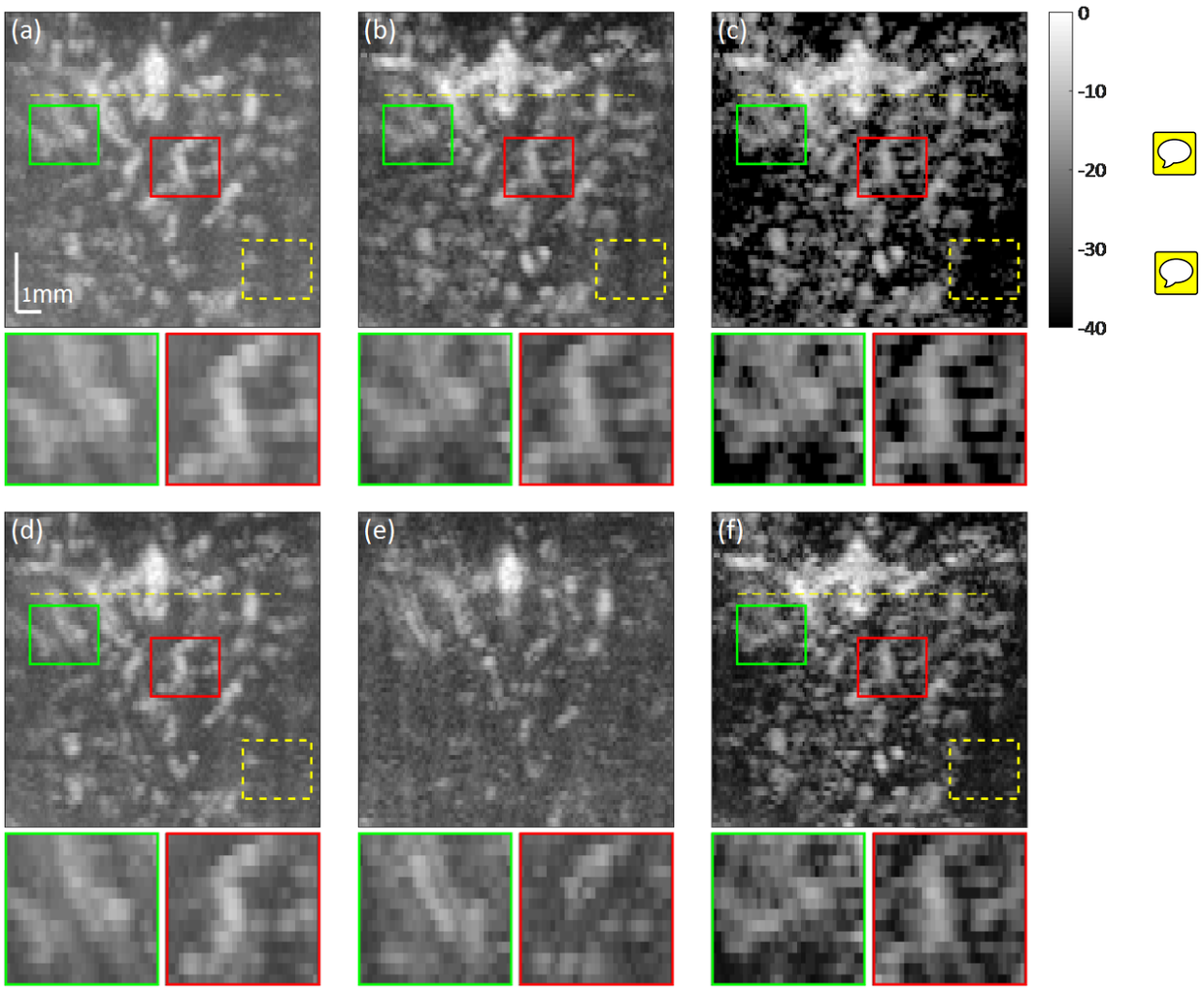


Fig. 4: Recovery of *in-vivo* CEUS signal depicting rat brain vasculature. (a) SVD based separation. (b) L+S FISTA separation. (c) Deep network separation, with the unfolded architecture of the FISTA algorithm. (d) Wall filtering with cutoff frequency of 0.2π (e) Wall filtering with cutoff frequency of 0.9π (f) ResNet. Color bar is in dB.

TABLE I: CNR values for the selected green and red rectangles of Fig. 4, as compared with the dashed yellow background rectangle in each corresponding panel. All values are in dB.

	SVD	**Wall filter**	**FISTA**	**ResNet**	**Unfolded**
Green box	-1.65	-2.02	-1.67	-2.17	-0.3
Red box	-4.8	-5.55	-3.52	-2.95	-1.13

TABLE II: CR values for the selected green and red rectangles of Fig. 4, as compared with the dashed yellow background rectangle in each corresponding panel. All values are in dB.

	SVD	**Wall filter**	**FISTA**	**ResNet**	**Unfolded**
Green box	4.68	4.5	5.52	7.92	15.24
Red box	4.56	4.1	5.24	7.55	14.88

To further quantify the performance of each method, we provide two metrics to assess the contrast ratio of their outputs, termed contrast to noise ratio (CNR) and contrast ratio (CR).

CNR is calculated between a selected patch, e.g. the red or green boxes in panels (a)-(f) and a reference patch, marked by the dashed yellow patches, which represents the background, for the same image. That is, for each panel we estimate the CNR values of the red - yellow and green - yellow boxes, where μ_s is the mean of the red/green box with variance σ_s^2 and μ_b is the mean of the dashed yellow patch with variance σ_b^2. The CNR is defined as

$$\text{CNR} = \frac{|\mu_s - \mu_b|}{\sqrt{\sigma_s^2 + \sigma_b^2}}.$$

Similarly, the CR is defined as

$$\text{CR} = \frac{\mu_s}{\mu_b}.$$

Table I and Table II provide the calculated CNR and CR values of each method, respectively.

In both metrics, higher values imply higher contrast ratios, which suggest better noise suppression and better signal

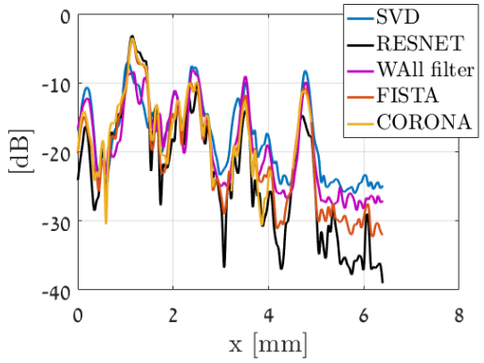


Fig. 5: Intensity profiles across the dashed yellow lines in panels (a)-(c) of Fig. 4. Regions in which CORONAs' curve is missing indicate a value of $-\infty$. Values are in dB.

depiction. Considering both tables, CORONA outperforms all other approaches. In most cases, its performance is an order of magnitude better than that SVD. The CR values of ResNet are also better than the baseline SVD, though lower than those of CORONA. Its CNR values however, are not always higher than those of the SVD. In terms of CR, the FISTA results are better than those of the SVD filter, though lower than the deep-learning based approaches. In terms of CNR, for the green box, FISTA is comparable to SVD and better than ResNet, while for the red box, its performance is the worst.

Both metrics support the previous conclusions, that by combining a proper model to the separation problem with a data-driven approach leads to improved separation of UCA and tissue signals, as well as noise reduction as compared to the popular SVD approach.

Finally, we also provide intensity cross-sections, taken along the horizontal yellow dashed line for each method, as presented in Fig. 5. Considering the intensity cross-section of Fig. 5, it is evident that all methods reconstruct the peaks with good correspondence. The FISTA and deep-learning networks' profiles exhibit higher contrast than the SVD and wall filter (deeper "cavities"). In some areas, the unfolded (yellow) profiles seems to vanish. This is because the attained value is $-\infty$. The supporting materials file contains additional comparisons. Section III presents the training and validation losses of the networks, as well as the evolution of the regularization coefficients of CORONA as a function of epoch number. Section IV discusses the importance of training the networks on both simulations and *in-vivo data* when applying CORONA on *in-vivo* experiments, while Section V presents the training and execution times for both networks.

V. Discussion and conclusions

In this work, we proposed a low-rank plus sparse model for tissue/UCA signal separation, which exploits both spatio-temporal relations in the data, as well as the sparse nature of the UCA signal. This model leads to a solution in the form of an iterative algorithm, which outperforms the commonly practiced SVD approach. We further suggested to improve both execution time and reconstructed image quality by unfolding the iterative algorithm into a deep network, referred to as CORONA. The proposed architecture utilizes convolution layers instead of FC layers and a hybrid simulation-*in-vivo* training policy. Combined, these techniques allow CORONA to achieve improved performance over its iterative counterpart, as well as over other popular architectures, such as ResNet. We demonstrated the performance of all methods on both simulated and *in-vivo* datasets, showing improved vascular depiction in a rat's brain.

We conclude by discussing several points, regarding the performance and design of deep-learning based networks. First, we attribute the improved performance over the commonly practiced SVD filtering, wall filtering and FISTA to two main reasons. The first, is the fact that for application on *in-vivo* data, the networks are trained based on both *in-vivo* data and simulated data. The simulated data provides the networks with an opportunity to learn from "perfect" examples, without noise and with absolute separation of UCAs and their surroundings. In Section IV of the supplementary materials we show the effect on recovery when the network is trained with and without experimental data. The iterative algorithm, on the other hand, cannot learn or improve its performance on the *in-vivo* data from the simulated data. The second, is the fact that both networks rely on 2D complex convolutions. Contrary to FC layers, convolution layers reduce the number of learnable parameters considerably, thus help avoid over-fitting and achieve good performance even when the training sets are relatively low. Moreover, convolutions offer spatial invariance, which allows the network to capture spatially translated UCAs.

Focusing on patch-based training (Section II-C) over entire image training has several benefits. UCAs are used to image blood vessels, and as such entire images will include implicitly blood vessel structure. Thus, training over entire images may result in the network being biased towards the vessel trees presented in the (relatively small) training cohort. On the other hand, small patches are less likely to include meaningful structure, hence training on small patches will be less likely to bias the network towards specific blood vessel structures and enable the network to generalize better. Furthermore, as FISTA and CORONA employ SVD operations, processing the data in small batches improves execution time [27, 51].

Second, as was mentioned in the introduction, in the context of RPCA, a principled way to construct learnable pursuit architectures for structured sparse and robust low rank models was introduced in [36]. The proposed network was shown to faithfully approximate the RPCA solution with several orders of magnitude speed-up compared to its standard optimization algorithm counterpart. However, this approach is based on a non-convex formulation of the nuclear norm in which the rank (or an upper bound of it) is assumed to be known a priori.

The main idea in [36] is to majorize the non-differentiable nuclear norm with a differentiable term, such that the low-rank matrix is factorized as a product of two matrices, $\mathbf{L} = \mathbf{AB}$, where $\mathbf{A} \in \mathbb{R}^{n \times q}$ and $\mathbf{B} \in \mathbb{R}^{q \times m}$. Using this kind of factorization alleviates the need to compute the SVD product, but introduces another unknown parameter q which needs to

be set (typically by hand), and corresponds to the rank of the low-rank matrix. This poses a network design limitation, as the rank can vary between different applications or even different realizations of the same application, requiring the network to be re-trained per each new choice of q.

In fact, this is the same rank-thresholding parameter as in the standard SVD filtering technique, which we want to avoid hand-tuning. Moreover, this kind of factorization leads to a non-convex minimization problem, whose globally optimal stationary points depend on the choice of the regularization parameter λ_*. Since typically these parameters are chosen empirically, a wrong choice of λ_* may lead to suboptimal reconstruction results of the RPCA problem, which are then used as training data for the fixed complexity learned algorithm. Since we operate on the original convex problem, we train against optimal reconstruction results of the RPCA algorithm, without the need to a-priori estimate the low-rank degree, q.

Third, currently CORONA and ResNet offer a trade-off between them. By relying on convolutions, CORONA is trained with a considerable lower number of parameters (314 for 1 layer, 1796 for 10 layers) than the ResNet (25378). CORONA outperforms ResNet in both visual quality and quantifiable metrics, as presented in Section IV. However, its training and execution times are slower (see Section V in the supporting materials file). This performance-runtime trade-off is attributed to the fact that CORONA relies on SVD decomposition in each layer, which is a relatively computationally demanding operation. However, it allows the network to learn the rank of the low-rank matrix, without the need to upper bound it and restrict the architecture of the network. Incorporation of fast approximations for SVD computations, such as truncated or random SVD [51–54], can potentially expedite the network's performance and achieve faster execution than ResNet. It is also important to keep in mind that ResNet does not recover the tissue signal, only the UCA signal. In some applications, such as super-resolution CEUS imaging over long time durations, the tissue signal is used to correct for motion artifacts.

On a final note, the proposed iterative and deep methods were demonstrated on the extraction of CEUS signal from an acquired IQ movie, but in principle can also be applied to dynamic MRI sequences, as well as to the separation of blood from tissue, e.g. for Doppler processing. In the latter case, the dynamic range between the tissue signal and the blood signal will be greater than that of the tissue and UCA signal. In terms of the iterative algorithm, this would lead to more iterations for the separation process, but once the iterative algorithm has finished, its learned version could be trained on its output to achieve faster execution.

Acknowledgment

The authors would like to thank De Ma and Zhifei Dai from the Biomedical Engineering department of Peking university for help in performing the *in-vivo* experiments.

References

[1] A. Fenster and J. C. Lacefield, *Ultrasound Imaging and Therapy*. Taylor & Francis, 2015.

[2] B. Furlow, "Contrast-enhanced ultrasound," *Radiologic technology*, vol. 80, no. 6, pp. 547S–561S, 2009.

[3] T. Opacic, S. Dencks, B. Theek, M. Piepenbrock, D. Ackermann, A. Rix, T. Lammers, E. Stickeler, S. Delorme, G. Schmitz *et al.*, "Motion model ultrasound localization microscopy for preclinical and clinical multiparametric tumor characterization," *Nature communications*, vol. 9, no. 1, p. 1527, 2018.

[4] N. De Jong, F. Ten Cate, C. Lancee, J. Roelandt, and N. Bom, "Principles and recent developments in ultrasound contrast agents," *Ultrasonics*, vol. 29, no. 4, pp. 324–330, 1991.

[5] N. Lassau, L. Chami, B. Benatsou, P. Peronneau, and A. Roche, "Dynamic contrast-enhanced ultrasonography (dce-us) with quantification of tumor perfusion: a new diagnostic tool to evaluate the early effects of antiangiogenic treatment," *European Radiology Supplements*, vol. 17, no. 6, pp. 89–98, 2007.

[6] J. M. Hudson, R. Williams, C. Tremblay-Darveau, P. S. Sheeran, L. Milot, G. A. Bjarnason, and P. N. Burns, "Dynamic contrast enhanced ultrasound for therapy monitoring," *European journal of radiology*, vol. 84, no. 9, pp. 1650–1657, 2015.

[7] C. Tremblay-Darveau, R. Williams, L. Milot, M. Bruce, and P. N. Burns, "Combined perfusion and doppler imaging using plane-wave nonlinear detection and microbubble contrast agents," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 61, no. 12, pp. 1988–2000, 2014.

[8] ——, "Visualizing the tumor microvasculature with a nonlinear plane-wave doppler imaging scheme based on amplitude modulation," *IEEE Transactions on Medical Imaging*, vol. 35, no. 2, pp. 699–709, 2016.

[9] A. Bar-Zion, O. Solomon, C. Tremblay-Darveau, D. Adam, and Y. C. Eldar, "Sparsity-based ultrasound super-resolution hemodynamic imaging," *to appear in IEEE Transactions on Ultrasonics, Ferroelectrics and Frequency Control*, 2018.

[10] R. J. van Sloun, O. Solomon, Y. C. Eldar, H. Wijkstra, and M. Mischi, "Sparsity-driven super-resolution in clinical contrast-enhanced ultrasound," in *Ultrasonics Symposium (IUS), 2017 IEEE International*. IEEE, 2017, pp. 1–4.

[11] R. J. van Sloun, O. Solomon, M. Bruce, Z. Z. Khaing, H. Wijkstra, Y. C. Eldar, and M. Mischi, "Super-resolution ultrasound localization microscopy through deep learning," *arXiv preprint arXiv:1804.07661*, 2018.

[12] O. Solomon, R. J. van Sloun, H. Wijkstra, M. Mischi, and Y. C. Eldar, "Exploiting flow dynamics for super-resolution in contrast-enhanced ultrasound," *arXiv preprint arXiv:1804.03134*, 2018.

[13] C. Errico, J. Pierre, S. Pezet, Y. Desailly, Z. Lenkei, O. Couture, and M. Tanter, "Ultrafast ultrasound localization microscopy for deep super-resolution vascular imaging," *Nature*, vol. 527, no. 7579, pp. 499–502, 2015.

[14] K. Christensen-Jeffries, R. J. Browning, M.-X. Tang, C. Dunsby, and R. J. Eckersley, "In vivo acoustic super-resolution and super-resolved velocity mapping using microbubbles," *IEEE Transactions on Medical Imaging*, vol. 34, no. 2, pp. 433–440, 2015.

[15] S. Bjaerum, H. Torp, and K. Kristoffersen, "Clutter filter design for ultrasound color flow imaging," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 49, no. 2, pp. 204–216, 2002.

[16] L. Thomas and A. Hall, "An improved wall filter for flow imaging of low velocity flow," in *Ultrasonics Symposium, 1994. Proceedings., 1994 IEEE*, vol. 3. IEEE, 1994, pp. 1701–1704.

[17] Y. M. Yoo, R. Managuli, and Y. Kim, "Adaptive clutter filtering for ultrasound color flow imaging," *Ultrasound in Medicine & Biology*, vol. 29, no. 9, pp. 1311–1320, 2003.

[18] P. J. Frinking, A. Bouakaz, J. Kirkhorn, F. J. Ten Cate, and N. De Jong, "Ultrasound contrast imaging: current and new potential methods," *Ultrasound in Medicine and Biology*, vol. 26, no. 6, pp. 965–975, 2000.

[19] L. A. Ledoux, P. J. Brands, and A. P. Hoeks, "Reduction of the clutter component in doppler ultrasound signals based on singular value decomposition: a simulation study," *Ultrasonic Imaging*, vol. 19, no. 1, pp. 1–18, 1997.

[20] A. C. Yu and L. Lovstakken, "Eigen-based clutter filter design for ultrasound color flow imaging: a review," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, 2010.

[21] F. W. Mauldin, F. Viola, and W. F. Walker, "Complex principal components for robust motion estimation," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 57, no. 11, 2010.

[22] F. W. Mauldin, D. Lin, and J. A. Hossack, "The singular value filter: a general filter design strategy for pca-based signal separation in medical ultrasound imaging," *IEEE Transactions on Medical Imaging*, vol. 30, no. 11, pp. 1951–1964, 2011.

[23] C. M. Gallippi, K. R. Nightingale, and G. E. Trahey, "BSS-based filtering of physiological and arfi-induced tissue and blood motion," *Ultrasound in Medicine & Biology*, vol. 29, no. 11, pp. 1583–1592, 2003.

[24] L. Lovstakken, S. Bjaerum, K. Kristoffersen, R. Haaverstad, and H. Torp, "Real-time adaptive clutter rejection filtering in color flow imaging using power method iterations," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 53, no. 9, pp. 1597–1608, 2006.

[25] D. E. Kruse and K. W. Ferrara, "A new high resolution color flow system using an eigendecomposition-based adaptive filter for clutter rejection," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 49, no. 10, pp. 1384–1399, 2002.

[26] C. Demené, T. Deffieux, M. Pernot, B.-F. Osmanski, V. Biran, J.-L. Gennisson, L.-A. Sieu, A. Bergel, S. Franqui, J.-M. Correas *et al.*, "Spatiotemporal clutter filtering of ultrafast ultrasound data highly increases doppler and fultrasound sensitivity," *IEEE Transactions on Medical Imaging*, vol. 34, no. 11, pp. 2271–2285, 2015.

[27] P. Song, A. Manduca, J. D. Trzasko, and S. Chen, "Ultrasound small vessel imaging with block-wise adaptive local clutter filtering," *IEEE Transactions on Medical Imaging*, vol. 36, no. 1, pp. 251–262, 2017.

[28] A. J. Chee and C. Alfred, "Receiver-operating characteristic analysis of eigen-based clutter filters for ultrasound color flow imaging," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 65, no. 3, pp. 390–399, 2018.

[29] A. Urban, C. Dussaux, G. Martel, C. Brunner, E. Mace, and G. Montaldo, "Real-time imaging of brain activity in freely moving rats using functional ultrasound," *Nature Methods*, vol. 12, no. 9, p. 873, 2015.

[30] C. Errico, B.-F. Osmanski, S. Pezet, O. Couture, Z. Lenkei, and M. Tanter, "Transcranial functional ultrasound imaging of the brain using microbubble-enhanced ultrasensitive doppler," *NeuroImage*, vol. 124, pp. 752–761, 2016.

[31] R. Otazo, E. Candès, and D. K. Sodickson, "Low-rank plus sparse matrix decomposition for accelerated dynamic mri with separation of background and dynamic components," *Magnetic Resonance in Medicine*, vol. 73, no. 3, pp. 1125–1136, 2015.

[32] M. Bayat and M. Fatemi, "Concurrent clutter and noise suppression via low rank plus sparse optimization for non-contrast ultrasound flow doppler processing in microvasculature," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 1080–1084.

[33] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

[34] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, p. 436, 2015.

[35] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on International Conference on Machine Learning*. Omnipress, 2010, pp. 399–406.

[36] P. Sprechmann, A. M. Bronstein, and G. Sapiro, "Learning efficient sparse and low rank models," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 37, no. 9, pp. 1821–1833, 2015.

[37] H. Sreter and R. Giryes, "Learned convolutional sparse coding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2191–2195.

[38] R. Giryes, Y. C. Eldar, A. Bronstein, and G. Sapiro, "Tradeoffs between convergence speed and reconstruction accuracy in inverse problems," *IEEE Transactions on Signal Processing*, vol. 66, pp. 1676–1690, April 2018.

[39] R. Giryes, Y. C. Eldar, A. M. Bronstein, and G. Sapiro, "The learned inexact project gradient descent algorithm," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 6767–6771.

[40] N. Samuel and A. Wiesel, "Learning to detect," *arXiv preprint arXiv:1805.07631*, 2018.

[41] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016, pp. 770–778.

[42] N. Samuel, T. Diskin, and A. Wiesel, "Deep mimo detection," *arXiv preprint arXiv:1706.01151*, 2017.

[43] A. Bar-Zion, C. Tremblay-Darveau, O. Solomon, D. Adam, and Y. C. Eldar, "Fast vascular ultrasound imaging with enhanced spatial resolution and background rejection," *IEEE Transactions on Medical Imaging*, vol. 36, no. 1, pp. 169–180, 2017.

[44] E. J. Candès and B. Recht, "Exact matrix completion via convex optimization," *Foundations of Computational Mathematics*, vol. 9, no. 6, p. 717, 2009.

[45] Y. C. Eldar and G. Kutyniok, *Compressed sensing: theory and applications*. Cambridge University Press, 2012.

[46] D. P. Palomar and Y. C. Eldar, *Convex optimization in signal processing and communications*. Cambridge University Press, 2010.

[47] J.-J. Moreau, "Proximité et dualité dans un espace hilbertien," *Bulletin de la Société mathématique de France*, vol. 93, pp. 273–299, 1965.

[48] Z. Tan, Y. C. Eldar, A. Beck, and A. Nehorai, "Analysis Sparse Recovery," *IEEE Transactions on Signal Processing*, vol. 62, no. 7, pp. 1762–1774, 2014.

[49] J.-F. Cai, E. J. Candès, and Z. Shen, "A singular value thresholding algorithm for matrix completion," *SIAM Journal on Optimization*, vol. 20, no. 4, pp. 1956–1982, 2010.

[50] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm," *SIAM Journal on Imaging Sciences*, vol. 2, no. 1, pp. 183–202, 2009.

[51] P. Song, J. D. Trzasko, A. Manduca, B. Qiang, R. Kadirvel, D. F. Kallmes, and S. Chen, "Accelerated singular value-based ultrasound blood flow clutter filtering with randomized singular value decomposition and randomized spatial downsampling," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 64, no. 4, pp. 706–716, 2017.

[52] Q. H. Yuanyuan, Wang and J. Luo, "Fast randomized singular value decomposition based clutter filtering for shear wave imaging," in *IEEE International Ultrasonics Symposium (IUS)*, 2018.

[53] N. Halko, P.-G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Probabilistic algorithms for constructing approximate matrix decompositions," *SIAM review*, vol. 53, no. 2, pp. 217–288, 2011.

[54] P.-G. Martinsson and S. Voronin, "A randomized blocked algorithm for efficiently computing rank-revealing factorizations of matrices," *SIAM Journal on Scientific Computing*, vol. 38, no. 5, pp. S485–S507, 2016.

Deep Unfolded Robust PCA with Application to Clutter Suppression in Ultrasound supporting materials

I. Learning fast approximations via unfolding

To better understand the concept of unfolding an iterative algorithm, we briefly describe the basic ideas presented in [35]. Consider the following sparse recovery model

$$\mathbf{y} = \mathbf{A}\mathbf{x},$$

where $\mathbf{y}$ is a length-m measurement vector, $\mathbf{x}$ is a length-n sparse vector to be recovered, and $\mathbf{A}$ is the

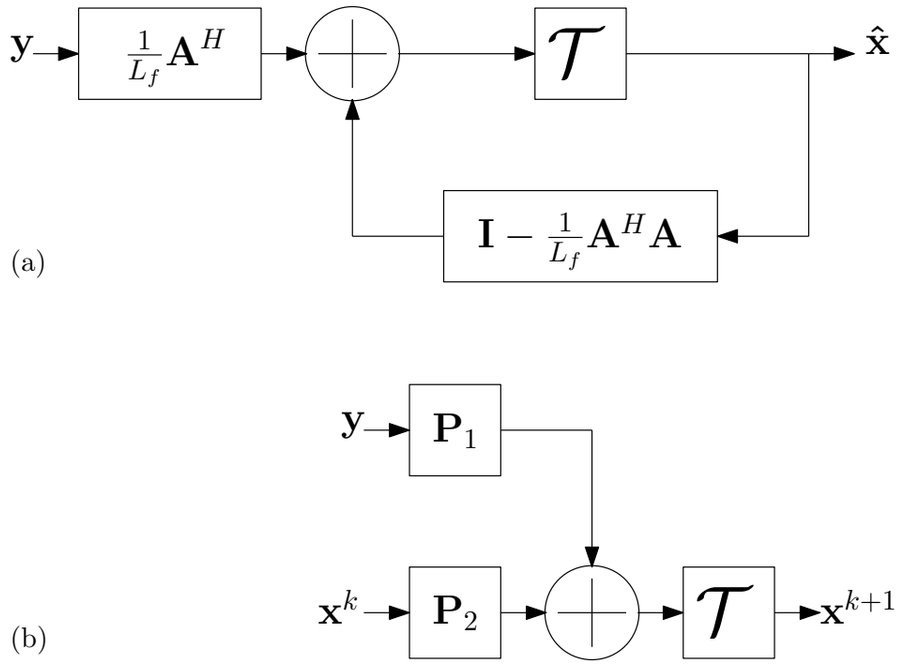


Fig. 6: ISTA iterative algorithm (panel (a)) compared with the learned ISTA (panel (b)). Each iteration in the iterative algorithm is replaced with a single layer in the learned algorithm. Instead of using the model parameters such as $\mathbf{A}$, these parameters are replaced with general matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ which are learned.

sensing matrix. Recovering $\mathbf{x}$ from $\mathbf{y}$ can be performed by formulating the following convex minimization problem

$$\min_{\mathbf{x}} \frac{1}{2}||\mathbf{y} - \mathbf{A}\mathbf{x}||_2^2 + \lambda||\mathbf{x}||_1, \tag{9}$$

where $\lambda > 0$ is a regularization parameter. A popular iterative algorithm which minimizes (9) is the ISTA algorithm, or its faster counterpart, the fast ISTA (FISTA). FISTA is guaranteed to converge, in the

worst case scenario, with a rate proportional to $1/k^2$, with k being the iteration number. As suggested in [35], this convergence can be sped up by proposing a learned version of ISTA (LISTA). Furthermore, the authors of [38] demonstrated that the unfolded architecture facilitates a trade-off between fast convergence and reconstruction accuracy of the sparse recovery problem.

More specifically, the iterative scheme of ISTA consists of the following iterative step

$$\mathbf{x}^{k+1} = \mathcal{T}_{\lambda/L_f}\left\{\left(\mathbf{I} - \frac{1}{L_f}\mathbf{A}^H\mathbf{A}\right)\mathbf{x}^k + \frac{1}{L_f}\mathbf{A}^H\mathbf{y}\right\},$$

with $\mathcal{T}_{\lambda/L_f}(\cdot)$ being the element-wise soft-thresholding operator with parameter λ/L_f and L_f is the spectral norm of $\mathbf{A}^H\mathbf{A}$. This iterative procedure is illustrated in panel (a) of Fig. 6, where $\hat{\mathbf{x}}$ is the output of the ISTA algorithm.

Conversely, we can consider each iteration of the iterative algorithm in panel (a) of Fig. 6 as a single layer in a feedforward network. Instead of using the known matrix $\mathbf{A}$ we replace the matrices in panel (a) with general matrices $\mathbf{P}_1$ and $\mathbf{P}_2$ to be learned, as well as the regularization parameter λ, as illustrated in panel (b) of Fig. 6. Thus, a single layer of this unfolded network is described by

$$\mathbf{x}^{k+1} = \mathcal{T}_{\lambda/L_f}\left\{\mathbf{P}_2\mathbf{x}^k + \mathbf{P}_1\mathbf{y}\right\}.$$

By concatenating several such layers (typically less than ten layers, corresponding to ten iterations are sufficient), a deep network is formed.

II. ResNet architecture

In this section we provide an additional result of ResNet applied to the simulated data (trained for 10 epochs on simulated data), as well as a detailed description of the complex ResNet architecture.

Figure 7 presents the ResNet recovery of the same simulated movie presented in Fig. 2 of the main paper. Visual inspection of Fig. 2 reveals that in the case of simulations, ResNet suppresses the tissue signal and reveals the UCA signal. However, comparing panel (c) of Fig. 7 to panel (c) of Fig. 2 of the main paper shows that the CORONA reconstruction achieves higher contrast, in line with the conclusions drawn in the main paper. Moreover, CORONA is able to recover the tissue signal as well as the UCA signal, whereas ResNet recovers the UCA signal only. Figure 8 shows the ResNet architecture used in this work. Here, Conv. layer is a complex convolution layer, and $16@5 \times 5$ refers to 16 convolution channels with a 5×5 pixels kernel.

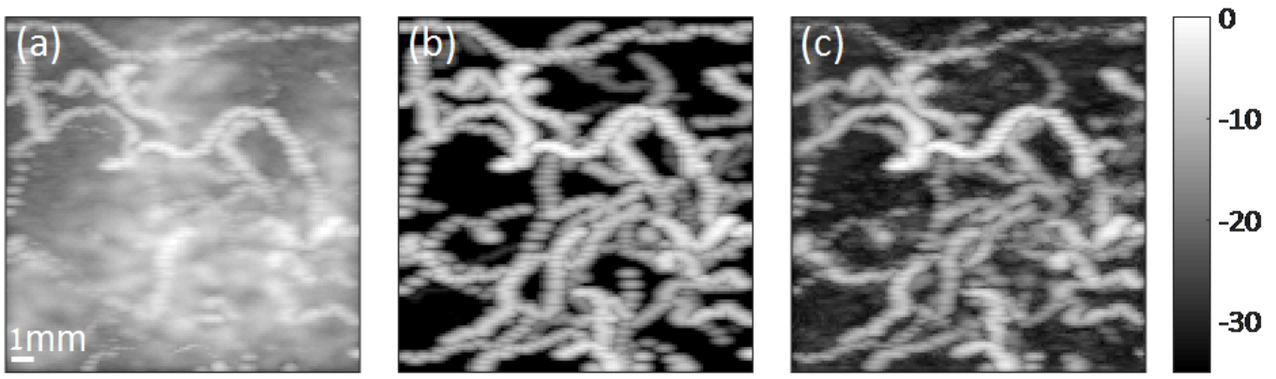


Fig. 7: Simulation results of ResNet. (a) MIP image of the same simulated movie used in the main paper. (b) Ground truth MIP image of the UCAs. (c) MIP image of the UCAs recovered by ResNet. Color bar is in dB.

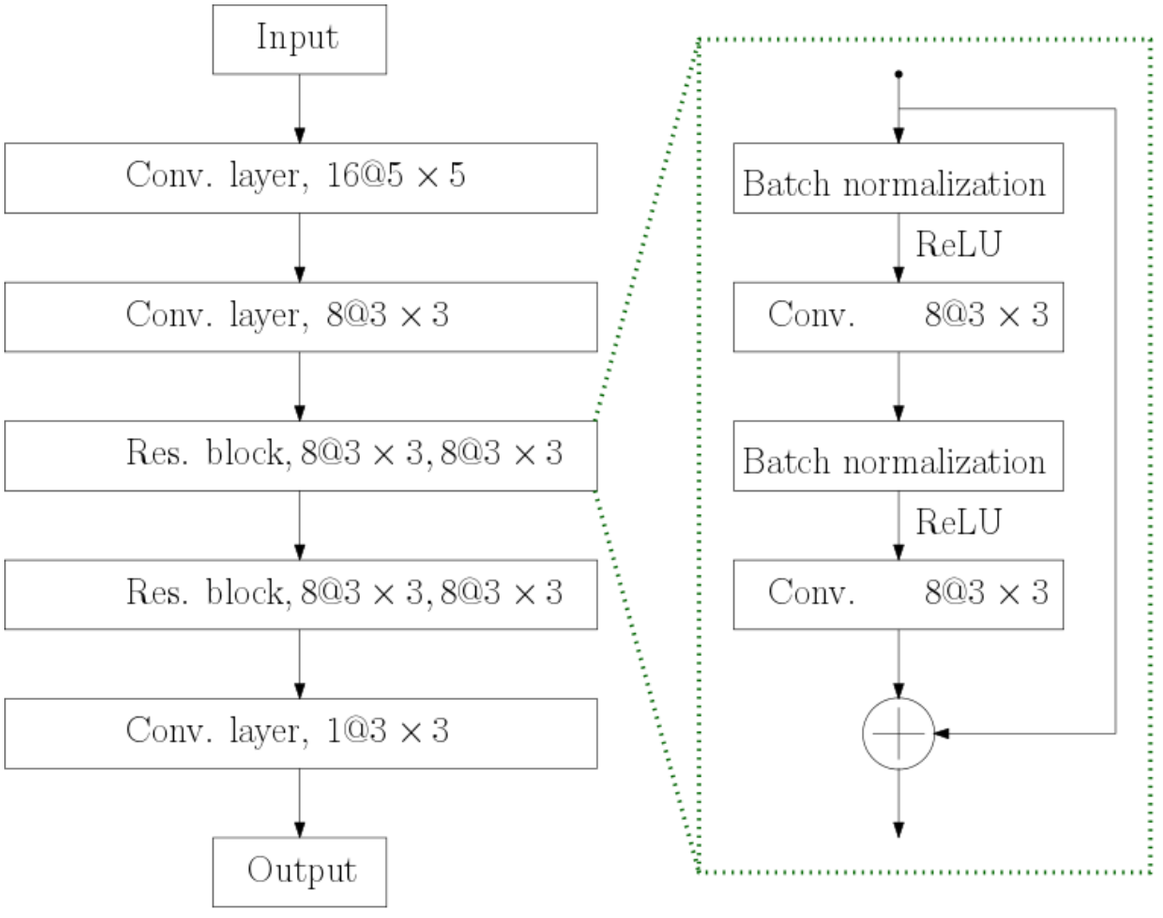
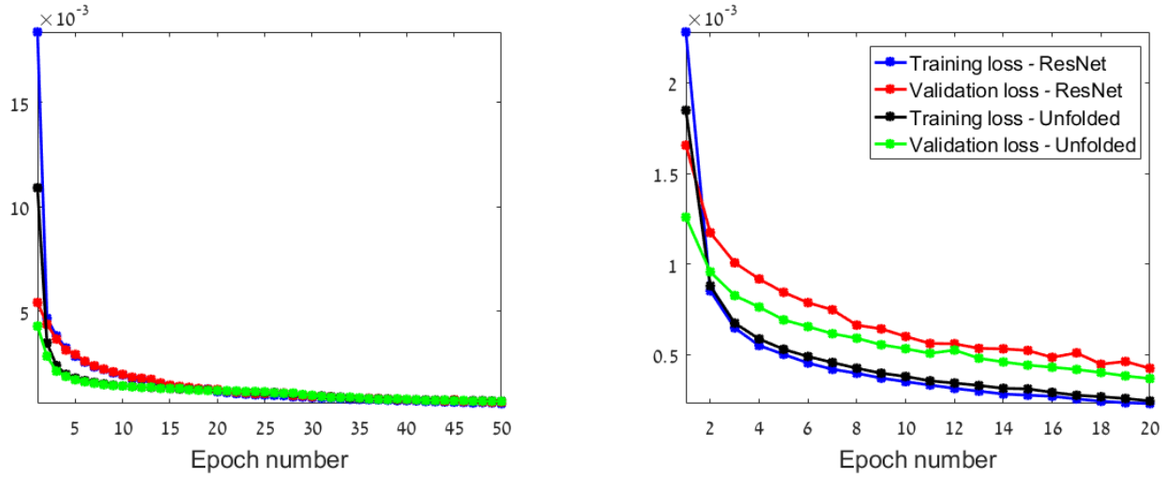


Fig. 8: ResNet architecture used in this work. Conv. layers are complex convolution layers.

III. Training loss functions and learned regularization parameters

In this section, we provide the training and validation losses for the training process of the unfolded network and ResNet. Training was performed in two stages. The first stage consisted of 50 training epochs over 2400 simulated movie patches (20 frames each), while the second stage included additional

20 training epochs over 2400 patches from the first rat (20 frames each). For *in-vivo* validation, 100 consecutive frames from the second rat were chosen randomly. MSE was calculated according to (8) in the main paper.



(a) Training on simulation data only.

(b) Training on simulation and *in-vivo* data.

Fig. 9: Training and validation losses for the unfolded network and ResNet. Left panel shows both training and validation losses for training the unfolded network (10 layers) and ResNet with simulation patches only for 50 epochs. Right panel presents both training and validation losses for the same networks trained with simulation patches for 50 epochs and additional 20 epochs on *in-vivo* data.

Considering Fig. 9, it is evident that when training on simulation data only, the validation curves follow the training loss curves for both networks and are comparable after 20 epochs. This behavior might suggest that the networks over-fit the simulated data, that is, they achieve the best possible recovery for simulated patches. However, in this case, the networks have yet to learn from actual data. In such a case, if the simulation does not represent the data precisely (e.g. different dynamic range, MB concentration, etc.), its performance will degrade when applied to *in-vivo data*, as presented in Section IV. Thus, additional training is performed, as shown in panel (b) of Fig. 9. In this case, the validation losses are higher than the training loss, but now, as presented in the main paper, the networks perform well on *in-vivo* data.

Figure 10 and Fig. 11 illustrate the learned values of λ_{L_i} and λ_{S_i} for the unfolded network, where $i = 1, \ldots, 10$ indicates the layer number when training on simulation data only and on simulation and *in-vivo* data together.

Considering both figures, it is evident that most of the regularization parameters do not change considerably when training on *in-vivo* data is performed. As the unfolded network is trained on both simulation and *in-vivo* data, the regularization parameters do not converge to the parameters used in the iterative FISTA algorithm. This also suggests that by performing combined learning on both simulation

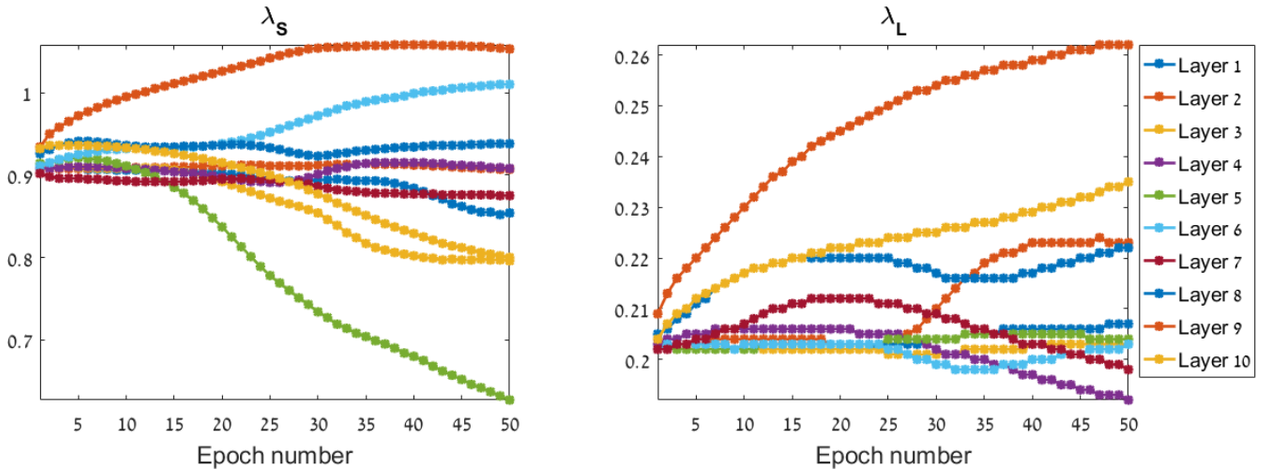


Fig. 10: Learned regularization parameters when training on simulation data only.

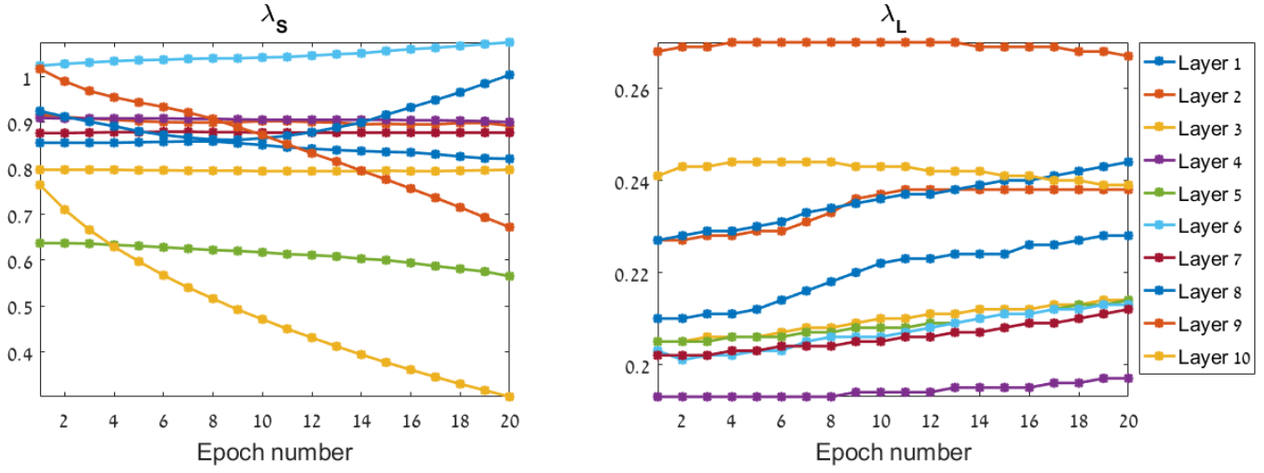


Fig. 11: Learned regularization parameters when training on simulation and *in-vivo* data.

and experimental data, the network further differs from its iterative counterpart, often leading to improved performance, as presented in the main paper.

IV. The importance of training on both simulations and in-vivo data

As was described in the main paper and in Section III, the unfolded network outperforms FISTA reconstruction due to the combined training on both simulations and *in-vivo* data. This joint training allows the network to learn both the ”ideal conditions” for MB/tissue separation from the simulations, as well as important features from the experimental data, and achieve robustness to noise and modeling mismatch.

In Fig. 12 we present *in-vivo* results of the network trained in two conditions. Panels (a) and (b) show the output of the network when trained solely on simulated data for 10 epochs and the output when

trained on both simulated and experimental data for 10 epochs each, respectively. Panels (c) and (d) show the output of the networks for the same two cases, only now the numbers of training epochs were 50 for simulated data and 20 for *in-vivo* data.

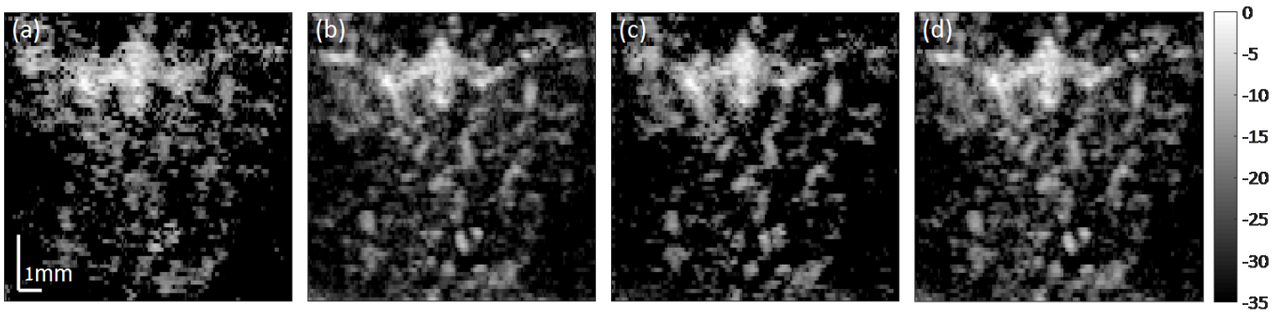


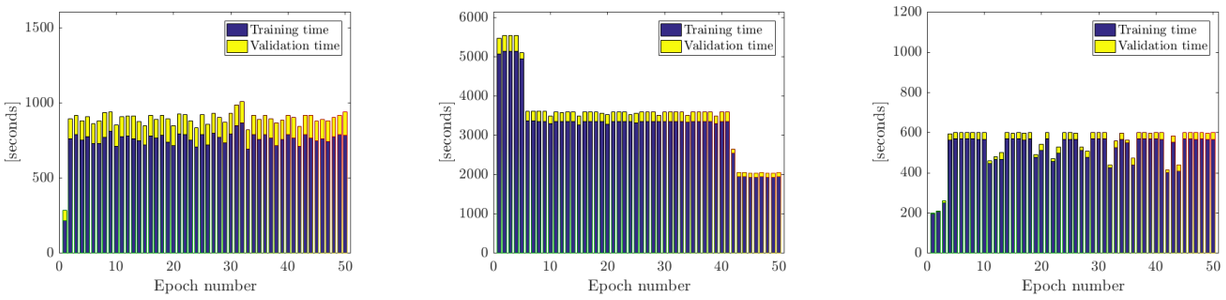
Fig. 12: *In-vivo* results when training on simulations only and on simulations and *in-vivo* data together. (a) Training on simulations for 10 epochs. (b) Training on both simulations and experimental data for 10 epochs each. (c) Training on simulations for 50 epochs. (d) Training on both simulations and experimental data for 50 and 20 epochs, respectively. Color bar is in dB.

Considering Fig. 12, clearly when training on a relatively low number of epochs (10), simulated data is not sufficient for good performance on experimental data. On the other hand, when combined with additional 10 epochs of training on *in-vivo* data, the performance of the network improves considerably, and is somewhat similar to the performance of the network result displayed in panel (d). Surprisingly, even when training on simulated data only for enough epochs, in this case 50, the network performs well in recovering the vascular bed of experimental data, as shown in panel (c). However, closer examination shows that albeit the image looks sparser than the image in panel (d), its texture looks more pixel-like than the FISTA and SVD images shown in Fig. 4 of the main paper.

The latter example suggests two things. First, that good results can be obtained by training the network on realistic simulations for enough training epochs. The second is that performance more similar in texture and visual quality to that of non-learning based techniques can be obtained by the combined training on both simulations and experimental data.

V. Runtime comparison

Here, we compare the run-time performance of both the unfolded network and ResNet, for both training phase and validation phase. Fig. 13 show the time in seconds each network required to train a single epoch and then validate its performance, in yellow, as a function of epoch number. Training was performed for 50 epochs on simulated data. Run-times results for training on *in-vivo* data were similar, and thus are omitted.



(a) Unfolded network, 1 layer (b) Unfolded network, 10 layers (c) ResNet

Fig. 13: Run-time results for training and validation of the unfolded network and ResNet.

Observing Fig. 13, it is evident that the training and validation of the unfolded network is slower compared with ResNet. The 10 layers network is slower by an order of magnitude, but the 1 layer network has a slightly slower runtime. The slower processing and training time of the unfolded network is attributed to the SVD operations required by the network, although faster and more efficient algorithms for SVD computations can be used, as discussed in the Discussion section of the main paper. This figure further supports the conclusions in the main paper. The unfolded network offers a flexible trade-off between execution time and performance, by allowing to choose its depth.

However, the unfolded network has an order of magnitude lower number of trainable parameters and achieves better CNR and CR values, as demonstrated in the main paper. It is also important to remember that the ResNet was not fully trained, rather only its last fully connected layers were trained. This transfer learning process considerably reduces the overall training time.

VI. Simulation description

As was indicated in the main paper, in this work we increase the number of training examples by training on both experimental and simulated data. In this section, we describe how the simulation was generated. In the simulations we used, pixel size is assumed to be $0.12 \times 0.12\text{mm}^2$ and the number of pixels is 128×128. Implementation was performed in Python 3.5.2.

A. MB signal generation

The overall number of MBs (as well as their initial positions) was generated randomly up to a maximum concentration of 130 MBs per cm^{-2}. MB amplitudes were drawn from a normal complex distribution.

MB velocity magnitudes were generated according to

$$v(x, y, t = 0) = \max(0, v_{\text{det}} \cdot \mathcal{N}(1, 1)),$$

where $v_{\text{det}} = 0.24\text{mm}/dt$, $dt = 0.01\text{s}$ is the imaging frame-rate and $\mathcal{N}(1,1)$ is a normal distribution with mean 1 and standard deviation 1. MB accelerations were generated according to

$$a_{x/y} = \mathcal{N}(0, \sigma_a),$$

with $\sigma_a = 0.05 \cdot 0.12/dt^2$ and x/y are the lateral and axial directions, respectively.

MB velocity directions are generated in each frame according to

$$v_x^k(t) = v_x^{k-1}(t)\text{cos}(\theta) - v_y^{k-1}(t)\text{sin}(\theta),$$

$$v_y^k(t) = v_x^{k-1}(t)\text{sin}(\theta) + v_y^{k-1}(t)\text{cos}(\theta),$$

with $\theta \sim \text{U}[-30^\circ, 30^\circ]$ and k indicates frame number. MB amplitudes are additionally multiplied by a random factor between 0.9 and 1.1 in each frame.

B. Tissue signal generation

To model the tissue signal, we start by generating a sum of five real 2D Gaussian matrices of the same size as the image frames (128×128 pixels) with random positions and variances. We then generate a complex random matrix to modulate the envelope of the tissue signal. The real and complex entries are both drawn from a normal distribution with zero mean and standard deviation 1. Both matrices are then multiplied element-wise, and the product is then low-pass filtered (2D real Gaussian matrix of 11×11 pixels). The resulting signal's envelope, denoted as $\mathbf{B} \in \mathcal{R}^{I \times J}$ mimics the texture of the tissue signal. Thus, the overall pixels' values are random, but locally they are correlated.

The next step involves the generation of a phase matrix, same size as before. It's entries are drawn from a Gaussian distribution in the following manner

$$\boldsymbol{\theta} \sim \mathcal{N}(\alpha, \sigma_\theta),$$

with a mean drawn from a uniform distribution in the range $\alpha \sim [0^\circ, 180^\circ]$ and standard deviation of $\sigma_\theta = 15^\circ$. The resulting complex tissue signal is given by

$$\mathbf{T}[i,j] = \mathbf{B}[i,j]e^{j\boldsymbol{\theta}[i,j]},\ i,j \in [I,J].$$

The next stage in generating the tissue signal is to apply spatial deformations in each frame, to mimic tissue movement during the acquisition period. To this end we start by generating four different 4×4

kernels, denoted as flow filters. The entries of those kernels, are positive and their sum equals to one. For each new frame, we generate additional four filters, with entries drawn from a Gaussian distribution with zero mean and standard deviation of 0.1. For each flow filter we add the corresponding new filter and for each pixel we take the maximum between the latter value and 0.1. The resulting new filter is normalized such that all entries sum to one and the entries are non-negative.

Once the flow filters have been updated for the current frame, they are convolved with $\mathbf{T}$, to get four different images. We then divide each of the four images into 4×4 blocks. The final image is generated by dividing an empty matrix into 4×4 blocks, and for each block choosing randomly one of the corresponding blocks from the four images. This process ensures that blocks in the same neighborhood share the same movement pattern, but in the whole image, the pattern is random.

C. Simulation of the PSF

Once the (complex) MB frame and tissue frame are generated and summed with complex Gaussian noise, the resulting frame is convolved with the PSF. The PSF is modeled as a 2D real Gaussian kernel with standard deviations of 0.14mm in the lateral and 0.32mm axial dimensions, taken from the *in-vivo* data.