Deep Learned-CoSaMP Network for Signal Denoising

Pagoti Reshma
Dept. of Humanities and Sciences
Indian Institute of Petroleum and Energy
Visakhapatnam, India
pagotireshma.maths@iipe.ac.in

Pradyumna Pradhan
Dept. of Humanities and Sciences
Indian Institute of Petroleum and Energy
Visakhapatnam, India
pradyumna11@iipe.ac.in

Hrithik Singh
Dept. of Chemical Engineering
Indian Institute of Petroleum and Energy
Visakhapatnam, India
singhh15@indianoil.in

Srinivas Tenneti
Dept. of Chemical Engineering
Indian Institute of Petroleum and Energy
Visakhapatnam, India
tennetisrinivas.che@iipe.ac.in

Pradip Sasmal
Dept. of Mathematics
Indian Institute of Technology
Jodhpur, India
psasmal@iitj.ac.in

Ramunaidu Randhi
Dept. of Humanities and Sciences
Indian Institute of Petroleum and Energy
Visakhapatnam, India
ramu.math@iipe.ac.in

***Abstract*—Deep Neural Networks (DNNs) have garnered considerable attention in the field of sparse signal recovery due to their powerful learning capabilities. However, they face challenges such as a lack of interpretability and a strong dependence on large training datasets. To address these issues, algorithm unrolling has emerged as a promising approach that systematically transforms iterative algorithms into neural network architectures. Recently, the unrolling of orthogonal matching pursuit (OMP), termed learned OMP (L-OMP), has demonstrated improved performance over existing unrolled methods. Nonetheless, L-OMP exhibits limitations in high-noise scenarios due to its slower dictionary learning process. To overcome this limitation, we propose the unrolling of compressive sampling matching pursuit (CoSaMP), leveraging its batch-wise support selection and noise-pruning capabilities. This method, termed learned CoSaMP (L-CoSaMP), effectively addresses noise-dominated components and accelerates dictionary learning. Experimental results indicate that L-CoSaMP consistently outperforms state-of-the-art unrolled networks, such as the learned iterative soft thresholding algorithm (LISTA) and L-OMP, particularly in high-noise environments. These findings highlight the robustness and efficiency of L-CoSaMP in signal denoising tasks.**

***Index Terms*—Sparse signal recovery, OMP, CoSaMP, Algorithm unrolling, Deep neural networks.**

I. Introduction

Linear inverse problems (LIPs) are fundamental aspects of various science and engineering applications. These problems involve the recovery of a target vector from an observation vector using a linear operator. Mathematically, the problem can be expressed as follows: Given a measurement vector $\mathbf{y} \in \mathbb{R}^{m\times 1}$, which can be modeled as the result of applying a linear operator $A \in \mathbb{R}^{m\times n}$, $(m < n)$ to an unknown target vector $\mathbf{x} \in \mathbb{R}^{n\times 1}$, such that

$$\mathbf{y} = A\mathbf{x} + \mathbf{e}, \tag{1}$$

The fifth author acknowledges the support from the Science and Engineering Research Board (SERB), Govt. of India (SRG/2023/002383). The sixth author acknowledges NBHM's financial assistance (Id:02011/26/2023/NBHM(R.P)/R&D II/ 5867).

where $\mathbf{e}$ is an error term due to noise such that $\|\mathbf{e}\|_2 \leq \epsilon$. Usually, an LIP is solved by applying certain constraints to the target vector. A commonly used constraint involves ensuring sparsity in the target vector, which transforms it into a sparse signal recovery problem [1]–[3]. In literature, it has been demonstrated that sparse signal recovery is useful for solving various problems such as denoising, deblurring, deraining, inpainting, demosaicing, image fusion, super-resolution, tomographic reconstruction, and MRI imaging [4]–[12]. In general, two broad approaches have been adopted to address sparse signal recovery problems: (A) model-based and (B) data-driven methods.

The model-based method leverages mathematical frameworks to characterize the relationship between observations and target information. These techniques are known for their simplicity, computational efficiency, and robustness to noise, but their effectiveness relies on accurate knowledge of the underlying model [13], [14]. Examples of this method include orthogonal matching pursuit (OMP) [15], compressed sampling matching pursuit (CoSaMP) [16], the iterative soft thresholding algorithm (ISTA) [17], and others.

The data-driven method employs machine learning techniques to learn a mapping between the observed and the target data. This approach proves beneficial when the underlying model of the system is either unknown or difficult to model explicitly. In this approach, a training dataset is first collected, which consists of pairs of observed data and corresponding target data. Subsequently, a machine learning model is trained on this dataset using supervised techniques. Once the model is trained, it can then predict the target signal for newly and unseen observed data. These methods have achieved state-of-the-art performance in various fields, including signal processing, medical imaging, and computer vision [18]–[24]. However, despite their advantages, they have several limitations, including the need for a large amount of training data, computational cost, and the potential lack of an

interpretable solution [13], [14], [25].

To address these issues, a new approach called model-aware data-driven/ algorithm unrolling has been introduced in literature [25], [26]. This approach uses both the mathematical formulation of the model and machine learning abilities. This method involves designing a neural network architecture through an iterative algorithm, where each step in the iteration corresponds to a single layer in the neural network. Ultimately, the unrolling algorithm consolidates all these layers to form a deep neural network (DNN). It has been observed that unrolling methods outperform existing approaches in various domains [27]–[30]. In particular, the learned ISTA (LISTA) [26] and learned OMP (L-OMP) [31] are two widely used unrolled algorithms, demonstrating their utility in applications such as ultrasound imaging [28], image super-resolution [32], and other related domains.

Despite the widespread use of LISTA and L-OMP networks, they exhibit several limitations. Specifically, these networks are sensitive to noise and often struggle to accurately identify the correct support during recovery. Additionally, they require significant time for training and are relatively slow in learning the underlying dictionary. In contrast, the CoSaMP algorithm employs a batch-wise support selection strategy combined with a pruning step, effectively removing noise-dominated components from the recovery process, making it particularly suitable for noisy scenarios.

Building on this, we propose the unrolled CoSaMP algorithm, referred to as learned CoSaMP (L-CoSaMP). This approach leverages the adaptability of batch-wise support selection and the computational efficiency of network unrolling, aiming to enhance performance and robustness to noise. The L-CoSaMP network is designed using MSPT and SAtoS units, which are essential for ensuring the smoothness of the model during training. Using synthetic data, we demonstrate that L-CoSaMP exhibits superior denoising performance compared to LISTA and L-OMP networks. Furthermore, the learned dictionary in the L-CoSaMP network converges more rapidly to the true dictionary during training, highlighting its efficiency and robustness relative to LISTA and L-OMP under high noise conditions.

The paper is organized as follows: Section II gives a brief introduction to the CoSaMP algorithm. Section III designs the L-CoSaMP network architecture. Experimental results are presented in Section IV. Finally, Section V draws conclusions.

Throughout this paper, the following notations are employed. The set of real numbers is defined by $\mathbb{R}$. We use bold lowercase letters for vectors and capital letters for matrices. Symbol $\|\mathbf{x}\|$ denotes the ℓ_2-norm of vector $\mathbf{x}$. Support of $\mathbf{x}$ denoted as $\text{supp}(\mathbf{x})$, which is defined as $\{i : \mathbf{x}_i \neq 0\}$. Let $\mathcal{T} = \{i_1, i_2, \dots i_t\} \subseteq [n]$ be the set of indices, then $\mathbf{x}_\mathcal{T} = [x_{i_1}, x_{i_2}, \dots x_{i_t}]^T \in \mathbb{R}^{|\mathcal{T}|}$ and $A_\mathcal{T} = [\mathbf{a}_{i_1}, \mathbf{a}_{i_2}, \dots \mathbf{a}_{i_t}] \in \mathbb{R}^{m \times |\mathcal{T}|}$, where $\mathbf{a}_{i_j}$ is the $i_j{}^{th}$ column of A. The complement of the set $\mathcal{T}$ is represented as $\mathcal{T}^c$. The transpose of the matrix A is denoted as A^T. For $m > |\mathcal{T}|$, the pseudo inverse of $A_\mathcal{T} \in \mathbb{R}^{m \times |\mathcal{T}|}$ is $A_\mathcal{T}^\dagger = (A_\mathcal{T}^T A_\mathcal{T})^{-1} A_\mathcal{T}^T$.

II. CoSaMP Algorithm

CoSaMP [16] is an iterative algorithm designed to recover a sparse signal $\mathbf{x}$ from a measurement vector $\mathbf{y}$ by solving (1). Specifically, the algorithm iteratively identifies a subset of components (supports) corresponding to the atoms that are most correlated with the residual. Subsequently, it projects the measurement vector onto the subspace spanned by the columns of the measurement matrix associated with the selected support set. Finally, it refines the estimate by pruning the components dominated by noise in the projected vector, yielding an approximation of the target sparse signal.

CoSaMP recovers the s-sparse vector $\mathbf{x} \in \mathbb{R}^{n \times 1}$ from a given noisy measurement vector $\mathbf{y} \in \mathbb{R}^{m \times 1}$ by using a linear operator $A \in \mathbb{R}^{m \times n}$. A comprehensive summary of CoSaMP [16] is given in Algorithm 1.

Algorithm 1: CoSaMP Algorithm [16]

Input: Linear operator A, noisy measurement vector $\mathbf{y}$, sparsity s

Output: s-sparse vector $\mathbf{x}$

Initialization: $\mathbf{x}^{(0)} = \mathbf{0}$, $\mathbf{r}^{(0)} = \mathbf{y}$

for $k = 0, 1, 2, \dots$ *until stopping criteria* **do**

$\mathbf{u}^{(k)} = A^\top \mathbf{r}^{(k)}$

$\Omega = \text{supp}(\mathbf{u}^{(k)}_{2s})$; support of $2s$ largest component of $\mathbf{u}^{(k)}$

$\mathcal{T} = \Omega \cup \text{supp}(\mathbf{x}^{(k)})$

$\mathbf{b}_\mathcal{T} = A_\mathcal{T}^\dagger \mathbf{y}$

$\mathbf{b}_{\mathcal{T}^c} = \mathbf{0}$

combine $\mathbf{b}_\mathcal{T}$ and $\mathbf{b}_{\mathcal{T}^c}$ at corresponding indices to get $\mathbf{b}$

$\mathcal{S}$ = Indices of the s largest components of $\mathbf{b}$

$$\mathbf{x}^{(k+1)} = \begin{cases} \mathbf{b}_\mathcal{S} & \text{on support} \\ \mathbf{0} & \text{off support} \end{cases}$$

$\mathbf{r}^{(k+1)} = \mathbf{y} - A\mathbf{x}^{(k+1)}$

end

$\mathbf{x} = \mathbf{x}^{(k+1)}$

III. Learned CoSaMP (L-CoSaMP)

Now, we discuss our method of converting the CoSaMP into a neural network architecture. In this approach, each iteration of CoSaMP is mapped into a layer within the proposed architecture. The index sets Ω and $\mathcal{T}$ in Algorithm 1 contain information that is carried from one iteration to the next. However, incorporating this set of indices into a network poses challenges in terms of differentiability. To address this, we implement the following.

Maximal S Projection-thresholding (MSPT) unit [31]: This unit is responsible for determining which s atoms are added to the support in each layer. Given a vector $\mathbf{u}$, let $\mathcal{I} = \{i_1, i_2, \dots, i_s\}$ denote the set of indices corresponding to the s-largest entries of $|\mathbf{u}|$. The MSPT function of the vector $\mathbf{u}$ is defined as

$$P = \text{MSPT}(\mathbf{u}) = [\mathbf{p}_1, \mathbf{p}_2, \dots, \mathbf{p}_s] \in \mathbb{R}^{m \times s},$$

where $\mathbf{p}_j$ include zeros except for i_j-th entry, which includes $\mathbf{u}_{i_j}$. Specifically, when the input is the correlation vector $A^\top \mathbf{r}$,

the resulting output is a matrix with s columns. Each column $\mathbf{p}_j$ in the matrix contains zeros in all positions except for the i_j-th index, which corresponds to one of the s most correlated atoms with the residual.

S Atom selecting (SAtoS) unit [31]: This unit is designed to extract a specific set of s atoms. It operates on the output of the MSPT function $P \in \mathbb{R}^{m\times s}$, and the dictionary A. As a result, this unit produces a collection of s atoms from A that align with the s indices of non-zero values in the columns of P. The SAtoS is defined as

$$\text{SAtoS}(A, P) = [\text{AtoS}(A, \mathbf{p}_1), \ldots, \text{AtoS}(A, \mathbf{p}_s)]\,,$$

where $\text{AtoS}(A, \mathbf{p}_i) = \frac{1}{\|\mathbf{p}_i\|} A \cdot |\mathbf{p}_i|, \quad \forall i \in [s]$.

We now describe the neural network architecture for the CoSaMP algorithm by using MSPT and SAtoS units. As discussed earlier, each iteration of the CoSaMP algorithm corresponds to a layer in the proposed neural network architecture, and the DNN is formed by stacking these layers. The details are discussed in the following.

A. L-CoSaMP Architecture

In L-CoSaMP, different layers exchange the aggregated sub-dictionary instead of passing the support Ω and $\mathcal{T}$. The k-th layer of the network receives the following inputs: an observation signal $\mathbf{y} \in \mathbb{R}^{m\times 1}$, a global dictionary $A \in \mathbb{R}^{m\times n}$, the aggregated sub-dictionary $A_{\Gamma(k-1)} \in \mathbb{R}^{m\times s}$ and the restored signal $\mathbf{y}^{(k-1)} \in \mathbb{R}^{m\times 1}$ from the previous layer. The k-th layer inference of L-CoSaMP is described in algorithm 2. It starts by identifying the $2s$ most correlated atoms with the current residual, using both the M2SPT[1] and 2SAtoS[2] units. These selected atoms are then appended to the temporal sub-dictionary $A_{\tilde{\Gamma}(k)}$, along with atoms from the previous layer. Subsequently, $\hat{\mathbf{x}}_{\tilde{\Gamma}(k)}$ is computed under the pseudo inverse of $A_{\tilde{\Gamma}(k)}$. Next, $A_{\Gamma(k)}$ and I^* are determined by selecting the atoms corresponding to the s highest magnitudes in $\hat{\mathbf{x}}_{\tilde{\Gamma}(k)}$, achieved through a combination of MSPT and SAtoS. Utilizing the updated support atoms of I^*, $\hat{\mathbf{x}}^{(k)}$ is evaluated. Finally, the restored signal $\mathbf{y}^{(k)}$ is obtained as the product of $A_{\Gamma(k)}$ and $\hat{\mathbf{x}}^{(k)}$. Ultimately, all layers are stacked to construct the L-CoSaMP network. The L-CoSaMP network architecture is given in Algorithm 3, which uses L-CoSaMP layer inference in each layer of the network.

In the next section, we illustrate the performance of the proposed L-CoSaMP via simulations.

IV. Experimental Results

In this section, we compare the denoising performance of L-CoSaMP with other unrolled networks like L-OMP [31] and LISTA [26]. We consider the following setup for our experiments.

Data generation and Training setup: We consider a random dictionary A of size 100×200, whose non-zero entries follow mean zero and variance of one. Next, we generate a sparse vector $\mathbf{x} \in \mathbb{R}^{200\times 1}$ with sparsity $s = 5, 7, 10$ for each signal. The non-zero positions of these vectors are chosen uniformly at random, with the values drawn from a Gaussian distribution with mean 0 and variance 1. Each measurement vector is then generated by multiplying the dictionary A with the corresponding sparse vector $\mathbf{x}$. To create input-output training pairs $(\mathbf{y}, \mathbf{y}^*)$, additive white Gaussian noise with a mean of 0 and a standard deviation of 0.04 is added to each measurement vector $\mathbf{y}^*$.

We generate a total of 10000 training signals, each with a sparsity of $s = 5, 7, 10$. The model is initialized with a random Gaussian dictionary and trained using the Adam optimizer [33]. The training process minimizes the squared loss, defined as:

$$L = \sum_{(\mathbf{y},\mathbf{y}^*)\in \text{training set}} \|\mathbf{y}^* - \hat{\mathbf{y}}\|_2^2, \tag{2}$$

where $\mathbf{y}$, $\mathbf{y}^*$, and $\hat{\mathbf{y}}$ are the noisy, cleaned, and predicted signals, respectively.

Next, we evaluate the denoising capabilities of various unrolled networks, including L-OMP, LISTA, and L-CoSaMP. Specifically, the L-OMP, LISTA, and L-CoSaMP networks are

Algorithm 2: L-CoSaMP k^{th}Layer

Input: $\mathbf{y}, \mathbf{y}^{(k-1)} \in \mathbb{R}^{m\times 1}, A \in \mathbb{R}^{m\times n}, A_{\Gamma(k-1)} \in \mathbb{R}^{m\times s}, I = 3s \times 3s$ Identity matrix
Output: $\hat{\mathbf{x}}^{(k)} \in \mathbb{R}^{s\times 1}, \mathbf{y}^{(k)} \in \mathbb{R}^{m\times 1}, A_{\Gamma(k)} \in \mathbb{R}^{m\times s}$
begin
- $\mathbf{r}^{(k-1)} = \mathbf{y} - \mathbf{y}^{(k-1)}$
- $\mathbf{u}^{(k)} = W_A A^\top \mathbf{r}^{(k-1)}; {}_{W_A = diag^{-1}(\|a_1\|, \|a_2\|, \ldots, \|a_n\|)}$
- $A_{tmp} = 2\text{SAtoS}(A, \text{M2SPT}(\mathbf{u}^{(k)}))$
- $A_{\tilde{\Gamma}(k)} = \left[A_{\Gamma(k-1)}, A_{tmp}\right]$
- $\hat{\mathbf{x}}_{\tilde{\Gamma}(k)} = A^\dagger_{\tilde{\Gamma}(k)} \mathbf{y}$
- $A_{\Gamma(k)} = \text{SAtoS}(A_{\tilde{\Gamma}(k)}, \text{MSPT}(W_{A_{\tilde{\Gamma}(k)}} \mathbf{x}^{\tilde{\Gamma}(k)})$
- $I^* = \text{SAtoS}(I, \text{MSPT}(\hat{\mathbf{x}}_{\tilde{\Gamma}(k)})$;
- $\hat{\mathbf{x}}^{(k)} = (I^*)^T \hat{\mathbf{x}}_{\tilde{\Gamma}(k)}$
- $\mathbf{y}^{(k)} = A_{\Gamma(k)}\, \hat{\mathbf{x}}^{(k)}$

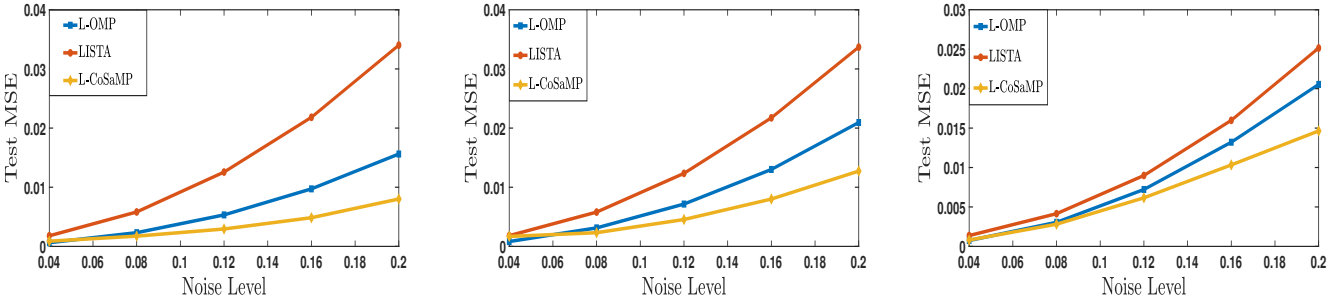
end

Algorithm 3: L-CoSaMP Network

Input: $\mathbf{y} \in \mathbb{R}^{m\times 1}$
Output: $\hat{\mathbf{y}} \in \mathbb{R}^{m\times 1}$
Initialization:
$\mathbf{y}^{(0)} = \mathbf{r}^{(0)} = \mathbf{y}, \quad A_{\Gamma(0)} = \emptyset$
for $k = 1, 2, \ldots$ **do**
- $\{\hat{\mathbf{x}}^{(k)}, \mathbf{y}^{(k)}, A_{\Gamma(k)}\}$ = L-CoSaMP k^{th}Layer$(\mathbf{y}, \mathbf{y}^{(k-1)}, A, A_{\Gamma(k-1)})$
- $\mathbf{r}^{(k)} = \mathbf{y} - \mathbf{y}^{(k)}$
- **if** $\|\mathbf{r}^{(k)}\|_2 > \|\mathbf{r}^{(k-1)}\|_2$ **then**
  - $\hat{\mathbf{y}} = \hat{\mathbf{y}}^{(k-1)}$
  - break
- **end**

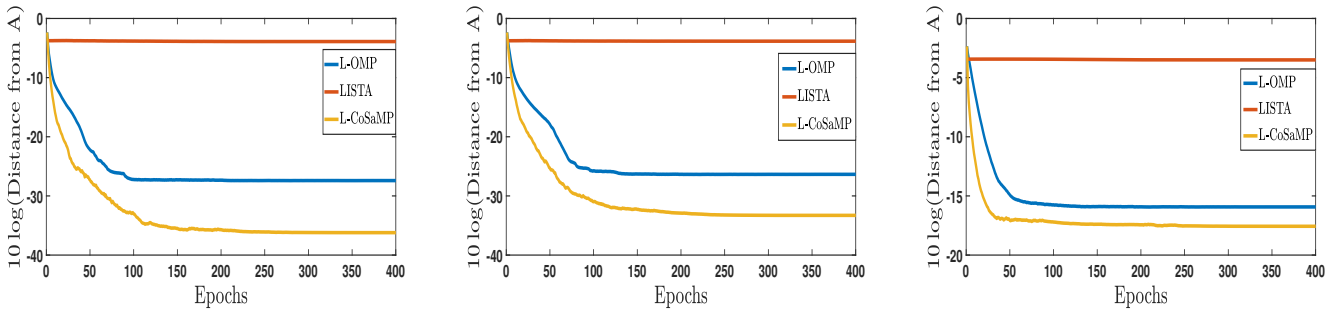
end

[1] M2SPT adds 2s atoms to the support in each layer.
[2] Similar to M2SPT, 2SAtoS extracts 2s specific atoms from dictionary A.



(a) Test MSE for sparsity 5 (b) Test MSE for sparsity 7 (c) Test MSE for sparsity 10

Fig. 1: Denoising performance comparison of various unrolled networks under different sparsity levels during testing. In all scenarios, L-CoSaMP outperforms both L-OMP and LISTA networks.



(a) Dictionary learning for sparsity level 5 (b) Dictionary learning for sparsity level 7 (c) Dictionary learning for sparsity level 10

Fig. 2: Comparison of dictionary learning performance across networks at varying sparsity levels. In all scenarios, L-CoSaMP surpasses other unrolled networks.

configured with 15, 10, and 10 layers, respectively. These networks are trained using the previously described training data over 400 epochs and a batch size of 50. The learning rates are set to 0.002, 0.00001, and 0.004 for the L-OMP, LISTA, and L-CoSaMP networks, respectively.

Test MSE Result: First, we consider noise levels of $\{0.04, 0.08, 0.12, 0.16, 0.2\}$ for the testing datasets. For each noise level, we construct 2000 test data points by adding a noise vector to the original signal $\mathbf{y}$, where the entries of the noise vector follow a Gaussian distribution with a mean of zero and a variance corresponding to the noise level. The test mean squared error (MSE) is computed for the 2000 new test datasets using the trained networks. Figure 1 presents the comparison of noise levels and test MSE for the L-OMP, LISTA, and L-CoSaMP networks for different sparsity levels. As evident from the figure, the L-CoSaMP network achieves superior test MSE performance compared to LISTA and L-OMP. Therefore, we conclude that the L-CoSaMP network is more effective at denoising signals under higher noise levels.

Dictionary Learning: Figure 2 presents the logarithmic-scale distance between the learned dictionary and the true dictionary for the given networks in different sparsity scenarios. The dictionary distance is calculated as the average distance between each atom in $A_{\text{true}} = A$ and its closest atom in A_{learned}, using the formula $\frac{1}{n}\sum_{i=1}^{n} \min(\mathbf{1} - |A_{\text{learned}}^{\top}\mathbf{a}_i|)$, where $\mathbf{1}$ represents a scalar vector whose all the entries are set to be 1 and $\mathbf{a}_i$ denotes the i-th column of A. The results reveal that the dictionary learned by L-CoSaMP converges more effectively to the true dictionary than those learned by L-OMP and LISTA for all different sparsity scenarios. This superior convergence explains why L-CoSaMP exhibits enhanced denoising performance relative to the other networks (see Figure 1).

V. Conclusions

In this paper, we proposed the unrolling of the CoSaMP algorithm, termed L-CoSaMP, by incorporating MSPT and SAtoS units. Through experiments on synthetic datasets, we demonstrated that L-CoSaMP exhibits superior denoising capabilities compared to LISTA and L-OMP, particularly in high-noise scenarios. Additionally, we observed that the dictionary learned by L-CoSaMP converges more rapidly to the true dictionary than those learned by LISTA and L-OMP, underscoring its effectiveness and robustness in signal denoising tasks.

References

[1] D. Donoho, "Compressed sensing," *IEEE Transactions on Information Theory*, vol. 52, no. 4, pp. 1289–1306, 2006.

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

[3] S. Foucart, H. Rauhut, S. Foucart, and H. Rauhut, *An invitation to compressive sensing*. Springer, 2013.

[4] M. Elad and M. Aharon, "Image denoising via sparse and redundant representations over learned dictionaries," *IEEE Transactions on Image Processing*, vol. 15, no. 12, pp. 3736–3745, 2006.

[5] W. Dong, L. Zhang, G. Shi, and X. Wu, "Image deblurring and super-resolution by adaptive sparse domain selection and adaptive regularization," *IEEE Transactions on Image Processing*, vol. 20, no. 7, pp. 1838–1857, 2011.

[6] Y. Luo, Y. Xu, and H. Ji, "Removing rain from a single image via discriminative sparse coding," in *2015 IEEE International Conference on Computer Vision (ICCV)*, pp. 3397–3405, 2015.

[7] R. Giryes and M. Elad, "Sparsity based poisson inpainting," in *2014 IEEE International Conference on Image Processing (ICIP)*, pp. 2839–2843, 2014.

[8] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Transactions on Image Processing*, vol. 17, no. 1, pp. 53–69, 2008.

[9] B. Huang, H. Song, H. Cui, J. Peng, and Z. Xu, "Spatial and spectral image fusion using sparse matrix factorization," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 52, no. 3, pp. 1693–1704, 2014.

[10] J. Yang, J. Wright, T. S. Huang, and Y. Ma, "Image super-resolution via sparse representation," *IEEE Transactions on Image Processing*, vol. 19, no. 11, pp. 2861–2873, 2010.

[11] L. Pfister and Y. Bresler, "Tomographic reconstruction with adaptive sparsifying transforms," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6914–6918, 2014.

[12] M. Lustig, D. L. Donoho, J. M. Santos, and J. M. Pauly, "Compressed sensing mri," *IEEE Signal Processing Magazine*, vol. 25, no. 2, pp. 72–82, 2008.

[13] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning," *Proceedings of the IEEE*, 2023.

[14] N. Shlezinger, J. Whang, Y. C. Eldar, and A. G. Dimakis, "Model-based deep learning: Key approaches and design guidelines," in *2021 IEEE Data Science and Learning Workshop (DSLW)*, pp. 1–6, IEEE, 2021.

[15] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition," in *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pp. 40–44 vol.1, 1993.

[16] D. Needell and J. A. Tropp, "Cosamp: Iterative signal recovery from incomplete and inaccurate samples," *Applied and computational harmonic analysis*, vol. 26, no. 3, pp. 301–321, 2009.

[17] I. Daubechies, M. Defrise, and C. De Mol, "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, vol. 57, no. 11, pp. 1413–1457, 2004.

[18] S. Nah, T. Hyun Kim, and K. Mu Lee, "Deep multi-scale convolutional neural network for dynamic scene deblurring," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 3883–3891, 2017.

[19] K. Zhang, W. Zuo, Y. Chen, D. Meng, and L. Zhang, "Beyond a gaussian denoiser: Residual learning of deep cnn for image denoising," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3142–3155, 2017.

[20] C. Dong, C. C. Loy, K. He, and X. Tang, "Image super-resolution using deep convolutional networks," *IEEE transactions on pattern analysis and machine intelligence*, vol. 38, no. 2, pp. 295–307, 2015.

[21] H. Yao, F. Dai, S. Zhang, Y. Zhang, Q. Tian, and C. Xu, "Dr2-net: Deep residual reconstruction network for image compressive sensing," *Neurocomputing*, vol. 359, pp. 483–493, 2019.

[22] X. Sun, N. M. Nasrabadi, and T. D. Tran, "Supervised deep sparse coding networks for image classification," *IEEE Transactions on Image Processing*, vol. 29, pp. 405–418, 2020.

[23] H. Zhang, Z. Dong, Z. Wang, L. Guo, and Z. Wang, "Csnet: A deep learning approach for ecg compressed sensing," *Biomedical Signal Processing and Control*, vol. 70, p. 103065, 2021.

[24] H.-J. Yoo, "Deep convolution neural networks in computer vision: a review," *IEIE Transactions on Smart Processing and Computing*, vol. 4, no. 1, pp. 35–43, 2015.

[25] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing," *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.

[26] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th international conference on international conference on machine learning*, pp. 399–406, 2010.

[27] G. Revach, N. Shlezinger, X. Ni, A. L. Escoriza, R. J. Van Sloun, and Y. C. Eldar, "KalmanNet: Neural network aided kalman filtering for partially known dynamics," *IEEE Transactions on Signal Processing*, vol. 70, pp. 1532–1547, 2022.

[28] R. J. Van Sloun, R. Cohen, and Y. C. Eldar, "Deep learning in ultrasound imaging," *Proceedings of the IEEE*, vol. 108, no. 1, pp. 11–29, 2019.

[29] S. B. Shah, P. Pradhan, W. Pu, R. Randhi, M. R. Rodrigues, and Y. C. Eldar, "Optimization guarantees of unfolded ISTA and ADMM networks with smooth soft-thresholding," *IEEE Transactions on Signal Processing*, 2024.

[30] Y. B. Sahel, J. P. Bryan, B. Cleary, S. L. Farhi, and Y. C. Eldar, "Deep unrolled recovery in sparse biological imaging: Achieving fast, accurate results," *IEEE Signal Processing Magazine*, vol. 39, no. 2, pp. 45–57, 2022.

[31] R. Khatib, D. Simon, and M. Elad, "Learned greedy method (lgm): A novel neural architecture for sparse coding and beyond," *Journal of Visual Communication and Image Representation*, vol. 77, p. 103095, 2021.

[32] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in *Proceedings of the IEEE international conference on computer vision*, pp. 370–378, 2015.

[33] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, 12 2014.