REST: ROBUST LEARNED SHRINKAGE-THRESHOLDING UNROLLED NETWORK

Wei Pu[⋆] *Chao Zhou*[⋆] *Yonina C. Eldar*[†] *Miguel R.D. Rodrigues*[⋆]

[⋆] Department of Electronic and Electrical Engineering, University College London, UK
[†] Weizmann Institute of Science, Rehovot, Israel.

ABSTRACT

We consider compressive sensing problems with model mismatch where one wishes to recover a sparse high-dimensional vector from low-dimensional observations subject to uncertainty in the measurement operator. In particular, we design a new robust deep neural network architecture by applying algorithm unfolding techniques to a robust version of the underlying recovery problem. Our proposed network – named Robust lErned Shrinkage-Thresholding (REST) – exhibits additional features including enlarged number of parameters and normalization processing compared to state-of-the-art deep architecture Learned Iterative Shrinkage-Thresholding Algorithm (LISTA), leading to the reliable recovery of the signal under sample-wise varying model mismatch. Our proposed network is also shown to outperform LISTA in compressive sensing problems under sample-wise varying model mismatch.

***Index Terms*—** Inverse Problems, Compressive Sensing Problems, Model Mismatch, Robustness, Deep Learning

1. INTRODUCTION

Inverse problems – involving the recovery of a high-dimensional vector of interest from a low-dimensional observation vector – arise in various relevant signal and image processing applications such as compressive sensing [1, 2], compressive radar [3, 4], medical imaging [5], and many more. However, without any assumptions, it is not generally possible to recover the quantities of interest from the observations because the problem is typically highly ill-posed [6].

Three major classes of approaches have therefore been developed over the years to solve such inverse problems. Classical model-based techniques leverage knowledge of the underlying linear model to solve inverse problems via the formulation of optimization problems that include two terms in the objective: (1) a data fidelity term and (2) a data regularization. The fidelity term encourages the solution to be consistent with the observations whereas the regularization one encourages solutions that conform to a certain postulated data prior. For example, variational methods use a regularizer that promotes smoothness of the solutions [7, 8] whereas sparsity-driven methods use regularizers that promote sparse solutions in some transform domain [9, 10].

More recent data-driven techniques "solve" an inverse problem by using powerful neural networks that learn how to map the model output to the model input based on a number of input-output examples [11, 12]. These data-driven approaches require rich enough datasets – which may not be always available in various domains – in order to learn how to solve the inverse problem. These approaches however can offer superior performance compared with model based ones [12].

Finally, in view of the fact that the underlying inverse problem model is often (approximately) known in various scenarios, there has been an increased interest in model-aware data-driven approaches to inverse problems via the adoption of algorithm unfolding (or unrolling) techniques [13–15]. This line of work dates back to the seminal work by Gregor and LeCun [13] showing that it is possible to map certain inverse problems iterative solvers onto a deep network architecture whose parameters can be further learnt to deliver state-of-the-art performance. It has seen a renaissance in recent years [15] because they can combine the merits of model based approaches (interpretability) and data driven ones (state of the art performance).

However, the presence of any mismatch between the actual model and the postulated one – underlying the operation of the different approaches – can lead to a serious performance degradation for model-based techniques and data-driven approaches. The impact of model mismatch in inverse problems such as compressive sensing has also been studied in detail in [17]. Such mismatches might arise in practice due erroneous assumptions about the measurement operator [16], erroneous modelling assumptions [17], or other problems.

Model-aware data driven approaches such as LISTA can cope well with scenarios where the model mismatch is fixed for different data samples. In particular, since the assumed measurement model is utilized as a initialization, any deviation between the postulated measurement model and the true one can be learnt through the training procedure. However, the situation is totally different where model mismatch varies between different data samples or between training and testing data. It is very hard to estimate sample-wise varying model mismatch in the existing model-aware data-driven approaches, and correspondingly, the recovery performance will seriously decline.

There has been some work on how to design model-based approaches that can effectively mitigate the impact of model mismatches in inverse problems. These include total-least squares based recovery algorithms [18] along with the matrix-uncertainty generalized approximate message-passing (MU-GAMP) algorithm [19]. However, there appears to be less work on how to design data-driven approaches – specially deep learning ones – that can effectively mitigate the challenge with sample-wise varying model mismatch. In fact, it has often been shown that, in the presence of mismatches, the performance of model based approaches to in-

verse problems can be significantly better than deep learning based ones [20].

In this paper, we show however that it is possible to design robust neural network architectures – leveraging unfolding techniques –allowing one to recover a high-dimensional vector from low-dimensional measurements subject to uncertainty in the compressive measurement operator within the context of compressive sensing problems. In particular, by building upon robust model-based algorithms, we develop a Robust lErned Shrinkage-Thresholding (REST) network by unrolling one iteration in a proposed robust ISTA algorithm into one layer and stacking several layers together. In addition, various experiments and observations are showing that REST outperform other model-based approach and learning based approaches such as LISTA in the presence of sample-wise varying model mismatch in inverse problems. This could capture scenarios where one desires to use a single network to solve different compressive sensing tasks where the measurement matrices might differ slightly.

2. PROBLEM FORMULATION

Consider a conventional linear inverse problem given by:

$$y = Ax + e, \tag{1}$$

where $y \in \mathbb{R}^{M\times 1}$ is an observation vector, $x \in \mathbb{R}^{N\times 1}$ is the vector of interest, and $e \in \mathbb{R}^{M\times 1}$ is measurement noise. The matrix $A \in \mathbb{R}^{M\times N}$ models the forward operator where $N > M$.

It is not generally possible to recover x from y when $N > M$ unless one makes additional assumptions about the structure of the vector. In particular, by postulating that the vector of interest x is sparse, a popular approach to recover x from y involves using the least-absolute shrinkage and selection operator (Lasso) [21] given as the solution to:

$$\min_x \|y - Ax\|_2 + \lambda\|x\|_1, \tag{2}$$

where $\|\cdot\|_2$ is the l_2 norm and $\|\cdot\|_1$ denotes ℓ_1-norm. This optimization problem can be solved using the well-known iterative soft thresholding algorithm (ISTA) [22] or alternating direction method of multipliers (ADMM) [23].

Alternatively, one can adopt algorithm unfolding or unrolling techniques [15] to map such solvers onto a neural network architecture, whose parameters can then be further tuned using gradient descent or some variant based on the availability of a series of examples $\{(x_i, y_i\}_{i=1}^n$. Networks derived from ISTA or ADMM known as LISTA [13] and ADMM-CSNet [24] respectively, have been shown to perform much better than purely learnt networks (e.g. [12]) or ISTA [22] or ADMM [23].

Consider now a more challenging scenario where the observation vector $y \in \mathbb{R}^{M\times 1}$ is related to the vector of interest $x \in \mathbb{R}^{N\times 1}$ as follows:

$$y = (A + E)x + e. \tag{3}$$

The model in (1) differs from the model in (3) in that the forward operator A – which is assumed to be known – is now also contaminated by an error matrix E – which is assumed to be unknown. Therefore, in addition to noise, one now needs to recover the vector of interest from the observation vector in the presence of model mismatch.

To recover x in this case, one may consider a robust version of LASSO, or an ℓ_1 regularized version of total least squares [18]:

$$\min_{x,e,E} \|e\|_2 + \|E\|_F + \lambda\|x\|_1, \quad \text{s.t.} \quad y = (A + E)x + e. \tag{4}$$

Our goal is to build upon this formulation in order to design an unfolded network that can be used to recover x reliably from y in the presence of model mismatch that can vary from data sample to data sample..

3. ROBUST LEARNED SHRINKAGE-THRESHOLDING NETWORK

3.1. Robust ISTA

We start by designing an ISTA-like iterative algorithm in order to recover x from y in the presence of model mismatch.

First, as shown in [18], we can convert the optimization problem in (4) that delivers both x, e and E into an optimization problem only on x as follows:

$$\min_x \frac{\|y - Ax\|_2^2}{1 + \|x\|_2^2} + \lambda\|x\|_1. \tag{5}$$

We use a proximal gradient methods in order to design an iterative algorithm to recover x from y. In particular, by taking the gradient on the first term in (5) and executing a proximal step on the second term, we end up with a series of iterations:

$$x^{k+1} = \mathcal{S}_{\mu\lambda}\left\{x^k - \mu g^k\right\}, \tag{6}$$

where x^{k+1} represents the $(k+1)$-th iterate, x^k is the k-th iterate, $\mu > 0$ is a step size, and g^k is the gradient of the first term of (5) evaluated at x^k and is given by

$$g^k = \frac{2}{(1 + \|x^k\|_2^2)^2}\left[(1 + \|x^k\|_2^2)A^T(Ax^k - y) - \|y - Ax^k\|_2^2 x^k\right]. \tag{7}$$

The operator $S_{\mu\lambda}(\cdot)$ is the soft thresholding operator, and is applied element-wise on its vector argument as $\mathcal{S}_{\mu\lambda}\{x\} = \text{sign}(x) \cdot \max(|x| - \mu\lambda, 0)$.

The iterative algorithm in (6) can be seen as a robust version of the ISTA algorithm, because both ISTA and robust ISTA in (6) solve a ℓ_1 regularized version of a least square problem using proximal gradient methods. However, robust ISTA faces a robust problem in (4) and (5), wherein model mismatch is taken into consideration.

We next adopt unfolding techniques in order to map this robust recovery algorithm onto a robust neural network that can be used to recover high-dimensional sparse vectors from low-dimensional noisy measurements in the presence of model mismatch.

3.2. Robust Learned Shrinkage-Thresholding Network

3.2.1. Architecture

To derive our network architecture, we first plug (7) into (6) which results in

$$x^{k+1} = \mathcal{S}_{\mu\lambda}\left(\frac{1-2\mu\|y-Ax^k\|_2^2}{(1+\|x^k\|_2^2)^2}x^k + \frac{2\mu A^T A}{1+\|x^k\|_2^2}x^k - \frac{2\mu A^T}{1+\|x^k\|_2^2}y\right) \tag{8}$$

We next also re-write (8) as follows

$$x^{k+1} = \mathcal{S}_{\mu\lambda}\left(\frac{1-2\mu\|y-A_1x^k\|_2^2}{(1+\|x^k\|_2^2)^2}x^k + \frac{2\mu A_2}{1+\|x^k\|_2^2}x^k - \frac{2\mu A_3}{1+\|x^k\|_2^2}y\right) \tag{9}$$

where we have replaced A in the first term by A_1, A^TA in the second term by A_2, and A^T in the third term by A_3.

We can now immediately map the iterative algorithm in (8) onto a neural network architecture using unfolding techniques. In particular,

- We map each iteration of the algorithm onto a layer of the neural network. Such a layer produces an output x^{k+1} – which is input to the next layer – based on an input x^k that undergoes a series of transformation such as linear transformations, linear scalings, and soft-thresholding operations.
- We then stack various such layers in order to produce an overall neural network. Such a network consisting of K layers corresponds to K iterations of the original robust ISTA algorithm, it accepts as input an initialization x^0, and it delivers as output an approximate solution x^K.
- Finally, we use different learnable parameters per network layer corresponding to the original parameters. Concretely, we use learnable parameters λ^{k+1}, μ^{k+1}, A_1^{k+1}, A_2^{k+1} and A_3^{k+1} associated with the k-th layer in lieu of the parameters λ, μ, A_1, A_2 and A_3 associated with our learning algorithm.

The architecture of our network is depicted in Fig. 1.

3.2.2. Learning Algorithm

To optimize the learnable parameters we rely on a dataset containing a series of pairs (x_l, y_l), $l = 1, ..., L$ where x_l corresponds to the original sparse vector and y_l corresponds to the observation vector

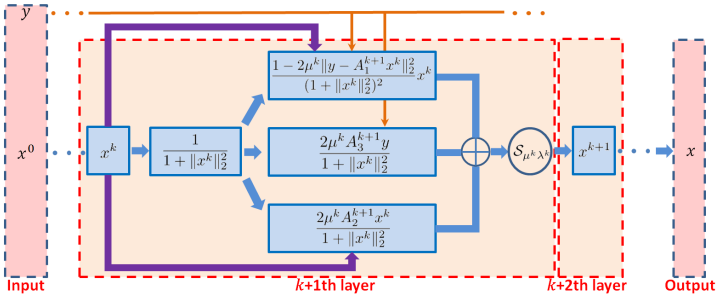


Fig. 1. REST Neural Network Architecture.

derived from the linear model in (3) for some fixed forward operator A and some unknown forward operator perturbation E (that may vary from example to example).

We also rely on a standard cost function measuring the difference between the network prediction of the vector of interest and the ground truth vector of interest as follows:

$$\min_{\theta} \frac{1}{L}\sum_{l=1}^{L}\|\hat{x}^l - x^l\|_2^2 \tag{10}$$

where $\hat{x}_l$ corresponds to the network output given network input y_l, x_l corresponds to the ground truth and θ aggregates the various learnable parameters. We can then adopt standard gradient descent algorithms in order to learn the network parameters $\lambda^k, \mu^k, A_1^k, A_2^k$ and A_3^k.

3.3. REST vs LISTA

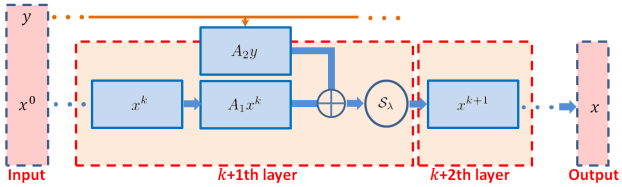


Fig. 2. LISTA Neural Network Architecture.

We will see in the sequel that REST can perform substantially better than other unfolding approaches such as LISTA in the presence of inverse problems subject to model uncertainty. It is therefore instructive to compare the REST network architecture to the LISTA architecture (cf. Fig 1 vs Fig 2).

The main difference between a layer in LISTA and in REST lies in two additional processing operations: 1) one corresponds to the first term in (8) and 2) the other corresponds to the normalization operation by $1 + \|x\|_2^2$. In fact, without these operations, the REST architecture immediately reduces to the LISTA one. It thus plausible that these additional operations play a critical role in mitigation of sample-wise model mismatch::

- the first operation critically enlarges number of parameters, allowing for a much better fit in the presence of model mis-

match. We will indeed see in the sequel that LISTA does not fit well to the training data whereas REST does much better.

- the second operation seems to play a role in the regularization of the distribution of the network output. With the normalization by $1 + \|x\|_2^2$, the network output seems to be more uniformly distributed – hence mimicking better the network input distribution. Without such normalization, the network output exhibits a much more skewed distribution.

4. EXPERIMENTS

We now compare the performance of REST to well-known LISTA for a simple compressive sensing task where one wishes to recover a vector of interest from a vector of noisy linear measurements, in the presence of model mismatch that may differ for different linear measurements. We also report results for Basis Pursuit (BP), ISTA and robust ISTA.

4.1. Experiment setup

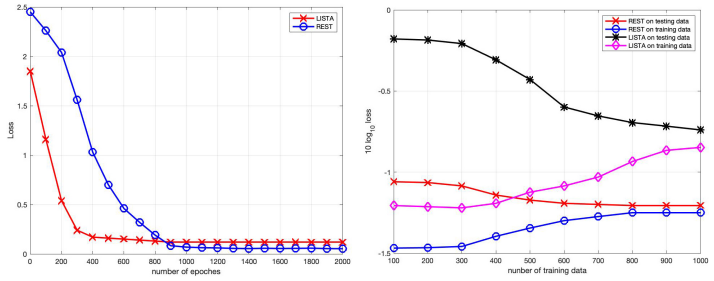
Our experimental set-up involves the generation of synthetic data, where $y \in \mathbb{R}^M$ is generated via the model in (3), $x \in \mathbb{R}^{25}$ is a vector with sparsity equal to four with the non-zero elements randomly chosen within the interval $(0, 1]$, and e is a Gaussian vector with zero mean and variance $\sigma^2 = 0.03$. The measurement matrix $A \in \mathbb{R}^{M\times 25}$ is Gaussian with $\|A\|_F = 10$ and the perturbation matrix $E \in \mathbb{R}^{M\times 25}$ is also Gausssian with $\|E\|_F$ varying between $(0, 12]$.

We generate 1000 different sample pairs (x, y) where the matrix is A remains fixed, the matrix E varies from sample to sample, and the noise vector e also varies from sample to sample. For the learning based approaches REST and LISTA, a fraction of these sample pairs is used for training purposes and the remaining samples are used for testing purposes (i.e. evaluation of the performance of the approaches). For the model based BP, ISTA and robust ISTA, we use 100 samples only to evaluate the performance. We adopt as performance metric the mean squared error between the original vectors and the reconstructed ones.

4.2. Experimental Results

Fig. 3 (a) depicts the evolution of the loss in terms of the number of training epochs for REST and LISTA. We observe that LISTA converges faster than REST possibly because the number of learnable parameters is lower in LISTA than REST. [1] However, we can also observe that REST fits the data much better than LISTA in the presence of model mismatch.

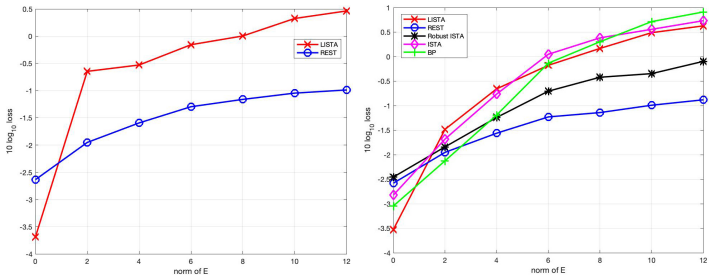
Fig. 3 (b) depicts training and testing error versus number of training samples for REST and LISTA. We note again that training error is much lower for REST than LISTA, especially for a larger number of training points. Likewise, the testing error is also much lower for REST in comparison with LISTA. Interestingly, in the presence of model mismatch, the generalization error – corresponding to the difference between testing and training errors – is also smaller for REST than LISTA.



(a) (b)

Fig. 3. REST vs LISTA: (a) Convergence (b) Performance for $M = 13$, $\|E\|_F = 5$.

Figs. 4 (a) and (b) depicts how the training error and the testing error behaves as a function of the level of mismatch for a range of approaches. It is clear that with the increase of mismatch LISTA training and testing errors become substantially worse than REST training and testing errors. Importantly, REST also outperforms model based algorithms such as BP, ISTA and notably robust ISTA.



(a) (b)

Fig. 4. Performance with different levels of model mismatch for $M = 10$. (a). training error. (b). testing error.

5. CONCLUSION

Deep learning has achieved significant successes in various signal and image processing tasks, but it is also known deep learning is very vulnerable to various perturbations. We show – by adopting algorithm unrolling techniques – that it is possible to design neural network architectures that can reliably recover high-dimensional sparse vectors from low-dimensional noisy linear measurements in the presence of a challenging sample-wise model mismatch setting. We also show the proposed REST network outperforms LISTA in view of various additional operations that seem to endow the architecture with some inherent robustness to mismatches.

[1] Here, both REST and LISTA are set to contain 5 layers. In each layer, REST has 2 learnable parameters as well as 3 learnable matrices, while LISTA has 1 learnable parameter and 2 learnable matrices.

6. REFERENCES

[1] Y. C. Eldar, and G. Kutyniok, "Compressed Sensing: Theory and Applications," Cambridge University Press, 2012.

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

[3] W. Pu, X. Wang, J. Wu, Y. Huang, J. Yang, "Video SAR Imaging Based on Low-Rank Tensor Recovery," *IEEE Trans. Neural Netw. Learn. Syst.*, 2020, doi:10.1109/TNNLS.2020.2978017.

[4] W. Pu, J. Wu, "OSRanP: A Novel Way for Radar Imaging Utilizing Joint Sparsity and Low-Rankness," *IEEE Trans. Comp. Imag.*, vol. 6, pp. 868–882, 2020.

[5] M. Azghani, P. Kosmas and F. Marvasti, "Microwave Medical Imaging Based on Sparsity and an Iterative Method With Adaptive Thresholding," *IEEE Trans. Medical Imag.*, vol. 34, no. 2, pp. 357–365, Feb. 2015

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

[7] L. I. Rudin, S. Osher and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1992.

[8] A. N., Tikhonov, V. Y. Arsenin, "Solutions of Ill-Posed Problems", 1977, New York: Winston.

[9] D. L. Donoho, M. Elad and V. N. Temlyakov, "Stable recovery of sparse overcomplete representations in the presence of noise," *IEEE Trans. Inf. Theory*, vol. 52, no. 1, pp. 6–18, Jan. 2006

[10] R. Tibshirani, "Regression Shrinkage and Selection via the Lasso," *Journal of the Royal Statistical Society, Series B (Methodological)*, vol. 73, pp. 267–288, 1996.

[11] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis and R. Willett, "Deep Learning Techniques for Inverse Problems in Imaging," *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 39–56, May 2020

[12] Chao Dong, Chen Change Loy, Kaiming He and Xiaoou Tang, "Image super-resolution using deep convolutional networks", *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 38, no. 2, pp. 295–307, 2016.

[13] K. Gregor and Y. LeCun, Learning fast approximations of sparse coding, in *Proc. 27th Int. Conf. Mach. Learn. (ICML)*, pp. 399–406, 2010.

[14] Y. Li, M. Tofighi, J. Geng, V. Monga, and Y. C. Eldar, "Efficient and interpretable deep blind image deblurring via algorithm unrolling," *IEEE Trans. Comp. Imag.*, vol. 6, pp. 666–681.

[15] V. Monga, Y. Li, Y. C. Eldar. Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing. [Online]. Available: https://arxiv.org/abs/1912.10557?context=cs, 2020.

[16] M. A. Herman and T. Strohmer, "General Deviants: An Analysis of Perturbations in Compressed Sensing," *IEEE J. Sel. Top. Signal Process.*, vol. 4, no. 2, pp. 342-349, April 2010

[17] Y. Chi, L. L. Scharf, A. Pezeshki and A. R. Calderbank, "Sensitivity to Basis Mismatch in Compressed Sensing," *IEEE Trans.Signal Process.*, vol. 59, no. 5, pp. 2182–2195, May 2011

[18] H. Zhu, G. Leus, G. B. Giannakis, "Sparsity-cognizant total least-squares for perturbed compressive sampling", *IEEE Trans.Signal Process.*, vol. 59, no. 5, pp. 2002–2016, 2011.

[19] J. T. Parker, V. Cevher, P. Schniter, "Compressive sensing under matrix uncertainties: Anapproximate message passing approach",in *Proceedings of Asilomar Conference Signals, Systems and Computers*, Pacific Grove, CA, USA, 2011, pp. 804–808.

[20] P. Song, X. Deng, J. F. C. Mota, N. Deligiannis, P. L. Dragotti and M. R. D. Rodrigues, "Multimodal Image Super-Resolution via Joint Sparse Representations Induced by Coupled Dictionaries," *IEEE Trans. Comp. Imag.*, vol. 6, pp. 57–72, 2020

[21] Tibshirani, and J. Ryan, "The Lasso Problem and Uniqueness," *Electronic Journal of Statistics* vol. 7, no. 1, pp. 1456–1490, 2013.

[22] B. Amir, and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM journal on imaging sciences*, vol. , no. 1, pp. 183–202, 2009.

[23] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2011.

[24] Y. Yan , et al. "ADMM-CSNet: A Deep Learning Approach for Image Compressive Sensing." *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no.3, pp. 521–538, 202.

[25] R. Arablouei,S.Werner,K.Do?an?ay, Analysis of the gradient-decent total least-squares algorithm, *IEEE Trans. Signal Process.*, vol. 62, no. 5, pp. 1256–1264, 2014.