Ada-LISTA: Learned Solvers Adaptive to Varying Models

Aviad Aberdam, Alona Golts, and Michael Elad, *Fellow, IEEE*

Abstract—Neural networks that are based on the unfolding of iterative solvers as LISTA (Learned Iterative Soft Shrinkage), are widely used due to their accelerated performance. These networks, trained with a fixed dictionary, are inapplicable in varying model scenarios, as opposed to their flexible non-learned counterparts. We introduce, Ada-LISTA, an adaptive learned solver which receives as input both the signal and its corresponding dictionary, and learns a universal architecture to serve them all. This scheme allows solving sparse coding in linear rate, under varying models, including permutations and perturbations of the dictionary. We provide an extensive theoretical and numerical study, demonstrating the adaptation capabilities of our approach, and its application to the task of natural image inpainting.

Index Terms—Sparse coding, learned solvers, lista, deep learning modeling

1 Introduction

Sparse coding is the task of representing a noisy signal $\mathbf{y} \in \mathbb{R}^n$ as a combination of few base signals (called "atoms"), taken from a matrix $\mathbf{D} \in \mathbb{R}^{n\times m}$ – the "dictionary". This is represented as the need to compute $\mathbf{x} \in \mathbb{R}^m$ such that $\mathbf{y} \approx \mathbf{D}\mathbf{x}$, such that $\|\mathbf{x}\|_0 \le s$, where the L^0-norm counts the non-zero elements, s is the cardinality of the representation, and $\mathbf{D}$ is often redundant ($m \ge n$). Among the various approximation methods for handling this NP-hard task, an appealing approach is a relaxation of the L^0 to an L^1-norm using Lasso or Basis-Pursuit [1], [2],

$$\underset{\mathbf{x}}{\text{minimize}} \frac{1}{2}\|\mathbf{y}-\mathbf{D}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1. \tag{1}$$

An effective way to address this optimization problem uses an iterative algorithm such as ISTA (Iterative Soft Thresholding Algorithm) [3], where the solution is obtained by iterations of the form

$$\mathbf{x}_{k+1} = \mathcal{S}_{\frac{\lambda}{L}}\left(\mathbf{x}_k + \frac{1}{L}\mathbf{D}^T(\mathbf{y}-\mathbf{D}\mathbf{x}_k)\right), k=0,1,.. \tag{2}$$

where $\frac{1}{L}$ is the step size, determined by the maximal eigenvalue of the Gram matrix $\mathbf{D}^T\mathbf{D}$, and $\mathcal{S}_\theta(x_i) = \text{sign}(x_i)(|x_i| - \theta_i)$ is the soft shrinkage function. Fast-ISTA (FISTA) [4] is a Nesterov momentum speed-up of the above iterative algorithm.

Note that ISTA has a much wider perspective: When aiming to minimize a function of the form

$$F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}), \tag{3}$$

where f, g are convex and g is possibly non-smooth, the solution is given by the proximal gradient method [5], [6]:

$$\mathbf{x}_{k+1} = \underset{g/L}{\text{prox}}\left(\mathbf{x}_k - \frac{1}{L}\nabla f(\mathbf{x}_k)\right), \tag{4}$$

where

$$\underset{g}{\text{prox}}(\mathbf{u}) = \arg\min_{\mathbf{v}} \frac{1}{2}\|\mathbf{v}-\mathbf{u}\|_2^2 + g(\mathbf{v}). \tag{5}$$

The above fits various optimization problems such as matrix completion [7], portfolio optimization [8], and non-negative matrix factorization [9].

Returning to the realm of sparse coding, the seminal work of LISTA (Learned-ISTA) [10] has shown that by unfolding K iterations of ISTA and freeing its parameters to be learned, one can achieve a substantial speedup over ISTA (and FISTA). Particularly, LISTA uses the following re-parametrization:

$$\mathbf{x}_{k+1} = \mathcal{S}_\theta(\mathbf{W}_1\mathbf{y} + \mathbf{W}_2\mathbf{x}_k), \quad k = 0, 1, ..., K-1, \tag{6}$$

where $\mathbf{W}_1$ and $\mathbf{W}_2$ re-parametrize the matrices $\frac{1}{L}\mathbf{D}^T$ and $(\mathbf{I} - \frac{1}{L}\mathbf{D}^T\mathbf{D})$ correspondingly. These two matrices and the scalar thresholding value θ are collectively referred to as $\Theta = (\mathbf{W}_1, \mathbf{W}_2, \theta)$ – the parameters to be learned. The model, denoted as $\mathcal{F}_K(\mathbf{y};\Theta)$, is trained by minimizing the squared error between the predicted sparse representations at the Kth unfolding $\mathbf{x}_K = \mathcal{F}_K(\mathbf{y};\Theta)$, and the optimal codes $\mathbf{x}$ obtained by running ISTA itself (necessarily using the dictionary $\mathbf{D}$),

$$\underset{\Theta}{\text{minimize}} \sum_{i=1}^{N} \left\|\mathcal{F}_K(\mathbf{y}^{(i)};\Theta) - \mathbf{x}^{(i)}\right\|_2^2, \tag{7}$$

where the superscript (i) denotes the sample index and N the number of training examples. During inference, LISTA

- *Aviad Aberdam is with the Department of Electrical Engineering, Technion Institute of Technology, Haifa 3200003, Israel. E-mail: aaberdam@cs.technion.ac.il.*
- *Alona Golts and Michael Elad are with the Department of Computer Science, Technion Institute of Technology, Haifa 3200003, Israel. E-mail: {salonaz, elad}@cs.technion.ac.il.*

Manuscript received 24 Jan. 2021; revised 24 Oct. 2021; accepted 29 Oct. 2021. Date of publication 4 Nov. 2021; date of current version 3 Nov. 2022.
This work was supported in part by the Israel Science Foundation (ISF) under Grant 335/18 and in part by the Technion Hiroshi Fujiwara Cyber Security Research Center and the Israel Cyber Bureau.
(Corresponding author: Aviad Aberdam.)
Recommended for acceptance by J. Mairal.
Digital Object Identifier no. 10.1109/TPAMI.2021.3125041

requires only the test signals, without their underlying dictionary. LISTA generalizes well for signals of the same distribution as the train set, allowing a significant speedup versus its non-learned counterparts [10]. A possible explanation for LISTA's success is that it fits itself to the input distribution, whereas non-learned solvers do not assume anything on the input data. Specifically, in sparse coding, the input signals are restricted to a union of low-dimensional Gaussians, as they are generated by a linear combination of few atoms. By focusing solely on such signals, LISTA achieves its acceleration. Note that the original dictionary is hard-coded into the model weights, via the ground truth solutions used during supervised training. Given a new test sample emerging from a slightly deviated (yet known) model/dictionary, LISTA will likely deteriorate in performance, whereas ISTA, using the relevant dictionary, is expected to provide a robust and consistent result.

Another drawback of LISTA is its relevance to a single dictionary, requiring separate and renewed training if the model evolves over time. Such is the case in video related applications as enhancement [11] or surveillance [12]. Similarly, in some image restoration problems, the model encapsulated by the dictionary is often corrupted by an additional constant perturbation, e.g., the sensing matrix in compressive sensing [13], the blur kernel in non-blind image deblurring [14], and a spatially-varying mask in image inpainting [15]. In all these cases, deployment of the classic framework of LISTA necessitates a newly trained network for each new dictionary. An alternative to the above is incorporating LISTA as a fixed black-box denoiser, and merging it within the plug-and-play [16] or RED [17] schemes, significantly increasing the inference complexity.

1.1 Main Contributions

Our aim in this work is to extend the applicability of LISTA to scenarios of model perturbations and varying signal distributions. More specifically,

- We bridge the gap between the efficiency and fast convergence rate of LISTA, and the adaptivity and applicability of ISTA, by introducing "Ada-LISTA" (Adaptive-LISTA). Our training is based on pairs of signals and corresponding dictionaries, learning a generic architecture that wraps the dictionary by two auxiliary weight matrices. At inference, our model can accommodate both the signal and its dictionary, allowing a variety of model modifications without repetitive re-training.
- We perform extensive numerical experiments, demonstrating the robustness of our model to three types of dictionary perturbations: column permutations, additive Gaussian noise, and completely renewed random dictionaries. We show that Ada-LISTA can handle complex and varying signal models while still providing an impressive advantage over both learned and non-learned solvers.
- We theoretically prove that our scheme achieves a linear convergence rate under a constant dictionary. More importantly, we allow for noisy modifications and random permutations of the dictionary and prove that robustness remains, with an ability to reconstruct the ideal sparse representations with the same linear rate.
- We demonstrate our approach on natural image inpainting, which cannot be used directly with fixed models as LISTA. We show a clear advantage of Ada-LISTA versus its non-learned counterparts.

More broadly, our study contributes to the understanding of learned solvers and their ability to accelerate convergence. Past work suggests that the signal model $p(\mathbf{y})$ should be structured and fixed for successful learning of such solvers. Our work reveals that effective learning can be achieved with a weaker constraint – a fixed conditional distribution of the data given the model $p(\mathbf{y}|\mathbf{D})$.

2 Proposed Method

Thus far, one could either benefit from a high convergence rate using a learned solver as LISTA, while restricting the signals to a specific model, or employ a less effective, non-learned solver as ISTA, capable of handling any pair of signal and its generative model. We hereby introduce our novel "Ada-LISTA" architecture, combining both benefits. Beyond enjoying the acceleration of learned solvers, we incorporate the dictionary as part of the input at both training and inference time, allowing for adaptivity to different models. Fig. 1 provides our architecture, based on the following:

Definition 1 (Ada-LISTA). *The Ada-LISTA solver is defined by the following iterative step:*

$$\mathbf{x}_{k+1} = \mathcal{S}_{\theta_{k+1}}\Big(\big(\mathbf{I} - \gamma_{k+1}\mathbf{D}^T\mathbf{W}_2^T\mathbf{W}_2\mathbf{D}\big)\mathbf{x}_k + \gamma_{k+1}\mathbf{D}^T\mathbf{W}_1^T\mathbf{y}\Big). \tag{8}$$

The inputs are $\mathbf{y}$ *and* $\mathbf{D}$, *while* $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{n\times n}$, $\{\gamma_k, \theta_k\}_{k=1}^K$ *are the learned parameters* (Θ).

Note that the particular placement of the learned matrices versus the dictionary in (8), and the subsequent architecture of the unfolded network, are both flexible and can be altered according to the desired application. In scenarios where the dictionary is noticeably overcomplete, i.e. $m \gg n$, the suggested framework may lack the required amount of parameters, since $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{n\times n}$. In this case, the location of either $\mathbf{W}_1, \mathbf{W}_2$ can be swapped, as later suggested in Section 5.3. Overall, Ada-LISTA is a holistic approach, suggesting that both the signal and the dictionary serve as inputs to the network, allowing for model adaptivity.

The inference and training for Ada-LISTA and its accelerated LFISTA version are detailed in Algorithms 1,2. We consider a similar loss as Eq. (7), while incorporating the dictionaries,

$$\underset{\Theta}{\text{minimize}} \sum_{i=1}^{N} \left\| \mathcal{F}_K(\mathbf{y}^{(i)}, \mathbf{D}^{(i)}; \Theta) - \mathbf{x}^{(i)} \right\|_2^2. \tag{9}$$

This learning regime is supervised, requiring reference representations $\mathbf{x}^{(i)}$ to be computed using ISTA. An unsupervised alternative can be envisioned, as in [9], [18], where the loss is

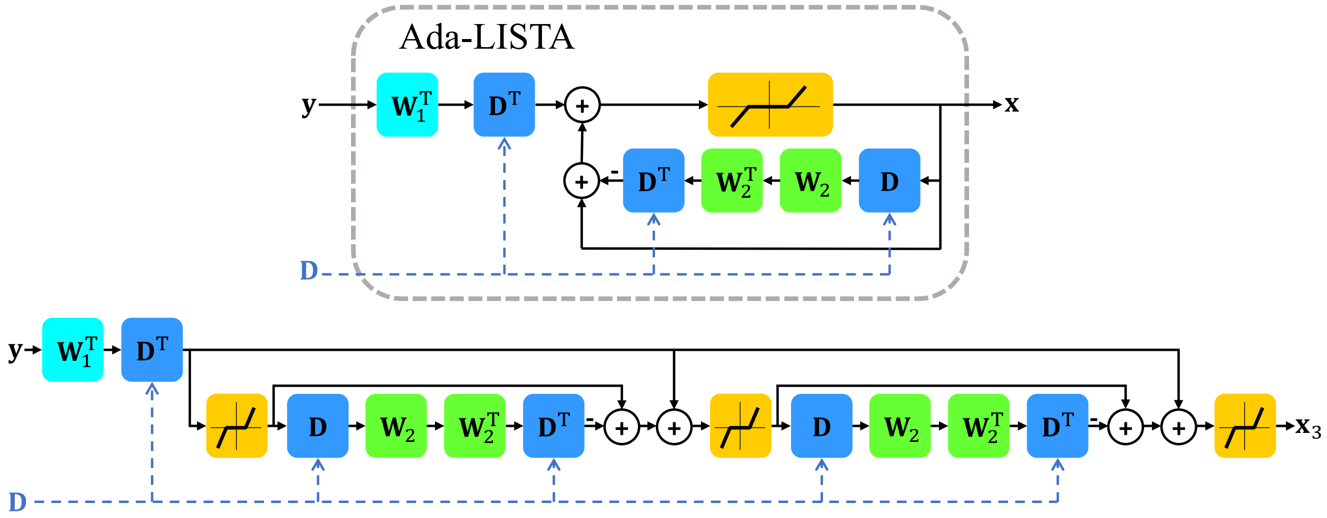


Fig. 1. Ada-LISTA architecture as an iterative model (top), and its unfolded version for three iterations (bottom). The input dictionary $\mathbf{D}$ is embedded in the architecture, while the matrices $\mathbf{W}_1, \mathbf{W}_2$ are free to be learned. For clarity, the sample indices of $\mathbf{y}, \mathbf{D}, \mathbf{x}_3$ have been omitted.

$$\min_{\Theta} \sum_{i=1}^{N} \left\| \mathbf{y}^{(i)} - \mathbf{D}^{(i)} \mathcal{F}_K(\mathbf{y}^{(i)}, \mathbf{D}^{(i)}; \Theta) \right\|_2^2 + \lambda \left\| \mathcal{F}_K(\mathbf{y}^{(i)}, \mathbf{D}^{(i)}; \Theta) \right\|_1. \tag{10}$$

In this work we focus on the supervised learning option. Several key questions arise on the applicability of Ada-LISTA: Does it work? Does it compromise performance, when compared to LISTA trained on each separate model? Can it handle completely random models? Can theoretical guarantees be provided on its convergence rate, or adaptation capability? We aim to answer these questions, starting by proving *linear rate* convergence under a *varying* model.

Algorithm 1. Ada-L(F)ISTA Inference

```
Input: signal $\mathbf{y}$, dictionary $\mathbf{D}$
Init: $\mathbf{x}_0 = \mathbf{0}$, $\mathbf{z}_0 = \mathbf{0}$, $t_0 = 1$
for $k = 0$ to $K - 1$ do
  $\mathbf{x}_{k+1} = \mathcal{S}_{\theta_{k+1}}\big((\mathbf{I} - \gamma_{k+1}\mathbf{D}^T\mathbf{W}_2^T\mathbf{W}_2\mathbf{D})\mathbf{z}_k$
      $+\gamma_{k+1}\mathbf{D}^T\mathbf{W}_1^T\mathbf{y}\big)$
  if Ada-LISTA then
    $\mathbf{z}_{k+1} = \mathbf{x}_{k+1}$
  else if Ada-LFISTA then
    $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
    $\mathbf{z}_{k+1} = \mathbf{x}_{k+1} + \frac{t_k - 1}{t_{k+1}}(\mathbf{x}_{k+1} - \mathbf{x}_k)$
Return: $\mathcal{F}_K(\mathbf{y}, \mathbf{D}; \Theta) = \mathbf{x}_K$
```

Algorithm 2. Ada-LISTA Training

```
Input: pairs of signals and dictionaries $\{\mathbf{y}^{(i)}, \mathbf{D}^{(i)}\}_{i=1}^N$, and the reference value of $\lambda$
Preprocessing: find $\mathbf{x}^{(i)}$ for each pair $(\mathbf{y}^{(i)}, \mathbf{D}^{(i)})$ by solving Eq. (1) using ISTA with $\lambda$
Goal: learn $\Theta = (\mathbf{W}_1, \mathbf{W}_2, \{\theta_k, \gamma_k\}_{k=1}^K)$
Init: $\mathbf{W}_1, \mathbf{W}_2 = \mathbf{I}$, $\theta_k, \gamma_k = 1, \ \forall\, k \in \{1, \ldots, K\}$
for each batch $\{\mathbf{y}^{(i)}, \mathbf{D}^{(i)}, \mathbf{x}^{(i)}\}_{i=1}^{N_B}$ do
  update $\Theta$ by $\partial_\Theta \sum_{i \in N_B} \left\| \mathcal{F}_K(\mathbf{y}^{(i)}, \mathbf{D}^{(i)}; \Theta) - \mathbf{x}^{(i)} \right\|_2^2$
```

3 Ada-LISTA: Theoretical Study

For the following theoretical study, we consider a reduced scheme of Ada-LISTA with a single weight matrix, so as to avoid complication in theorem conditions. Note that if $\mathbf{W}_1 = \mathbf{W}$ and $\mathbf{W}_2^T\mathbf{W}_2 = \mathbf{W}$, then Definition 1 collapses to this scheme. Similar, yet more cumbersome, claims can be derived for our original scheme.

Definition 2 (Ada-LISTA – Single Matrix). *Ada-LISTA with a single weight matrix is defined by*

$$\mathbf{x}_{k+1} = \mathcal{S}_{\theta_{k+1}}(\mathbf{x}_k + \mathbf{D}^T\mathbf{W}^T(\mathbf{y} - \mathbf{D}\mathbf{x}_k)). \tag{11}$$

Recall the definition of mutual coherence between two matrices:

Definition 3 (Mutual Coherence). *Given two matrices,* $\mathbf{A}$ *and* $\mathbf{B}$*, if the diagonal elements of* $\mathbf{A}^T\mathbf{B}$ *are equal to* 1[1]*, then the mutual coherence is defined as*

$$\mu(\mathbf{A}, \mathbf{B}) = \max_{i \neq j} |\mathbf{a}_i^T\mathbf{b}_j|, \tag{12}$$

where $\mathbf{a}_i$ *and* $\mathbf{b}_j$ *are the ith and jth columns of* $\mathbf{A}$ *and* $\mathbf{B}$.

We first prove that Ada-LISTA is capable of solving sparse coding in linear rate. We show that if all signals emerge from the same dictionary $\mathbf{D}$, there exists a weight matrix $\mathbf{W}$ and threshold values such that the recovery error decreases linearly over iterations. Note that the theorems below guarantee convergence to the representation $\mathbf{x}_*$, generating the signal $\mathbf{y} = \mathbf{D}\mathbf{x}_*$; however, under the theorem assumptions this vector is equal to the ISTA solution. The following theorem indicates that if Ada-LISTA's training reaches its global minimum, the rate would be at least linear. The proof for Theorem 1, appearing in Section 6.1, follows the steps in [19], generalizing [20] for noisy signals.

Theorem 1 (Ada-LISTA Convergence Guarantee). *Consider a noisy input* $\mathbf{y} = \mathbf{D}\mathbf{x}_* + \mathbf{e}$*. If* $\mathbf{x}_*$ *is sufficiently sparse,*

[1] If the diagonal elements of $\mathbf{A}^T\mathbf{B}$ are not equal to 1, then define $\tilde{\mathbf{A}}$, such that $\tilde{\mathbf{a}}_i = \frac{1}{|\mathbf{a}_i^T\mathbf{b}_i|}\mathbf{a}_i$, and the mutual coherence is $\mu(\tilde{\mathbf{A}}, \mathbf{B})$.

$$s = \|\mathbf{x}_*\|_0 < \frac{1}{2\widetilde{\mu}}, \text{ with } \widetilde{\mu} \triangleq \mu(\mathbf{WD}, \mathbf{D}), \tag{13}$$

and the thresholds satisfy

$$\theta_k = \theta_{\max}\gamma^{-k} > \theta_{\min} = \frac{\|\mathbf{A}^T\mathbf{e}\|_\infty}{1 - 2\gamma\widetilde{\mu}s} \tag{14}$$

with $1 < \gamma < (2\widetilde{\mu}s)^{-1}$, $\mathbf{A} \triangleq \mathbf{WD}$, *and* $\theta_{\max} \geq \|\mathbf{A}^T\mathbf{y}\|_\infty$, *then the support in the kth iteration of Ada-LISTA is included in that of* $\mathbf{x}_*$, *and its values satisfy*

$$\|\mathbf{x}_k - \mathbf{x}_*\|_\infty \leq 2\max\{\theta_{\max}\gamma^{-k}, \theta_{\min}\}. \tag{15}$$

Theorem 1 guarantees linear convergence to $\mathbf{x}_*$ in the noiseless case, as studied in [20], [21], since $\theta_{min} = 0$. In the noisy case, convergence is achieved only up to a certain sphere surrounding the ideal solution, a result in-line with other theoretical guarantees in the related literature [22]. Note that Theorem 1 uses the mutual coherence, a common and easily verified tool, but quite restrictive as it considers worst-case analysis [19], [20], [21], [22]. As a consequence, there is indeed an unavoidable gap between the theoretical guarantees obtained and actual performance (see Section 5).

We proceed by showing that Ada-LISTA can be adaptive to model variations. In this setting, we argue that the signal can originate from different models, and nonetheless there exist global parameters such that Ada-LISTA will converge in linear rate. Our Theorem exposes the key idea that, as opposed to LISTA which corresponds to a single dictionary, Ada-LISTA can be flexible to various models, while still providing a good generalization. Section 6.2 contains the proof of the following Theorem.

Theorem 2 (Ada-LISTA – Applicable Dictionaries). *Consider a trained Ada-LISTA network with fixed* $\mathbf{W}$, *and noisy input* $\mathbf{y} = \mathbf{D}\mathbf{x}_* + \mathbf{e}$. *If the following conditions hold:*

1) *The diagonal elements of* $\mathbf{G} \triangleq \mathbf{D}^T\mathbf{W}^T\mathbf{D}$ *are close to 1:*

$$\max_i |G_{ii} - 1| \leq \epsilon_d; \tag{16}$$

2) *The off-diagonals are bounded:*

$$\max_{i \neq j} |G_{ij}| \leq \bar{\mu}; \tag{17}$$

3) $\mathbf{x}_*$ *is sufficiently sparse:*

$$s = \|\mathbf{x}_*\|_0 < \frac{1}{2\mu} - \frac{\epsilon_d}{\mu}; \tag{18}$$

4) *The thresholds satisfy:*

$$\theta_k = \theta_{\max}\gamma^{-k} > \theta_{\min} = \frac{\|\mathbf{A}^T\mathbf{e}\|_\infty}{1 - 2\gamma\epsilon_d - 2\gamma\mu s}, \tag{19}$$

with $1 < \gamma < 0.5(\mu s + \epsilon_d)^{-1}$, $\mathbf{A} \triangleq \mathbf{WD}$, *and* $\theta_{\max} \geq \|\mathbf{A}^T\mathbf{y}\|_\infty$,

the support of the kth iteration of Ada-LISTA is included in that of $\mathbf{x}_*$, *with values satisfying*

$$\|\mathbf{x}_k - \mathbf{x}_*\|_\infty \leq 2\max\{\theta_{\max}\gamma^{-k}, \theta_{\min}\}. \tag{20}$$

An interesting question arising is: Once Ada-LISTA has been trained and $\mathbf{W}$ is fixed, which dictionaries can be effectively served with the same parameters, without additional training? Theorem 2 reveals that as long as the effective matrix $\mathbf{G} = \mathbf{D}^T\mathbf{W}^T\mathbf{D}$ is sufficiently close to identity, linear convergence is guaranteed. This holds for two interesting scenarios, proven in Sections 6.3 and 6.4:

1) *Random permutations* – If Ada-LISTA converges for signals emerging from $\mathbf{D}$, it can also converge for signals originating from any permutation of $\mathbf{D}$'s atoms.
2) *Noisy dictionaries* – If Ada-LISTA converges given a clean dictionary $\mathbf{D}$, satisfying

$$\mu(\mathbf{WD}, \mathbf{D}) < \mu, \tag{21}$$

it also converges for noisy models $\tilde{\mathbf{D}} = \mathbf{D} + \mathbf{E}$, with some probability, depending on the distribution of $\mathbf{E}$.

To the best of our knowledge, Theorem 2 provides the first linear rate convergence guarantee, in the presence of *model variations*, depending on small enough cardinality and low mutual coherence $\widetilde{\mu}$.

Admittedly, our hypothesis requires a low mutual coherence $\tilde{\mu}$ of the effective matrix $\mathbf{G} = \mathbf{D}^T\mathbf{W}^T\mathbf{D}$. This, however, is a weaker requirement than common worst-case analysis in the field of sparse representation, in which the mutual coherence of $\mathbf{D}$ is assumed to be low. Furthermore, the weight matrix $\mathbf{W}$ can compensate for a higher mutual coherence of $\mathbf{D}$, and the condition will still hold. Although Theorem 2 does not address completely random dictionaries, characterized with higher mutual coherence, our experiments in Section 5.1.4 show that our method succeeds in such a scenario as well.

4 Related Work

LISTA's concept of unfolding the iterations of a classical optimization scheme into an RNN-like neural network and freeing its parameters to be learned appears in many works. These include an unsupervised, online training procedure [9], a multi-layer version [23], a gated mechanism, compensating shrinkage artifacts [21], as well as reduced-parameter schemes [20], [24]. This paradigm has been brought to various applications, including compressed sensing, super-resolution, communication and MRI reconstruction [25], [26], [27], [28], [29], [30].

A prominent line of work investigates the success of learned solvers from a theoretical point of view [19], [31], [32], [33]. [19], [20], [21], [24] have recently shown learned solvers can achieve linear convergence, under specific conditions on sparsity and mutual coherence, inspiring our derivation of Theorem 1. This work generalizes these guarantees to a varying model scenario, proving the same weight matrix can serve different models while still reaching linear convergence.

4.1 Relation to Robust-ALISTA [20]

While the literature on LISTA is abundant, the most relevant work to ours is "robust-ALISTA" [20], introducing adaptivity to dictionary perturbations. Robust-ALISTA models the clean dictionary $\mathbf{D}$ as having small perturbations of the form $\tilde{\mathbf{D}} = \mathbf{D} + \mathbf{E}$, where $\mathbf{E}$ is an interference matrix. The robustness is achieved in two stages, the first, computing a weight matrix $\tilde{\mathbf{W}}$ for each noisy model $\tilde{\mathbf{D}}$ by minimizing the mutual coherence $\mu(\tilde{\mathbf{W}}, \tilde{\mathbf{D}})$:

$$\tilde{\mathbf{W}} = \arg\min_{\mathbf{W}} \left\| \mathbf{W}^T \tilde{\mathbf{D}} \right\|_F^2, \text{ s.t. } \mathbf{w}_i^T \tilde{\mathbf{d}}_i = 1, \ \forall i \in [1, m], \tag{22}$$

where $\mathbf{w}_i, \tilde{\mathbf{d}}_i$ are the ith columns of $\mathbf{W}$ and $\tilde{\mathbf{D}}$ respectively. Instead of solving Eq. (22) analytically, the authors in [20] suggest computing $\tilde{\mathbf{W}}$ using an additional unfolded encoder network. This improves inference time, but complicates training and adds an additional trained matrix to the set of learned parameters. The encoder, computing $\tilde{\mathbf{W}}$ given $\tilde{\mathbf{D}}$, is minimized via the following softened version of:

$$\tilde{\mathbf{W}} = \arg\min_{\mathbf{W}} \left\| \mathbf{Q} \odot (\tilde{\mathbf{D}}^T \mathbf{W} - \mathbf{I}) \right\|_F^2, \tag{23}$$

where $\odot$ is the Hadamard product, $\mathbf{I}$ is the identity matrix, $\mathbf{Q}$ is a constant matrix, penalizing errors on the diagonal, $\tilde{\mathbf{D}}$ is given and $\tilde{\mathbf{W}}$ is learned.The next stage in Robust-ALISTA's pipeline is embedding the matrices $\tilde{\mathbf{W}}, \mathbf{D}$ into the ALISTA architecture:

$$\mathbf{x}_{k+1} = \mathcal{S}_{\theta_{k+1}} \Big(\mathbf{x}_k - \gamma_{k+1} \tilde{\mathbf{W}}^T (\mathbf{D}\mathbf{x}_k - \mathbf{y}) \Big), \tag{24}$$

where the step sizes and thresholds $\{\gamma_k, \theta_k\}$ are learned parameters. ALISTA's initial scheme of computing $\tilde{\mathbf{W}}$ analytically leads to a remarkably reduced set of parameters, $2K$. Its learned version in Eq. (23), however, does contain an additional trained matrix $\tilde{\mathbf{W}}$. To increase stability, the authors in [20] use the clean dictionary $\mathbf{D}$, instead of $\tilde{\mathbf{D}}$ in Eq. (24), and perform curriculum learning of gradually increasing the noise levels, until reaching the target noise level. Note that the clean version of the dictionary does not exist in many practical real-world settings.

Given the brief summary of Robust-ALISTA, we provide a numeric comparison in Section 4.1, and hereby highlight the conceptual differences to our approach:

1) Robust-ALISTA computes the weight matrix $\tilde{\mathbf{W}}$ such that the mutual coherence with $\tilde{\mathbf{D}}$ is minimized. Although mutual coherence is a theoretical tool in performing worst-case analysis in the field of sparse representations, it is not a precondition of the success of a dictionary in practice, but a mere proxy of it. We thus do not enforce a small mutual coherence between the weights and input dictionary, but instead suggest a flexible scheme in which both weight matrices are free to be learned over the input data.
2) Both alternatives of stage 1 in robust-ALISTA (analytic and encoder-based) lead to longer inference times, as compared to Ada-LISTA. Computing $\tilde{\mathbf{W}}$ analytically adds a quadratic optimization problem to the inference of each input signal, whereas using a trained encoder increases both runtime and memory usage. Conversely, our Ada-LISTA is trained a-priori for dealing with varied dictionaries, requiring a forward pass over a single unfolded network.
3) Training-wise, Ada-LISTA is both faster and simpler. Robust-ALISTA features an additional analytic computation/encoder network, which cannot be trained straightforward on a target noise level, but requires a gradual curriculum learning training regime. Our training is much simpler and elegant, requiring a single end-to-end unfolded network.
4) While [20] deals with Gaussian noise perturbations, we offer a wider set of perturbation scenarios, including random column permutations, Gaussian noise, and most importantly, completely random dictionaries. In fact, the current curriculum learning of robust-ALISTA will most likely fail in dealing with permuted columns or random dictionaries, since there is no notion of a "clean" dictionary.
5) Finally, Robust-ALISTA's training targets the original sparse representations that generated the signals. This makes [20] both impractical to real-world scenarios and restricted to sparse coding applications. Ada-LISTA, on the other hand, operates with accessible ISTA/FISTA solutions of Eq. (1), and thus can be used for any generic problem, solvable with ISTA (Eq. (4)), e.g., low-rank matrix models [9], acceleration of Eulerian fluid simulation [34] and feature learning [35].

5 Numerical Results

To show the effectiveness of our approach, we perform extensive numerical experiments, where our goal is twofold. First, we examine Ada-LISTA on a variety of synthetic data scenarios, including column permutations of the input dictionary, additive noisy versions of it, and completely random dictionaries. Second, we provide a comparison between our method and Robust-ALISTA [20]. Finally, we showcase our robustness in a real-world task of natural image inpainting.[2]

5.1 Synthetic Experiments

5.1.1 Experiment Setting

We construct a dictionary $\mathbf{D} \in \mathbb{R}^{50 \times 70}$ with random entries drawn from a normal distribution, and normalize its columns to have a unit L^2-norm. Our signals $\mathbf{y} \in \mathbb{R}^{50}$ are created as sparse combinations of atoms over this dictionary, $\mathbf{y} = \mathbf{D}\mathbf{x}_*$. While the reported experiments in the following subsections assume no additive noise, Section 5.1.5 presents a series of similar tests with varying levels of noise, showing the same qualitative results. The representation vectors $\mathbf{x}_* \in \mathbb{R}^{70}$ are created by randomly choosing a support of cardinality $s = 4$ with coefficients selected from a normal distribution. Instead of using the true sparse representations $\mathbf{x}_*$ as ground truth for training, we compute the Lasso solution $\mathbf{x}$ with FISTA (100 iterations, $\lambda = 1$), using the obtained signals $\mathbf{y}$ and their corresponding dictionary $\mathbf{D}$. This is done in order to maintain a real-world setting, where one does not have access to the true sparse representations. We create in this manner $N = 20,000$ examples for training, and $N_{\text{test}} = 1,000$ for test. Our metric for comparison between different

[2] The code for reproducing all experiments is available at https://github.com/aaberdam/AdaLISTA.

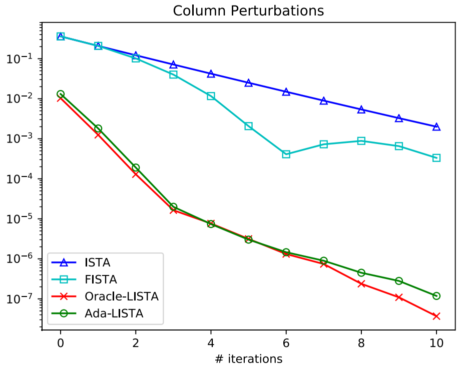


Fig. 2. MSE performance under column permutations.

methods is the MSE (Mean Square Error) between the ground truth $\mathbf{x}$ and the predicted sparse representations at K unfoldings, $\|\mathbf{x}-\mathbf{x}_K\|^2$. In all experiments, the Ada-LISTA weight matrices are both initialized as the identity matrix. In the following set of experiments we gradually diverge from the initial model, given by the dictionary $\mathbf{D}$, by applying different modifications to it.

5.1.2 Random Permutations

We start with a scenario in which the columns of the initial dictionary $\mathbf{D}$ are permuted randomly to create a new dictionary $\tilde{\mathbf{D}}^{(i)}$. This transformation can occur in the non-convex process of dictionary learning, in which different initializations might incur a different order of the resulting atoms. Although the signals' subspace remains intact, learned solvers as LISTA where the dictionary is hard-coded during training, will most likely fail, as they cannot predict the updated support.

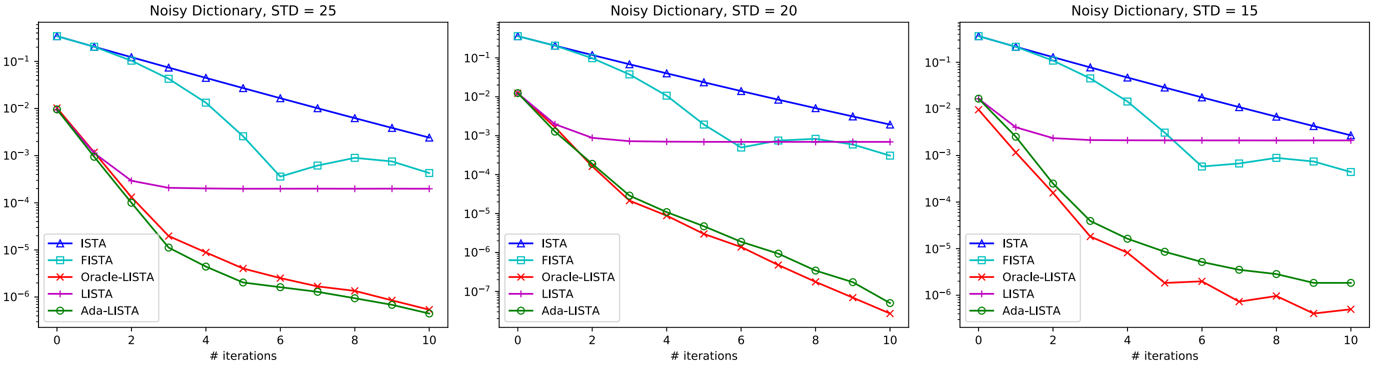
Here and below, we compare the results of four solvers: ISTA, FISTA, Oracle-LISTA and Ada-LISTA, all versus the number of iterations/unfoldings, K. For each training example in ISTA, FISTA and Ada-LISTA, we create new instances of a permuted dictionary $\mathbf{D}^{(i)}$ and its corresponding true representation, $\mathbf{x}_*^{(i)}$. We then apply FISTA for 100 iterations and obtain the ground truth representations $\mathbf{x}^{(i)}$ for the signal $\mathbf{y}^{(i)}=\tilde{\mathbf{D}}^{(i)}\mathbf{x}_*^{(i)}$. Then ISTA and FISTA are applied for only K iterations to solve for the pairs $\{\mathbf{y}^{(i)},\tilde{\mathbf{D}}^{(i)}\}$. Similarly, the supervised Ada-LISTA is given the ground truth $\{\mathbf{y}^{(i)},\tilde{\mathbf{D}}^{(i)},\mathbf{x}^{(i)}\}$ for training. In Oracle-LISTA *we solve a simpler problem* in which the dictionary is fixed ($\mathbf{D}$) for all training examples $\{\mathbf{y}^{(i)},\mathbf{x}^{(i)}\}$. The results in Fig. 2 clearly show that Ada-LISTA is much more efficient compared to ISTA/FISTA, capable of mimicking the performance of the Oracle-LISTA, which considers a single constant $\mathbf{D}$.

5.1.3 Noisy Dictionaries

We aim to show Ada-LISTA can handle a more challenging case in which the dictionary varies by $\tilde{\mathbf{D}}^{(i)}=\mathbf{D}+\mathbf{E}^{(i)}$. Each training signal $\mathbf{y}^{(i)}$ is created by drawing a different noisy instance of the dictionary $\tilde{\mathbf{D}}^{(i)}$ and a sparse representation $\mathbf{x}_*^{(i)}$, and solving FISTA to obtain $\mathbf{x}^{(i)}$. ISTA and FISTA receive the pairs $\{\mathbf{y}^{(i)},\tilde{\mathbf{D}}^{(i)}\}$, and Ada-LISTA receives the triplet $\{\mathbf{y}^{(i)},\tilde{\mathbf{D}}^{(i)},\mathbf{x}^{(i)}\}$. By vanilla LISTA, we refer to a learned solver that obtains $\{\mathbf{y}^{(i)},\mathbf{x}^{(i)}\}$, and trains a network while disregarding the changing models. Oracle-LISTA, as before, handles a simpler case in which the dictionary is fixed, being $\mathbf{D}$, and all signals are created from it. Fig. 3 presents the performance of the different solvers with a decreasing SNR (Signal to Noise Ratio) of the dictionary. The performance of ISTA and FISTA is agnostic to the noisy model, since they do not require prior training. Ada-LISTA again performs on-par with Oracle-LISTA, which has prior knowledge of the dictionary. LISTA's performance, however, deteriorates with the decrease of the dictionary SNR.

5.1.4 Random Dictionaries

In this setting, we diverge even further from a fixed model, and examine the capability of our method to handle completely random input dictionaries. This time, for each training example we create a different Gaussian normalized dictionary $\mathbf{D}^{(i)}$, and a corresponding representation vector with an increasing cardinality: $s=4,8,12$. The resulting signals, $\mathbf{y}^{(i)}=\mathbf{D}^{(i)}\mathbf{x}_*^{(i)}$, and their corresponding dictionaries are fed to FISTA to obtain the ground truth sparse representations for training, $\mathbf{x}^{(i)}$. We compare the performance of ISTA, FISTA, Ada-LISTA and Oracle-LISTA. Similarly to previous experiments, Ada-LISTA is fed during training with the triplet $\{\mathbf{y}^{(i)},\mathbf{D}^{(i)},\mathbf{x}^{(i)}\}_{i=1}^{N}$. Vanilla LISTA cannot handle such variation in the input distribution, and thus it is omitted. We again show the results of Oracle-LISTA where all training signals are created from the same dictionary.

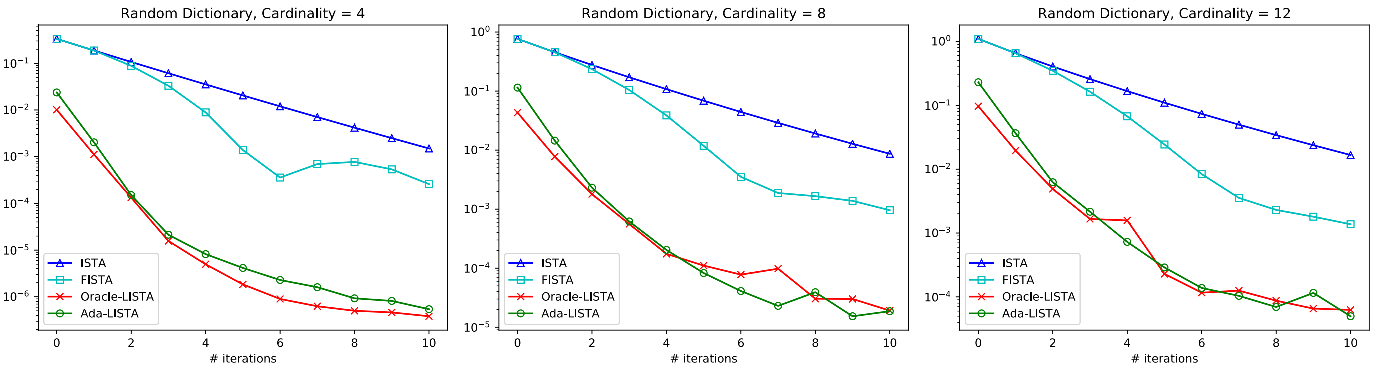


(a) SNR of 25 dB.

(b) SNR of 20 dB.

(c) SNR of 15 dB.

Fig. 3. MSE performance for noisy dictionaries with decreasing SNR values.



(a) $s = 4$.

(b) $s = 8$.

(c) $s = 12$.

Fig. 4. MSE performance for random dictionaries with increasing cardinality.

As can be seen in Fig. 4, for a small cardinality of $s = 4$, Oracle-LISTA is able to drastically lower the reconstruction error as compared to ISTA and FISTA. This result, however, has already been demonstrated in [10]. Ada-LISTA which deals with a much more complex scenario, still provides a similar improvement over both ISTA and FISTA. As the cardinality increases to $s = 8, 12$, the performance of both learned solvers deteriorates, and the improvement over their non-learned counterparts diminishes. The last experiment provides a valuable insight on the success of LISTA-like learned solvers.

Common belief suggests acceleration in convergence can be obtained when the signals are restricted to a union of low-dimensional subspaces, as opposed to the entire signal space. The above experiment suggests otherwise: Although the signals occupy the whole space, Ada-LISTA still achieves improved convergence. This implies that the underlying structure should be only of the *signal, given its generative model* $p(\mathbf{y}|\mathbf{D})$, as opposed to the *signal* model, $p(\mathbf{y})$. In the above, even if the dictionaries are random, the signals must be *sparse combinations of atoms*. As this assumption of structure weakens with the increased cardinality, the resulting acceleration becomes less prominent. We believe this conditional information is the key for improved convergence.

5.1.5 Noisy Signals

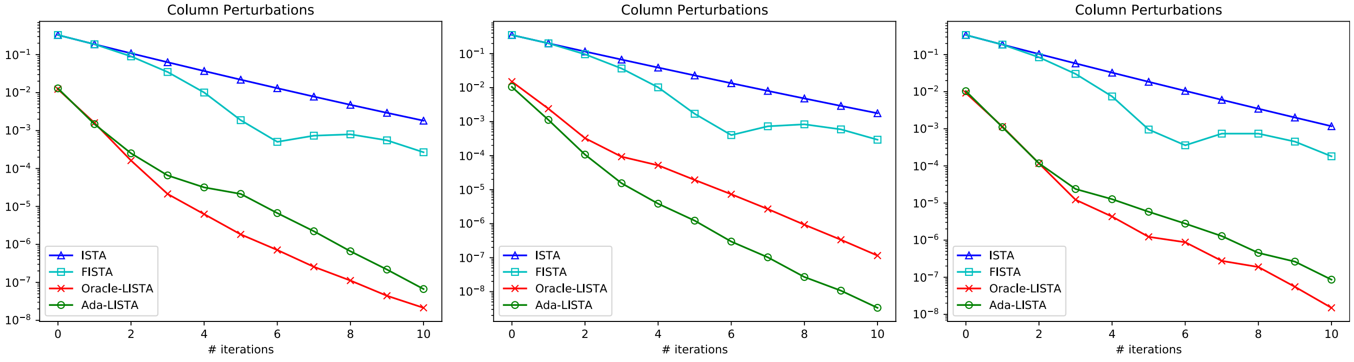
In this part we examine Ada-LISTA's performance for noisy signals by repeating the synthetic experiments above, with three levels of input SNR: 10, 20, and 30dB. Figs. 5, 6, and 7 respectively, present the results for column permutations, noisy dictionaries and random dictionaries. The same observations as in the noiseless case hold for noisy signals just as well. Learned solvers can achieve an acceleration even in the presence of noise in the input, and Ada-LISTA manages to mimic the oracle-LISTA, while coping with a much harder scenario of varying dictionaries.

5.2 Comparison to Robust-ALISTA

As discussed above in Section 4.1, we see the mutual coherence as a mere proxy for the dictionary, usually utilized for worst-case analysis, but not as a precondition of the success of a dictionary in practice. Therefore, our method does not set the weight matrices as the minimizers of the mutual coherence, but rather, frees the weights to be learned over the input data.

To validate our notion, we compare our algorithm to the vanilla robust-ALISTA[20] which consists of two stages. It first computes the 'ideal' weight matrix $\tilde{\mathbf{W}}$ for each dictionary $\tilde{\mathbf{D}}$ such that it minimizes the mutual coherence (Eq. (16) in [20]). Then, in stage 2, it uses triplets of $\mathbf{y}, \mathbf{x}, \tilde{\mathbf{W}}$ to train the network step-sizes and shrinkage parameters.

Fig. 8 presents the experiments of noisy dictionaries with SNR of $20dB$ as described above. As can be observed, in this scenario our method largely outperforms the robust-ALISTA approach which minimizes the mutual coherence. The relatively poor performance of ALISTA in the described scenario aligns with the results of [36].

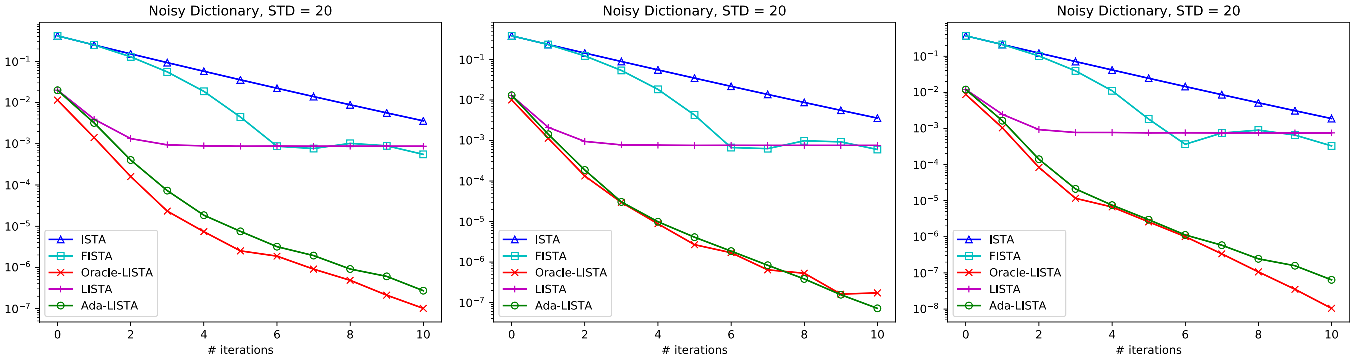


(a) Signal SNR: 10 dB.

(b) Signal SNR: 20 dB.

(c) Signal SNR: 30 dB.

Fig. 5. MSE performance under column permutations and noisy inputs.

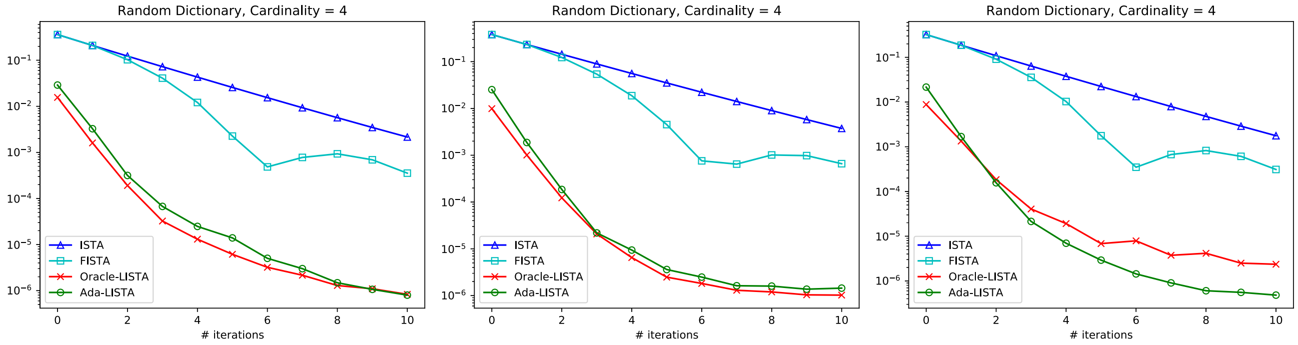


(a) Signal SNR: 10 dB.

(b) Signal SNR: 20 dB.

(c) Signal SNR: 30 dB.

Fig. 6. MSE performance for noisy dictionaries and noisy inputs.



(a) Signal SNR: 10 dB.

(b) Signal SNR: 20 dB.

(c) Signal SNR: 30 dB.

Fig. 7. MSE performance under random dictionaries and noisy inputs.

5.3 Natural Image Inpainting

We apply our method to a natural image inpainting task. We assume the image is corrupted by a known mask with a ratio of p missing pixels. The updated objective we minimize is

$$\frac{1}{2}\|\mathbf{y}-\mathbf{M}\mathbf{D}\mathbf{x}\|_2^2 + \lambda\|\mathbf{x}\|_1, \tag{25}$$

where $\mathbf{y} \in \mathbb{R}^n$ is a corrupt patch of the same size as the clean one, $\mathbf{D} \in \mathbb{R}^{n\times m}$ is a dictionary trained on clean image patches, and $\mathbf{M} \in \mathbb{R}^{n\times n}$ represents the mask, being an identity matrix with a percentage of p diagonal elements equal to zero. The effective dictionary $\mathbf{D}_{\text{eff}} = \mathbf{M}\mathbf{D}$ changes for every signal since $\mathbf{M}$ is unique for each patch.

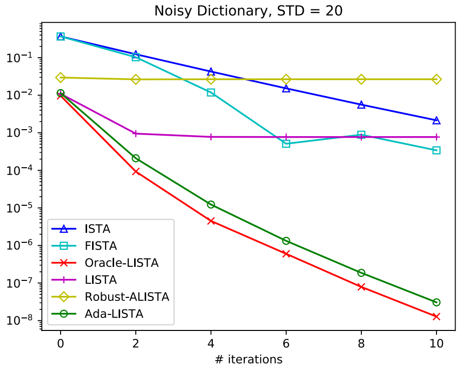


Fig. 8. MSE performance for the experiment of noisy dictionaries with SNR of $20dB$. Robust-ALISTA represents the clean version of the algorithm proposed in [21]. As can be seen, our proposed Ada-LISTA method significantly outperforms this algorithm while almost reaching the Oracle-LITSA performances.

5.3.1 *Updated Model*

We slightly change the formulation of the model described in Section 2, and reverse the roles of the input and learned matrices. The updated shrinkage step (Eq. (2)) is now

$$\mathbf{x}_{k+1} = \mathcal{S}_{\frac{\lambda}{L}}\left(\mathbf{x}_k + \frac{1}{L}\mathbf{D}^T\mathbf{M}^T(\mathbf{y}-\mathbf{M}\mathbf{D}\mathbf{x}_k)\right). \tag{26}$$

We consider the mask as input, while the dictionary $\mathbf{D}$ is learned with the following parameterization:

$$\frac{1}{L}\mathbf{D}^T\mathbf{M}^T\mathbf{M}\mathbf{D} \rightarrow \gamma_{k+1}\mathbf{W}_1^T\mathbf{M}^T\mathbf{M}\mathbf{W}_1^T, \tag{27}$$

$$\frac{1}{L}\mathbf{D}^T\mathbf{M}^T \rightarrow \gamma_{k+1}\mathbf{W}_2^T\mathbf{M}^T, \tag{28}$$

where $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{n\times m}$ are the same size as the dictionary $\mathbf{D}$, and initialized by it.

5.3.2 *Experiment Setting*

To collect natural image patches, we use the BSDS500 data-set [37] and divide it to $400, 50$ and 50 training, validation and test images correspondingly. To train the dictionary $\mathbf{D}$,

TABLE 1
PSNR Results for Image Inpainting With 50% Missing Pixels and $K = 20$ Unfoldings

	Barbara	Boat	House	Lena	Peppers	C.man	Couple	Finger	Hill	Man	Montage
ISTA	23.49	25.40	26.87	27.83	23.56	22.72	25.34	20.63	27.26	26.34	22.48
FISTA	24.93	28.18	30.53	31.02	26.75	25.25	28.09	25.45	29.64	29.03	25.08
Ada-LFISTA	**26.09**	**30.03**	**32.36**	**32.50**	**28.81**	**27.94**	**30.02**	**28.25**	**30.86**	**30.67**	**27.22**

Higher is better.

we extract 100,000 8×8 patches at random locations from the train images, subtract their mean and divide by the average standard deviation. The dictionary of size $\mathbf{D} \in \mathbb{R}^{64\times 256}$ is learned via `scikit-learn`'s function `MiniBatchDictionaryLearning` with $\lambda = 0.1$. To train our network, we randomly pick a subset of $N = 50,000$ training and $N_{\text{val}} = 1,000$ validation patches. We train the network to perform an image inpainting task with ratio of $p = 0.5$. Instead of using Ada-LISTA as before, we tweak the architecture described in Eq. (27) to unfold the FISTA algorithm, termed Ada-LFISTA, as described in Algorithm 1. The input to our network is triplets $\{\mathbf{y}^{(i)}, \mathbf{M}^{(i)}, \mathbf{x}^{(i)}\}_{i=1}^{N}$ of the corrupt train patches $\mathbf{y}^{(i)}$, their corresponding mask $\mathbf{M}^{(i)}$, and the solutions $\mathbf{x}^{(i)}$ of the FISTA solver applied for 300 iterations on the corrupt signals. The output is the reconstructed representations $\mathbf{x}_K^{(i)}$. We evaluate the performance of our method on images from the popular `Set11`, corrupted with the same inpainting ratio of $p = 0.5$, and compare between ISTA, FISTA and Ada-LFISTA for a fixed number of $K = 20$ iterations/unfoldings. The performance is measured in PSNR between the clean images and the reconstruction of their corrupt version. The patch-wise validation error versus the the number of unfoldings is given in Fig. 10; numerical results are given in Table 1, and select qualitative results are shown in Fig. 9. There is a clear advantage to Ada-LFISTA over the non-learned ISTA and FISTA solvers. In this setting of 50% missing pixels, a hard-coded solver with a fixed $\mathbf{D}$, such as LISTA, cannot deal with the changing mask of each patch. To validate the above claim, we perform the same experiment with only 5% missing pixels ($p = 0.05, \lambda = 1, T = 17$). In this setting, LISTA achieves a PSNR of only 23.9dB, while ISTA, FISTA and Ada-LISTA correspondingly, reach $23, 27.4, \mathbf{30.7}$dB as presented in Fig. 11.

6 Proofs of Main Theorems

6.1 Proof of Theorem 1

Proof. This proof follows the steps from [19], with slight modifications to fit our scheme. Following the notations in Theorem 1, $\mathbf{x}_*$ denotes the true sparse representation of the signal $\mathbf{y}$, and $\mathbf{A} = \mathbf{WD}$. In addition, we define $\text{Supp}(\cdot)$ as the support of a vector.

Induction Hypothesis. For any iteration $k \geq 0$ the following hold

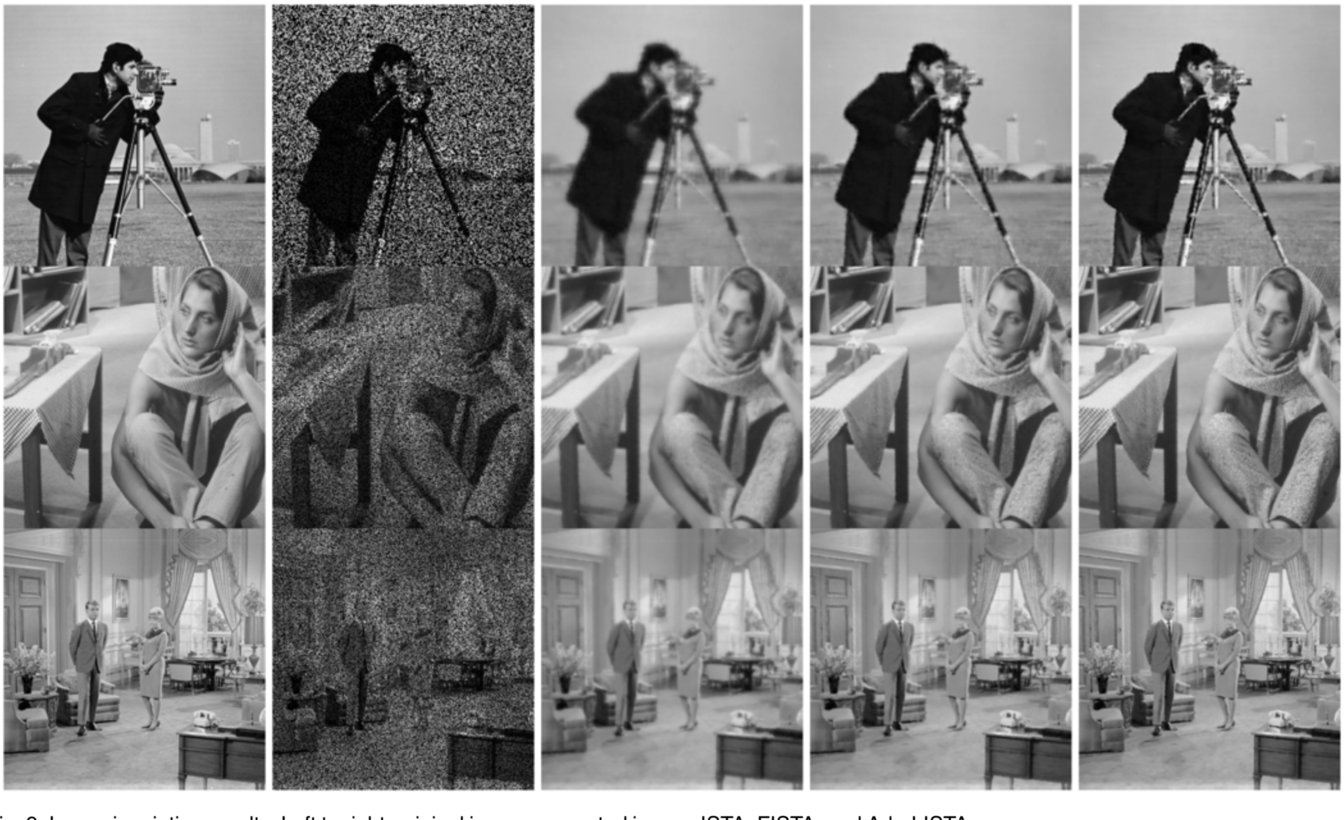


Fig. 9. Image inpainting results. Left to right: original image, corrupted image, ISTA, FISTA, and Ada-LISTA.

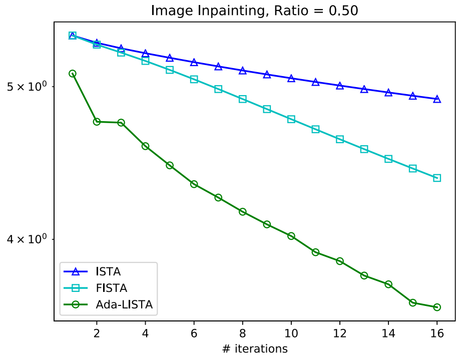


Fig. 10. Patch-wise validation error versus unfoldings.

1) The estimated *support* is contained in the true support,

$$\mathrm{Supp}(\mathbf{x}_k) \subseteq \mathrm{Supp}(\mathbf{x}_*). \tag{29}$$

2) The *recovery error* is bounded by

$$\|\mathbf{x}_k - \mathbf{x}_*\|_\infty \leq 2\theta_k. \tag{30}$$

Base Case. We start by showing that the induction hypothesis holds for $k = 0$. Since $\mathbf{x}_0 = \mathbf{0}$ we get that the support is empty and the support hypothesis Eq. (29) holds. As for the recovery error, we get that

$$\|\mathbf{x}_0 - \mathbf{x}_*\|_\infty = \|\mathbf{x}_*\|_\infty. \tag{31}$$

Therefore, to verify Eq. (30) we need to show that

$$\|\mathbf{x}_*\|_\infty \leq 2\theta_0 = 2\theta_{\max}. \tag{32}$$

Since $\mathbf{y} = \mathbf{D}\mathbf{x}_* + \mathbf{e}$, for any index i we can write

$$\mathbf{y} = \mathbf{d}_i \mathbf{x}_*[i] + \sum_{j \neq i} \mathbf{d}_j \mathbf{x}_*[j] + \mathbf{e}, \tag{33}$$

where $\mathbf{d}_i$ denotes the ith column in $\mathbf{D}$ and $\mathbf{x}[i]$ denotes the ith element in $\mathbf{x}$. Multiplying each side by $\mathbf{a}_i^T$ we get

$$\mathbf{x}_*[i]\mathbf{a}_i^T\mathbf{d}_i = \mathbf{a}_i^T\mathbf{y} - \sum_{j \neq i} \mathbf{x}_*[j]\mathbf{a}_i^T\mathbf{d}_j - \mathbf{a}_i^T\mathbf{e}. \tag{34}$$

Since by assumption $\mathbf{a}_i^T\mathbf{d}_i = 1$, the left term becomes $\mathbf{x}_*[i]$. In addition, since, by assumption, there are no more than s nonzeros in $\mathbf{x}_*$ and $\left|\mathbf{a}_i^T\mathbf{d}_j\right|$ is bounded by $\tilde{\mu}$, we get the following bound

$$|\mathbf{x}_*[i]| \leq \left|\mathbf{a}_i^T\mathbf{y}\right| + s\tilde{\mu}\|\mathbf{x}_*\|_\infty + \left|\mathbf{a}_i^T\mathbf{e}\right|. \tag{35}$$

By taking a maximum over i we obtain

$$(1 - s\tilde{\mu})\|\mathbf{x}_*\|_\infty \leq \left\|\mathbf{A}^T\mathbf{y}\right\|_\infty + \left\|\mathbf{A}^T\mathbf{e}\right\|_\infty. \tag{36}$$

Since we have assumed that $\left\|\mathbf{A}^T\mathbf{y}\right\|_\infty \leq \theta_{\max}$, and

$$\left\|\mathbf{A}^T\mathbf{e}\right\|_\infty = \theta_{\min}(1 - 2\gamma\tilde{\mu}s) < \theta_{\max}(1 - 2\gamma\tilde{\mu}s), \tag{37}$$

we get

$$(1 - s\tilde{\mu})\|\mathbf{x}_*\|_\infty \leq 2\theta_{\max}(1 - \gamma\tilde{\mu}s). \tag{38}$$

Finally, since $s\tilde{\mu} \leq \frac{1}{2}$, and $\gamma > 1$, we get

$$\|\mathbf{x}_0 - \mathbf{x}_*\|_\infty = \|\mathbf{x}_*\|_\infty \leq 2\theta_{\max}, \tag{39}$$

as in Eq. (30), and therefore the recovery error hypothesis holds for the base case.

Inductive Step. Assuming the induction hypothesis holds for iteration k, we show that it also holds for the next iteration $k + 1$. We define $\mathcal{I} \triangleq \mathrm{Supp}(\mathbf{x}_*) - \{i\}$ and denote by $\mathbf{D}_\mathcal{I}$ the subset $\mathcal{I}$ of columns in $\mathbf{D}$.

We start by proving the support hypothesis (Eq. (29)). By Definition 2, the following holds for any index i:

$$\mathbf{x}_{k+1}[i] = \mathcal{S}_{\theta_{k+1}}(\mathbf{x}_k[i] + \mathbf{a}_i^T(\mathbf{y} - \mathbf{D}\mathbf{x}_k)). \tag{40}$$

Placing $\mathbf{y} = \mathbf{D}\mathbf{x}_* + \mathbf{e}$, we get

$$\mathbf{x}_{k+1}[i] = \mathcal{S}_{\theta_{k+1}}(\mathbf{x}_k[i] + \mathbf{a}_i^T\mathbf{D}(\mathbf{x}_* - \mathbf{x}_k) + \mathbf{a}_i^T\mathbf{e}). \tag{41}$$

Since $\mathbf{a}_i^T\mathbf{d}_i = 1$, the following holds:

$$\mathbf{x}_k[i] + \mathbf{a}_i^T\mathbf{D}(\mathbf{x}_* - \mathbf{x}_k) = \mathbf{x}_*[i] + \mathbf{a}_i^T\mathbf{D}_\mathcal{I}(\mathbf{x}_*[\mathcal{I}] - \mathbf{x}_k[\mathcal{I}]).$$

Therefore, Eq. (41) becomes

$$\mathbf{x}_{k+1}[i] = \mathcal{S}_{\theta_{k+1}}(\underbrace{\mathbf{x}_*[i] + \mathbf{a}_i^T\mathbf{D}_\mathcal{I}(\mathbf{x}_*[\mathcal{I}] - \mathbf{x}_k[\mathcal{I}]) + \mathbf{a}_i^T\mathbf{e}}_{\triangleq r}). \tag{42}$$

We aim to show that for any $i \notin \mathrm{Supp}(\mathbf{x}_*)$, $\mathbf{x}_{k+1}[i] = 0$, as the support hypothesis suggests. Since $\mathbf{x}_*[i] = 0$, we can bound the input argument of the soft threshold by

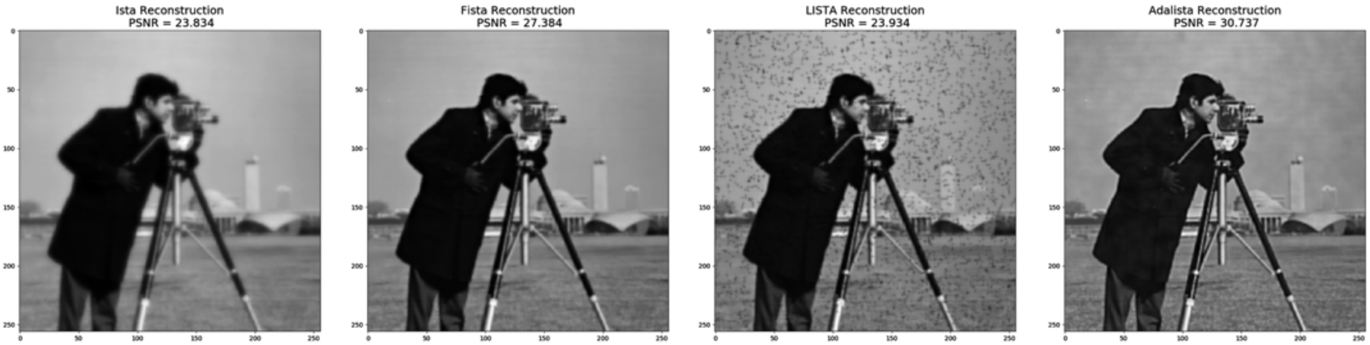


Fig. 11. Image inpainting results with only 5% missing pixels. As can be seen, LISTA cannot handle the varying mask of every patch while Ada-LISTA significantly outperforms the non-learned solvers ISTA and FISTA.

$$|r| \leq \left|\mathbf{a}_i^T \mathbf{D}_{\mathcal{I}}(\mathbf{x}_*[\mathcal{I}] - \mathbf{x}_k[\mathcal{I}])\right| + \left\|\mathbf{A}^T \mathbf{e}\right\|_\infty. \tag{43}$$

Using the induction assumption on the support, $\mathrm{Supp}(\mathbf{x}_k) \in \mathrm{Supp}(\mathbf{x}_*)$, we can upper bound the first term in the right-hand-side,

$$\left|\mathbf{a}_i^T \mathbf{D}_{\mathcal{I}}(\mathbf{x}_*[\mathcal{I}] - \mathbf{x}_k[\mathcal{I}])\right| \leq s\widetilde{\mu}\|\mathbf{x}_* - \mathbf{x}_k\|_\infty. \tag{44}$$

Using the induction assumption on the recovery error (Eq. (30)), we have $\|\mathbf{x}_* - \mathbf{x}_k\|_\infty \leq 2\theta_k$. Thus, we get

$$|r| \leq 2s\widetilde{\mu}\theta_k + \left\|\mathbf{A}^T \mathbf{e}\right\|_\infty. \tag{45}$$

However, by our assumptions,

$$\left\|\mathbf{A}^T \mathbf{e}\right\|_\infty = \theta_{\min}(1 - 2\gamma\widetilde{\mu}s) \ < \ \theta_{k+1}(1 - 2\gamma\widetilde{\mu}s). \tag{46}$$

Therefore,

$$|r| \leq 2s\widetilde{\mu}\theta_k + \theta_{k+1}(1 - 2\gamma\widetilde{\mu}s), \tag{47}$$

and by placing $\theta_k = \gamma\theta_{k+1}$ we get

$$|r| \leq \theta_{k+1}. \tag{48}$$

Since r is the input to the soft threshold operator $\mathcal{S}_{\theta_{k+1}}$, and it is no bigger than the threshold, we get that $\mathbf{x}_{k+1}[i] = 0$, and the support hypothesis holds.

We proceed by proving that the recovery error hypothesis also holds (Eq. (30)). We use the fact that for any scalar triplet, $(\mathbf{x}_1, \mathbf{x}_2, \theta)$, the soft threshold satisfies

$$|\mathcal{S}_\theta(\mathbf{x}_1 + \mathbf{x}_2) - \mathbf{x}_1| \leq \theta + |\mathbf{x}_2|. \tag{49}$$

Therefore, following Eq. (42) we get

$$|\mathbf{x}_{k+1}[i] - \mathbf{x}_*[i]| \leq \theta_{k+1} + \left|\mathbf{a}_i^T \mathbf{D}_{\mathcal{I}}(\mathbf{x}_*[\mathcal{I}] - \mathbf{x}_k[\mathcal{I}])\right| + \left\|\mathbf{A}^T \mathbf{e}\right\|_\infty.$$

As before, since $\mathrm{Supp}(\mathbf{x}_k) \in \mathrm{Supp}(\mathbf{x}_*)$, we have

$$\left|\mathbf{a}_i^T \mathbf{D}_{\mathcal{I}}(\mathbf{x}_*[\mathcal{I}] - \mathbf{x}_k[\mathcal{I}])\right| \leq 2s\widetilde{\mu}\theta_k. \tag{50}$$

Therefore, by using Eq. (46) we get

$$|\mathbf{x}_{k+1}[i] - \mathbf{x}_*[i]| \leq \theta_{k+1} + 2s\widetilde{\mu}\theta_k + \theta_{k+1}(1 - 2\gamma\widetilde{\mu}s), \tag{51}$$

and by placing $\theta_k = \gamma\theta_{k+1}$ we obtain

$$|\mathbf{x}_{k+1}[i] - \mathbf{x}_*[i]| \leq 2\theta_{k+1}. \tag{52}$$

By taking a maximum over i, we establish the recovery error hypothesis (Eq. (30)), concluding the proof. □

6.2 Proof of Theorem 2

We define an effective matrix $\mathbf{G} = \mathbf{D}^T\mathbf{W}^T\mathbf{D}$. In this part, we aim to prove that linear convergence is guaranteed for any dictionary $\mathbf{D}$, satisfying two conditions: (i) the diagonal elements of $\mathbf{G}$ are close to 1, and (ii) the off-diagonal elements of $\mathbf{G}$ are bounded.

Proof. This proof is based on Section 6.1, with the following two modifications: The mutual coherence $\widetilde{\mu}$ is replaced with μ, and the diagonal element $\mathbf{a}_i^T\mathbf{d}_i$ is not assumed to be equal to 1, but rather bounded from below by $1 - \epsilon_d$. The base case of the induction (Eq. (36)) now becomes:

$$\|\mathbf{x}_*\|_\infty(1 - \epsilon_d - \mu s) \leq \left\|\mathbf{A}^T \mathbf{y}\right\|_\infty + \left\|\mathbf{A}^T \mathbf{e}\right\|_\infty. \tag{53}$$

Since we assume $\left\|\mathbf{A}^T \mathbf{y}\right\|_\infty \leq \theta_{\max}$, and

$$\left\|\mathbf{A}^T \mathbf{e}\right\|_\infty < \theta_{\max}(1 - 2\gamma\epsilon_d - 2\gamma\mu s), \tag{54}$$

we get

$$\|\mathbf{x}_*\|_\infty(1 - \epsilon_d - \mu s) \leq 2\theta_{\max}(1 - \gamma\epsilon_d - \gamma\mu s). \tag{55}$$

As $\gamma > 1$, $\|\mathbf{x}_*\|_\infty < 2\theta_{\max}$, therefore the induction hypothesis holds for the base case.

Moving to the inductive step, the proof of the support hypothesis remains almost the same, apart from replacing $\widetilde{\mu}$ with μ. This is due to the fact that if $i \notin \mathrm{Supp}(\mathbf{x}_*)$, then $\mathbf{x}_*[i] = \mathbf{x}_k[i] = 0$, and therefore the diagonal elements $\mathbf{a}_i^T\mathbf{d}_i$ multiply zero elements.

As to the recovery error hypothesis, we need to upper bound $\|\mathbf{x}_* - \mathbf{x}_{k+1}\|_\infty$ for $i \in \mathrm{Supp}(\mathbf{x}_*)$. Since $\mathbf{a}_i^T\mathbf{d}_i \neq 1$ we need to modify Eq. (41):

$$\begin{aligned}\mathbf{x}_{k+1}[i] = \mathcal{S}_{\theta_{k+1}}\big(&\mathbf{x}_*[i] + \mathbf{a}_i^T\mathbf{D}_{\mathcal{I}}(\mathbf{x}_* - \mathbf{x}_k)_{\mathcal{I}} \\ &+ \mathbf{a}_i^T\mathbf{e} + (1 - \mathbf{a}_i^T\mathbf{d}_i)(\mathbf{x}_k[i] - \mathbf{x}_*[i])\big).\end{aligned} \tag{56}$$

Using Eq. (49) we get that $|\mathbf{x}_{k+1}[i] - \mathbf{x}_*[i]|$ is upper bounded by

$$\theta_{k+1} + \mu s\|\mathbf{x}_* - \mathbf{x}_k\|_\infty + \|\tilde{\mathbf{A}}^T\mathbf{e}\|_\infty + \left|(1 - \tilde{\mathbf{a}}_i^T\mathbf{d}_i)\right||\mathbf{x}_k[i] - \mathbf{x}_*[i]|, \tag{57}$$

which in turn is upper bounded by

$$\theta_{k+1} + \mu s 2\theta_k + \theta_{k+1}(1 - 2\gamma\epsilon_d - 2\gamma\mu s) + 2\epsilon_d\theta_k. \tag{58}$$

Placing $\theta_k = \gamma\theta_{k+1}$ results in

$$|\mathbf{x}_{k+1}[i] - \mathbf{x}_*[i]| \leq 2\theta_{k+1}. \tag{59}$$

Taking a maximum over i establishes the recovery error assumption, proving the induction hypothesis. □

6.3 Proof for Random Permutations

We show that if the weight matrix $\mathbf{W}$ leads to linear convergence for signals generated by $\mathbf{D}$, then linear convergence is also guaranteed for signals originating from $\tilde{\mathbf{D}} = \mathbf{DP}$, where $\mathbf{P}$ is a permutation matrix. The proof is straightforward, as the permutation matrix does not flip diagonal and off-diagonal elements in the effective matrix $\mathbf{P}^T\mathbf{GP}$. Thus, the mutual coherence does not change and the conditions of Theorem 2 hold, establishing linear convergence.

6.4 Proof for Noisy Dictionaries

We now consider signals from noisy models, $\mathbf{y} = \tilde{\mathbf{D}}\mathbf{x}_* + \mathbf{e}$, where $\tilde{\mathbf{D}} = \mathbf{D} + \mathbf{E}$, and the model deviations are of Gaussian distribution, $E_{ij} \sim \mathcal{N}(0, \sigma^2)$. Given pairs of $(\mathbf{y}, \tilde{\mathbf{D}})$, we show that Ada-LISTA recovers the original representations $\mathbf{x}_*$, with respect to their model $\tilde{\mathbf{D}}$ in linear rate.

Theorem 3 (Ada-LISTA Convergence – Noisy Model). *Consider a noisy input $\mathbf{y} = \tilde{\mathbf{D}}\mathbf{x}_* + \mathbf{e}$, where $\tilde{\mathbf{D}} = \mathbf{D} + \mathbf{E}$, $E_{ij} \sim \mathcal{N}(0, \sigma^2/n)$. If for some constants $\tau_{\mathrm{d}}, \tau_{\mathrm{od}} > 0$, $\mathbf{x}_*$ is sufficiently sparse,*

$$s = \|\mathbf{x}_*\|_0 < \frac{1}{2\mu} - \frac{\epsilon_d}{\mu}, \quad \mu \triangleq \widetilde{\mu} + \tau_{\text{od}}, \tag{60}$$

and the thresholds satisfy

$$\theta_k = \theta_{\max} \gamma^{-k} \; > \; \theta_{\min} = \frac{\|\tilde{\mathbf{A}}^T \mathbf{e}\|_\infty}{1 - 2\gamma\epsilon_d - 2\gamma\mu s}, \tag{61}$$

with $1 < \gamma < 0.5(\mu s + \epsilon_d)^{-1}$, $\epsilon_d \triangleq w_{\text{d}} + \tau_{\text{d}} < \frac{1}{2}$, $w_{\text{d}} \triangleq \frac{\sigma^2}{n} \sum_{k=1}^n W_{kk}$, $\tilde{\mathbf{A}} \triangleq \mathbf{W}\tilde{\mathbf{D}}$, *and* $\theta_{\max} \geq \|\tilde{\mathbf{A}}^T \mathbf{y}\|_\infty$, *then, with probability of at least* $(1 - p_1 p_2)$, *the support of the kth iteration of Ada-LISTA is included in the support of* $\mathbf{x}_*$ *and its values satisfy*

$$\|\mathbf{x}_k - \mathbf{x}_*\|_\infty \leq 2\max\{\theta_{\max} \gamma^{-k}, \theta_{\min}\}. \tag{62}$$

Proof. The proof for this theorem consists of two stages. First, we study the effect of model perturbations on the effective matrix $\tilde{\mathbf{G}} = \tilde{\mathbf{D}}^T \mathbf{W}^T \tilde{\mathbf{D}}$, deriving probabilistic bounds for the changes in the diagonal and off-diagonal elements. Then, we place these bounds in Theorem 2 to guarantee linear rate.

We start by bounding the changes in $\tilde{\mathbf{G}} = \tilde{\mathbf{D}}^T \mathbf{W}^T \tilde{\mathbf{D}}$. These deviations modify the off-diagonal elements, which are no longer bounded by $\widetilde{\mu}$, and the diagonal elements that are not equal to 1 anymore. Define $\mathbf{G}$ as:

$$\mathbf{G} = \tilde{\mathbf{G}} - \mathbf{G} = \mathbf{D}^T \mathbf{W}^T \mathbf{E} + \mathbf{E}^T \mathbf{W}^T \mathbf{D} + \mathbf{E}^T \mathbf{W}^T \mathbf{E}. \tag{63}$$

This implies $\mathbf{G}_{ij}$ is equal to:

$$\begin{aligned} \mathbf{G}_{ij} = \underbrace{\sum_{k=1}^n \sum_{l=1}^n D_{ki} W_{lk} E_{lj}}_{\triangleq T^a_{ij}} + \underbrace{\sum_{k=1}^n \sum_{l=1}^n E_{ki} W_{lk} D_{lj}}_{\triangleq T^b_{ij}} \\ + \underbrace{\sum_{k=1}^n \sum_{l=1}^n E_{ki} W_{lk} E_{lj}}_{\triangleq T^c_{ij}}. \end{aligned} \tag{64}$$

Since $\mathbb{E}[E_{ij}^2] = \frac{\sigma^2}{n}$ and the elements in $\mathbf{E}$ are independent, the expected value of $\mathbf{G}_{ij}$ is

$$\mathbb{E}[\mathbf{G}_{ij}] = \begin{cases} \frac{\sigma^2}{n} \sum_{k=1}^n W_{kk}, & \text{if } i = j \\ 0, & \text{if } i \neq j. \end{cases} \tag{65}$$

To bound the changes in $\mathbf{G}_{ij}$ we aim to use Cantelli's inequality, but first, we need to find the variance of $\mathbf{G}_{ij}$:

$$\begin{aligned} \text{Var}[\mathbf{G}_{ij}] = \mathbb{E}[T^a_{ij}]^2 + \mathbb{E}[T^b_{ij}]^2 + \mathbb{E}[T^c_{ij} - \mathbb{E}[\mathbf{G}_{ij}]]^2 + 2\mathbb{E}[T^a_{ij} T^b_{ij}] \\ + 2\mathbb{E}[T^a_{ij}(T^c_{ij} - \mathbb{E}[\mathbf{G}_{ij}])] + 2\mathbb{E}[T^b_{ij}(T^c_{ij} - \mathbb{E}[\mathbf{G}_{ij}])]. \end{aligned}$$

In what follows we calculate each term in the right-hand-side, starting with $\mathbb{E}[T^a_{ij}]^2$:

$$\begin{aligned} \mathbb{E}[T^a_{ij}]^2 &= \mathbb{E}\Big[\sum_{k=1}^n \sum_{l=1}^n D_{ki} W_{lk} E_{lj} \sum_{k'=1}^n \sum_{l'=1}^n D_{k'i} W_{l'k'} E_{l'j}\Big] \\ &= \frac{\sigma^2}{n} \underbrace{\mathbb{E}\Big[\sum_{k=1}^n \sum_{l=1}^n \sum_{k'=1}^n D_{ki} W_{lk} D_{k'i} W_{lk'}\Big]}_{\triangleq C^a_{ij}}. \end{aligned} \tag{66}$$

Moving on to $\mathbb{E}[T^b_{ij}]^2$, we get

$$\begin{aligned} \mathbb{E}[T^b_{ij}]^2 &= \mathbb{E}\Big[\sum_{k=1}^n \sum_{l=1}^n E_{ki} W_{lk} D_{lj} \sum_{k'=1}^n \sum_{l'=1}^n E_{k'i} W_{l'k'} D_{l'j}\Big] \\ &= \frac{\sigma^2}{n} \underbrace{\mathbb{E}\Big[\sum_{k=1}^n \sum_{l=1}^n \sum_{l'=1}^n W_{lk} D_{lj} W_{l'k} D_{l'j}\Big]}_{\triangleq C^b_{ij}}. \end{aligned} \tag{67}$$

As for $\mathbb{E}[T^c_{ij} - \mathbb{E}[\mathbf{G}_{ij}]]^2$, if $i \neq j$ then

$$\begin{aligned} \mathbb{E}\Big[\sum_{k=1}^n \sum_{l=1}^n E_{ki} W_{lk} E_{lj} \sum_{k'=1}^n \sum_{l'=1}^n E_{k'i} W_{l'k'} E_{l'j}\Big] \\ = \frac{\sigma^4}{n^2} \underbrace{\mathbb{E}\Big[\sum_{k=1}^n \sum_{l=1}^n W_{lk}^2\Big]}_{\triangleq C^c_{ij}}. \end{aligned} \tag{68}$$

Whereas, if $i = j$, then $\mathbb{E}[T^c_{ij} - \mathbb{E}[\mathbf{G}_{ij}]]^2$ becomes

$$\begin{aligned} \mathbb{E}\Big[\sum_{k=1}^n \sum_{l=1\neq k}^n E_{ki}^2 W_{lk}^2 E_{lj}^2\Big] + \mathbb{E}\Big[\sum_{k=1}^n \sum_{l=1\neq k}^n E_{ki}^2 W_{lk} W_{kl} E_{lj}^2\Big] \\ + \mathbb{E}\Big[\sum_{k=1}^n \sum_{k'=1\neq k}^n E_{ki} W_{kk} E_{kj} E_{k'i} W_{k'k'} E_{k'j}\Big] \\ + \mathbb{E}\Big[\sum_{k=1}^n E_{ki}^2 W_{kk}^2 E_{kj}^2\Big] - \frac{\sigma^4}{n^2}\Big(\sum_{k=1}^n W_{kk}\Big)^2 \end{aligned} \tag{69}$$

Using the fourth moment of Gaussian distribution, we obtain $\mathbb{E}[T^c_{ij} - \mathbb{E}[\mathbf{G}_{ij}]]^2$ is equal to

$$\frac{\sigma^4}{n^2} \underbrace{\left(\sum_{k=1}^n \sum_{l=1\neq k}^n W_{lk}^2 + \sum_{k=1}^n \sum_{l=1\neq k}^n W_{lk} W_{kl} + 2\sum_{k=1}^n W_{kk}^2\right)}_{\triangleq C^d_{ij}}. \tag{70}$$

Continuing with $2\mathbb{E}[T^a_{ij} T^b_{ij}]$, we get

$$2\mathbb{E}[T^a_{ij} T^b_{ij}] = 2\mathbb{E}\Big[\sum_{k=1}^n \sum_{l=1}^n \sum_{l'=1}^n D_{ki} W_{lk} E_{lj} E_{li} W_{l'l} D_{l'j}\Big]. \tag{71}$$

Therefore, if $i = j$ then $\mathbb{E}[T^a_{ij} T^b_{ij}] = 0$, and if $i \neq j$ then

$$2\mathbb{E}[T^a_{ij} T^b_{ij}] = \frac{\sigma^2}{n} \underbrace{2\sum_{k=1}^n \sum_{l=1}^n \sum_{l'=1}^n D_{ki} W_{lk} W_{l'l} D_{l'j}}_{\triangleq C^e_{ij}}. \tag{72}$$

As for $2\mathbb{E}[T^a_{ij}(T^c_{ij} - \mathbb{E}[\mathbf{G}_{ij}])]$, and $2\mathbb{E}[T^b_{ij}(T^c_{ij} - \mathbb{E}[\mathbf{G}_{ij}])]$, both are zero since the third moment of Gaussian variable is zero.

To conclude, we define the maximal variance of the off-diagonal elements as,

$$v_{\mathrm{od}} \triangleq \max_{i \neq j} \frac{\sigma^2}{n}\left(C^a_{ij} + C^b_{ij}\right) + \frac{\sigma^4}{n^2} C^c_{ij}, \tag{73}$$

and the maximal variance of the diagonal elements as,

$$v_{\mathrm{d}} \triangleq \max_{i=j} \frac{\sigma^2}{n}\left(C^a_{ij} + C^b_{ij} + C^e_{ij}\right) + \frac{\sigma^4}{n^2} C^d_{ij}. \tag{74}$$

Identifying the variance of $\mathbf{G}_{ij}$ enables to bound the changes in the effective matrix using Cantelli's inequality. Starting with the off-diagonal elements, we obtain

$$p(\left|\mathbf{G}_{ij}\right| \geq \tau_{\mathrm{od}}) \leq \frac{2{v_{\mathrm{od}}}^2}{{v_{\mathrm{od}}}^2 + {\tau_{\mathrm{od}}}^2}. \tag{75}$$

Taking the maximum over all off-diagonal elements, we get

$$p(\max_{i,j \neq i}\left|\mathbf{G}_{ij}\right| \geq \tau_{\mathrm{od}}) \leq p_1, \tag{76}$$

with

$$p_1 \triangleq 1 - \left(\frac{{\tau_{\mathrm{od}}}^2 - {v_{\mathrm{od}}}^2}{{v_{\mathrm{od}}}^2 + {\tau_{\mathrm{od}}}^2}\right)^{n(n-1)}. \tag{77}$$

Moving on to the diagonal elements, we have

$$p\left(\left|\mathbf{G}_{ii} - \frac{\sigma^2}{n}\sum_{k=1}^{n} W_{kk}\right| \geq \tau_{\mathrm{d}}\right) \leq \frac{2{v_{\mathrm{d}}}^2}{{v_{\mathrm{d}}}^2 + {\tau_{\mathrm{d}}}^2}. \tag{78}$$

Taking the maximum over all diagonal elements, we get

$$p\left(\max_i\left|\mathbf{G}_{ii} - \frac{\sigma^2}{n}\sum_{k=1}^{n} W_{kk}\right| \geq \tau_{\mathrm{d}}\right) \leq p_2, \tag{79}$$

with

$$p_2 \triangleq 1 - \left(\frac{{\tau_{\mathrm{d}}}^2 - {v_{\mathrm{d}}}^2}{{v_{\mathrm{d}}}^2 + {\tau_{\mathrm{d}}}^2}\right)^{n}. \tag{80}$$

Therefore, with probability of at least $1 - p_1 p_2$, we obtain that the matrix $\tilde{\mathbf{G}} = \mathbf{W}^T \tilde{\mathbf{D}}^T \tilde{\mathbf{D}}$ satisfies the following:

- The off-diagonal elements are bounded:

$$\max_{i,j \neq i}\left|\tilde{G}_{ij}\right| \leq \tilde{\mu} + \tau_{\mathrm{od}}. \tag{81}$$

- The diagonal elements are close to 1:

$$\max_i\left|\tilde{G}_{ii} - 1\right| \leq w_{\mathrm{d}} + \tau_{\mathrm{d}}, \quad w_{\mathrm{d}} \triangleq \frac{\sigma^2}{n}\sum_{k=1}^{n} W_{kk}. \tag{82}$$

Finally, we apply Theorem 2 with the constants

$$\mu = \tilde{\mu} + \tau_{\mathrm{od}}, \quad \epsilon_d = w_{\mathrm{d}} + \tau_{\mathrm{d}}, \tag{83}$$

and establish linear convergence, with probability of at least $(1 - p_1 p_2)$. □

7 Conclusion

We have introduced Ada-LISTA, a new extension of LISTA which receives both the signals and their dictionaries as input, and learns a universal architecture that can cope with varying models. This modification shows great flexibility in working with changing dictionaries, leveling the playing field with non-learned solvers such as ISTA and FISTA that are agnostic to the entire signal distribution, while enjoying the acceleration and convergence benefits of learned solvers. We have substantiated the validity of our method, both in a comprehensive theoretical study, and with extensive synthetic and real-world experiments. Future work includes further investigation of the discussed rationale, and an extension to additional applications.

Acknowledgments

A. Aberdam thanks the Azrieli foundation for providing additional research support. A. Aberdam and A. Golts contributed equally to this work.

References

[1] R. Tibshirani, "Regression shrinkage and selection via the lasso," *J. Roy. Stat. Society*, vol. 58, pp. 267–288, 1996.

[2] S. S. Chen, D. L. Donoho, and M. A. Saunders, "Atomic decomposition by basis pursuit," *SIAM Rev.*, vol. 43, no. 1, pp. 129–159, 2001.

[3] I. Daubechies, M. Defrise, and C. De Mol , "An iterative thresholding algorithm for linear inverse problems with a sparsity constraint," *Commun. Pure Appl. Math. J. Issued Courant Inst. Math. Sci.*, vol. 57, no. 11, pp. 1413–1457, 2004.

[4] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM J. Imag. Sci.*, vol. 2, no. 1, pp. 183–202, 2009.

[5] P. L. Combettes and V. R. Wajs, "Signal recovery by proximal forward-backward splitting," *Multiscale Model. Simul.*, vol. 4, no. 4, pp. 1168–1200, 2005.

[6] A. Beck, *First-Order Methods in Optimization*. Philadelphia, PA, USA: SIAM, 2017.

[7] R. Mazumder, T. Hastie, and R. Tibshirani, "Spectral regularization algorithms for learning large incomplete matrices," *J. Mach. Learn. Res.*, vol. 11, pp. 2287–2322, 2010.

[8] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[9] P. Sprechmann, A. Bronstein, and G. Sapiro, "Learning efficient sparse and low rank models," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 37, no. 9, pp. 1821–1833, Sep. 2015.

[10] K. Gregor and Y. LeCun , "Learning fast approximations of sparse coding," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 399–406.

[11] M. Protter and M. Elad, "Image sequence denoising via sparse and redundant representations," *IEEE Trans. Image Process.*, vol. 18, no. 1, pp. 27–35, Jan. 2009.

[12] B. Zhao, L. Fei-Fei , and E. P. Xing, "Online detection of unusual events in videos via dynamic sparse coding," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2011, pp. 3313–3320.

[13] K. Kulkarni, S. Lohit, P. Turaga, R. Kerviche, and A. Ashok, "Reconnet: Non-iterative reconstruction of images from compressively sensed measurements," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2016, pp. 449–458.

[14] S. Tang, W. Gong, W. Li, and W. Wang, "Non-blind image deblurring method by local and nonlocal total variation models," *Signal Process.*, vol. 94, pp. 339–349, 2014.

[15] J. Mairal, M. Elad, and G. Sapiro, "Sparse representation for color image restoration," *IEEE Trans. Image Process.*, vol. 17, no. 1, pp. 53–69, Jan. 2008.

[16] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, "Plug-and-play priors for model based reconstruction," in *Proc. IEEE Glob. Conf. Signal Inf. Process.*, 2013, pp. 945–948.

[17] Y. Romano, M. Elad, and P. Milanfar, "The little engine that could: Regularization by denoising (red)," *SIAM J. Imag. Sci.*, vol. 10, no. 4, pp. 1804–1844, 2017.

[18] A. Golts, D. Freedman, and M. Elad, "Deep energy: Task driven training of deep neural networks," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 2, pp. 324–338, 2021.

[19] J. Zarka, L. Thiry, T. Angles, and S. Mallat, "Deep network classification by scattering and homotopy dictionary learning," 2019, *arXiv:1910.03561*.

[20] J. Liu, X. Chen, Z. Wang, and W. Yin, "ALISTA: Analytic weights are as good as learned weights in LISTA," in *Proc. Int. Conf. Learn. Representation*, 2019.

[21] K. Wu, Y. Guo, Z. Li, and C. Zhang, "Sparse coding with gated learned {ISTA}," in *Proc. Int. Conf. Learn. Representations*, 2020.

[22] A. M. Bruckstein, D. L. Donoho, and M. Elad, "From sparse solutions of systems of equations to sparse modeling of signals and images," *SIAM review*, vol. 51, no. 1, pp. 34–81, 2009.

[23] J. Sulam, A. Aberdam, A. Beck, and M. Elad, "On multi-layer basis pursuit, efficient algorithms and convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 42, no. 8, pp. 1968–1980, Aug. 2020.

[24] X. Chen, J. Liu, Z. Wang, and W. Yin, "Theoretical linear convergence of unfolded ista and its practical weights and thresholds," in *Proc. Neural Inf. Process. Syst.*, 2018, pp. 9061–9071.

[25] J. Zhang and B. Ghanem, "Ista-net: Interpretable optimization-inspired deep network for image compressive sensing," in *Proc. Int. Conf. Comput. Vis.*, 2018, pp. 1828–1837.

[26] C. Metzler, A. Mousavi, and R. Baraniuk, "Learned d-amp: Principled neural network based compressive image recovery," in *Proc. Neural Inf. Process. Syst.* 2017, pp. 1772–1783.

[27] Z. Wang, D. Liu, J. Yang, W. Han, and T. Huang, "Deep networks for image super-resolution with sparse prior," in Proc. *Int. Conf. Comput. Vis.*, 2015, pp. 370–378.

[28] M. Borgerding, P. Schniter, and S. Rangan, "Amp-inspired deep networks for sparse linear inverse problems," *IEEE Trans. Signal Process.*, vol. 65, no. 16, pp. 4293–4308, Aug. 2017.

[29] J. Sun *et al.*, "Deep admm-net for compressive sensing MRI," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 10–18.

[30] J. R. Hershey, J. L. Roux, and F. Weninger, "Deep unfolding: Model-based inspiration of novel deep architectures," 2014, *arXiv:1409.2574*.

[31] B. Xin, Y. Wang, W. Gao, D. Wipf, and B. Wang, "Maximal sparsity with deep networks?," in *Proc. Neural Inf. Process. Syst.*, 2016, pp. 4340–4348.

[32] Z. Wang, Q. Ling, and T. S. Huang, "Learning deep ℓ_0 encoders," in *Proc. AAAI Conf. Artif. Intell.*, vol. 30, no. 1, 2016.

[33] R. Giryes, Y. C. Eldar, A. M. Bronstein, and G. Sapiro, "Tradeoffs between convergence speed and reconstruction accuracy in inverse problems," *IEEE Trans. Signal Process.*, vol. 66, no. 7, pp. 1676–1690, Apr. 2018.

[34] J. Tompson, K. Schlachter, P. Sprechmann, and K. Perlin, "Accelerating eulerian fluid simulation with convolutional networks," in *Proc. Int. Conf. Mach. Learn., 2017*, pp. 3424–3433.

[35] M. Andrychowicz *et al.*, "Learning to learn by gradient descent by gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2016, pp. 3981–3989.

[36] P. Ablin, T. Moreau, M. Massias, and A. Gramfort, "Learning step sizes for unfolded sparse coding," 2019, *arXiv:1905.11071*.

[37] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics," in *Proc. Int. Conf. Comput. Vis.*, 2011, pp. 416–423.

Aviad Aberdam received the BSc, MSc, and PhD degrees from the Department of Electrical Engineering, Technion – Israel Institute of Technology, Haifa, Israel, in 2017, 2019, and 2021 respectively. He is currently an applied scientist with Amazon AWS AI. His research interests include machine learning, optimization and signal and image processing, inverse problems, and sparse representations. He was the recipient of 2020–2022 Azrieli fellowship, 2019 Fine fellowship, and 2017–2018 Meyer fellowship.

Alona Golts received the BSc degree from the Department of Electrical Engineering and Physics in 2010 and the MSc degree from the Department of Electrical Engineering, Technion–Israel Institute of Technology, Haifa, Israel, in 2015. She is currently working toward the PhD degree with the Department of Computer Science, Technion supervised by Prof. Michael Elad. Her research interests include signal and image processing, computer vision, deep learning, medical image analysis, inverse problems, and sparse representations.

Michael Elad (Fellow, IEEE) is currently a faculty with Computer-Science Department, Technion. He has authored hundreds of publications in leading venues, many of which were exceptionally high-impact. His book *Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing* is a leading publication in this field. His research interests include the fields of signal and image processing and machine learning, specializing in inverse problems, sparse representations, and deep learning. He was an associate editor for *IEEE Transactions on Image Processing*, *IEEE Transactions on Information Theory*, *Applied and Computational Harmonic Analysis*, and *SIAM Imaging Sciences*. He held a senior editorial role for *IEEE Signal Processing Letters* from 2012 to 2014 and since January 2016, he has been the editor-in-chief for *SIAM Journal on Imaging Sciences*. He was the recipient of numerous teaching and research awards and grants, including an ERC advanced grant (2013), the Henri Taub Prize for Academic Excellence (2008 and 2015), and Hershel-Rich Prize for Innovation (2017). He has been the SIAM fellow since 2018.