Ada-LISTA: Learned Solvers Adaptive to Varying Models

Aviad Aberdam [1] Alona Golts [2] Michael Elad [2]

Abstract

Neural networks that are based on unfolding of an iterative solver, such as LISTA (learned iterative soft threshold algorithm), are widely used due to their accelerated performance. Nevertheless, as opposed to non-learned solvers, these networks are trained on a certain dictionary, and therefore they are inapplicable for varying model scenarios. This work introduces an adaptive learned solver, termed Ada-LISTA, which receives pairs of signals and their corresponding dictionaries as inputs, and learns a universal architecture to serve them all. We prove that this scheme is guaranteed to solve sparse coding in linear rate for varying models, including dictionary perturbations and permutations. We also provide an extensive numerical study demonstrating its practical adaptation capabilities. Finally, we deploy Ada-LISTA to natural image inpainting, where the patch-masks vary spatially, thus requiring such an adaptation.

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

$$\mathbf{y} \approx \mathbf{D}\mathbf{x}, \quad \text{s.t. } \|\mathbf{x}\|_0 \leq s, \tag{1}$$

where the L^0-norm counts the non-zero elements, s is the cardinality of the representation, and $\mathbf{D}$ is often redundant ($m \geq n$). 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 (Tibshirani, 1996; Chen et al., 2001),

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

[1]Department of Electrical Engineering, Technion Institute of Technology, Israel. [2]Department of Computer Science, Technion Institute of Technology, Israel. Correspondence to: Aviad Aberdam <aaberdam@cs.technion.ac.il>, Alona Golts <salonaz@cs.technion.ac.il>, Michael Elad <elad@cs.technion.ac.il>.
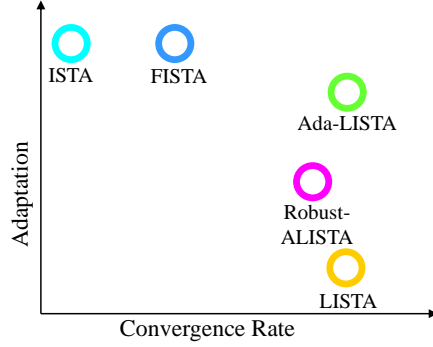
Figure 1: We propose "Ada-LISTA", a fusion between the flexible ISTA and FISTA schemes, receiving both the signal and dictionary at inference time, and the highly efficient learned solvers, LISTA and ALISTA.

An effective way to address this optimization problem uses an iterative algorithm such as ISTA (Iterative Soft Thresholding Algorithm) (Daubechies et al., 2004), where the solution is obtained by iterations of the form

$$\mathbf{x}_{k+1} = \mathcal{S}_{\frac{\lambda}{L}} \left(\mathbf{x}_k + \frac{1}{L} \mathbf{D}^T (\mathbf{y} - \mathbf{D}\mathbf{x}_k) \right), k = 0, 1, .. \tag{3}$$

where $\frac{1}{L}$ is the step size determined by the largest eigenvalue of the Gram matrix $\mathbf{D}^T \mathbf{D}$, and $\mathcal{S}_\theta(\mathbf{x}_i) = \text{sign}(\mathbf{x}_i)(|\mathbf{x}_i| - \theta_i)$ is the soft shrinkage function. Fast-ISTA (FISTA) (Beck & Teboulle, 2009) is a speed-up of the above iterative algorithm, which should remind the reader of the momentum method in optimization.

As a side note, we mention that ISTA has a much wider perspective when aiming to minimize a function of the form

$$F(\mathbf{x}) = f(\mathbf{x}) + g(\mathbf{x}), \tag{4}$$

where f and g are convex functions, with g possibly non-smooth. The solution is given by the proximal gradient method (Combettes & Wajs, 2005; Beck, 2017):

$$\mathbf{x}_{k+1} = \underset{g}{\text{prox}} \left(\mathbf{x}_k - \frac{1}{L} \nabla f(\mathbf{x}_k) \right),$$
$$\underset{g}{\text{prox}}(\mathbf{u}) = \arg\min_{\mathbf{v}} \frac{1}{2} \|\mathbf{v} - \mathbf{u}\|_2^2 + g(\mathbf{v}). \tag{5}$$

The above fits various optimization problems such as a projected gradient descent over an indicator function g, the ma-

trix completion problem (Mazumder et al., 2010), portfolio optimization (Boyd & Vandenberghe, 2004), non-negative matrix factorization (Sprechmann et al., 2015), and more.

Returning to the realm of sparse coding, the seminal work of LISTA (Learned-ISTA) (Gregor & LeCun, 2010) 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:

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

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,

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

Once trained, LISTA requires only the test signals during inference, without their underlying dictionary. It has been shown in (Gregor & LeCun, 2010) that LISTA generalizes well for signals of the same distribution as in the train set, allowing a significant speedup versus its non-learned counterparts. This may be explained by the fact that while non-learned solvers do not make any assumption on the input signals, LISTA fits itself to the input distribution. More 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 on such signals solely, this allows LISTA to achieve its acceleration. Note, however, that the original dictionary is hard-coded into the model weights via the ground truth solutions used during the supervised training. Given a new test sample that emerges from a slightly deviated (yet known) model/dictionary, LISTA will most likely deteriorate in performance, whereas ISTA and FISTA are expected to provide a robust and consistent result, as they are agnostic to the input signals and dictionary.

From a different point of view, a drawback of LISTA is its relevance to a single dictionary, requiring a separate and renewed training if the model evolves over time. Such is the case in video related applications as enhancement (Protter & Elad, 2008) or surveillance (Zhao et al., 2011), where the dictionary should vary along time. 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

(Kulkarni et al., 2016), the blur kernel in non-blind image deblurring (Tang et al., 2014), and a spatially-varying mask in image inpainting (Mairal et al., 2007). 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 (Venkatakrishnan et al., 2013) or RED (Romano et al., 2017) schemes, significantly increasing the inference complexity.

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 the fast convergence rate of LISTA, and the high adaptivity and applicability of ISTA (and FISTA), by introducing "Ada-LISTA" (Adaptive-LISTA). Our training is based on pairs of signals and their corresponding dictionaries, learning a generic architecture that wraps the dictionary by two auxiliary weight matrices. At inference, our model can accommodate the signal and its corresponding dictionary, allowing to handle 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: permuted columns, additive Gaussian noise, and completely renewed random dictionaries. We demonstrate the ability of Ada-LISTA to handle complex and varying signal models while still providing an impressive advantage over both learned and non-learned solvers.

- We prove that our modified scheme achieves a linear convergence rate under a constant dictionary. More importantly, we allow for noisy modifications and random permutations to the dictionary and prove that robustness remains, with an ability to reconstruct the ideal sparse representations with the same linear rate.

- We demonstrate the use of our approach on natural image inpainting, which cannot be directly used with hard-coded models as LISTA. We show a clear advantage of Ada-LISTA versus its non-learned counterparts.

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

The LISTA concept of unfolding the iterations of a classical optimization scheme into an RNN-like neural network, and freeing its parameters to be learned over the training data, appears in many works. These include an unsupervised and online training procedure (Sprechmann et al., 2015), a multi-layer version (Sulam et al., 2019), a gated mechanism compensating shrinkage artifacts (Wu et al., 2020), as well as reduced-parameter schemes (Chen et al., 2018; Liu et al., 2019). This paradigm has been brought to various applications, such as compressed sensing, super-resolution, communication, MRI reconstruction (Zhang & Ghanem, 2018; Metzler et al., 2017; Wang et al., 2015; Borgerding et al., 2017; Sun et al., 2016; Hershey et al., 2014), and more. A prominent line of work investigates the success of such learned solvers from a theoretical point of view (Xin et al., 2016; Wang et al., 2016; Moreau & Bruna, 2016; Giryes et al., 2018; Zarka et al., 2019). Most of these consider a fixed signal model, with the exception of "robust-ALISTA" (Liu et al., 2019) that introduces an adaptive variation of LISTA. This scheme, however, is restricted to small model perturbations, and cannot address more complicated model variations. A more detailed discussion of the relevant literature in relation of our study appears in Section 4.

2. Proposed Method

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**
$\quad \mathbf{x}_{k+1} = \mathcal{S}_{\theta_{k+1}} \big((\mathbf{I} - \gamma_{k+1} \mathbf{D}^T \mathbf{W}_2^T \mathbf{W}_2 \mathbf{D}) \mathbf{z}_k$
$\quad\quad\quad\quad\quad\quad\quad\quad\quad\quad + \gamma_{k+1} \mathbf{D}^T \mathbf{W}_1^T \mathbf{y} \big)$
$\quad$ **if** Ada-LISTA **then**
$\quad\quad \mathbf{z}_{k+1} = \mathbf{x}_{k+1}$
$\quad$ **else if** Ada-LFISTA **then**
$\quad\quad t_{k+1} = \frac{1 + \sqrt{1 + 4t_k^2}}{2}$
$\quad\quad \mathbf{z}_{k+1} = \mathbf{x}_{k+1} + \frac{t_k - 1}{t_{k+1}} (\mathbf{x}_{k+1} - \mathbf{x}_k)$
$\quad$ **end if**
end for
Return: $\mathcal{F}_K(\mathbf{y}, \mathbf{D}; \Theta) = \mathbf{x}_K$

Algorithm 2 Ada-LISTA Training

Input: pairs of signals and dictionaries $\{\mathbf{y}_i, \mathbf{D}_i\}_{i=1}^N$
Preprocessing: find $\mathbf{x}_i$ for each pair $(\mathbf{y}_i, \mathbf{D}_i)$ by solving Equation 2 using ISTA
Goal: learn $\Theta = (\mathbf{W}_1, \mathbf{W}_2, \theta_k, \gamma_k)$
Init: $\mathbf{W}_1, \mathbf{W}_2 = \mathbf{I}$, $\theta_k, \gamma_k = 1$
for each batch $\{\mathbf{y}_i, \mathbf{D}_i, \mathbf{x}_i\}_{i=1}^{N_B}$ **do**
$\quad$ update Θ by $\partial_\Theta \sum_{i \in N_B} \|\mathcal{F}_K(\mathbf{y}_i, \mathbf{D}_i; \Theta) - \mathbf{x}_i\|_2^2$
end for

Thus far, as depicted in Figure 1, one could either bene-

fit from a high convergence rate using a learned solver as LISTA, while restricting the signals to a specific model, or employ a non-learned and less effective solver as ISTA/FISTA that is capable of handling any pair of signal and its generative model. In this paper we introduce a novel architecture, termed "Adaptive-LISTA" (Ada-LISTA), combining both benefits. Beyond enjoying the acceleration benefits of learned solvers, we incorporate the dictionary as part of the input at both training and inference time, allowing for adaptivity to different models. Figure 2 provides our suggested architecture, based on the following:

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

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

The signal $\mathbf{y}$ and the dictionary $\mathbf{D}$ are the inputs, and the learned parameters are $\mathbf{W}_1, \mathbf{W}_2 \in \mathbb{R}^{n \times n}$ and $\{\gamma_k, \theta_k\}$.

The inference (for both ISTA and FISTA) and the training procedures of Ada-LISTA are detailed in Algorithms 1 and 2 correspondingly. We consider a similar loss as in Equation 7, while also incorporating the concurrent dictionaries,

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

This learning regime is supervised, requiring reference representations $\mathbf{x}_i$ to be computed using ISTA. An unsupervised alternative can be envisioned, as in (Sprechmann et al., 2015; Golts et al., 2018), where the loss is

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

In this paper we shall focus on the supervised mode of learning, leaving the unsupervised alternative to future work.

Several key questions arise on the applicability of the above learned solver: Does it work? and if so, is performance compromised by Ada-LISTA, as opposed to training LISTA for each separate model? To what extent can it be used? Can it handle completely random models? Can theoretical guarantees be provided on its convergence rate, or adaptation capability? We aim to answer these questions, and we start with a theorem on the robustness of our scheme by proving *linear rate* convergence under *varying* model.

---

[1]Although the above definition corresponds to the sparse coding problem, the Ada-LISTA method can be applied to any convex problem formulated as Equation 4, by swapping the soft-threshold with a different proximal operator (Equation 5).
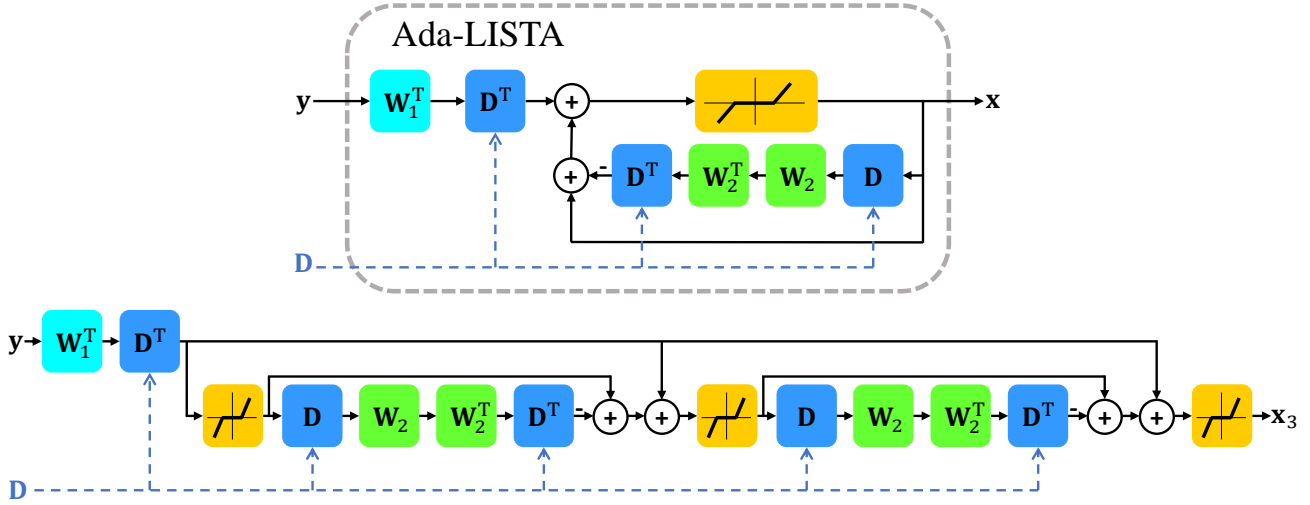
Figure 2: Ada-LISTA architecture as an iterative model (top), and its unfolded version for three iterations (bottom). The input dictionary $\mathbf{D}$ is embedded in the architecture, while the matrices $\mathbf{W}_1, \mathbf{W}_2$ are free to be learned.

3. Ada-LISTA: Theoretical Study

For the following study, we consider a reduced scheme of Ada-LISTA with a single weight matrix, so as to avoid complication in theorem conditions. We emphasize, however, that the same claims can be derived for our original scheme.

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

$$\mathbf{x}_{k+1} = \mathcal{S}_{\theta_{k+1}}(\mathbf{x}_k + \mathbf{D}^T \mathbf{W}^T (\mathbf{y} - \mathbf{D}\mathbf{x}_k)). \tag{10}$$

We start by recalling 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, then the mutual coherence is defined as*

$$\mu(\mathbf{A}, \mathbf{B}) = \max_{i \neq j} |\mathbf{a}_i^T \mathbf{b}_j|, \tag{11}$$

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

Our first goal is to prove that Ada-LISTA is capable of solving the sparse coding problem in linear rate. We show that if all the 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. The following theorem indicates that if Ada-LISTA's training reaches its global minimum, the rate would be at least linear. In this part, we follow the steps in (Zarka et al., 2019), which generalize the proof of ALISTA (Liu et al., 2019) to noisy signals. The proof for Theorem 1 appears in Appendix A.

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

$$s = \|\mathbf{x}^*\|_0 < \frac{1}{2\widetilde{\mu}}, \quad \widetilde{\mu} \triangleq \mu(\mathbf{W}\mathbf{D}, \mathbf{D}), \tag{12}$$

and the thresholds satisfy the condition

$$\theta_k = \theta_{\max} \gamma^{-k} > \theta_{\min} = \frac{\|\mathbf{A}^T \mathbf{e}\|_\infty}{1 - 2\gamma \widetilde{\mu} s}, \tag{13}$$

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

$$\|\mathbf{x}_k - \mathbf{x}^*\|_\infty \leq 2\,\theta_{\max}\,\gamma^{-k}. \tag{14}$$

We proceed by claiming 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 to the original representation. 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 good generalization. Appendix B contains the proof of the following Theorem.

Theorem 2 (Ada-LISTA – The Applicable Dictionaries). *Consider a trained Ada-LISTA network with a 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: $\max_i |G_{ii} - 1| \leq \epsilon_d$;

2. The off-diagonals are bounded: $\max_{i \neq j} |G_{ij}| \leq \bar{\mu}$;

3. $\mathbf{x}^*$ is sufficiently sparse: $s = \|\mathbf{x}^*\|_0 < \frac{1}{2\bar{\mu}}$; and

4. The thresholds satisfy

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

with $1 < \gamma < (2\bar{\mu}s)^{-1}$, $\mathbf{A} \triangleq \mathbf{WD}$, and $\theta_{\max} \geq \|\mathbf{A}^T \mathbf{y}\|_\infty$,

then the support of the kth iteration of Ada-LISTA is included in the support of $\mathbf{x}^*$, and its values satisfy

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

An interesting question arising is the following: Once Ada-LISTA has been trained and the matrix $\mathbf{W}$ 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 the identity matrix, linear convergence is guaranteed. This holds in particular for two interesting scenarios, proven in Appendices C and D:

1. **Random permutations** – If Ada-LISTA converges for signals emerging from $\mathbf{D}$, it also converges 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 $\mu(\mathbf{WD}, \mathbf{D}) < \bar{\mu}$, 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 convergence guarantee in the presence of *model variations*, claiming that linear rate convergence is guaranteed, depending on the availability of small enough cardinality and low mutual-coherence $\tilde{\mu}$. Note that the above claim, as in previous work (Liu et al., 2019; Zarka et al., 2019), addresses the core capability of reaching linear convergence rate while disregarding both training and generalization errors.

4. Related Work

As already mentioned, the literature discussing LISTA and its successors, is abundant. In this section we aim to discuss relevant work to provide better context to our contribution.

The most relevant work to ours is "robust-ALISTA" (Liu et al., 2019), introducing adaptivity to dictionary perturbations. Their work assumes that every signal $\mathbf{y}_i$ comes from a different noisy model $\tilde{\mathbf{D}}_i = \mathbf{D} + \mathbf{E}_i$, where $\mathbf{E}_i$ is an interference matrix. For each noisy dictionary $\tilde{\mathbf{D}}_i$ this method computes an analytic matrix $\tilde{\mathbf{W}}_i$ that minimizes the mutual coherence $\mu(\tilde{\mathbf{W}}_i, \tilde{\mathbf{D}}_i)$. Then, $\tilde{\mathbf{W}}_i$ and $\mathbf{D}$ are embedded in the architecture, and the training is performed over the step sizes and the thresholds only, leading to a considerable reduction in the number of trained parameters.

While Robust-ALISTA considers model perturbations only, we show empirically that our method can handle more complicated model deviations, as dictionary permutations and totally random models. Additionally, in terms of computational complexity, robust-ALISTA has a complicated calculation of the analytic matrices during inference time, a limitation that does not exist in our scheme. We refer the reader to Appendix F for a more detailed discussion on the difference between both methods.

As to the theoretical aspect of our study, (Chen et al., 2018; Liu et al., 2019; Zarka et al., 2019; Wu et al., 2020) have recently shown that learned solvers can achieve linear convergence, under specific conditions on the sparsity level and mutual coherence. These results are the inspiration behind Theorem 1. This work, however, generalizes these guarantees to a varying model scenario, proving that the same weight matrix can serve different models while still reaching linear convergence.

5. Numerical Results

To demonstrate the effectiveness of our approach, we perform extensive numerical experiments, where our goal is two-fold. 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 input dictionaries. Second, we perform a natural image inpainting experiment, showcasing our robustness to a real-world task[2].

5.1. Synthetic Experiments

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, $\mathbf{y} = \mathbf{Dx}^*$. While the reported experiments in this section assume no additive noise, Appendix E 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 Gaussian coefficients, $\mathbf{x}^*_{i \in \text{support}} \in \mathcal{N}(0, 1)$. 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 methods is the MSE (Mean Square Error)

---

[2]The code for reproducing all experiments is available at https://github.com/aaberdam/AdaLISTA.
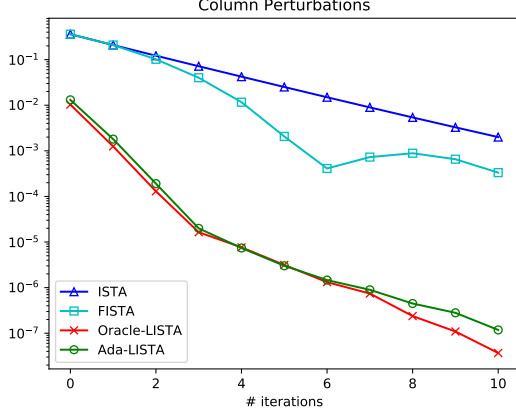
Figure 3: MSE performance under column permutations.

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 degradation/modifications to it.

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 $\tilde{\mathbf{D}}$. 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 $\tilde{\mathbf{D}}_i$ 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 $\mathbf{y}_i = \tilde{\mathbf{D}}\mathbf{x}_i^*$. 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 Figure 3 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 $\mathbf{D}$.

Noisy Dictionaries. In this experiment we aim to show that Ada-LISTA can handle a more challenging case in which the dictionary varies by $\tilde{\mathbf{D}}_i = \mathbf{D} + \mathbf{E}_i$. Each train-

ing 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 the 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.

Figure 4 presents the performance of the different solvers with a decreasing SNR (Signal to Noise Ratio) of the dictionary[3]. The performance of ISTA and FISTA is agnostic to the noisy model, since they do not require prior training. The Ada-LISTA again performs on-par with Oracle-LISTA, which has a prior knowledge of the dictionary. LISTA's performance, however, deteriorates with the decrease of the dictionary SNR. At SNR = 25dB it still provides a computational gain over ISTA and FISTA, but loses its advantage for lower SNRs and higher number of iterations.

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. For reference, we show the results of Oracle-LISTA in which all of the training signals are created from the same dictionary.

As can be seen in Figure 5, 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 (Gregor & LeCun, 2010). 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. The common belief is that 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

---

[3]Note that the noise is inflicted on the model (i.e., the dictionary) without an additive noise on the resulting signals
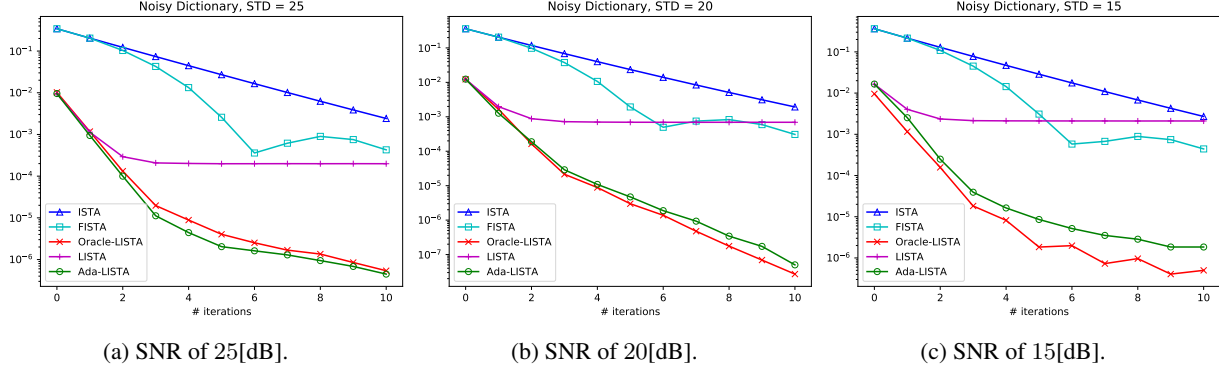
(a) SNR of 25[dB]. (b) SNR of 20[dB]. (c) SNR of 15[dB].

Figure 4: MSE performance for noisy dictionaries with decreasing SNR values.

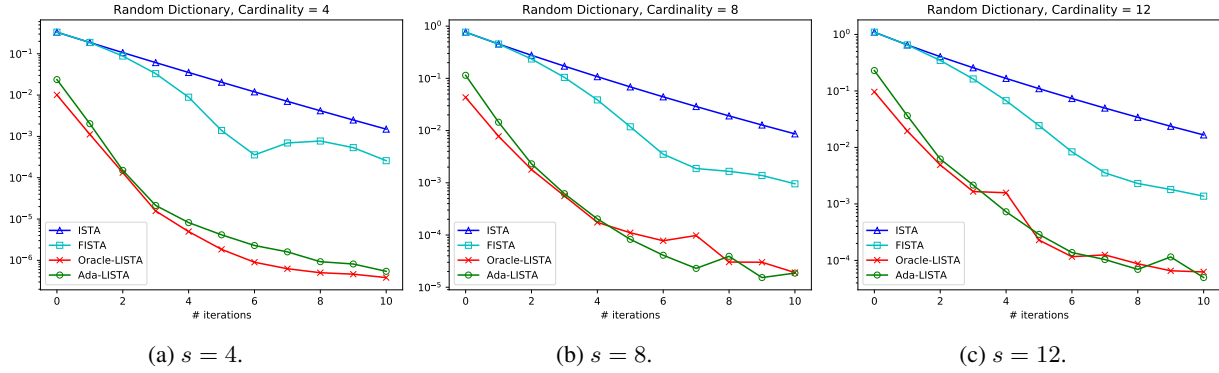(a) $s = 4$. (b) $s = 8$. (c) $s = 12$.

Figure 5: MSE performance for random dictionaries with increasing cardinality.

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 that this conditional information is the key for improved convergence.

5.2. Natural Image Inpainting

In this section 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. Thus, the updated objective we wish to solve is

$$\underset{\mathbf{x}}{\text{minimize}} \; \frac{1}{2}\|\mathbf{y} - \mathbf{MDx}\|_2^2 + \lambda\|\mathbf{x}\|_1, \quad (16)$$

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. Thus, the dictionary is constant, but each patch

has a different (yet known) inpainting mask, and thus the effective dictionary $\mathbf{D}_{\text{eff}} = \mathbf{MD}$ changes for each signal.

Updated Model. We slightly change the formulation of the model described in Section 2, and reverse the roles of the input and learned matrices. Specifically, the updated shrinkage step (Equation 3) for image inpainting is

$$\mathbf{x}_{k+1} = \mathcal{S}_{\frac{\lambda}{L}}\left(\mathbf{x}_k + \frac{1}{L}\mathbf{D}^T\mathbf{M}^T(\mathbf{y} - \mathbf{MDx}_k)\right). \quad (17)$$

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

$$\begin{aligned}
\frac{1}{L}\mathbf{D}^T\mathbf{M}^T\mathbf{MD} &\rightarrow \gamma_{k+1}\mathbf{W}_1^T\mathbf{M}^T\mathbf{MW}_1^T, \\
\frac{1}{L}\mathbf{D}^T\mathbf{M}^T &\rightarrow \gamma_{k+1}\mathbf{W}_2^T\mathbf{M}^T,
\end{aligned} \quad (18)$$

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.

Experiment Setting. In order to collect natural image patches, we use the BSDS500 dataset (Martin et al., 2001) and divide it to $400, 50$ and 50 training, validation and

Figure 6: Image inpainting with 50% missing pixels. From left to right: corrupted image, ISTA, FISTA, and Ada-LISTA.

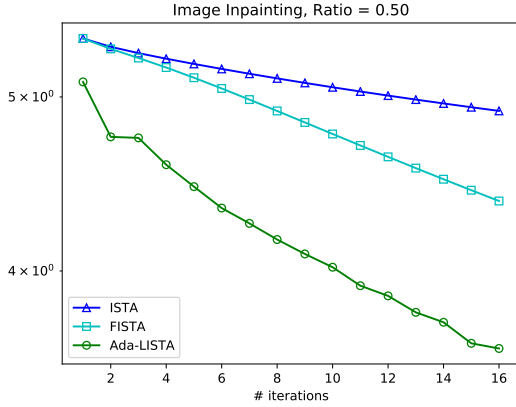|	Barbara	Boat	House	Lena	Peppers	C.man	Couple	Finger	Hill	Man	Montage
ISTA	23.49	25.40	26.87	27.83	23.56	22.72	25.34	20.63	27.26	26.34	22.48
FISTA	24.93	28.18	30.53	31.02	26.75	25.25	28.09	25.45	29.64	29.03	25.08
Ada-LFISTA	**26.09**	**30.03**	**32.36**	**32.50**	**28.81**	**27.94**	**30.02**	**28.25**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

Table 1: Image inpainting with 50% missing pixels and $K = 20$ unfoldings.

Figure 7: Patch-wise validation error versus unfoldings.

test images correspondingly. To train the dictionary $\mathbf{D}$, 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 Equation (18) 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. We extract all overlapping patches in each image, subtract the mean and divide by the standard deviation, apply each solver, un-normalize the patches and return their mean, and then place them in their correct position in the image and average over overlaps. The quality of the results 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 Figure 7; numerical results are given in Table 1, and select qualitative results are shown in Figure 6 and more in Appendix G. 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.

6. Conclusions

We have introduced a new extension of LISTA, termed Ada-LISTA, which receives both the signals and their dictionaries as input, and learns a universal architecture that can cope with the varying models. This modification produces 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.

References

Andrychowicz, M., Denil, M., Gomez, S., Hoffman, M. W., Pfau, D., Schaul, T., Shillingford, B., and De Freitas, N. Learning to learn by gradient descent by gradient descent. In *Advances in neural information processing systems*, pp. 3981–3989, 2016.

Beck, A. *First-order methods in optimization*, volume 25. SIAM, 2017.

Beck, A. and Teboulle, M. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1):183–202, 2009.

Borgerding, M., Schniter, P., and Rangan, S. Amp-inspired deep networks for sparse linear inverse problems. *IEEE Transactions on Signal Processing*, 65(16):4293–4308, 2017.

Boyd, S. and Vandenberghe, L. *Convex optimization*. Cambridge university press, 2004.

Chen, S. S., Donoho, D. L., and Saunders, M. A. Atomic decomposition by basis pursuit. *SIAM review*, 43(1): 129–159, 2001.

Chen, X., Liu, J., Wang, Z., and Yin, W. Theoretical linear convergence of unfolded ista and its practical weights and thresholds. In *Advances in Neural Information Processing Systems*, pp. 9061–9071, 2018.

Combettes, P. L. and Wajs, V. R. Signal recovery by proximal forward-backward splitting. *Multiscale Modeling & Simulation*, 4(4):1168–1200, 2005.

Daubechies, I., Defrise, M., and De Mol, C. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 57(11):1413–1457, 2004.

Giryes, R., Eldar, Y. C., Bronstein, A. M., and Sapiro, G. Tradeoffs between convergence speed and reconstruction accuracy in inverse problems. *IEEE Transactions on Signal Processing*, 66(7):1676–1690, 2018.

Golts, A., Freedman, D., and Elad, M. Deep energy: Using energy functions for unsupervised training of dnns. *arXiv preprint arXiv:1805.12355*, 2018.

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

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

Kulkarni, K., Lohit, S., Turaga, P., Kerviche, R., and Ashok, A. Reconnet: Non-iterative reconstruction of images from compressively sensed measurements. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 449–458, 2016.

Liu, J., Chen, X., Wang, Z., and Yin, W. ALISTA: Analytic weights are as good as learned weights in LISTA. In *International Conference on Learning Representations*, 2019.

Mairal, J., Elad, M., and Sapiro, G. Sparse representation for color image restoration. *IEEE Transactions on image processing*, 17(1):53–69, 2007.

Martin, D., Fowlkes, C., Tal, D., Malik, J., et al. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. Iccv Vancouver:, 2001.

Mazumder, R., Hastie, T., and Tibshirani, R. Spectral regularization algorithms for learning large incomplete matrices. *Journal of machine learning research*, 11(Aug): 2287–2322, 2010.

Metzler, C., Mousavi, A., and Baraniuk, R. Learned d-amp: Principled neural network based compressive image recovery. In *Advances in Neural Information Processing Systems*, pp. 1772–1783, 2017.

Moreau, T. and Bruna, J. Understanding trainable sparse coding via matrix factorization. *arXiv preprint arXiv:1609.00285*, 2016.

Protter, M. and Elad, M. Image sequence denoising via sparse and redundant representations. *IEEE Transactions on Image Processing*, 18(1):27–35, 2008.

Romano, Y., Elad, M., and Milanfar, P. The little engine that could: Regularization by denoising (red). *SIAM Journal on Imaging Sciences*, 10(4):1804–1844, 2017.

Sprechmann, P., Bronstein, A., and Sapiro, G. Learning efficient sparse and low rank models. *IEEE transactions on pattern analysis and machine intelligence*, 37(9):1821–1833, 2015.

Sulam, J., Aberdam, A., Beck, A., and Elad, M. On multi-layer basis pursuit, efficient algorithms and convolutional neural networks. *IEEE transactions on pattern analysis and machine intelligence*, 2019.

Sun, J., Li, H., Xu, Z., et al. Deep admm-net for compressive sensing mri. In *Advances in neural information processing systems*, pp. 10–18, 2016.

Tang, S., Gong, W., Li, W., and Wang, W. Non-blind image deblurring method by local and nonlocal total variation models. *Signal processing*, 94:339–349, 2014.

Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (methodological)*, 58:267288, 1996.

Tompson, J., Schlachter, K., Sprechmann, P., and Perlin, K. Accelerating eulerian fluid simulation with convolutional networks. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, pp. 3424–3433. JMLR. org, 2017.

Venkatakrishnan, S. V., Bouman, C. A., and Wohlberg, B. Plug-and-play priors for model based reconstruction. In *2013 IEEE Global Conference on Signal and Information Processing*, pp. 945–948. IEEE, 2013.

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

Wang, Z., Ling, Q., and Huang, T. S. Learning deep ℓ_0 encoders. In *Thirtieth AAAI Conference on Artificial Intelligence*, 2016.

Wu, K., Guo, Y., Li, Z., and Zhang, C. Sparse coding with gated learned {ista}. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=BygPO2VKPH.

Xin, B., Wang, Y., Gao, W., Wipf, D., and Wang, B. Maximal sparsity with deep networks? In *Advances in Neural Information Processing Systems*, pp. 4340–4348, 2016.

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

Zhang, J. and Ghanem, B. Ista-net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1828–1837, 2018.

Zhao, B., Fei-Fei, L., and Xing, E. P. Online detection of unusual events in videos via dynamic sparse coding. In *CVPR 2011*, pp. 3313–3320. IEEE, 2011.

A. Proof of Theorem 1

Proof. This proof follows the steps from (Zarka et al., 2019), 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 $\mathrm{Supp}(\cdot)$ as the support of a vector.

Induction hypothesis: For any iteration $k \geq 0$ the following hold

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

$$\mathrm{Supp}(\mathbf{x}_k) \subseteq \mathrm{Supp}(\mathbf{x}^*). \qquad (19)$$

2. The *recovery error* is bounded by

$$\|\mathbf{x}_k - \mathbf{x}^*\|_\infty \leq 2\theta_k. \qquad (20)$$

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 Equation 19 holds. As for the recovery error, we get that

$$\|\mathbf{x}_0 - \mathbf{x}^*\|_\infty = \|\mathbf{x}^*\|_\infty. \qquad (21)$$

Therefore, to verify Equation 20 we need to show that

$$\|\mathbf{x}^*\|_\infty \leq 2\theta_0 = 2\theta_{\max}. \qquad (22)$$

Since $\mathbf{y} = \mathbf{Dx}^* + \mathbf{e}$, for any index i we can write

$$\mathbf{y} = \mathbf{d}_i \mathbf{x}^*[i] + \sum_{j \neq i} \mathbf{d}_j \mathbf{x}^*[j] + \mathbf{e}, \qquad (23)$$

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

$$\mathbf{x}^*[i] \mathbf{a}_i^T \mathbf{d}_i = \mathbf{a}_i^T \mathbf{y} - \sum_{j \neq i} \mathbf{x}^*[j] \mathbf{a}_i^T \mathbf{d}_j - \mathbf{a}_i^T \mathbf{e}. \qquad (24)$$

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 $|\mathbf{a}_i^T \mathbf{d}_j|$ is bounded by $\widetilde{\mu}$, we get the following bound

$$|\mathbf{x}^*[i]| \leq |\mathbf{a}_i^T \mathbf{y}| + s\widetilde{\mu} \|\mathbf{x}^*\|_\infty + |\mathbf{a}_i^T \mathbf{e}|. \qquad (25)$$

By taking a maximum over i we obtain

$$(1 - s\widetilde{\mu}) \|\mathbf{x}^*\|_\infty \leq \|\mathbf{A}^T \mathbf{y}\|_\infty + \|\mathbf{A}^T \mathbf{e}\|_\infty. \qquad (26)$$

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

$$\|\mathbf{A}^T \mathbf{e}\|_\infty = \theta_{\min}(1 - 2\gamma\widetilde{\mu}s) < \theta_{\max}(1 - 2\gamma\widetilde{\mu}s), \qquad (27)$$

we get

$$(1 - s\widetilde{\mu}) \|\mathbf{x}^*\|_\infty \leq 2\theta_{\max}(1 - \gamma\widetilde{\mu}s). \qquad (28)$$

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

$$\|\mathbf{x}_0 - \mathbf{x}^*\|_\infty = \|\mathbf{x}^*\|_\infty \leq 2\theta_{\max}, \qquad (29)$$

as in Equation 20, 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 $\mathcal{I} \triangleq \text{Supp}(\mathbf{x}^*) - \{i\}$ and denote by $\mathbf{D}_{\mathcal{I}}$ the subset $\mathcal{I}$ of columns in $\mathbf{D}$.

We start by proving the support hypothesis (Equation 19). By Definition 2, the following holds for any index i:

$$\mathbf{x}_{k+1}[i] = \mathcal{S}_{\theta_{k+1}}(\mathbf{x}_k[i] + \mathbf{a}_i^T(\mathbf{y} - \mathbf{D}\mathbf{x}_k)). \quad (30)$$

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

$$\mathbf{x}_{k+1}[i] = \mathcal{S}_{\theta_{k+1}}(\mathbf{x}_k[i] + \mathbf{a}_i^T\mathbf{D}(\mathbf{x}^* - \mathbf{x}_k) + \mathbf{a}_i^T\mathbf{e}). \quad (31)$$

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, Equation 31 becomes

$$\mathbf{x}_{k+1}[i] = \mathcal{S}_{\theta_{k+1}}(\underbrace{\mathbf{x}^*[i] + \mathbf{a}_i^T\mathbf{D}_{\mathcal{I}}(\mathbf{x}^*[\mathcal{I}] - \mathbf{x}_k[\mathcal{I}]) + \mathbf{a}_i^T\mathbf{e}}_{\triangleq r}).$$
$$(32)$$

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

$$|r| \leq \left|\mathbf{a}_i^T\mathbf{D}_{\mathcal{I}}(\mathbf{x}^*[\mathcal{I}] - \mathbf{x}_k[\mathcal{I}])\right| + \left\|\mathbf{A}^T\mathbf{e}\right\|_\infty. \quad (33)$$

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

$$\left|\mathbf{a}_i^T\mathbf{D}_{\mathcal{I}}(\mathbf{x}^*[\mathcal{I}] - \mathbf{x}_k[\mathcal{I}])\right| \leq s\widetilde{\mu} \left\|\mathbf{x}^* - \mathbf{x}_k\right\|_\infty. \quad (34)$$

Using the induction assumption on the recovery error (Equation 20), we have $\|\mathbf{x}^* - \mathbf{x}_k\|_\infty \leq 2\theta_k$. Therefore, we get

$$|r| \leq 2s\widetilde{\mu}\theta_k + \left\|\mathbf{A}^T\mathbf{e}\right\|_\infty. \quad (35)$$

However, by our assumptions,

$$\left\|\mathbf{A}^T\mathbf{e}\right\|_\infty = \theta_{\min}(1 - 2\gamma\widetilde{\mu}s) < \theta_{k+1}(1 - 2\gamma\widetilde{\mu}s). \quad (36)$$

Therefore,

$$|r| \leq 2s\widetilde{\mu}\theta_k + \theta_{k+1}(1 - 2\gamma\widetilde{\mu}s), \quad (37)$$

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

$$|r| \leq \theta_{k+1}. \quad (38)$$

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 (Equation 20). We use the fact that for any scalar triplet, $(\mathbf{x}_1, \mathbf{x}_2, \theta)$, the soft threshold satisfies

$$|\mathcal{S}_\theta(\mathbf{x}_1 + \mathbf{x}_2) - \mathbf{x}_1| \leq \theta + |\mathbf{x}_2|. \quad (39)$$

Therefore, following Equation 32 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 $\text{Supp}(\mathbf{x}_k) \in \text{Supp}(\mathbf{x}^*)$, we have

$$\left|\mathbf{a}_i^T\mathbf{D}_{\mathcal{I}}(\mathbf{x}^*[\mathcal{I}] - \mathbf{x}_k[\mathcal{I}])\right| \leq 2s\widetilde{\mu}\theta_k. \quad (40)$$

Therefore, by using Equation 36 we get

$$|\mathbf{x}_{k+1}[i] - \mathbf{x}^*[i]| \leq \theta_{k+1} + 2s\widetilde{\mu}\theta_k + \theta_{k+1}(1 - 2\gamma\widetilde{\mu}s), \quad (41)$$

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

$$|\mathbf{x}_{k+1}[i] - \mathbf{x}^*[i]| \leq 2\theta_{k+1}. \quad (42)$$

By taking a maximum over i, we establish the recovery error hypothesis (Equation 20), concluding the proof. $\square$

B. 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 Appendix A, with the following two modifications: The mutual coherence $\widetilde{\mu}$ is replaced with $\bar{\mu}$, 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 (Equation 26) now becomes:

$$\left\|\mathbf{x}^*\right\|_\infty(1 - \epsilon_d - \bar{\mu}s) \leq \left\|\mathbf{A}^T\mathbf{y}\right\|_\infty + \left\|\mathbf{A}^T\mathbf{e}\right\|_\infty. \quad (43)$$

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

$$\left\|\mathbf{A}^T\mathbf{e}\right\|_\infty < \theta_{\max}(1 - 2\gamma\epsilon_d - 2\gamma\bar{\mu}s), \quad (44)$$

we get

$$\left\|\mathbf{x}^*\right\|_\infty(1 - \epsilon_d - \bar{\mu}s) \leq 2\theta_{\max}(1 - \gamma\epsilon_d - \gamma\bar{\mu}s). \quad (45)$$

As $\gamma > 1$, $\left\|\mathbf{x}^*\right\|_\infty < 2\theta_{\max}$, 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 $\bar{\mu}$. This is due to the fact that if $i \notin \text{Supp}(\mathbf{x}^*)$, 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 $i \in \text{Supp}(\mathbf{x}^*)$. Since $\mathbf{a}_i^T\mathbf{d}_i \neq 1$ we need to modify Equation 31:

$$\mathbf{x}_{k+1}[i] = \mathcal{S}_{\theta_{k+1}}\big(\mathbf{x}^*[i] + \mathbf{a}_i^T\mathbf{D}_{\mathcal{I}}(\mathbf{x}^* - \mathbf{x}_k)_{\mathcal{I}} + \mathbf{a}_i^T\mathbf{e}$$
$$+ (1 - \mathbf{a}_i^T\mathbf{d}_i)(\mathbf{x}_k[i] - \mathbf{x}^*[i])\big). \quad (46)$$

Using Equation 39 we get that $|\mathbf{x}_{k+1}[i] - \mathbf{x}^*[i]|$ is upper bounded by

$$\theta_{k+1} + \bar{\mu}s\|\mathbf{x}^* - \mathbf{x}_k\|_\infty + \|\tilde{\mathbf{A}}^T\mathbf{e}\|_\infty \\ + \left|(1 - \tilde{\mathbf{a}}_i^T\mathbf{d}_i)\right| |\mathbf{x}_k[i] - \mathbf{x}^*[i]|, \quad (47)$$

which in turn is upper bounded by

$$\theta_{k+1} + \bar{\mu}s2\theta_k + \theta_{k+1}(1 - 2\gamma\epsilon_d - 2\gamma\bar{\mu}s) + 2\epsilon_d\theta_k. \quad (48)$$

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

$$|\mathbf{x}_{k+1}[i] - \mathbf{x}^*[i]| \leq 2\theta_{k+1}. \quad (49)$$

Taking a maximum over i establishes the recovery error assumption, proving the induction hypothesis. $\quad\square$

C. 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.

D. 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,*

$$s = \|\mathbf{x}^*\|_0 < \frac{1}{2\bar{\mu}}, \quad \bar{\mu} \triangleq \tilde{\mu} + \tau_{\mathrm{od}}, \quad (50)$$

and the thresholds satisfy

$$\theta_k = \theta_{\max}\gamma^{-k} > \theta_{\min} = \frac{\|\tilde{\mathbf{A}}^T\mathbf{e}\|_\infty}{1 - 2\gamma\epsilon_d - 2\gamma\bar{\mu}s}, \quad (51)$$

with $1 < \gamma < (2\bar{\mu}s)^{-1}$, $\epsilon_d \triangleq w_{\mathrm{d}} + \tau_{\mathrm{d}} < \frac{1}{2}$, $w_{\mathrm{d}} \triangleq \frac{\sigma^2}{n}\sum_{k=1}^n W_{kk}$, $\tilde{\mathbf{A}} \triangleq \mathbf{W}\tilde{\mathbf{D}}$, and $\theta_{\max} \geq \|\tilde{\mathbf{A}}^T\mathbf{y}\|_\infty$, then, with probability of at least $(1 - p_1p_2)$, the support of the kth iteration of Ada-LISTA is included in the support of $\mathbf{x}^*$ and its values satisfy

$$\|\mathbf{x}_k - \mathbf{x}^*\|_\infty \leq 2\,\theta_{\max}\gamma^{-k}. \quad (52)$$

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

$$\bar{\mathbf{G}} = \tilde{\mathbf{G}} - \mathbf{G} = \mathbf{D}^T\mathbf{W}^T\mathbf{E} + \mathbf{E}^T\mathbf{W}^T\mathbf{D} + \mathbf{E}^T\mathbf{W}^T\mathbf{E}. \quad (53)$$

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

$$\bar{G}_{ij} = \underbrace{\sum_{k=1}^n\sum_{l=1}^n D_{ki}W_{lk}E_{lj}}_{\triangleq T_{ij}^a} + \underbrace{\sum_{k=1}^n\sum_{l=1}^n E_{ki}W_{lk}D_{lj}}_{\triangleq T_{ij}^b} \\ + \underbrace{\sum_{k=1}^n\sum_{l=1}^n E_{ki}W_{lk}E_{lj}}_{\triangleq T_{ij}^c}. \quad (54)$$

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

$$\mathbb{E}[\bar{G}_{ij}] = \begin{cases} \frac{\sigma^2}{n}\sum_{k=1}^n W_{kk}, & \text{if } i = j \\ 0, & \text{if } i \neq j. \end{cases} \quad (55)$$

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

$$\mathrm{Var}[\bar{G}_{ij}] = \mathbb{E}[T_{ij}^a]^2 + \mathbb{E}[T_{ij}^b]^2 + \mathbb{E}[T_{ij}^c - \mathbb{E}[\bar{G}_{ij}]]^2 + 2\mathbb{E}[T_{ij}^aT_{ij}^b] \\ + 2\mathbb{E}[T_{ij}^a(T_{ij}^c - \mathbb{E}[\bar{G}_{ij}])] + 2\mathbb{E}[T_{ij}^b(T_{ij}^c - \mathbb{E}[\bar{G}_{ij}])].$$

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

$$\mathbb{E}[T_{ij}^a]^2 = \mathbb{E}\Big[\sum_{k=1}^n\sum_{l=1}^n D_{ki}W_{lk}E_{lj}\sum_{k'=1}^n\sum_{l'=1}^n D_{k'i}W_{l'k'}E_{l'j}\Big] \\ = \frac{\sigma^2}{n}\mathbb{E}\Big[\underbrace{\sum_{k=1}^n\sum_{l=1}^n\sum_{k'=1}^n D_{ki}W_{lk}D_{k'i}W_{lk'}}_{\triangleq C_{ij}^a}\Big]. \quad (56)$$

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

$$
\begin{aligned}
\mathbb{E}[T_{ij}^b]^2 &= \mathbb{E}\Big[\sum_{k=1}^n \sum_{l=1}^n E_{ki} W_{lk} D_{lj} \sum_{k'=1}^n \sum_{l'=1}^n E_{k'i} W_{l'k'} D_{l'j} \Big] \\
&= \frac{\sigma^2}{n} \mathbb{E}\Big[\underbrace{ \sum_{k=1}^n \sum_{l=1}^n \sum_{l'=1}^n W_{lk} D_{lj} W_{l'k} D_{l'j} }_{\triangleq C_{ij}^b} \Big].
\end{aligned}
$$
$$(57)$$

As for $\mathbb{E}[T_{ij}^c - \mathbb{E}[\bar{G}_{ij}]]^2$, if $i \neq j$ then

$$
\begin{aligned}
\mathbb{E}\Big[\sum_{k=1}^n \sum_{l=1}^n &E_{ki} W_{lk} E_{lj} \sum_{k'=1}^n \sum_{l'=1}^n E_{k'i} W_{l'k'} E_{l'j} \Big] \\
&= \frac{\sigma^4}{n^2} \mathbb{E}\Big[\underbrace{ \sum_{k=1}^n \sum_{l=1}^n W_{lk}^2 }_{\triangleq C_{ij}^c} \Big].
\end{aligned}
$$
$$(58)$$

Whereas, if $i = j$, then $\mathbb{E}[T_{ij}^c - \mathbb{E}[\bar{G}_{ij}]]^2$ becomes

$$
\begin{aligned}
\mathbb{E}\Big[\sum_{k=1}^n \sum_{l=1\neq k}^n &E_{ki}^2 W_{lk}^2 E_{lj}^2 \Big] + \mathbb{E}\Big[\sum_{k=1}^n \sum_{l=1\neq k}^n E_{ki}^2 W_{lk} W_{kl} E_{lj}^2 \Big] \\
&+ \mathbb{E}\Big[\sum_{k=1}^n \sum_{k'=1\neq k}^n E_{ki} W_{kk} E_{kj} E_{k'i} W_{k'k'} E_{k'j} \Big] \\
&+ \mathbb{E}\Big[\sum_{k=1}^n E_{ki}^2 W_{kk}^2 E_{kj}^2 \Big] - \frac{\sigma^4}{n^2} \Big(\sum_{k=1}^n W_{kk} \Big)^2
\end{aligned}
$$
$$(59)$$

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

$$
\frac{\sigma^4}{n^2} \Big(\underbrace{ \sum_{k=1}^n \sum_{l=1\neq k}^n W_{lk}^2 + \sum_{k=1}^n \sum_{l=1\neq k}^n W_{lk} W_{kl} + 2 \sum_{k=1}^n W_{kk}^2 }_{\triangleq C_{ij}^d} \Big).
$$
$$(60)$$

Continuing with $2\mathbb{E}[T_{ij}^a T_{ij}^b]$, we get

$$
2\mathbb{E}[T_{ij}^a T_{ij}^b] = 2\mathbb{E}\Big[\sum_{k=1}^n \sum_{l=1}^n \sum_{l'=1}^n D_{ki} W_{lk} E_{lj} E_{li} W_{l'l} D_{l'j} \Big].
$$
$$(61)$$

Therefore, if $i = j$ then $\mathbb{E}[T_{ij}^a T_{ij}^b] = 0$, and if $i \neq j$ then

$$
2\mathbb{E}[T_{ij}^a T_{ij}^b] = \frac{\sigma^2}{n} 2 \underbrace{ \sum_{k=1}^n \sum_{l=1}^n \sum_{l'=1}^n D_{ki} W_{lk} W_{l'l} D_{l'j} }_{\triangleq C_{ij}^e}.
$$
$$(62)$$

As for $2\mathbb{E}[T_{ij}^a (T_{ij}^c - \mathbb{E}[\bar{G}_{ij}])]$, and $2\mathbb{E}[T_{ij}^b (T_{ij}^c - \mathbb{E}[\bar{G}_{ij}])]$, 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,

$$
v_{\text{od}} \triangleq \max_{i\neq j} \frac{\sigma^2}{n} \big(C_{ij}^a + C_{ij}^b \big) + \frac{\sigma^4}{n^2} C_{ij}^c,
$$
$$(63)$$

and the maximal variance of the diagonal elements as,

$$
v_{\text{d}} \triangleq \max_{i=j} \frac{\sigma^2}{n} \big(C_{ij}^a + C_{ij}^b + C_{ij}^e \big) + \frac{\sigma^4}{n^2} C_{ij}^d.
$$
$$(64)$$

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

$$
p(|\bar{G}_{ij}| \geq \tau_{\text{od}}) \leq \frac{2v_{\text{od}}^2}{v_{\text{od}}^2 + \tau_{\text{od}}^2}.
$$
$$(65)$$

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

$$
p(\max_{i,j\neq i} |\bar{G}_{ij}| \geq \tau_{\text{od}}) \leq p_1,
$$
$$(66)$$

with

$$
p_1 \triangleq 1 - \left(\frac{\tau_{\text{od}}^2 - v_{\text{od}}^2}{v_{\text{od}}^2 + \tau_{\text{od}}^2} \right)^{n(n-1)}.
$$
$$(67)$$

Moving on to the diagonal elements, we have

$$
p\left(\Big| \bar{G}_{ii} - \frac{\sigma^2}{n} \sum_{k=1}^n W_{kk} \Big| \geq \tau_{\text{d}} \right) \leq \frac{2v_{\text{d}}^2}{v_{\text{d}}^2 + \tau_{\text{d}}^2}.
$$
$$(68)$$

Taking the maximum over all diagonal elements, we get

$$
p\left(\max_i \Big| \bar{G}_{ii} - \frac{\sigma^2}{n} \sum_{k=1}^n W_{kk} \Big| \geq \tau_{\text{d}} \right) \leq p_2,
$$
$$(69)$$

with

$$
p_2 \triangleq 1 - \left(\frac{\tau_{\text{d}}^2 - v_{\text{d}}^2}{v_{\text{d}}^2 + \tau_{\text{d}}^2} \right)^n.
$$
$$(70)$$

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:

$$
\max_{i,j\neq i} \Big| \tilde{G}_{ij} \Big| \leq \tilde{\mu} + \tau_{\text{od}}.
$$
$$(71)$$

- The diagonal elements are close to 1:

$$
\max_i \Big| \tilde{G}_{ii} - 1 \Big| \geq w_{\text{d}} + \tau_{\text{d}}, \quad w_{\text{d}} \triangleq \frac{\sigma^2}{n} \sum_{k=1}^n W_{kk}.
$$
$$(72)$$

Finally, we apply Theorem 2 with the constants

$$
\bar{\mu} = \tilde{\mu} + \tau_{\text{od}}, \quad \epsilon_d = w_{\text{d}} + \tau_{\text{d}},
$$
$$(73)$$

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

$\square$

E. Synthetic Experiments on Noisy Signals

In this part we examine Ada-LISTA's performance for noisy signals by repeating the synthetic experiments in subsection 5.1, with three levels of input SNR: 10, 20, and 30[dB]. Figures 8, 9, and 10 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.

F. Comparison to Robust-ALISTA

A similar concept of robustness to model noise is suggested in "robust-ALISTA" (Liu et al., 2019), which models the clean dictionary $\mathbf{D}$ as having small perturbations of the form $\tilde{\mathbf{D}} = \mathbf{D} + \mathbf{E}$. This robustness is achieved in two stages, the first, computing a weight matrix $\tilde{\mathbf{W}}$ for each noisy model $\tilde{\mathbf{D}}$ by minimizing:

$$\tilde{\mathbf{W}} = \arg\min_{\mathbf{W}} \left\| \mathbf{W}^T \tilde{\mathbf{D}} \right\|_F^2, \quad \text{s.t.} \quad \mathbf{w}_i^T \tilde{\mathbf{d}}_i = 1, \ \forall i \in [1, m], \tag{74}$$

where $\mathbf{w}_i, \tilde{\mathbf{d}}_i$ are the ith columns of $\mathbf{W}$ and $\tilde{\mathbf{D}}$ respectively. Secondly, the matrix $\tilde{\mathbf{W}}$ is inserted into the ALISTA scheme:

$$\mathbf{x}_{k+1} = \mathcal{S}_{\theta_{k+1}} \left(\mathbf{x}_k - \gamma_{k+1} \tilde{\mathbf{W}}^T (\mathbf{D}\mathbf{x}_k - \mathbf{y}) \right), \tag{75}$$

where the step sizes and thresholds $\{\gamma_k, \theta_k\}$ are learned parameters. The advantage of this approach is the remarkably reduced number of trained parameters, $2K$. This method, however, suffers from several drawbacks. First, compared to Ada-LISTA, ALISTA is restricted to small model perturbations, and cannot handle more general scenarios, such as random dictionaries or even column perturbations. Second, in terms of computational complexity, robust-ALISTA has a complicated calculation of the analytic matrices during both training and inference (Equation 74), a limitation that does not exist in our scheme. Lastly, robust-ALISTA's training targets the original sparse representations that generated the signals. This makes ALISTA 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 Equation 2, and thus can be used for any generic problem, solvable with ISTA (Equation 4), e.g., low-rank matrix models (Sprechmann et al., 2015), acceleration of Eulerian fluid simulation (Tompson et al., 2017) and feature learning (Andrychowicz et al., 2016).

G. Image Inpainting Results

Figures 11, 12 present the qualitative inpainting results on the rest of the Set11 images, presented in subsection 5.2.
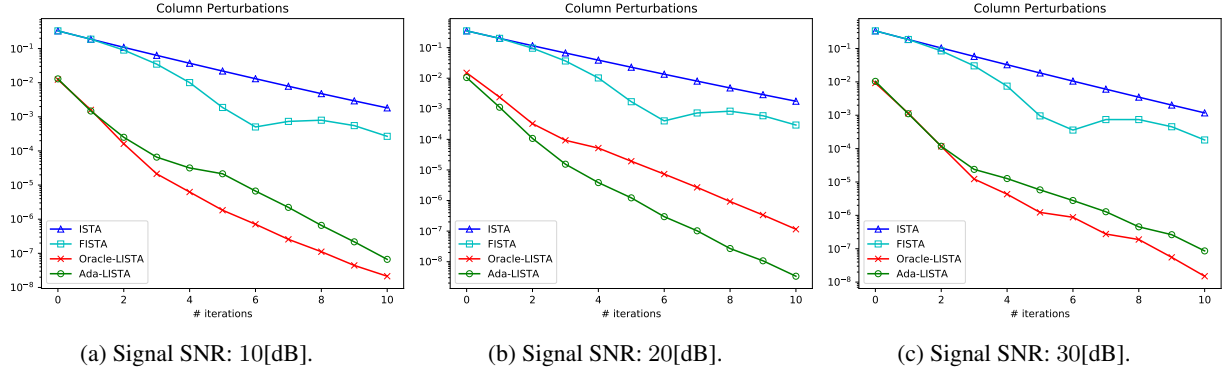
(a) Signal SNR: 10[dB]. (b) Signal SNR: 20[dB]. (c) Signal SNR: 30[dB].

Figure 8: MSE performance under column permutations and noisy inputs.

(a) Signal SNR: 10[dB]. (b) Signal SNR: 20[dB]. (c) Signal SNR: 30[dB].

Figure 9: MSE performance for noisy dictionaries and noisy inputs.

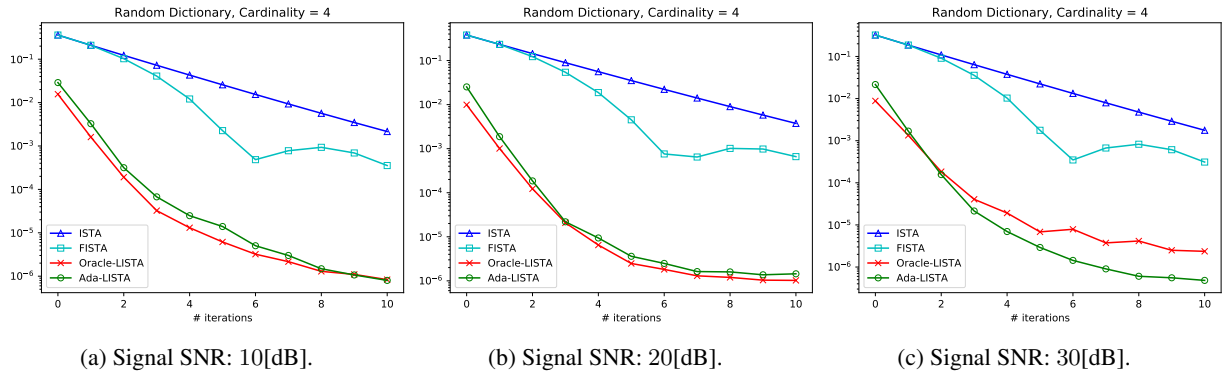(a) Signal SNR: 10[dB]. (b) Signal SNR: 20[dB]. (c) Signal SNR: 30[dB].

Figure 10: MSE performance under random dictionaries and noisy inputs.
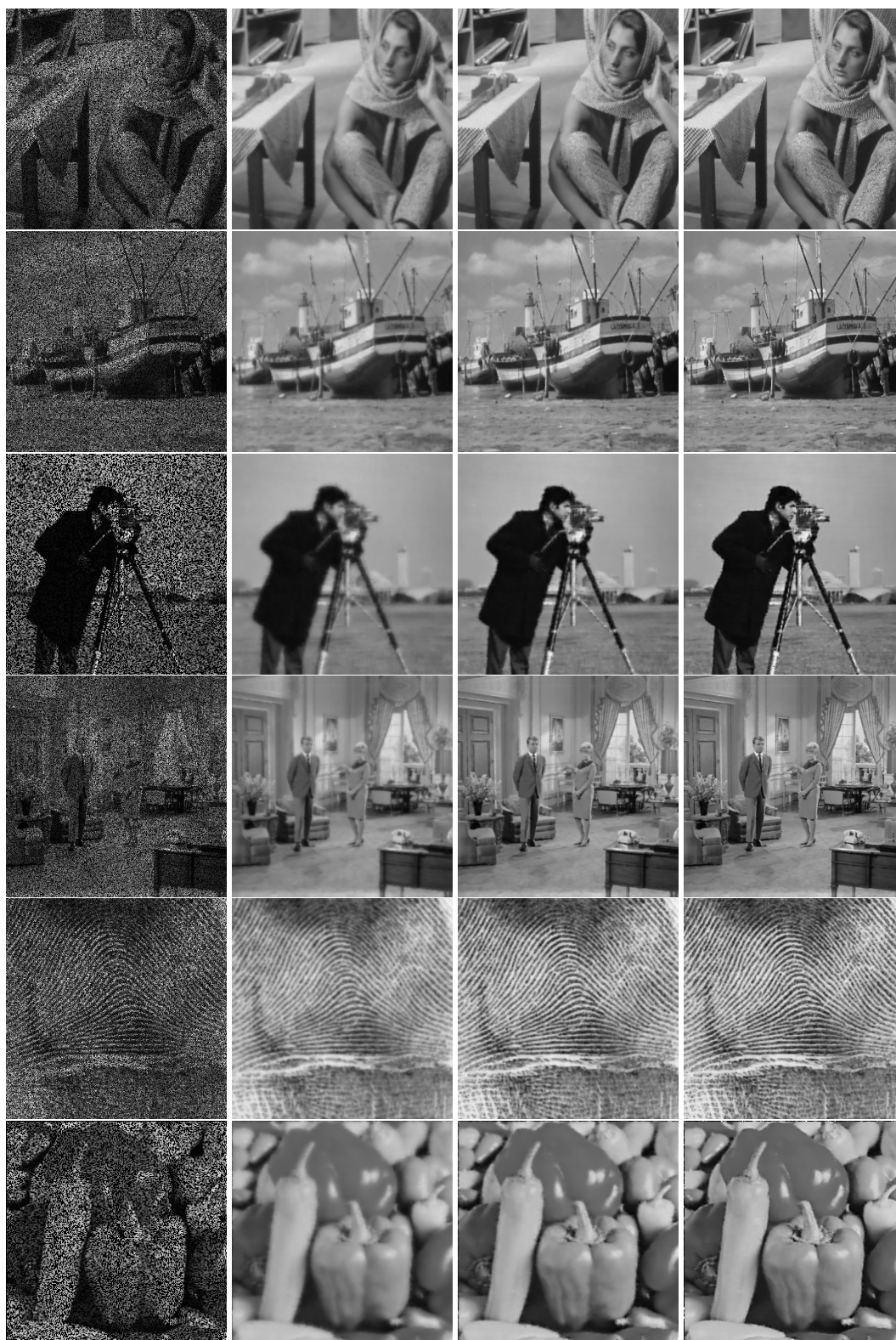
Figure 11: Image inpainting with 50% missing pixels. From left to right: corrupted image, ISTA, FISTA, and Ada-LISTA.

Figure 12: Image inpainting with 50% missing pixels. From left to right: corrupted image, ISTA, FISTA, and Ada-LISTA.