Learned Extragradient ISTA with Interpretable Residual Structures for Sparse Coding

Paper ID: 8787

Abstract

Recently, the study on learned iterative shrinkage thresholding algorithm (LISTA) has attracted increasing attentions. A large number of experiments as well as some theories have proved the high efficiency of LISTA for solving sparse coding problems. However, existing LISTA methods are all serial connection. To address this issue, we propose a novel extragradient based LISTA (ELISTA), which has a residual structure and theoretical guarantees. Moreover, most LISTA methods use the soft thresholding function, which has been found to cause large estimation bias. Therefore, we propose a thresholding function for ELISTA instead of soft thresholding. In the theoretical aspect, we prove that our method attains linear convergence. In addition, through ablation experiments, the improvements of our method on the network structure and the thresholding function are verified. Extensive empirical results verify the advantages of our method.

1 Introduction

In this paper, we mainly consider the following problem, which is to recover a sparse vector $x^* \in \mathbb{R}^n$ from an observation vector $y \in \mathbb{R}^m$ with noise $\varepsilon \in \mathbb{R}^m$ (e.g., additive Gaussian white noise):

$$y = Ax^* + \varepsilon, \tag{1}$$

where $A \in \mathbb{R}^{m\times n}$ ($m \ll n$) is the dictionary matrix. Generally, (1) is an ill-posed problem. Therefore, some prior information such as sparsity or low-rankness needs to be incorporated, for example, x^* is sparse, i.e., the number of elements of the support set of x^*, $S = \{i | x_i^* \neq 0\}$, is much smaller than the dimension n. A common way to estimate x^* is to solve the Lasso problem (Tibshirani 1996):

$$\min_{x\in\mathbb{R}^n} P(x) = f(x) + g(x) = \frac{1}{2}\|y - Ax\|_2^2 + \lambda\|x\|_1, \tag{2}$$

where $\lambda \geq 0$ is a regularization parameter. There are many methods for solving the problem of sparse coding, such as least angle regression (Efron et al. 2004), iterative shrinkage thresholding algorithm (ISTA) (Daubechies, Defrise, and De Mol 2004; Blumensath and Davies 2008) and approximate message passing (AMP) (Donoho, Maleki, and Montanari 2009). The update rule of ISTA is

$$x^{t+1} = \mathrm{ST}\left(x^t + \frac{1}{L}A^T(y - Ax^t), \frac{\lambda}{L}\right), \quad t = 0, 1, 2, ..., \tag{3}$$

where $\mathrm{ST}(\cdot, \theta)$ is the soft-thresholding function (ST) with the threshold θ, $\frac{1}{L}$ is the step size which should be taken in $(0, \frac{2}{L})$, where L is usually taken as the largest eigenvalue of A^TA, and $A^T(Ax^t - y)$ is actrually equal to $\nabla f(x^t)$.

ISTA converges slowly with only a sublinear rate (Beck and Teboulle 2009). Inspired by ISTA and Deep Neural Networks (DNNs) (LeCun, Bengio, and Hinton 2015), Gregor and LeCun (2010) viewed ISTA as a recurrent neural network (RNN) and proposed a learning-based model named Learned ISTA (LISTA):

$$x^{t+1} = \mathrm{ST}(W_1^t y + W_2^t x^t, \theta^t), \quad t = 0, 1, 2, ..., \tag{4}$$

where W_1^t, W_2^t and θ^t are initialized as $\frac{1}{L}A^T$, $I - \frac{1}{L}A^TA$ and $\frac{\lambda}{L}$, respectively. All the parameters $\Theta = \{W_1^t, W_2^t, \theta^t\}$ are learnable and data-driven. Many empirical and theoretical results (Aberdam, Golts, and Elad 2020; Giryes et al. 2018) have shown that T-layer LISTA or its variants can recover x^* from y more accurately and use one or two order-of-magnitude fewer iterations than the original ISTA. Moreover, the CSC version of LISTA can be used to explain the CNN in series (Papyan, Romano, and Elad 2017).

On one hand, inspired by (Gregor and LeCun 2010), many learnable network methods such as (Wang, Ling, and Huang 2016; Sprechmann, Bronstein, and Sapiro 2015; Ito, Takabe, and Wadayama 2019; Borgerding, Schniter, and Rangan 2017; Sreter and Giryes 2018) have been proposed and successfully used in different fields, and got satisfactory experimental results.

On the other hand, many works (Xin et al. 2016; Giryes et al. 2018; Moreau and Bruna 2017; Chen et al. 2018; Liu et al. 2019; Wu et al. 2020; Ablin et al. 2019) discussed LISTA and its variants from a theoretical point of view. Among them, Xin et al. (2016) first discussed LIHT (Wang, Ling, and Huang 2016), which was obtained by unfolding the iterative hard thresholding (IHT) (Blumensath and Davies 2009) inspired by (Gregor and LeCun 2010), in terms of improving the restricted isometry property (RIP) constant. Inspired by (Xin et al. 2016), He et al. (2017)

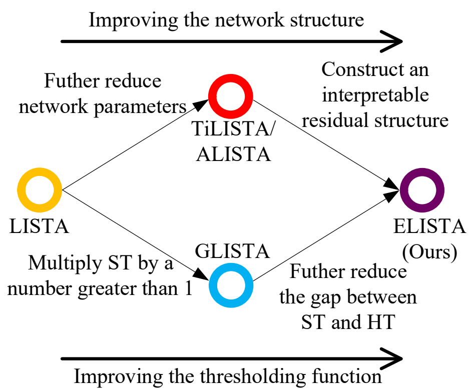


Figure 1: Subsequent improvements on LISTA

connected sparse Bayesian learning (SBL) (Tipping 2001) with long short-term memory (LSTM) (Gers, Schraudolph, and Schmidhuber 2002), and Moreau and Bruna (2017) explained the mechanism of LISTA by re-factorizing the Gram matrix of dictionary. Other works (Chen et al. 2018; Liu et al. 2019; Wu et al. 2020; Ablin et al. 2019) related to this paper will be detailed in Section 1.1.

A series of studies on LISTA have attracted increasing attentions and inspired many subsequent works in different aspects, including learning based optimization (Xie et al. 2019; Sun et al. 2016), design of DNNs (Metzler, Mousavi, and Baraniuk 2017; Zhang and Ghanem 2018; Zhou et al. 2018; Chen et al. 2020; Rick Chang et al. 2017; Zhang et al. 2020; Simon and Elad 2019) and interpreting the DNNs (Zarka et al. 2020; Papyan, Romano, and Elad 2017; Aberdam, Sulam, and Elad 2019; Sulam et al. 2018, 2019).

1.1 Related Works

Chen et al. (2018) proved the coupling relationship between W_1^t and W_2^t, i.e., $W_2^t \rightarrow (I - W_1^t A)$ when $t \rightarrow \infty$, which greatly reduced the number of learnable parameters of LISTA. They also first provided the rigorous proof of the linear convergence of LISTA, which is the basis of the subsequent works. Moreover, the subsequent improvements of LISTA can be divided into two categories: improvements of the network structure and the thresholding function.

For the improvement of the network structure, Liu et al. (2019) further reduced the number of learnable parameters by proposing a noval algorithm, whose update rule is $x^{t+1} = \mathrm{ST}(x^t - \alpha^t W(Ax^t - y), \theta^t)$, where α^t is a learnable scaler. They proposed TiLISTA when W is a learnable parameter and ALISTA when W is obtained by solving a data-independent optimization problem. For the improvement of the thresholding function, Wu et al. (2020) argued that the code components in LISTA estimations may be lower than expected, i.e., the algorithms require gains. Inspired by gated recurrent unit (GRU) (Cho et al. 2014; Chung et al. 2015), Wu et al. (2020) proposed GLISTA, which can be viewed as multiplying ST by a coefficient greater than 1 to reduce the gap between ST and HT. All the improvements of LISTA in different aspects above are shown in Figure 1, where ELISTA is an innovative algorithm proposed in this paper, which will be described in detail in Section 2.

Moreover, Ablin et al. (2019) also discussed LISTA from the theoretical aspect. They proposed a simple step size strategy which can improve the convergence rate of ISTA by leveraging the space of the iterates, and presented a network named SLISTA to learn only the step size of ISTA for unsupervised training.

1.2 Motivations and Main Contributions

We attempt to answer the following questions, which are not fully addressed in literature yet:

• All the existing variants of LISTA with convergence proofs are serial, the residual network (Res-Net) (He et al. 2016), which is influential in deep learning has not been introduced into LISTA. An important reason is that changing the original structure of LISTA will destroy its excellent mathematical interpretability. Can we get a LISTA with a interpretable residual structure, which has a convergence guarantee?

• Recent studies (Fan and Li 2001; Gu, Wang, and Liu 2014; Xu and Gu 2016; Zhu and Gu 2015; Lederer 2013; Deledalle et al. 2017) have shown that ST may cause large estimation bias, and incurs worse empirical performance than the hard-thresholding function (HT), which means there are some limitations by using ST for sparse coding. Can we improve the thresholding function to reduce the gap between ST and HT?

Our Main Contributions: The main contributions of this paper are as follows:

• We propose a novel variant of LISTA with residual structure by introducing the idea of extragradient into LISTA and establishing the relationship with Res-Net, which is an improvment about the network structure for solving the sparse coding problem. To the best of our knowledge, this is the first residual structure LISTA with theoretical guarantee.

• We design a new thresholding function, called Multistage-Thresholding function (MT), to reduce the gap between ST and HT. A large number of experiments show that MT can ensure the sparsity of the representation as low as possible and obtain effective sparse representation.

• Using extragradient and the MT operator, we propose a novel algorithm, named Extragradient based LISTA (ELISTA), and prove the convergence of ELISTA. Moreover, we conduct ablation experiments to verify the effectiveness of each of our improvements. Extensive experimental results show our ELISTA is superior to the state-of-the-art methods.

2 Extragradient Based LISTA and Multistage-thresholding

In this section, we first introduce the idea of extragradient into LISTA. Then we propose a new multistage-thresholding function (MT) and analyze its advantages. Finally, by combining the idea of extragradient and MT, we propose an innovative algorithm, named *Extragradient based LISTA* (ELISTA), and depict it in detail. Moreover, we also establish the relationship between ELISTA and Res-Net, which is one of the reasons why our algorithm is advantageous.

2.1 Extragradient Method

We note that iterative algorithms, such as ISTA, can actually be treated as a proximal gradient descent method, which is a first-order optimization algorithm, for special objective functions. Thus, we want to introduce the idea of extragradient into the related iterative algorithms. The extragradient method was first proposed by (Korpelevich 1976), which is a classical method for variational inequality problems. For optimization problems, the idea of extragradient was first used in (Nguyen et al. 2018), which proposed an extended extragradient method (EEG) by combining this idea with some first-order descent methods. In the t-th iteration of EEG, it first calculates the gradient at x^t, and updates x^t according to the gradient to get a middle point $x^{t+\frac{1}{2}}$, then calculates the gradient at $x^{t+\frac{1}{2}}$, and updates the original point x^t according to the gradient at the middle point $x^{t+\frac{1}{2}}$ to obtain x^{t+1}, which is the key idea of extragradient. Intuitively, the additional step in each iteration of EEG allows us to examine the geometry of the problem and consider its curvature information, which is one of the most important bottlenecks for first-order methods. Thus, by using the idea of extragradient, we can get a better result after each iteration. The update rules of EEG for Problem (2) can be rewritten as follows:

$$\begin{aligned} x^{t+\frac{1}{2}} &= \mathrm{ST}\left(x^t - \tfrac{1}{L}A^T(Ax^t - y), \tfrac{\lambda}{L}\right), \\ x^{t+1} &= \mathrm{ST}\left(x^t - \tfrac{1}{L}A^T(Ax^{t+\frac{1}{2}} - y), \tfrac{\lambda}{L}\right). \end{aligned} \tag{5}$$

This form of EEG is similar to ISTA, thus it can be regarded as a generalization of ISTA.

2.2 Multistage-thresholding

The nonlinear transformations in most LISTA related algorithms are realized by the standard ST. However, according to its definition, we know that ST has a weakness, i.e., $|x_i^t|$ obtained from the algorithms with ST is actually smaller than the real $|x_i^*|$, which was described by Proposition 1 in (Wu et al. 2020) and alleviated by (Wu et al. 2020) with the proposal of a gain gate (GG) and an algorithm called GLISTA, whose update rule is as follows:

$$x^{t+1} = \mathrm{ST}(W^t(g_t(x^t, y|\Lambda_g^t) \odot x^t) + U^t y, b^t),$$

where $g_t(x^t, y|\Lambda_g^t)$ is the gate function and greater than 1, and Λ_g^t is the set of its parameters to learn. Besides, W^t, U^t and b^t are also learnable parameters. We define $\tilde{x}^t \triangleq g_t(x^t, y|\Lambda_g^t) \odot x^t$, and obtain

$$\tilde{x}^{t+1} = g_{t+1}(x^{t+1}, y|\Lambda_g^{t+1}) \odot \mathrm{ST}(W^t \tilde{x}^t + U^t y, b^t),$$

which means that GLISTA multiplies ST by a number greater than 1, thus reducing the gap between ST and HT. Therefore, GLISTA can be treated as an improvement of ST. However, the proposal of GG in (Wu et al. 2020) is based on the assumption that there is no "false positive", which is not always true in reality. Therefore, GLISTA will increase some values that should be decreased, which will bring bad results. To address the issue, we design and propose an innovative thresholding function called *Multistage-Thresholding function (*MT*)*, which is defined as follows:

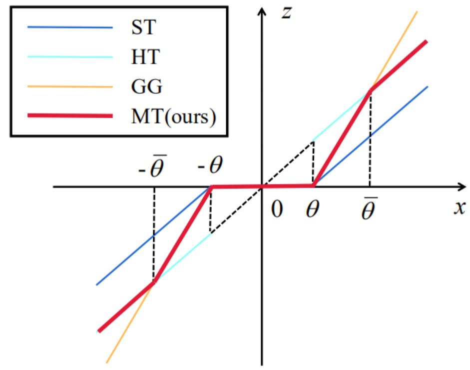


Figure 2: Different thresholding functions

$$z = \mathrm{MT}(x, \theta, \bar{\theta}) \triangleq \begin{cases} 0, & 0 \le |x| < \theta, \\ \frac{\bar{\theta}}{\bar{\theta}-\theta}\mathrm{sign}(x)(|x| - \theta), & \theta \le |x| < \bar{\theta}, \\ x, & |x| \ge \bar{\theta}. \end{cases} \tag{6}$$

Different thresholding functions are shown in Figure 2, from which we know that MT is equal to GG when $0 \le |x| < \bar{\theta}$, which plays the role of gain to ST, and when $|x| \ge \bar{\theta}$, it is equal to HT, which makes the result more accurate. Therefore, compared with other thresholding functions, MT can get a better result at each layer.

Our MT is similar to the function $\mathrm{HELU}_\sigma(\cdot)$ proposed in (Wang, Ling, and Huang 2016). However, the motivation of its proposal and the internal mathematical mechanism are different from those of MT. We will give detailed explanations and verifications in the Supplementary Material.

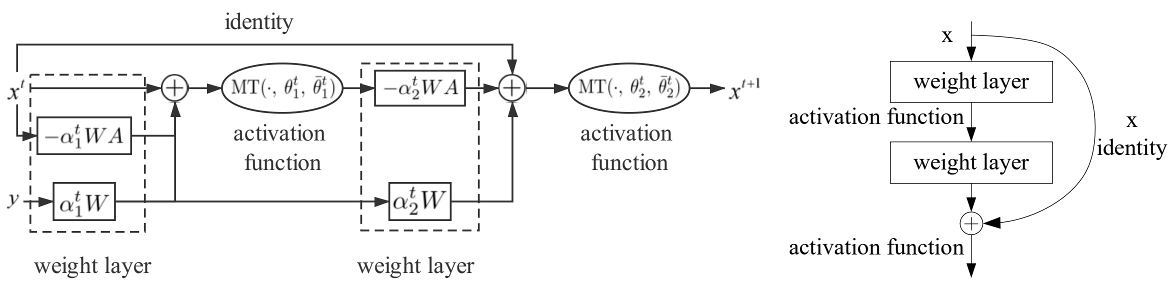
2.3 Extragradient Based LISTA and the Relationship with Res-Net

In order to speed up the convergence of EEG, we combine the algorithm with deep networks and regard $\frac{1}{L}A^T$ and two thresholds of two steps in (5) as learnable parameters, and get the following update rules:

$$\begin{aligned} x^{t+\frac{1}{2}} &= \mathrm{ST}(x^t - W_1^t(Ax^t - y), \theta_1^t), \\ x^{t+1} &= \mathrm{ST}(x^t - W_2^t(Ax^{t+\frac{1}{2}} - y), \theta_2^t). \end{aligned} \tag{7}$$

However, since the above scheme has two different matrices W_1^t and W_2^t to learn in each layer, the number of network parameters greatly increases and the training of the network slows down significantly. Therefore, to address this issue and further establish the connection between the two steps of (7), we convert W_1^t and W_2^t into $\alpha_1^t W^t$ and $\alpha_2^t W^t$, respectively, where α_1^t and α_2^t are two scalars to learn. Then, inspired by (Liu et al. 2019), we change the W^t of each layer into the same W and get a tied algorithm, which can significantly reduce the number of learnable parameters. By replacing ST with MT, we finally obtain the following update rules for our *Extragradient Based LISTA* (ELISTA):

$$\begin{aligned} x^{t+\frac{1}{2}} &= \mathrm{MT}(x^t - \alpha_1^t W(Ax^t - y),\ \theta_1^t,\ \bar{\theta}_1^t), \\ x^{t+1} &= \mathrm{MT}(x^t - \alpha_2^t W(Ax^{t+\frac{1}{2}} - y),\ \theta_2^t,\ \bar{\theta}_2^t), \end{aligned} \tag{8}$$



(a) ELISTA: a building layer. (b) Res-Net: a building block.

Figure 3: Comparison of the network structures of ELISTA and Res-Net.

Table 1: Comparison of the number of parameters to learn in different methods.

LISTA	LAMP	GLISTA	ELISTA-m-t	ELISTA-m	ELISTA-t	ELISTA
$\mathcal{O}(TMN+T)$	$\mathcal{O}(TMN+T)$	$\mathcal{O}(TMN+T)$	$\mathcal{O}(TMN+T)$	$\mathcal{O}(MN+T)$	$\mathcal{O}(TMN+T)$	$\mathcal{O}(MN+T)$

where $\bar{\theta}_1^t$ and $\bar{\theta}_2^t$ are also learnable parameters.

In order to make the algorithms in this paper easy to distinguish, we present the following naming system:

ELISTA is our main algorithm, which is obtained by introducing the idea of extragradient into LISTA and using MT, and it is a tied algorithm. It should be emphasized that we use +m or -m to represent using MT or not, and -t to indicate that the algorithm is untied. For example, ELISTA-m means ELISTA using ST instead of MT.

Besides, according to (8), we can get the network structure diagram of ELISTA. Through our observation and comparison, we find that the network structure of ELISTA is corresponding to the Res-Net. Since y is already given, we can regard y as a bias. Thus, from Figure 3, we can see that the structure of the network obtained by ELISTA is the same as that of Res-Net, including weight layer, activation function and identity. As we all know, Res-Net can obtain a better performance by improving the network structure. Therefore, it is meaningful to discuss and study the explanation for the internal mathematical mechanism of Res-Net. On the one hand, to some extend, our algorithm may be regarded as a mathematical explanation of the reason for the superiority of Res-Net. On the other hand, the connection and combination of ELISTA and Res-Net might be able to explain why our algorithm has better performance than existing methods.

Moreover, the comparison on the number of parameters of the network corresponding to different algorithms is shown in Table 1, where LAMP (Borgerding, Schniter, and Rangan 2017) is an algorithm to transform AMP (Donoho, Maleki, and Montanari 2009) into a neural network inspired by (Gregor and LeCun 2010).

3 Convergence Analysis

In this section, we provide the convergence analysis of our algorithms. We first give a basic assumption and two useful definitions. Then we provide the convergence property of ELISTA, and that of ELISTA-t is similar. We note that our analysis, like that of Theorems 3 and 4 of (Wu et al. 2020), is proved under the existence of "false positive", while the theoretical analysis of (Chen et al. 2018; Liu et al. 2019) was provided under the assumption of no "false positive", which is difficult to satisfy in reality.

Assumption 1 (Basic assumption)**.** *The signal x^* is sampled from the following set:*

$$x^* \in \mathcal{X}(B,s) \overset{\triangle}{=} \{x^* \| x_i^*| \leq B, \forall i, \|x^*\|_0 \leq s\}.$$

In other words, x^* is bounded and s-sparse ($s \geq 2$). Furthermore, we assume $\varepsilon = 0$.

We note that this assumption is a basic assumption for this class of algorithms. Almost all the related algorithms need to satisfy this assumption, for example (Liu et al. 2019; Wu et al. 2020).

Definition 1 ((Liu et al. 2019))**.** *Given a matrix $A \in \mathbb{R}^{m \times n}$, its generalized mutual coherence is defined as follows:*

$$\mu(A) = \inf_{W \in \mathbb{R}^{n \times m}, W_{i,:}A_{:,i}=1, \forall i} \left\{ \max_{i \neq j, 1 \leq i,j \leq n} W_{i,:}A_{:,j} \right\}.$$

We let $\mathcal{W}(A)$ denote a set of all matrices with the generalized mutual coherence $\mu(A)$, which means that

$$\mathcal{W}(A) = \left\{ W | \max_{i \neq j, 1 \leq i,j \leq n} W_{i,:}A_{:,j} = \mu(A), W_{i,:}A_{:,i} = 1, \forall i \right\}.$$

A weight matrix W is "good" if $W \in \mathcal{W}(A)$.

This definition is also described in Definition 1 in (Liu et al. 2019). From Lemma 1 in (Chen et al. 2018), we know $\mathcal{W}(A) \neq \varnothing$.

Definition 2. *Given a model with Θ, in which*

$$\theta_1^t = \Gamma \mu(A) \sup_{x^*} \|x^t - x^*\|_1,\ \theta_2^t = \Gamma \mu(A) \sup_{x^*} \|x^{t+\frac{1}{2}} - x^*\|_1,$$

Table 2: The results of ablation experiments. We use +m or -m to represent using MT or not, and -t to indicate that the algorithm is untied.

		Verify the network structure			Verify the thresholding function		Ours	
	LISTA	TiLISTA	ELISTA-m-t	ELISTA-m	GLISTA	LISTA+m	ELISTA-t	ELISTA
NMSE	-36.01	-50.28	-51.82	-65.66	-63.73	-62.21	-77.03	$\mathbf{-107.48}$
FLSNE	0.16	0.02	0.10	0.02	0.02	0.12	0.04	**0.00**
SPERR	147.12	46.26	3.23	2.35	57.22	0.80	0.15	**0.01**

we use $\omega_{t+\frac{1}{2}}(k_{t+\frac{1}{2}}|\Theta)$ and $\omega_{t+1}(k_{t+1}|\Theta)$ to characterize its relationship with the "false positive" rate, which is

$$\begin{aligned}
&\omega_{k+\frac{1}{2}}(k_{t+\frac{1}{2}}|\Theta)\\
&=\sup\nolimits_{\forall x^*,|supp(\check{x}^{t+\frac{1}{2}})\bigcup supp(x^*)|\leq|supp(x^*)|+k_{t+\frac{1}{2}}}\Gamma,\\
&\omega_{k+1}(k_{t+1}|\Theta)\\
&=\sup\nolimits_{\forall x^*,|supp(\check{x}^{t+1})\bigcup supp(x^*)|\leq|supp(x^*)|+k_{t+1}}\Gamma,
\end{aligned}$$

where $\check{x}^{t+\frac{1}{2}} := \mathrm{MT}((I-\alpha_1^t WA)(x^{t+\frac{1}{2}}-x^*),\theta_1^t)$, $\check{x}^{t+1} := \mathrm{MT}((I-\alpha_2^t WA)(x^{t+1}-x^*),\theta_2^t)$ and $k_{t+\frac{1}{2}}$ and k_{t+1} are the desired maximal number of "false positive" of $x^{t+\frac{1}{2}}$ and x^{t+1}, respectively.

This definition is similar to Definition 2 in (Wu et al. 2020). Besides, this definition is only an example for ELISTA. For our ELISTA-t, we can also easily get a similar definition.

Based on the assumption and these two definitions, we can get the linear convergence of ELISTA, which can be given by the following theorem.

Theorem 1 (Linear Convergence for ELISTA)**.** *If Assumption 1 holds, $W\in\mathcal{W}(A)$ can be satisfied by selecting W properly,*

$$\begin{aligned}
\theta_1^t&=\alpha_1^t\omega_{t+\frac{1}{2}}(k_{t+\frac{1}{2}}|\Theta)\mu(A)\sup\nolimits_{x^*}\|x^t-x^*\|_1,\\
\theta_2^t&=\alpha_2^t\omega_{t+1}(k_{t+1}|\Theta)\mu(A)\sup\nolimits_{x^*}\|x^{t+\frac{1}{2}}-x^*\|_1,
\end{aligned}\tag{9}$$

$\bar{\theta}_1^t\geq\theta_1^t+|\tilde{x}_i^{t+\frac{1}{2}}|$, $\bar{\theta}_2^t\geq\theta_2^t+|\tilde{x}_i^{t+1}|$ are achieved, α_1^t and α_2^t belong to specific bounded intervals for different cases, and s is small enough, then for the sequences generated by ELISTA, the following result holds

$$\|x^t-x^*\|_2\leq sB\exp\left(\sum_{i=1}^{t}c_i^{'}\right)<sB\exp(c^{'}t),\tag{10}$$

where $c^{'}=\max_{i=1,2,\ldots,t}\{c_i^{'}\}$. $\exists t_0=\lceil-\log(\frac{sB}{\sigma})/c\rceil$, for $i\geq t_0$, $0<k_{i-\frac{1}{2}},k_i<s$, if $\gamma^{i-\frac{1}{2}}=\gamma^i=0$, then $c_i^{'}<0$, and for $i>t_0$, $k_{i-\frac{1}{2}}=k_i=0$, if $1-\omega_{i-\frac{1}{2}}(s|\Theta)<\gamma^{i-\frac{1}{2}}\leq 1$ and $1-\omega_i(s|\Theta)<\gamma^i\leq 1$, then $c_i^{'}<0$. Thus, $c^{'}<0$.

The definitions of $\gamma^{i-\frac{1}{2}}$ and γ^i are given in the detailed proof of this theorem in the Supplementary Material. Here we give a simple sketch of the full proof:

To prove Theorem 1, we first need to obtain the relationship between $\|x^{t+1}-x^*\|_2$ and $\|x^t-x^*\|_2$. To calculate all non-zero elements of $x^{t+\frac{1}{2}}-x^*$, we divide them into three parts: $i\in\bar{S}^{(t+\frac{1}{2})}$, $i\in S\backslash\bar{S}^{(t+\frac{1}{2})}$ and $i\in S^{(t+\frac{1}{2})}$, where $S\triangleq supp(x^*)$, $\bar{S}^{(t+\frac{1}{2})}\triangleq S\cap supp(x^{t+\frac{1}{2}})$ and $S^{(t+\frac{1}{2})}\triangleq\{i|i\in supp(x^{t+\frac{1}{2}}), i\notin S\}$, and then sum the results to obtain the relationship between $\|x^{t+\frac{1}{2}}-x^*\|_1$ and $\|x^t-x^*\|_1$. In a similar way, we can get the relationship between $\|x^{t+1}-x^*\|_1$, $\|x^{t+\frac{1}{2}}-x^*\|_1$ and $\|x^t-x^*\|_1$. Then, we can obtain the relationship between $\|x^{t+1}-x^*\|_1$ and $\|x^t-x^*\|_1$, and thus the relationship between $\|x^{t+1}-x^*\|_2$ and $\|x^t-x^*\|_2$. Finally, Theorem 1 can be proved by the recursion in terms of t.

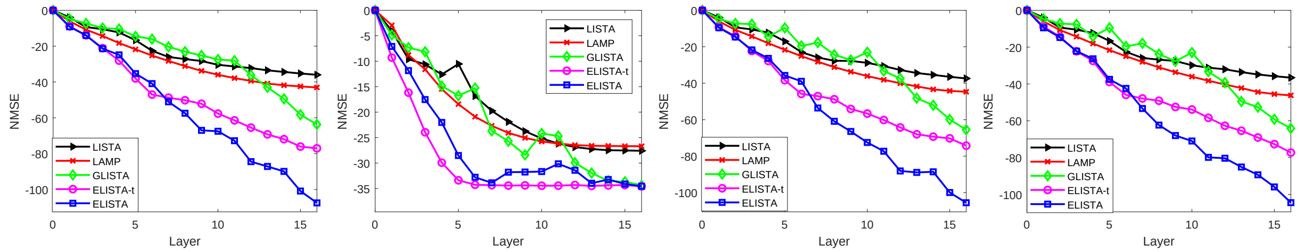
4 Numerical Results

In this section, we first perform ablation experiments to verify the effectiveness of our method and provide the justification of some parameters in the algorithms and the verification of an assumption. Then we evaluate our ELISTA and ELISTA-t in terms of sparse representation performance, natural image inpainting, 3D geometry recovery via photometric stereo, support set accuracy and unsupervised experiment as in (Ablin et al. 2019). All the experimental settings are the same as previous works (Chen et al. 2018; Liu et al. 2019; Wu et al. 2020; Borgerding, Schniter, and Rangan 2017). We find that Support Selection (SS) (Chen et al. 2018) can generally improve the performance of related networks including ours. However, the performance of SS is greatly affected by the hyper parameters, and it is necessary to know the sparsity of x^* in advance to set the hyper parameters, which is difficult to get in real situations. Thus, in order to more fairly compare the impact of the network itself on performance, all the networks do not use SS. All training follows (Chen et al. 2018) (The details are provided in the Supplementary Material). For all our methods, α_1^t and α_2^t are initialized as 1.0. θ_1^t and θ_2^t are initialized as $\frac{\lambda}{L}$ when using ST, while θ_1^t and θ_2^t are initialized as $\frac{\lambda}{L}-0.1$, $\bar{\theta}_1^t$ and $\bar{\theta}_2^t$ are initialized as $\frac{\lambda}{L}$ when using MT. All the results are obtained by running ten times and averaged. Verification of the parameters and the assumption, support set accuracy and unsupervised experiment are presented in the Supplementary Material.

4.1 Ablation Experiments

In this subsection, by controlling variables, we compare our ELISTA-m with LISTA (Gregor and LeCun 2010; Chen et al. 2018) and TiLISTA (Liu et al. 2019), and compare LISTA+m[1] with LISTA (Gregor and LeCun 2010; Chen

[1]LISTA+m is an algorithm which replaces ST in LISTA with MT.



(a) $\kappa = 0$, SNR $= \infty$ (b) $\kappa = 0$, SNR $= 30$ (c) $\kappa = 5$, SNR $= \infty$ (d) $\kappa = 50$, SNR $= \infty$

Figure 4: Comparison of sparse representation with different layers under different SNR and κ.

Table 3: The PSNR of natural image inpainting

	Barbara	Boat	House	Lena	Peppers	C.man	Couple	Finger	Hill	Man	Montage
ISTA	23.51	25.38	26.88	26.11	23.53	22.73	25.33	20.64	27.28	24.25	21.29
LISTA	24.52	27.29	29.50	27.84	25.78	24.51	27.20	23.60	28.92	26.32	22.50
GLISTA	25.30	28.95	30.95	29.97	27.64	25.76	27.48	26.29	29.53	28.14	24.31
LFISTA	26.01	29.68	32.06	32.12	28.57	26.77	29.77	28.10	**30.69**	30.22	26.94
ELISTA	**26.60**	**30.33**	**32.76**	**32.75**	**29.61**	**27.67**	**30.09**	**28.20**	30.41	**30.36**	**28.49**

et al. 2018) and GLISTA (Wu et al. 2020) in the noiseless condition to verify the improvement of the network structure and that of the thresholding function, respectively. For TiLISTA, we set

$$x^{t+\frac{1}{2}} = \mathrm{ST}(x^t - \alpha_1^t W(Ax^t - y), \theta_1^t)$$
$$x^{t+1} = \mathrm{ST}(x^{t+\frac{1}{2}} - \alpha_2^t W(Ax^{t+\frac{1}{2}} - y), \theta_2^t). \tag{11}$$

as one layer[2]. We set $m = 250$, $n = 500$ and $T = 16$. α_1^t and α_2^t in TiLISTA are also initialized as 1.0. We sample the elements of the dictionary matrix A randomly from a standard Gaussian distribution in simulations, the ground-truth x^* is also generated by the standard Gaussian distribution and we use Bernoulli distribution with a probability of 0.1 to ensure the sparsity. y is produced by A, x^* and noise ε. All experimental results are on the test set. The sparse representation performance is evaluated by NMSE (in dB):

$$\mathrm{NMSE}\left(\hat{x}, x^*\right) = 10 \log_{10}\left(\frac{\mathbb{E}\left\|\hat{x} - x^*\right\|^2}{\mathbb{E}\left\|x^*\right\|^2}\right). \tag{12}$$

We use NMSE, FLSNE and SPERR as indicators to evaluate the networks, where NMSE is defined in (12), FLSNE is the number of "false negative" and SPERR denotes the number of support error.

From Table 2, we can find that: (i) Because of the residual structure brought by the extragradient, ELISTA-m is superior to LISTA and TiLISTA in terms of NMSE and SPERR, where the two latter are serial connection. (ii) ST tends to expand the size of the support set to get a smaller FLSNE, however this also leads to a very large SPERR and a worse NMSE. GG can obtain better results than ST by narrowing the gap between ST and HT, but the SPERR of GLISTA is still large. That is, ST and GG expand the size of the support set in order to obtain a better sparse representation, so as to obtain a sparse representation that is not sparse. The residual structure induced by the extragradient can alleviate the problem of ST. Since MT is closer to HT, it can obtain a more sparse representation, which in turn enhances NMSE. Because our ELISTA is an improved algorithm combining these two improvements, it outperforms all the other algorithms, which also shows the effectiveness of the residual structure and the improvement of our thresholding function.

[2]The definition of one layer is different from that of (Liu et al. 2019). The purpose of this change is to control variables to verify the validity of our ELISTA.

4.2 Sparse Representation Performance

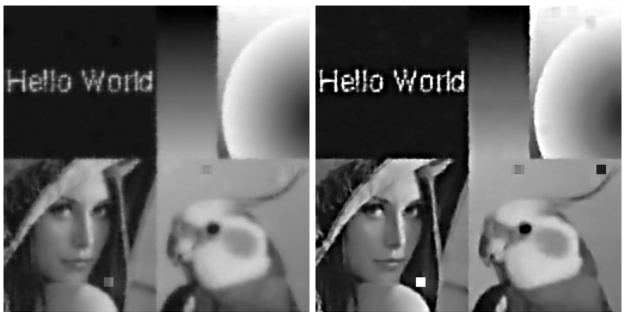
In this subsection, we compare our ELISTA and ELISTA-t with the state-of-the-art methods: LISTA (Gregor and LeCun 2010; Chen et al. 2018), LAMP (Borgerding, Schniter, and Rangan 2017) and GLISTA (Wu et al. 2020). We train the networks with three different noise levels: SNR (Signal-to-Noise Ratio) = 30, 40, ∞ and three different ill conditioned matrices A with condition numbers κ = 5, 30, 50.

Figure 4 shows that our methods are obviously better than the compared methods in terms of both convergence speed and accuracy in the noiseless case. Especially, compared with LISTA, the NMSE performance of our methods is nearly twice better than that of LISTA. In the presence of noise, our methods achieve the state-of-the-art convergence accuracy and are obviously better than other methods in terms of convergence speed. We note that due to the limitation of space, only part of the results are given here, and more results are reported in the Supplementary Material.

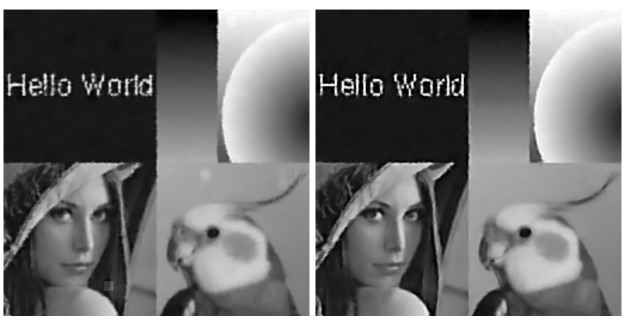
4.3 Natural Image Inpainting

In this subsection, we apply our algorithm to solve the natural image inpainting problem, and comparing it with LISTA

(a) corrupted image (b) ISTA, PSNR=21.19



(c) LISTA, PSNR=22.50 (d) GLISTA, PSNR=24.31



(e) LFISTA, PSNR=26.94 (f) ELISTA, PSNR=28.49

Figure 5: Image inpainting with 50% missing pixels on Montage.

(Chen et al. 2018), LFISTA (Moreau and Bruna 2017; Aberdam, Golts, and Elad 2020) and GLISTA (Wu et al. 2020). The training dataset is BSDS500 and the test dataset is Set 11. For LFISTA (Aberdam, Golts, and Elad 2020), we use the code provided by this work and for the other algorithms, we implement them ourselves. The PSNR of different algorithms are shown in Table 3, the qualitative results on the Montage image are shown in Figure 5 and the other qualitative results are shown in the Supplementary Material. In addition, detailed experimental setup and other details are also given in the Supplementary Material.

From Table 3, Figure 5 and all the other qualitative results in the Supplementary Material, we can see that our ELISTA outperforms other algorithms in most cases.

Table 4: The mean angular error of 3D geometry recovery via photometric stereo

q	LISTA	GLISTA	ELISTA-t	ELISTA
35	0.06836	0.06249	0.03534	**0.02754**
25	0.09664	0.10033	0.05885	**0.04947**
15	0.69334	0.63967	**0.47569**	0.60010

4.4 3D Geometry Recovery via Photometric Stereo

In this subsection, we compare our ELISTA and ELISTA-t with the state-of-the-art methods: LISTA (Gregor and LeCun 2010; Chen et al. 2018) and GLISTA (Wu et al. 2020) for 3D Geometry Recovery via Photometric Stereo. Photometric stereovision is a powerful technique used to recover high resolution surface normals from a 3D scene using appearance changes of 2D images in different lighting (Woodham 1980). In practice, however, the estimation process is often interrupted by non-lambert effects, such as highlights, shadows, or image noise. This problem can be solved by decomposing the observation matrix of the superimposed image under different lighting conditions into ideal lambert components and sparse error terms (Wu et al. 2010; Ikehata et al. 2012), i.e., $o = \rho L n + e$, where $o \in \mathbb{R}^q$ denotes the resulting measurements, $n \in \mathbb{R}^3$ denotes the true surface normal, $L \in \mathbb{R}^{q\times 3}$ defines a lighting direction, ρ is the diffuse albedo, acting here as a scalar multiplier and $e \in \mathbb{R}^q$ is an unknown sparse vector. By multiplying both sides of $o = \rho L n + e$ by the orthogonal complement to L, we can get $Proj_{null_{[L^\top]}}(o) = Proj_{null_{[L^\top]}}(e)$. Let $Proj_{null_{[L^\top]}}(o)$ be y and $Proj_{null_{[L^\top]}}(e)$ be Ax, e can be obtained by solving the sparse coding problem. Then we can use $L^\dagger(o-e)$ to recover n. The main experimental settings follow (Xin et al. 2016; Wu et al. 2020; He et al. 2017). Tests are performed using the 32-bit HDR gray-scale images of objects “Bunny” as in (Xin et al. 2016; Wu et al. 2020; He et al. 2017) with $q = 35, 25, 15$ and 40% of the elements of the sparse noise e are non-zero. From Table 4, we can find that our methods perform much better than LISTA and GLISTA, which is similar to the conclusion we came to in Section 4.2.

5 Conclusions

In this paper, we consider a sparse representation problem. We proposed an innovative algorithm called ELISTA with interpretable residual structure and a better thresholding function. Moreover, we proved that ELISTA can achieve linear convergence in theory. Extensive empirical results verified the high efficiency of our method. One limitation of this paper is that in the theoretical analysis, we use the same assumption as in the previous work (Chen et al. 2018; Liu et al. 2019; Wu et al. 2020), that s, i.e., the sparsity of x^*, is small enough. Removing this common assumption of the related algorithms is our future work.

References

Aberdam, A.; Golts, A.; and Elad, M. 2020. Ada-LISTA: Learned Solvers Adaptive to Varying Models. *arXiv preprint arXiv:2001.08456* .

Aberdam, A.; Sulam, J.; and Elad, M. 2019. Multi-layer sparse coding: The holistic way. *SIAM Journal on Mathematics of Data Science* 1(1): 46–77.

Ablin, P.; Moreau, T.; Massias, M.; and Gramfort, A. 2019. Learning step sizes for unfolded sparse coding. In *Advances in Neural Information Processing Systems*, 13100–13110.

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

Blumensath, T.; and Davies, M. E. 2008. Iterative thresholding for sparse approximations. *Journal of Fourier analysis and Applications* 14(5-6): 629–654.

Blumensath, T.; and Davies, M. E. 2009. Iterative hard thresholding for compressed sensing. *Applied and Computational Harmonic Analysis* 27(3): 265–274.

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

Chen, X.; Li, Y.; Umarov, R.; Gao, X.; and Song, L. 2020. Rna secondary structure prediction by learning unrolled algorithms. In *Proceedings of the International Conference on Learning Representations*.

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

Cho, K.; Van Merriënboer, B.; Gulcehre, C.; Bahdanau, D.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using RNN encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078* .

Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2015. Gated feedback recurrent neural networks. In *International Conference on Machine Learning*, 2067–2075.

Daubechies, I.; Defrise, M.; and De Mol, C. 2004. 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.

Deledalle, C.-A.; Papadakis, N.; Salmon, J.; and Vaiter, S. 2017. Clear: Covariant least-square refitting with applications to image restoration. *SIAM Journal on Imaging Sciences* 10(1): 243–284.

Donoho, D. L.; Maleki, A.; and Montanari, A. 2009. Message-passing algorithms for compressed sensing. *Proceedings of the National Academy of Sciences* 106(45): 18914–18919.

Efron, B.; Hastie, T.; Johnstone, I.; Tibshirani, R.; et al. 2004. Least angle regression. *The Annals of statistics* 32(2): 407–499.

Fan, J.; and Li, R. 2001. Variable selection via nonconcave penalized likelihood and its oracle properties. *Journal of the American statistical Association* 96(456): 1348–1360.

Gers, F. A.; Schraudolph, N. N.; and Schmidhuber, J. 2002. Learning precise timing with LSTM recurrent networks. *Journal of Machine Learning Research* 3(Aug): 115–143.

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

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

Gu, Q.; Wang, Z.; and Liu, H. 2014. Sparse pca with oracle property. In *Advances in Neural Information Processing Systems*, 1529–1537.

He, H.; Xin, B.; Ikehata, S.; and Wipf, D. 2017. From Bayesian sparsity to gated recurrent nets. In *Advances in Neural Information Processing Systems*, 5554–5564.

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

Ikehata, S.; Wipf, D.; Matsushita, Y.; and Aizawa, K. 2012. Robust photometric stereo using sparse regression. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 318–325. IEEE.

Ito, D.; Takabe, S.; and Wadayama, T. 2019. Trainable ISTA for sparse signal recovery. *IEEE Transactions on Signal Processing* 67(12): 3113–3125.

Korpelevich, G. 1976. The extragradient method for finding saddle points and other problems. *Matecon* 12: 747–756.

LeCun, Y.; Bengio, Y.; and Hinton, G. 2015. Deep learning. *Nature* 521(7553): 436–444.

Lederer, J. 2013. Trust, but verify: benefits and pitfalls of least-squares refitting in high dimensions. *arXiv preprint arXiv:1306.0113* .

Liu, J.; Chen, X.; Wang, Z.; and Yin, W. 2019. Alista: Analytic weights are as good as learned weights in lista. In *Proceedings of the International Conference on Learning Representations*.

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

Moreau, T.; and Bruna, J. 2017. Understanding trainable sparse coding via matrix factorization. In *Proceedings of the International Conference on Learning Representations*.

Nguyen, T. P.; Pauwels, E.; Richard, E.; and Suter, B. W. 2018. Extragradient method in optimization: Convergence and complexity. *Journal of Optimization Theory and Applications* 176(1): 137–162.

Papyan, V.; Romano, Y.; and Elad, M. 2017. Convolutional neural networks analyzed via convolutional sparse coding. *The Journal of Machine Learning Research* 18(1): 2887–2938.

Rick Chang, J.; Li, C.-L.; Poczos, B.; Vijaya Kumar, B.; and Sankaranarayanan, A. C. 2017. One Network to Solve Them All–Solving Linear Inverse Problems Using Deep Projection Models. In *Proceedings of the IEEE International Conference on Computer Vision*, 5888–5897.

Simon, D.; and Elad, M. 2019. Rethinking the csc model for natural images. In *Advances in Neural Information Processing Systems*, 2271–2281.

Sprechmann, P.; Bronstein, A. M.; and Sapiro, G. 2015. Learning efficient sparse and low rank models. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(9): 1821–1833.

Sreter, H.; and Giryes, R. 2018. Learned convolutional sparse coding. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2191–2195. IEEE.

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

Sulam, J.; Papyan, V.; Romano, Y.; and Elad, M. 2018. Multilayer convolutional sparse modeling: Pursuit and dictionary learning. *IEEE Transactions on Signal Processing* 66(15): 4090–4104.

Sun, J.; Li, H.; Xu, Z.; et al. 2016. Deep ADMM-Net for compressive sensing MRI. In *Advances in Neural Information Processing Systems*, 10–18.

Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* 58(1): 267–288.

Tipping, M. E. 2001. Sparse Bayesian learning and the relevance vector machine. *Journal of Machine Learning Research* 1(Jun): 211–244.

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

Woodham, R. J. 1980. Photometric method for determining surface orientation from multiple images. *Optical engineering* 19(1): 191139.

Wu, K.; Guo, Y.; Li, Z.; and Zhang, C. 2020. SPARSE CODING WITH GATED LEARNED ISTA. In *Proceedings of the International Conference on Learning Representations*.

Wu, L.; Ganesh, A.; Shi, B.; Matsushita, Y.; Wang, Y.; and Ma, Y. 2010. Robust photometric stereo via low-rank matrix completion and recovery. In *Asian Conference on Computer Vision*, 703–717. Springer.

Xie, X.; Wu, J.; Zhong, Z.; Liu, G.; and Lin, Z. 2019. Differentiable linearized ADMM. In *Proceedings of the 27th International Conference on International Conference on Machine Learning*.

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

Xu, P.; and Gu, Q. 2016. Semiparametric differential graph models. In *Advances in Neural Information Processing Systems*, 1064–1072.

Zarka, J.; Thiry, L.; Angles, T.; and Mallat, S. 2020. Deep Network classification by Scattering and Homotopy dictionary learning. In *Proceedings of the International Conference on Learning Representations*.

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

Zhang, Q.; Ye, X.; Liu, H.; and Chen, Y. 2020. A Novel Learnable Gradient Descent Type Algorithm for Nonconvex Non-smooth Inverse Problems. *arXiv preprint arXiv:2003.06748* .

Zhou, J. T.; Di, K.; Du, J.; Peng, X.; Yang, H.; Pan, S. J.; Tsang, I. W.; Liu, Y.; Qin, Z.; and Goh, R. S. M. 2018. SC2Net: Sparse LSTMs for sparse coding. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhu, R.; and Gu, Q. 2015. Towards a lower sample complexity for robust one-bit compressed sensing. In *International Conference on Machine Learning*, 739–747.