Learned Extragradient ISTA with Interpretable Residual Structures for Sparse Coding

Yangyang Li\textsuperscript{1*}, Lin Kong\textsuperscript{1*}, Fanhua Shang\textsuperscript{1,2†}, Yuanyuan Liu\textsuperscript{†}, Hongying Liu\textsuperscript{1}, Zhouchen Lin\textsuperscript{3}

\textsuperscript{1}Key Lab. of Intelligent Perception and Image Understanding of Ministry of Education, School of Artificial Intelligence, Xidian University; \textsuperscript{2}Peng Cheng Lab; \textsuperscript{3}Key Lab. of Machine Perception (MoE), School of EECS, Peking University 1615401247li@gmail.com, xdkonglin0511@163.com, \{fhshang, yyliu, hyliu\}@xidian.edu.cn, zlin@pku.edu.cn

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 a large estimation bias. Therefore, we propose a thresholding function for ELISTA instead of soft thresholding. From a theoretical perspective, we prove that our method attains linear convergence. Through ablation experiments, the improvements of our method on the network structure and the thresholding function are verified in practice. 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 $\epsilon \in \mathbb{R}^m$ (e.g., additive Gaussian white noise):

$$y = Ax^* + \epsilon,$$

where $A \in \mathbb{R}^{m \times n}$ ($m \ll n$ in general) 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):

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

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

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

where $\text{ST}(\cdot, \theta)$ is the soft-thresholding (ST) function with the threshold $\theta$, $\frac{L}{2}$ is the step size which should be taken in $(0, \frac{L}{2})$, where $L$ is the largest singular value of the dictionary matrix, and $A^T(Ax^t - y)$ is actually 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):

$$x^{t+1} = \text{ST}(W_1^t y + W_2^t x^t, \theta^t), \quad t = 0, 1, 2, \ldots,$$

where $W_1^t, W_2^t$ and $\theta^t$ are initialized as $\frac{1}{L} A^T, I - \frac{1}{L} A^T A$ and $\frac{L}{2}$, 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 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, similar to the Ordinary Differential Equation (ODE) that can be used to explain some neural networks (Chen et al. 2018a), the convolutional sparse coding version of LISTA can be used to explain the convolutional neural network in series (Papyan, Romano, and Elad 2017).

On the 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; Sreret and Giryes 2018) have been proposed and successfully used in different fields, and achieved satisfactory experimental results.

On the other hand, many works (Xin et al. 2016; Giryes et al. 2018; Moreau and Bruna 2017; Chen et al. 2018b; 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 learned iterative hard thresholding (LIHT) (Wang, Ling, and Huang 2016),
which was obtained by unfolding iterative hard thresholding (IHT) (Blumensath and Davies 2009) inspired by (Gregor and LeCun 2010), in terms of improving the restricted isometry property constant. Inspired by (Xin et al. 2016), He et al. (2017) connected sparse Bayesian learning (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. 2018b; 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; Papan, Romano, and Elad 2017; Abendam, Sulam, and Elad 2019; Sulam et al. 2018, 2019).

1.1 Related Works

Chen et al. (2018b) proved the coupling relationship between $W_1$ and $W_2$, i.e., $W_2 \rightarrow (I - W_1 A)$ 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: the improvements of the network structure and thresholding functions.

For the improvement of the network structure, Liu et al. (2019) further reduced the number of learnable parameters by proposing a novel algorithm, whose update rule is $x_{t+1} = ST(x_t - \alpha t W(A x_t - y), \theta_t)$, where $\alpha t$ 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 thresholding functions, 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 hard thresholding (HT). The improvements of LISTA mentioned above are shown in Figure 1, where ELISTA is an innovative algorithm proposed in this paper (see details in Section 2).

Moreover, Ablin et al. (2019) also discussed LISTA from a theoretical perspective. They proposed a simple step size strategy which can improve the convergence rate of IHT. Then we propose a new multistage-Thresholding (MT) function, which can improve the thresholding function to reduce the gap between ST and HT.

Our Main Contributions: The main contributions of this paper are listed 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 improvement about the network structure for solving sparse coding problems. To the best of our knowledge, this is the first residual structure LISTA with theoretical guarantees.
• We design a new thresholding function, called the Multistage-Thresholding (MT) function, 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 technique 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 technique of extragradient into LISTA. Then we propose a new multistage-thresholding (MT) function and analyze its advantages. Finally, by combining the techniques of extragradient and MT,
we propose an innovative algorithm, named *Extragradient based LISTA* (ELISTA), and depict it in detail. Moreover, we establish the relationship between ELISTA and Res-Net, which is one of the reasons why ELISTA is advantageous.

### 2.1 Extragradients 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 extragradients 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 at $x^t+\frac{1}{2}$, 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:

$$
x^{t+\frac{1}{2}} = ST\left(x^t - \frac{1}{T}A^T(Ax^t - y), \frac{1}{T}\right),
$$

$$
x^{t+1} = ST\left(x^t - \frac{1}{T}A^T(Ax^{t+\frac{1}{2}} - y), \frac{1}{T}\right).
$$

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., $|x|_1$ obtained from the algorithms with ST is actually smaller than the real $|x^*_t|$, 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:

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

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

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

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 this issue, we design and propose an innovative thresholding function called *Multistage-Thresholding (MT) function*, which is defined as follows:

$$
z = MT(x, \theta, \bar{\theta}) \triangleq \begin{cases} 
0, & 0 \leq |x| < \theta, \\
\frac{\bar{\theta}}{\theta - \bar{\theta}} \text{sign}(x)(|x| - \theta), & \theta \leq |x| < \bar{\theta}, \\
x, & |x| \geq \bar{\theta}.
\end{cases}
$$

Different thresholding functions are shown in Figure 2, from which we know that MT is equal to GG when $0 \leq |x| < \theta$, which plays the role of gain to ST, and when $|x| \geq \theta$, 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 functions of HELU$_{\sigma}(\cdot)$ (Wang, Ling, and Huang 2016), SCAD (Li 2001) and MCP (Zhang 2010). However, there are some differences between MT, HELU$_{\sigma}(\cdot)$, SCAD and MCP in terms of the motivation of proposal and the internal mathematical mechanism. The detailed discussions are given in the Supplementary Material.

### 2.3 Extragradients 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 $\frac{1}{T}A^T$ and two thresholds of two steps in (5) as learnable parameters, and get the following update rules:

$$
x^{t+\frac{1}{2}} = ST(x^t - W^t_1(Ax^t - y), \theta^t_1),
$$

$$
x^{t+1} = ST(x^t - W^t_2(Ax^{t+\frac{1}{2}} - y), \theta^t_2).
$$

However, since the above scheme has two different matrices $W^t_1$ and $W^t_2$ 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 $W^t_1$ and $W^t_2$ into $\alpha^t_1 W^t$ and $\alpha^t_2 W^t$, respectively, where $\alpha^t_1$ and $\alpha^t_2$ 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{align}
x^{t+\frac{1}{2}} &= MT(x^t - \alpha_1^t W(Ax^t - y), \theta_1, \theta_2), \\
x^{t+1} &= MT(x^t - \alpha_1^t W(Ax^{t+\frac{1}{2}} - y), \theta_2),
\end{align}
(8)
where $\theta_1$ and $\theta_2$ 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 extent, 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. Besides, there is a lot of work using ODE to interpret the network by considering ODE as a continuous equivalent of Residual Networks (ResNets) (Chen et al. 2018a). However, we found that ODE can only explain the networks with linear connection blocks, while ours is nonlinear. But, the form of our blocks are less general than those of ODE.

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. 2018b; 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:
\[ x^* \in \mathcal{X}(B, s) \triangleq \{x^* \mid \|x^*\|_0 \leq s \}. \]
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:
\[
\mu(A) = \inf_{W \in \mathbb{R}^{m \times n}, \|W\|_{\infty} = 1} \left\{ \max_{1 \leq i,j \leq n} W_i;A_{i,j} \right\}.
\]
We let $\mathcal{W}(A)$ denote a set of all matrices with the generalized mutual coherence $\mu(A)$, which means that
\[
\mathcal{W}(A) = \left\{ W \mid \max_{1 \leq i,j \leq n} W_i;A_{i,j} = \mu(A), W_i;A_{i,i} = 1, \forall i \right\}.
\]
Verify the thresholding function

<table>
<thead>
<tr>
<th>LISTA</th>
<th>TiLISTA</th>
<th>ELISTA-m-t</th>
<th>ELISTA-m</th>
<th>GLISTA</th>
<th>LISTA+m</th>
<th>ELISTA-t</th>
<th>ELISTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>NMSE</td>
<td>-36.01</td>
<td>-50.28</td>
<td>-51.82</td>
<td>-65.66</td>
<td>-63.73</td>
<td>-62.21</td>
<td>-17.03</td>
</tr>
<tr>
<td>FLSNE</td>
<td>0.16</td>
<td>0.02</td>
<td>0.10</td>
<td>0.02</td>
<td>0.02</td>
<td>0.12</td>
<td>0.04</td>
</tr>
<tr>
<td>SPERR</td>
<td>147.12</td>
<td>46.26</td>
<td>3.23</td>
<td>2.35</td>
<td>57.22</td>
<td>0.80</td>
<td>0.15</td>
</tr>
</tbody>
</table>

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. 2018b), we know $\mathcal{W}(A) \neq \emptyset$.

**Definition 2.** Given a model with $\Theta$, in which

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

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

$$\omega_{t+\frac{1}{2}}(k_{t+\frac{1}{2}}|\Theta) = \sup_{\frac{x^t}{x^*} \sup(x^t) \leq \sup(x^*) + k_{t+\frac{1}{2}}} \Gamma,$$

$$\omega_{t+1}(k_{t+1}|\Theta) = \sup_{\frac{x^t}{x^*} \sup(x^t) \leq \sup(x^*) + k_{t+1}} \Gamma,$$

where $x^{t+\frac{1}{2}} := MT((I - \alpha_1 W A)(x^{t+\frac{1}{2}} - x^*), \theta_1^t)$, $x^{t+1} := MT((I - \alpha_2^t W A)(x^{t+1} - x^*), \theta_2^t)$ 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.

$$\theta_1^t = \alpha_1 t \omega_{t+\frac{1}{2}}(k_{t+\frac{1}{2}}|\Theta) \mu(A) \sup_{x^* \sup(x^*)} \| x^t - x^* \|_1, \quad \theta_2^t = \alpha_2^t \omega_{t+1}(k_{t+1}|\Theta) \mu(A) \sup_{x^* \sup(x^*)} \| x^{t+\frac{1}{2}} - x^* \|_1,$$

$$\theta_1^i \geq \theta_1^t + \| x^{t+\frac{1}{2}} \|, \quad \theta_2^i \geq \theta_2^t + \| x^{t+1} \|$$

are achieved, $\alpha_1\xi$ and $\alpha_2\xi$ belong to specific bounded intervals for different cases, and $s$ is small enough, then for the sequences generated by ELISTA, the following result holds

$$\| x^t - x^* \|_2 \leq sB \exp \left( \sum_{i=1}^t \frac{c_i}{sB} \right) < sB \exp(c^t),$$

where $c^t = \max_{i=1,2,\ldots,t} (c_i)$. \exists \theta_0 = [-\log(\frac{\| x^t \|}{c})/c]$, where $c = \log((2s - 1)\mu(A))$, $\sigma = \| x^t \|$, for $i \geq \theta_0$, $0 < k_{i-\frac{1}{2}} \leq k_i < s$, if $\gamma^{i-\frac{1}{2}} = \gamma^i = 0$, then $c_i < 0$, and for $i > \theta_0$, $k_{i-\frac{1}{2}} = k_i = 0$, 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, $c^t < 0$.

The definitions of $\gamma^{i-\frac{1}{2}}$ and $\gamma^i$ 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: $i \in S^{t+\frac{1}{2}}, i \notin S^{t+\frac{1}{2}}$ and $i \in S^{t+\frac{1}{2}} \cup S^{t+\frac{1}{2}} \cup S^{t+\frac{1}{2}} \cup S^{t+\frac{1}{2}} \cup S^{t+\frac{1}{2}} \cup S^{t+\frac{1}{2}} \cup S^{t+\frac{1}{2}} \cup S^t$. Then, we can get the results to obtain the relationship between $\| x^{t+\frac{1}{2}} - x^* \|_1$ and $\| x^t - x^* \|_1$ and $\| x^t - x^* \|_1$. Then, we can get the relationship between $\| x^{t+1} - x^* \|_1$ and $\| x^t - x^* \|_1$ and $\| x^t - x^* \|_1$. 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. 2018b; Liu et al. 2019; Wu et al. 2020; Borgerding, Schniter, and Rangan 2017). We find that Support Selection (SS) (Chen et al. 2018b) 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 $x^t$ 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. 2018b) (The details are provided in the Supplementary Material). For all our methods, $\alpha_1$ and $\alpha_2$ are initialized as 1.0. $\theta_1$ and $\theta_2$ are initialized as $\frac{1}{\xi}$ when using ST, while $\theta_1$ and $\theta_2$ are initialized as $\frac{1}{2} - 0.1$, $\theta_1$ and $\theta_2$ are initialized as $\frac{1}{2}$ 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.
Figure 4: Comparison of sparse representation with different layers under different SNR and $\kappa$.

**Table 3: The PSNR results of the methods for natural image inpainting tasks.**

<table>
<thead>
<tr>
<th>Method</th>
<th>Barbara</th>
<th>Boat</th>
<th>Lena</th>
<th>Peppers</th>
<th>C.man</th>
<th>Couple</th>
<th>Finger</th>
<th>Hill</th>
<th>Man</th>
<th>Montage</th>
</tr>
</thead>
<tbody>
<tr>
<td>LISTA</td>
<td>24.52</td>
<td>27.29</td>
<td>29.50</td>
<td>27.84</td>
<td>25.78</td>
<td>24.51</td>
<td>27.20</td>
<td>23.60</td>
<td>22.50</td>
<td></td>
</tr>
<tr>
<td>GLISTA</td>
<td>25.30</td>
<td>28.95</td>
<td>30.95</td>
<td>29.97</td>
<td>27.64</td>
<td>25.76</td>
<td>27.48</td>
<td>26.29</td>
<td>24.31</td>
<td></td>
</tr>
<tr>
<td>LFIISTA</td>
<td>26.01</td>
<td>29.68</td>
<td>32.06</td>
<td>32.12</td>
<td>28.57</td>
<td>26.77</td>
<td>29.77</td>
<td>28.10</td>
<td>24.31</td>
<td></td>
</tr>
<tr>
<td>ELISTA</td>
<td><strong>26.60</strong></td>
<td><strong>30.33</strong></td>
<td><strong>32.76</strong></td>
<td><strong>32.75</strong></td>
<td><strong>29.61</strong></td>
<td><strong>27.67</strong></td>
<td><strong>30.09</strong></td>
<td><strong>28.20</strong></td>
<td><strong>30.69</strong></td>
<td></td>
</tr>
</tbody>
</table>

4.1 Ablation Experiments

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

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

(11)

as one layer\(^2\). We set $m = 250, n = 500$ and $T = 16, \alpha_1^t$ and $\alpha_2^t$ 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 $\varepsilon$. All experimental results are on the test set. The sparse representation performance is evaluated by NMSE (in dB):

$$NMSE(x, x^*) = 10 \log_{10} \left( \frac{E[\|\hat{x} - x^*\|^2]}{E[\|x^*\|^2]} \right).$$

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 second-order curvature information and 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.

4.2 Sparse Representation Performance

In this subsection, we compare our ELISTA and ELISTA-t with the state-of-the-art methods: LISTA, LAMP (Borgerding, Schniter, and Rangan 2017) and GLISTA. We train the networks with three different noise levels: SNR (Signal-to-Noise Ratio) = 30, 40, $\infty$ and three different ill-conditioned matrices $A$ with condition numbers $\kappa = 5, 30, 50$.

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, LFISTA (Moreau and Bruna 2017; Aberdam, Golts, and Eldad 2020) and GLISTA. The training dataset is BSDS500 and the test dataset is Set 11. For LFISTA, 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.

<table>
<thead>
<tr>
<th>q</th>
<th>LISTA</th>
<th>GLISTA</th>
<th>ELISTA-t</th>
<th>ELISTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>0.06836</td>
<td>0.06249</td>
<td>0.03534</td>
<td><strong>0.02754</strong></td>
</tr>
<tr>
<td>25</td>
<td>0.09664</td>
<td>0.10033</td>
<td>0.05885</td>
<td><strong>0.04947</strong></td>
</tr>
<tr>
<td>15</td>
<td>0.69334</td>
<td>0.63967</td>
<td><strong>0.47569</strong></td>
<td>0.60010</td>
</tr>
</tbody>
</table>

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 and GLISTA 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., \( o = \rho Ln + e \), 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, \( \rho \) 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 \( o = \rho Ln + e \) by the orthogonal complement to \( L \), we can get \( \text{Proj}_{\text{null}(L^\top)}(o) = \text{Proj}_{\text{null}(L^\top)}(e) \). Let \( \text{Proj}_{\text{null}(L^\top)}(o) \) be \( y \) and \( \text{Proj}_{\text{null}(L^\top)}(e) \) be \( Ax \), \( e \) can be obtained by solving the sparse coding problem. Then we can use \( L^\top(o - e) \) 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.

5 Conclusions

In this paper, we proposed a novel extragradient based learned iterative shrinkage thresholding algorithm (called ELISTA) with interpretable residual structure and a better thresholding function. Moreover, we proved that ELISTA can achieve linear convergence. Extensive empirical results verified the high efficiency of our method. This could have both theoretical and practical impacts to the relationship between new neural network architectures and advanced algorithms, and potentially deepen our understanding to interpretability of deep learning models. One limitation of this paper is that in theory, we use the same assumption as in the previous work (Chen et al. 2018b; Liu et al. 2019; Wu et al. 2020), that the sparsity of \( x^* \) is small enough. Removing this common assumption is our future work.
Acknowledgments

This work was supported by the National Natural Science Foundation of China (Nos. 61876220, 61876221, 61976164, 61836009, U1701267, and 61871310), the Project supported the Foundation for Innovative Research Groups of the National Natural Science Foundation of China (No. 61621005), the Major Research Plan of the National Natural Science Foundation of China (Nos. 91438201 and 91438103), the Program for Cheung Kong Scholars and Innovative Research Team in University (No. IRT_15R53), the Fund for Foreign Scholars in University Research and Teaching Programs (the 111 Project) (No. B07048), the Science Foundation of Xidian University (Nos. 10251180018 and 10251180019), the National Science Basic Research Plan in Shaanxi Province of China (Nos. 2019JQ-657 and 2020JM-194), and the Key Special Project of China High Resolution Earth Observation System-Young Scholar Innovation Fund. Z. Lin is supported by NSF China (grant nos. 61625301 and 61731018), Major Scientific Research Project of Zhejiang Province of China (Nos. 2019JQ-657 and 2019KB0AC01 and 2019KB0AB02), Beijing Academy of Artificial Intelligence, and Qualcomm.

References


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*.


