Non-negative Learned ISTA with Reflected-ReLU-Augmented ℓ_1 Regularization

Haruki Esaki*, Towa Yasui*, Seisuke Kyochi*
* Kogakuin University, Japan
E-mail: kyochi@cc.kogakuin.ac.jp

***Abstract*—Deep unfolding has become a powerful paradigm for converting iterative sparse-optimization solvers into trainable neural architectures. However, most existing unfolded networks —such as Learned ISTA (LISTA)—inherit the soft-thresholding operator and therefore cannot enforce explicit non-negativity, which is essential in many signal processing and imaging applications. We propose the reflected-ReLU-augmented ℓ_1 regularizer, which simultaneously promotes sparsity while penalizing negative coefficients. We derive its closed-form proximal mapping and show that it reduces to a piece-wise affine "reflected-ReLU" shrinkage. By unfolding an ISTA scheme equipped with this proximal operator, we obtain non-negative LISTA (NNLISTA), a deep-unfolded network that learns sparse and non-negative representations end-to-end. Experiments on sparse coding and compressed sensing tasks that require non-negative signal recovery consistently demonstrate that NNLISTA outperforms standard LISTA in estimation accuracy.**

I. Introduction

Over the past decades, convex optimization based on sparse modeling—hereafter simply sparse optimization—has become a unifying mathematical framework for signal processing, machine learning, and inverse problems, powering applications such as denoising, deblurring, compressed sensing, regression, and classification [1]–[8]. The canonical sparsity-promoting ℓ_1-norm is non-differentiable at the origin, which once made ℓ_1-regularized problems more difficult than their smooth counterparts; the advent of proximal splitting methods has since rendered such problems tractable and spurred increasingly rich sparse models. In high-dimensional restoration tasks, for instance, group-sparse modeling with mixed norms (e.g. the $\ell_{2,1}$-norm) or the total-variation (TV) regularizer is now standard practice for capturing piecewise-smooth structure in images and videos [6]–[9].

Beyond sparsity alone, many real-world signals are *a priori* non-negative—pixel intensities, amplitude spectrograms, Poisson event rates, or regression coefficients with physical meaning. Incorporating this prior, either as a hard constraint enforced by projected methods (ISTA, FISTA, ADMM, or primal–dual splitting) or via one-sided/asymmetric ℓ_1 penalties that softly suppress negative components [10], [11], improves interpretability, identifiability, and numerical stability. Classical examples include non-negative matrix factorization (NMF) for speech and acoustic processing [12], [13] and non-negative LASSO for interpretable sparse regression.

First-order algorithms such as ISTA, FISTA, and ADMM typically require tens of iterations to reach acceptable accuracy —too slow for latency-critical tasks. Deep unrolling (a.k.a. unfolding) alleviates this bottleneck by interpreting each iteration as a neural network layer whose parameters are learned end-to-end, thereby retaining much of the interpretability and theoretical guarantees of the underlying algorithm while accelerating inference by orders of magnitude. Seminal examples include LISTA [14], Deep-ADMM-Net [15], and Deep-AMP [16]. Yet almost all existing unfolded networks either ignore non-negativity or impose it crudely with a post-layer ReLU, which destroys the proximal structure and eliminates any tunable trade-off between sparsity and positivity.

In this work, we propose a convex composite regularizer that promotes both sparsity and non-negativity while admitting a closed-form proximity operator suitable for deep unfolding. Its proximity operator is a three-segment positive soft-threshold that smoothly interpolates between ordinary LASSO and fully constrained non-negative LASSO. Embedding this operator in a forward–backward (proximal-gradient) scheme and unrolling the iterations yields non-negative LISTA (NNLISTA), which learns the sparsity–positivity balance from data while retaining the speed and interpretability of deep unfolding.

Contributions

1) We introduce the reflected-ReLU-augmented ℓ_1 regularizer (RR-ℓ_1 regularizer), the sparsity-plus-non-negativity penalty with a closed-form proximal mapping, enabling end-to-end learnable deep unfolding.
2) We derive the corresponding forward–backward algorithm and unroll it into NNLISTA, thereby preserving convexity, interpretability, and fast inference.
3) Experiments on sparse-coding and compressed-sensing tasks requiring non-negative recovery show that NNLISTA consistently outperforms standard LISTA while maintaining computational efficiency.

It is worth noting that while we demonstrate the proposed regularizer using LISTA, it can be incorporated into any deep unrolling approach.

The remainder of the paper is organized as follows. Section II reviews convex analysis and proximal algorithms. Section III details the RR-ℓ_1 regularizer, its proximal operator, and the NNLISTA architecture. Experimental results are in Section IV, and conclusions are in Section V.

979-8-3315-7206-8/25/$31.00

A. Notation

Bold uppercase letters denote matrices, whereas bold lowercase letters denote vectors. $\mathbb{N}$ is the set of natural numbers and $\mathbb{R}$ the set of real numbers. We write $\mathbb{R}_+ := \{a \in \mathbb{R} \mid a \geq 0\}$, $\mathbb{R}_{++} := \{a \in \mathbb{R} \mid a > 0\}$, for the non-negative and positive reals, respectively. The set of N-dimensional real vectors is $\mathbb{R}^N$, and the set of real matrices of size $N_r \times N_c$ ($N_r, N_c \in \mathbb{N}$) is $\mathbb{R}^{N_r \times N_c}$. $\mathbf{I}$ denotes the identity matrix. For $\mathbf{X} \in \mathbb{R}^{N_c \times N_r}$, its transpose is $\mathbf{X}^\top \in \mathbb{R}^{N_r \times N_c}$.

For a vector $\mathbf{x} \in \mathbb{R}^N$ and $p \in [1, \infty)$, the ℓ_p-norm $\|\cdot\|_p : \mathbb{R}^N \to \mathbb{R}_+$ is

$$\|\mathbf{x}\|_p \;=\; \Big(\sum_{n=1}^{N} |x_n|^p\Big)^{1/p}. \tag{1}$$

II. Preliminaries

A. Convex functions, convex sets, and convex optimization problems [4]

A function $f : \mathbb{R}^N \to \mathbb{R}$ is called convex if, for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ and any $\alpha \in (0, 1)$,

$$f(\alpha\mathbf{x} + (1-\alpha)\mathbf{y}) \leq \alpha f(\mathbf{x}) + (1-\alpha) f(\mathbf{y}). \tag{2}$$

Regarding vectors $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$, the inequality $\mathbf{x} \leq \mathbf{y}$ means $x_n \leq y_n$ for all n. A vector-valued function $f : \mathbb{R}^N \to \mathbb{R}^M$ that satisfies (2) is also called convex. For functions $f : \mathbb{R}^M \to \mathbb{R}$ and $g : \mathbb{R}^N \to A \subset \mathbb{R}^M$, their composition $f \circ g : \mathbb{R}^N \to \mathbb{R}$ is convex if and only if both f and g are convex and f is non-decreasing on A, i.e. $\mathbf{x} \leq \mathbf{y} \Rightarrow f(\mathbf{x}) \leq f(\mathbf{y})$. A set $C \subset \mathbb{R}^N$ is called convex if, for any $\mathbf{x}, \mathbf{y} \in \mathbb{R}^N$ and any $\alpha \in (0, 1)$, $\alpha\mathbf{x} + (1-\alpha)\mathbf{y} \in C$. Finding a minimizer (or maximizer) of a convex function $f : \mathbb{R}^N \to \mathbb{R}$ over a convex set $C \subset \mathbb{R}^N$,

$$\mathbf{x}^\star \in \operatorname*{argmin}_{\mathbf{x} \in C} f(\mathbf{x}), \tag{3}$$

is called a convex optimization problem. Convexity of the objective and feasible set guarantees that any solution obtained is globally optimal.

B. Lower-semicontinuity and proper convex functions [4]

A convex function $f : \mathbb{R}^N \to \mathbb{R}$ is lower-semicontinuous if every level set $\{\mathbf{x} \in \mathbb{R}^N \mid f(\mathbf{x}) \leq \alpha\}$ ($\forall \alpha \in \mathbb{R}$) is closed. It is proper if its effective domain $\operatorname{dom}(f) := \{\mathbf{x} \in \mathbb{R}^N \mid f(\mathbf{x}) < \infty\}$ is non-empty. The set of all lower-semicontinuous proper convex functions on $\mathbb{R}^N$ is denoted $\Gamma_0(\mathbb{R}^N)$.

C. Proximity operator [4]

For $f \in \Gamma_0(\mathbb{R}^N)$, the proximity operator $\operatorname{prox}_{\gamma f} : \mathbb{R}^N \to \mathbb{R}^N$ ($\gamma \in \mathbb{R}_{++}$) is defined by

$$\operatorname{prox}_{\gamma f}(\mathbf{x}) := \operatorname*{argmin}_{\mathbf{y} \in \mathbb{R}^N} \gamma f(\mathbf{y}) + \frac{1}{2}\|\mathbf{x} - \mathbf{y}\|_2^2. \tag{4}$$

Orthogonal projection onto a convex set $C \subset \mathbb{R}^N$, $\mathcal{P}_C(\mathbf{x}) : \mathbb{R}^N \to \mathbb{R}^N$, is a special case obtained by taking $f = \iota_C$, the indicator function $\iota_C(\mathbf{x}) = 0$ if $\mathbf{x} \in C$ and $\iota_C(\mathbf{x}) = \infty$ otherwise:

$$\operatorname{prox}_{\gamma \iota_C}(\mathbf{x}) = \operatorname*{argmin}_{\mathbf{y} \in C} \|\mathbf{x} - \mathbf{y}\|_2^2 =: \mathcal{P}_C(\mathbf{x}). \tag{5}$$

D. Subdifferential [4]

For $f \in \Gamma_0(\mathbb{R}^N)$, the subdifferential at $\mathbf{x} \in \mathbb{R}^N$ is

$$\partial f(\mathbf{x}) := \big\{\mathbf{u} \in \mathbb{R}^N \;\big|\; f(\mathbf{y}) \geq f(\mathbf{x}) + \langle \mathbf{u}, \mathbf{y} - \mathbf{x}\rangle,\ \forall \mathbf{y} \in \mathbb{R}^N\big\}. \tag{6}$$

If $f, g \in \Gamma_0(\mathbb{R}^N)$ and $\operatorname{ri}(\operatorname{dom}(f)) \cap \operatorname{ri}(\operatorname{dom}(g)) \neq \varnothing$, then $\partial(f+g)(\mathbf{x}) = \partial f(\mathbf{x}) + \partial g(\mathbf{x})$. Moreover, for $f \in \Gamma_0(\mathbb{R}^N)$,

$$\mathbf{x}^\star \in \operatorname*{argmin}_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) \iff \mathbf{0} \in \partial f(\mathbf{x}^\star). \tag{7}$$

E. Proximal gradient method [4]

Let $f, g \in \Gamma_0(\mathbb{R}^N)$ with f differentiable and its gradient β-Lipschitz, i.e. $\|\nabla f(\mathbf{x}) - \nabla f(\mathbf{y})\|_2 \leq \beta\|\mathbf{x} - \mathbf{y}\|_2$ for all $\mathbf{x}, \mathbf{y}$, and assume that the proximity operator of g is easy to compute. Then the minimizer

$$\mathbf{x}^\star \in \operatorname*{argmin}_{\mathbf{x} \in \mathbb{R}^N} f(\mathbf{x}) + g(\mathbf{x}) \tag{8}$$

can be obtained from any initial point $\mathbf{x}^{(0)} \in \mathbb{R}^N$ by the iteration

$$\mathbf{x}^{(k+1)} = \operatorname{prox}_{\alpha g}\big(\mathbf{x}^{(k)} - \alpha \nabla f(\mathbf{x}^{(k)})\big), \tag{9}$$

where $\alpha \in \big(0, \frac{2}{\beta}\big)$.

F. Sparse optimization problems and solution algorithms

In many signal-processing tasks such as compressed sensing, signal restoration, and regression, the observation model

$$\mathbf{y} = \mathbf{\Phi}\mathbf{x} + \mathbf{n} \tag{10}$$

is used, where $\mathbf{\Phi} \in \mathbb{R}^{M \times N}$ ($M < N$) represents the sensing process (e.g. a sampling matrix) or a basis/frame/dictionary, and $\mathbf{n} \in \mathbb{R}^M$ is Gaussian noise. Assuming sparsity of $\mathbf{x}$, one often formulates the estimation of $\mathbf{x}$ from $\mathbf{y}$ as

$$\mathbf{x}^\star := \operatorname*{argmin}_{\mathbf{x} \in \mathbb{R}^N} \tfrac{1}{2}\|\mathbf{\Phi}\mathbf{x} - \mathbf{y}\|_2^2 \;+\; \mu\|\mathbf{x}\|_1, \qquad \mu \in \mathbb{R}_{++}, \tag{11}$$

which is well known as LASSO. Problem (11) fits the framework of the proximal gradient method, and starting from any $\mathbf{x}^{(0)} \in \mathbb{R}^N$ one obtains the optimal solution via the iteration called ISTA [17]:

$$\mathbf{x}^{(k+1)} = \operatorname{Soft}_{\alpha\mu}\Big(\mathbf{x}^{(k)} - \alpha\mathbf{\Phi}^\top(\mathbf{\Phi}\mathbf{x}^{(k)} - \mathbf{y})\Big), \tag{12}$$

where $\alpha \in \big(0, 2/\sigma_{\max}(\mathbf{\Phi}^\top\mathbf{\Phi})\big)$ and $\operatorname{Soft}_\lambda : \mathbb{R}^N \to \mathbb{R}^N$ ($\lambda \in \mathbb{R}_{++}$) is the soft-thresholding operator, i.e. the proximity operator of the ℓ_1-norm:

$$\operatorname{Soft}_\lambda(\mathbf{x}) := \begin{bmatrix} s_\lambda(x_1) & \cdots & s_\lambda(x_N) \end{bmatrix}^\top,$$
$$s_\lambda(x) := \begin{cases} x - \lambda & (x > \lambda) \\ 0 & (-\lambda \leq x \leq \lambda) \\ x + \lambda & (x < -\lambda) \end{cases}. \tag{13}$$

As illustrated in Fig. 1, the soft-thresholding operator sets components with small magnitude (relative to the threshold) to zero, thereby promoting sparsity.

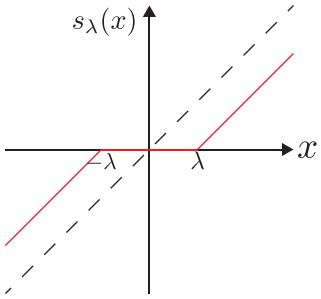
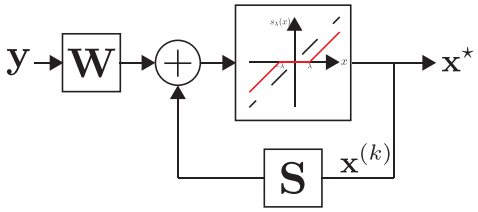
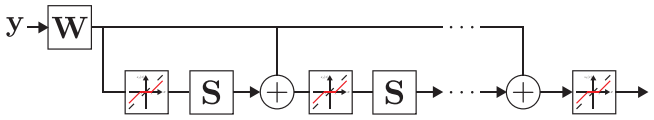


Fig. 1: Soft-thresholding operator (proximal mapping of the ℓ_1 norm)



(a) ISTA



(b) LISTA

Fig. 2: ISTA and its deep unrolling (LISTA)

G. Deep Unrolling

Deep unrolling unfolds the iterative steps of a proximal-splitting algorithm into a feed-forward neural network so that standard deep-learning techniques (back-propagation, stochastic gradient descent, etc.) can be applied to learn various parameters (step size, regularization weights) directly from data. Compared with a conventional optimization algorithm, deep unrolling often accelerates convergence. In this work, we focus on LISTA (Learned ISTA), a prototypical deep-unrolling method.

First rewrite ISTA in (12) as

$$\begin{aligned}\mathbf{x}^{(k+1)} &= \mathrm{Soft}_{\alpha\mu}\big((\mathbf{I}-\alpha\mathbf{\Phi}^\top\mathbf{\Phi})\mathbf{x}^{(k)}+\alpha\mathbf{\Phi}^\top\mathbf{y}\big)\\ &= \mathrm{Soft}_{\alpha\mu}\big(\mathbf{S}\mathbf{x}^{(k)}+\mathbf{W}\mathbf{y}\big),\end{aligned} \tag{14}$$

where $\mathbf{S} := \mathbf{I}-\alpha\mathbf{\Phi}^\top\mathbf{\Phi}$ and $\mathbf{W} := \alpha\mathbf{\Phi}^\top$. Equation (14) can be viewed, as in Fig. 2(a), as an algorithm that feeds back the updated estimate, but, when unrolled as in Fig. 2(b), it is equivalent to a neural-network architecture consisting of matrix multiplications and the non-linear soft-thresholding operation. Denoting this network by $\mathrm{Net}_\Theta : \mathbb{R}^M \to \mathbb{R}^N$, we learn its parameters Θ (matrices, step size, regularization weights) from training data $\{(\mathbf{x}_i,\mathbf{y}_i)\}_{i=1}^{N_{\text{train}}}$ by minimizing the loss

$$\Theta^\star = \operatorname*{argmin}_\Theta \sum_{i=1}^{N_{\text{train}}} \mathcal{L}\big(\mathrm{Net}_\Theta(\mathbf{y}_i),\,\mathbf{x}_i\big), \tag{15}$$

after which an input $\mathbf{y}$ is processed as $\mathrm{Net}_{\Theta^\star}(\mathbf{y})$.

III. Non-Negative LISTA with Reflected-ReLU-Augmented ℓ_1 Regularization

A. Reflected-ReLU-Augmented ℓ_1 Regularizer

We propose a convex formulation that considers both sparsity and non-negativity. We first construct a convex penalty for negative values using the rectified linear unit (ReLU). For a scalar input,

$$\mathcal{R}(x) := \max\{x, 0\}, \tag{16}$$

and for a vector input $\widehat{\mathcal{R}}(\mathbf{x}) := [\mathcal{R}(x_1),\ldots,\mathcal{R}(x_N)]^\top$. The reflected ReLU (RReLU) is

$$\begin{aligned}\mathcal{R}_r(x) &:= \mathcal{R}(-x) = \max\{-x, 0\},\\ \widehat{\mathcal{R}}_r(\mathbf{x}) &:= [\mathcal{R}_r(x_1),\ldots,\mathcal{R}_r(x_N)]^\top.\end{aligned} \tag{17}$$

Because both $\widehat{\mathcal{R}}_r$ and the ℓ_1-norm are convex and the latter is non-decreasing on $\mathbb{R}_+$, their composition $\|\widehat{\mathcal{R}}_r(\cdot)\|_1$ is convex and imposes a larger penalty when negative entries are present. Combining it with the ℓ_1-norm yields the RReLU-augmented ℓ_1 (RR-ℓ_1) regularizer.

$$\mathcal{F}_{\mu_1,\mu_2}(\mathbf{x}) := \mu_1\|\mathbf{x}\|_1 + \mu_2\|\widehat{\mathcal{R}}_r(\mathbf{x})\|_1, \tag{18}$$

where $\mu_1, \mu_2 \in \mathbb{R}_{++}$. Note that the RR-$\ell_1$ regularizer remains convex.

To illustrate how the reflected ReLU imposes a penalty on negative components, consider the following simple two-dimensional examples. Let $\mathbf{x}^{(1)} = [2,\ 1]^\top$ and $\mathbf{x}^{(2)} = [0,\ -1]^\top$. We compute $\widehat{\mathcal{R}}_r(\mathbf{x})$ and the corresponding $\|\widehat{\mathcal{R}}_r(\mathbf{x})\|_1$ for each case:

- For $\mathbf{x}^{(1)} = [2,\ 1]^\top$: $\widehat{\mathcal{R}}_r(\mathbf{x}^{(1)}) = [0,\ 0]^\top$, then $\|\widehat{\mathcal{R}}_r(\mathbf{x}^{(1)})\|_1 = 0$.
- For $\mathbf{x}^{(2)} = [0,\ -1]^\top$: $\widehat{\mathcal{R}}_r(\mathbf{x}^{(2)}) = [0,\ 1]^\top$, then $\|\widehat{\mathcal{R}}_r(\mathbf{x}^{(2)})\|_1 = 1$.

These examples clearly demonstrate that $\|\widehat{\mathcal{R}}_r(\mathbf{x})\|_1$ increases proportionally to the magnitude and number of negative components in $\mathbf{x}$. Thus, the RR-ℓ_1 regularizer acts as a convex and continuous penalty that discourages negative entries while preserving differentiability almost everywhere.

B. Proximity operator of RR-ℓ_1 regularizer

Theorem 1. For $\mathcal{F}_{\mu_1,\mu_2} : \mathbb{R}^N \to \mathbb{R}_+$ with $\mu_1, \mu_2 \in \mathbb{R}_{++}$,

$$\mathrm{prox}_{\gamma\mathcal{F}_{\mu_1,\mu_2}}(\mathbf{x}) = \big[f_{\gamma\mu_1,\gamma\mu_2}(x_1),\ldots,f_{\gamma\mu_1,\gamma\mu_2}(x_N)\big]^\top, \tag{19}$$

where

$$f_{\lambda_1,\lambda_2}(x) := \begin{cases} x-\lambda_1 & (x > \lambda_1),\\ 0 & (-\lambda_1-\lambda_2 \le x \le \lambda_1),\\ x+\lambda_1+\lambda_2 & (x < -\lambda_1-\lambda_2).\end{cases} \tag{20}$$

Proof: The function $\mathcal{F}_{\mu_1,\mu_2}$ is separable, so its proximity operator can be computed component-wise. We consider the scalar function

$$\mathcal{J}(y) := \gamma\mu_1|y| + \gamma\mu_2\max(-y, 0) + \frac{1}{2}(x-y)^2, \tag{21}$$

and aim to solve

$$\text{prox}_{\gamma\mathcal{F}_{\mu_1,\mu_2}}(x) = \underset{y\in\mathbb{R}}{\text{argmin}}\, \mathcal{J}(y). \tag{22}$$

We divide the real line into three regions and analyze the behavior of $\mathcal{J}(y)$ piecewise.

Case 1: $y > 0$
In this region, $|y| = y$ and $\max(-y, 0) = 0$. So:

$$\mathcal{J}(y) = \gamma\mu_1 y + \frac{1}{2}(x-y)^2. \tag{23}$$

Taking the derivative with respect to y and setting it to zero:

$$\frac{d\mathcal{J}}{dy} = \gamma\mu_1 - (x-y) = 0 \;\Rightarrow\; y = x - \gamma\mu_1. \tag{24}$$

We require $y > 0$, so this solution is valid if $x > \gamma\mu_1$.

Case 2: $y < 0$
In this region, $|y| = -y$ and $\max(-y, 0) = -y$. Thus:

$$\mathcal{J}(y) = -\gamma(\mu_1+\mu_2)y + \frac{1}{2}(x-y)^2. \tag{25}$$

Setting the derivative to zero:

$$\frac{d\mathcal{J}}{dy} = 0 \;\Rightarrow\; y = x + \gamma(\mu_1+\mu_2). \tag{26}$$

This is valid when $y < 0$, i.e., $x < -\gamma(\mu_1+\mu_2)$.

Case 3: $y = 0$
We check whether $y = 0$ is a minimizer by computing the subdifferential of $\mathcal{J}$ at $y = 0$:

$$\partial\mathcal{J}(0) = \gamma\mu_1[-1,1] + \gamma\mu_2[-1,0] - x. \tag{27}$$

This set is

$$[-\gamma(\mu_1+\mu_2) - x,\ \gamma\mu_1 - x]. \tag{28}$$

If $0 \in \partial\mathcal{J}(0)$, then

$$-\gamma(\mu_1+\mu_2) \le x \le \gamma\mu_1. \tag{29}$$

Thus, $y = 0$ is optimal when $x \in [-\gamma(\mu_1+\mu_2), \gamma\mu_1]$.

Combining all three cases and letting $\lambda_1 = \gamma\mu_1$, $\lambda_2 = \gamma\mu_2$, we obtain

$$f_{\lambda_1,\lambda_2}(x) = \begin{cases} x - \lambda_1 & \text{if } x > \lambda_1, \\ 0 & \text{if } -\lambda_1 - \lambda_2 \le x \le \lambda_1, \\ x + \lambda_1 + \lambda_2 & \text{if } x < -\lambda_1 - \lambda_2. \end{cases} \tag{30}$$

This completes the proof. ∎

C. Solution via the proximal-gradient method

Substituting $\mathcal{F}_{\mu_1,\mu_2}$ for the ℓ_1-norm in (11) gives

$$\mathbf{x}^\star := \underset{\mathbf{x}\in\mathbb{R}^N}{\text{argmin}}\, \tfrac{1}{2}\|\mathbf{\Phi x} - \mathbf{y}\|_2^2 + \mathcal{F}_{\mu_1,\mu_2}(\mathbf{x}), \tag{31}$$

solved by

$$\mathbf{x}^{(k+1)} = \text{prox}_{\alpha\mathcal{F}_{\mu_1,\mu_2}}\big(\mathbf{x}^{(k)} - \alpha\mathbf{\Phi}^\top(\mathbf{\Phi x}^{(k)} - \mathbf{y})\big), \tag{32}$$

which reduces to ISTA when $\mu_2 = 0$.

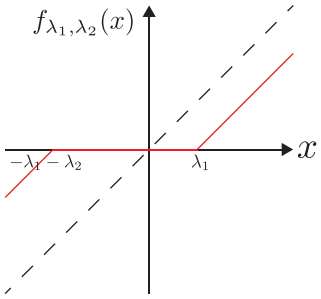


Fig. 3: Proximity operator f_{λ_1,λ_2}

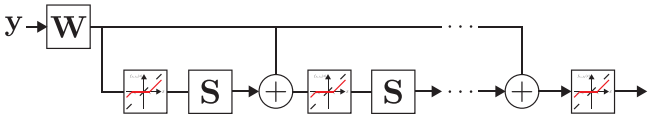


Fig. 4: Non-Negative LISTA (NNLISTA)

D. Deep-unrolled version: Non-Negative LISTA

Because NNISTA differs from ISTA only in its proximity operator, it can be unrolled analogously, yielding non-negative LISTA (NNLISTA). Rewriting

$$\mathbf{x}^{(k+1)} = \text{prox}_{\alpha\mathcal{F}_{\mu_1,\mu_2}}\big(\mathbf{S}\mathbf{x}^{(k)} + \mathbf{W}\mathbf{y}\big), \tag{33}$$

we unroll the iterations as in Fig. 4 and train the parameters by minimizing (15).

E. Comparison with conventional non-negative sparse-optimization

Conventional approaches impose non-negativity as an explicit constraint, leading to

$$\mathbf{x}^\star := \underset{\mathbf{x}\in\mathbb{R}^N}{\text{argmin}}\, \tfrac{1}{2}\|\mathbf{\Phi x} - \mathbf{y}\|_2^2 + \mu\|\mathbf{x}\|_1 \quad \text{s.t. } \mathbf{x} \ge \mathbf{0}, \tag{34}$$

which is typically solved by ADMM or PDS.

1) [Example (ADMM).]: For $f, g \in \Gamma_0(\mathbb{R}^N)$,

$$\underset{\mathbf{x},\mathbf{z}}{\text{argmin}}\, f(\mathbf{x}) + g(\mathbf{z}) \text{ s.t. } \mathbf{z} = \mathbf{L}\mathbf{x} \tag{35}$$

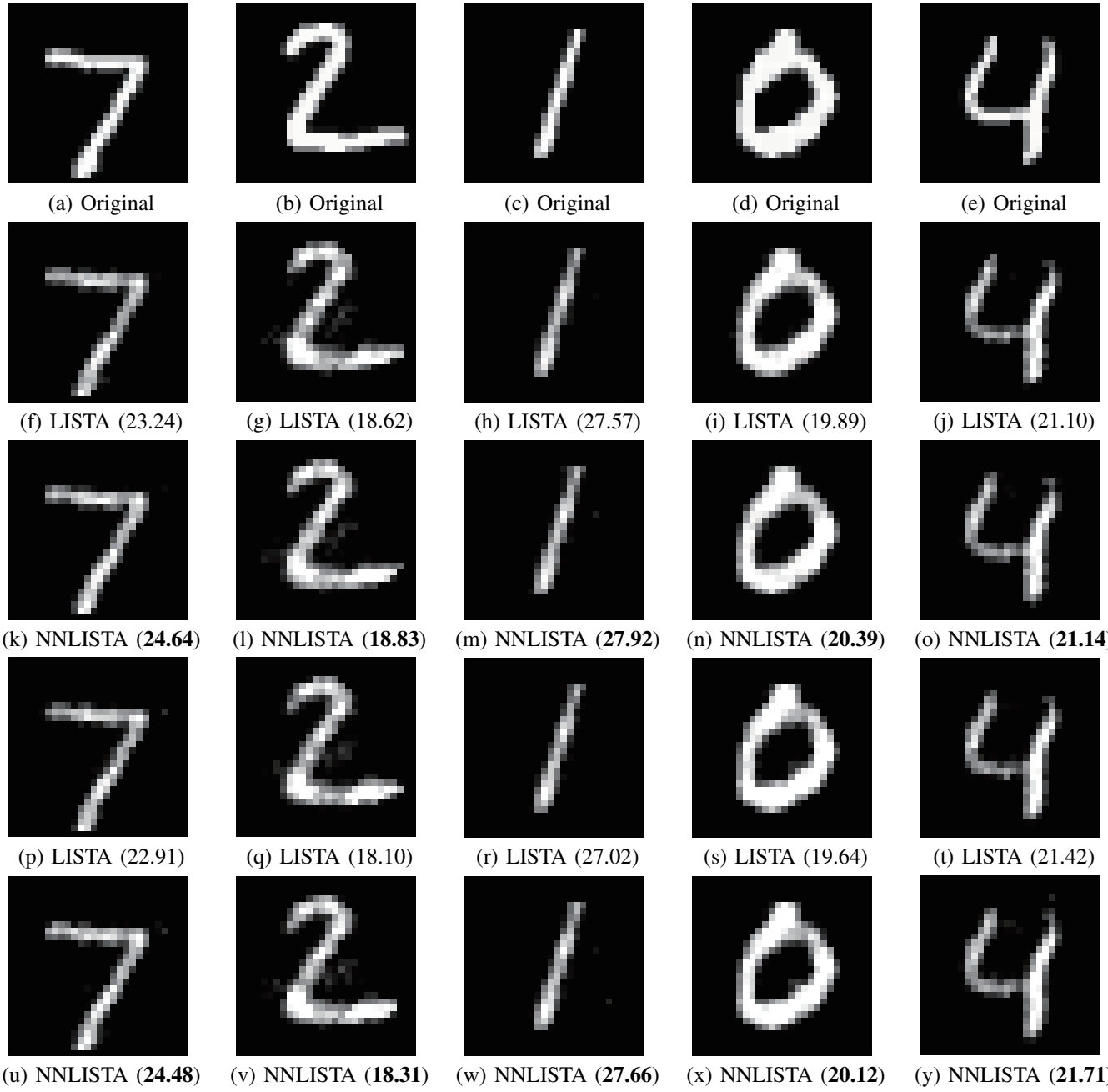
is solved iteratively via

$$\begin{cases} \mathbf{x}^{(k+1)} = \underset{\mathbf{x}}{\text{argmin}}\, f(\mathbf{x}) + \frac{\eta}{2}\|\mathbf{L}\mathbf{x} - \mathbf{z}^{(k)} + \mathbf{u}^{(k)}\|_2^2, \\ \mathbf{z}^{(k+1)} = \underset{\mathbf{z}}{\text{argmin}}\, g(\mathbf{z}) + \frac{\eta}{2}\|\mathbf{L}\mathbf{x}^{(k+1)} - \mathbf{z} + \mathbf{u}^{(k)}\|_2^2, \\ \mathbf{u}^{(k+1)} = \mathbf{u}^{(k)} + \mathbf{L}\mathbf{x}^{(k+1)} - \mathbf{z}^{(k+1)}. \end{cases}$$

Choosing $f(\mathbf{x}) = \frac{1}{2}\|\mathbf{\Phi x} - \mathbf{y}\|_2^2$, $g(\mathbf{z}) = \mu\|\mathbf{z}_1\|_1 + \iota_{\{\ge\mathbf{0}\}}(\mathbf{z}_2)$, $\mathbf{L} = [\mathbf{I}^\top\ \mathbf{I}^\top]^\top$ gives Algorithm 1. A drawback is the matrix inversion required in large-scale problems.

IV. Experimental Results

We evaluated the proposed NNLISTA through two experiments: sparse recovery of synthetic non-negative signals and compressed sensing reconstruction on MNIST images. In both cases, NNLISTA is compared against conventional LISTA.



(a) Original (b) Original (c) Original (d) Original (e) Original

(f) LISTA (23.24) (g) LISTA (18.62) (h) LISTA (27.57) (i) LISTA (19.89) (j) LISTA (21.10)

(k) NNLISTA (**24.64**) (l) NNLISTA (**18.83**) (m) NNLISTA (**27.92**) (n) NNLISTA (**20.39**) (o) NNLISTA (**21.14**)

(p) LISTA (22.91) (q) LISTA (18.10) (r) LISTA (27.02) (s) LISTA (19.64) (t) LISTA (21.42)

(u) NNLISTA (**24.48**) (v) NNLISTA (**18.31**) (w) NNLISTA (**27.66**) (x) NNLISTA (**20.12**) (y) NNLISTA (**21.71**)

Fig. 5: Compressed sensing reconstruction results on MNIST data using LISTA and NNLISTA. (f)–(o): $\sigma = 0.01$, (p)–(y): $\sigma = 0.1$. Numbers indicate PSNR [dB].

A. Sparse Recovery of Non-Negative Vectors

We first compared LISTA and NNLISTA under a synthetic sparse–coding setup. The dictionary $\mathbf{\Phi} \in \mathbb{R}^{70\times 100}$ was generated by sampling each entry from a zero-mean, unit-variance Gaussian distribution. The ground-truth coefficient vector $\mathbf{x} \in \mathbb{R}^{100}$ was constructed by selecting ten percent of the indices $\mathcal{N} \subset \{1, \ldots, 100\}$ uniformly at random and setting $x_n = 1$ for $n \in \mathcal{N}$, and $x_n = 0$ otherwise. Observations were generated as in (10), where $\mathbf{n}$ is additive white Gaussian noise with standard deviation $\sigma \in \{0.01, 0.03, 0.05\}$. We generated 100000 training pairs and 1000 test pairs. Both LISTA and NNLISTA were implemented with 15 layers. LISTA learned $\{\mu_k\}_{k=1}^{15}$, while NNLISTA learned both $\{\mu_{k,1}\}_{k=1}^{15}$ and $\{\mu_{k,2}\}_{k=1}^{15}$. Weight matrices $\mathbf{W}$ and $\mathbf{S}$ were trained and shared across layers. Training used the Huber loss and ran for 500 epochs. Performance was evaluated using the SNR [dB] between the reconstructed signal $\mathbf{\Phi}\widetilde{\mathbf{x}}$ and the ground truth $\mathbf{\Phi}\mathbf{x}^{(\text{Test})}$.

As shown in Table I, the proposed method (NNLISTA) achieves consistently higher reconstruction accuracy com-

Algorithm 1 ADMM for solving (34)

$k \leftarrow 0$. Initialize $\mathbf{x}^{(0)}, \mathbf{z}_1^{(0)}, \mathbf{z}_2^{(0)}, \mathbf{u}_1^{(0)}, \mathbf{u}_2^{(0)}$.
while $k = 0$ or $\|\mathbf{x}^{(k)} - \mathbf{x}^{(k-1)}\|_2 > \epsilon_{\text{stop}}$ **do**
  $\mathbf{x}^{(k+1)} \leftarrow (\mathbf{\Phi}^\top \mathbf{\Phi} + 2\eta \mathbf{I})^{-1}\big(\mathbf{\Phi}^\top \mathbf{y} + \eta(\mathbf{z}_1^{(k)} + \mathbf{z}_2^{(k)}) - \eta(\mathbf{u}_1^{(k)} + \mathbf{u}_2^{(k)})\big)$
  $\mathbf{z}_1^{(k+1)} \leftarrow \text{Soft}_{\mu/\eta}\big(\mathbf{x}^{(k+1)} + \mathbf{u}_1^{(k)}\big)$
  $\mathbf{z}_2^{(k+1)} \leftarrow \max\big(\mathbf{x}^{(k+1)} + \mathbf{u}_2^{(k)}, 0\big)$
  $\mathbf{u}_1^{(k+1)} \leftarrow \mathbf{u}_1^{(k)} + \mathbf{x}^{(k+1)} - \mathbf{z}_1^{(k+1)}$
  $\mathbf{u}_2^{(k+1)} \leftarrow \mathbf{u}_2^{(k)} + \mathbf{x}^{(k+1)} - \mathbf{z}_2^{(k+1)}$
  $k \leftarrow k + 1$
end while
Output $\mathbf{x}^{(k)}$.

TABLE I: Results of sparse coding

	Average SNR [dB]			Num. negative entries	
Noise	Observation	LISTA	NNLISTA	LISTA	NNLISTA
$\sigma = 0.01$	26.37	27.31	**27.95**	53	0
$\sigma = 0.03$	16.83	20.39	**20.73**	502	0
$\sigma = 0.05$	12.39	16.76	**17.06**	610	0

pared to LISTA. Since the true coefficients are non-negative, NNLISTA's explicit non-negativity modeling better matches the data and improves estimation performance. Notably, the total number of negative entries of all the reconstructed signals obtained by NNLISTA was zero, confirming the effectiveness of the non-negative regularization.

B. Compressed Sensing on MNIST Images

Next, we evaluated NNLISTA on compressed sensing reconstruction of MNIST digits. Each 28×28 image ($\mathbf{x} \in [0,1]^{784}$) was compressed into $M = 100$ random linear measurements using a Gaussian sensing matrix $\mathbf{\Phi} \in \mathbb{R}^{100\times 784}$. Additive Gaussian noise with standard deviation $\sigma \in \{0.01, 0.03, 0.05, 0.1\}$ was added to the compressed signals. A total of 100 images were sampled from the MNIST test set for evaluation. The training set consisted of 60000 images and the test set consisted of 10000 images. Both LISTA and NNLISTA were implemented with 15 layers and trained for 100 epochs using a learning rate of 5×10^{-4}. PSNR between the reconstructed image and the original ground-truth image was used as the evaluation metric. As shown in Table II, NNLISTA achieves slightly but consistently higher SNR compared to LISTA across all noise levels. Although NNLISTA produces a small number of negative entries, as indicated by the total number of images containing negative values and the total number of negative elements in Table II, the proportion is extremely small relative to the 10,000 test images. This confirms that the non-negative regularization is effectively enforced and is beneficial in realistic image recovery scenarios, especially when the target signals are known to be non-negative.

V. Conclusion

We proposed an approach to non-negative sparse optimization by combining the ℓ_1-norm with a reflected ReLU to build a convex regularizer that simultaneously promotes sparsity and non-negativity. We derived its proximity operator in closed form and developed an efficient proximal-gradient algorithm (NNISTA). By unrolling NNISTA, we obtained NNLISTA, a deep model that inherits the interpretability of optimization algorithms while benefiting from data-driven parameter learning. Numerical experiments on compressed-sensing data confirmed the effectiveness of the proposed method.

TABLE II: Results of compressed sensing

	Average SNR [dB]		Num. neg. images and entries	
Noise	LISTA	NNLISTA	LISTA	NNLISTA
$\sigma = 0.01$	21.65	**22.12**	382, 463	30, 34
$\sigma = 0.03$	21.79	**21.86**	469, 603	18, 23
$\sigma = 0.05$	21.58	**21.81**	597, 821	22, 25
$\sigma = 0.10$	20.99	**21.46**	679, 1010	22, 26

References

[1] F. Bach, R. Jenatton, J. Mairal, and G. Obozinski, "Optimization with sparsity-inducing penalties," *Found. Trends Mach. Learn.*, vol. 4, no. 1, pp. 1–106, Jan. 2012.

[2] P. L. Combettes and J.-C. Pesquet, "A proximal decomposition method for solving convex variational inverse problems," *Inverse Problems*, vol. 24, no. 6, p. 065014, Nov. 2008.

[3] M. V. Afonso, J. M. B.-Dias, and M. A. T. Figueiredo, "An augmented lagrangian approach to the constrained optimization formulation of imaging inverse problems," *IEEE Trans. Image Process.*, vol. 20, no. 3, pp. 681–695, Mar. 2011.

[4] H. H. Bauschke and P. L. Combettes, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*. New York, NY, USA: Springer-Verlag, 2011.

[5] A. Beck, *First-Order Methods in Optimization*. Philadelphia, PA: Society for Industrial and Applied Mathematics, 2017.

[6] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Phys. D*, vol. 60, no. 1-4, pp. 259–268, Nov. 1992.

[7] X. Bresson and T. F. Chan, "Fast dual minimization of the vectorial total variation norm and applications to color image processing," *Inverse Probl. Imag*, vol. 2, no. 4, pp. 455–484, Nov. 2008.

[8] R. H. Chan, Y. Dong, and M. Hintermuller, "An efficient two-phase L^1-TV method for restoring blurred images with impulse noise," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1731–1739, Jul. 2010.

[9] I. Bayram and M. E. Kamasak, "Directional total variation," *IEEE Signal Process. Letters*, vol. 19, no. 12, pp. 781–784, Sep. 2012.

[10] H. Zhang, X. Huang, and D. N. Metaxas, "Iterative positive thresholding algorithm for non-negative sparse regularization," *Optimization*, vol. 67, no. 9, pp. 1575–1596, 2018.

[11] T. Kawaguchi, K. Okada, and H. Fujisawa, "Asymmetric LASSO for variable selection with sign-specific shrinkage," *Journal of Data Science*, vol. 19, no. 3, pp. 453–472, 2021.

[12] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 6, pp. 1336–1353, 2013.

[13] Y. Iwase and D. Kitamura, "Supervised audio source separation based on nonnegative matrix factorization with cosine similarity penalty," *IEICE Transactions on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E105.A, no. 6, pp. 906–913, 2022.

[14] K. Gregor and Y. LeCun, "Learning fast approximations of sparse coding," in *Proceedings of the 27th International Conference on Machine Learning (ICML)*, 2010, pp. 399–406.

[15] Y. Yang, J. Sun, H. Li, and Z. Xu, "Deep ADMM-Net for compressive sensing MRI," in *Advances in Neural Information Processing Systems*, D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett, Eds., vol. 29. Curran Associates, Inc., 2016.

[16] Z. Zhang, Y. Liu, J. Liu, F. Wen, and C. Zhu, "AMP-Net: Denoising-based deep unfolding for compressive image sensing," *IEEE Trans. Image Process.*, vol. 30, pp. 1487–1500, 2021.

[17] A. Chambolle, R. De Vore, N.-Y. Lee, and B. Lucier, "Nonlinear wavelet image processing: variational problems, compression, and noise removal through wavelet shrinkage," *IEEE Transactions on Image Processing*, vol. 7, no. 3, pp. 319–335, 1998.