Learned Successive Convex Approximation for Sparse Signal Recovery

Moritz Hemsing
Technische Universität Darmstadt
Darmstadt, Germany
m.hemsing@nt.tu-darmstadt.de

Lukas Schynol
Technische Universität Darmstadt
Darmstadt, Germany
lschynol@nt.tu-darmstadt.de

Marius Pesavento
Technische Universität Darmstadt
Darmstadt, Germany
pesavento@nt.tu-darmstadt.de

***Abstract*—Sparse signal recovery using the least absolute shrinkage and selection operator (LASSO) and model-aided deep learning is considered. We propose the Learned Soft-Thresholding with Exact Line search Algorithm (LSTELA) deep network architecture which is based on unrolling a successive convex approximation algorithm. LSTELA incorporates debiasing through the use of the soft-thresholding with support selection operator and revised step size rules as well as instance-adaptive parameters. In addition, we introduce a novel piecewise differentiable approximation of the soft-thresholding with support selection operator, which allows all model parameters to be learned end-to-end, avoiding a computationally costly gradient-free search. We further propose a lightweight LSTELA variant to improve the computational efficiency. Compared to state-of-the-art model-aided network architectures, the proposed unrolling-based methods achieve a lower MSE in the case of noise-contaminated measurements while exhibiting excellent adaptation capabilities in case of changing data distributions.**

***Index Terms*—deep learning, deep unrolling, compressed sensing, successive convex approximation, sparsity, L1-tail-minimization**

I. Introduction

Sparse signal recovery is a critical task in signal processing [1], exploited in diverse applications such as image-processing [2], tomography [3], and communications [4]. In this context, the least absolute shrinkage and selection operator (LASSO) formulation is prevalent. Leveraging the convexity of the corresponding optimization problem, numerous low-complexity iterative solvers have been developed to obtain LASSO solutions. Classical examples include the Iterate Soft-Thresholding Algorithm (ISTA) [5], the Fast Iterate Soft-Thresholding Algorithm (FISTA) [6], and Least Angle Regression (LARS) [7]. Relying on the successive convex approximation (SCA) optimization approach, the alternative Soft-Thresholding with Exact Line search Algorithm (STELA) works particularly well in case of sparse problem instances [8]. As an extension, N-STELA has recently been proposed, employing a Nesterov-like momentum and empirically yielding a faster convergence rate compared to STELA and FISTA [9].

In recent years, deep unrolling has emerged as a powerful instrument in sparse signal recovery [10], [11]. Deep unrolling-based deep neural network (DNN) architectures rely on the truncation of classical iterative algorithms, where each iteration is reinterpreted as a DNN layer and then modified, thereby introducing learnable weights. Thus, deep unrolling-based DNNs retain the original algorithm's structure and interpretability while mitigating model mismatch by leveraging data and reducing the computational cost.

A well-known application of deep unrolling is Learned ISTA (LISTA) in [12], where linear operators and the sparsity regularization are replaced by learnable weights. The authors of [13] proposed Analytic LISTA (ALISTA), which exploits analytically derived linear operators to significantly reduce the number of learnable weights and increase the convergence rate. In addition, a support selection excludes a "trusted" proportion of entries from soft-thresholding, effectively reducing the bias of the LASSO-based estimate. In [14], HyperLISTA is proposed, which further incorporates momentum as well as instance-adaptive thresholds and support selection, thereby improving the robustness of the DNN against domain changes. However, the proposed architectures in [13] and [14] are non-differentiable w.r.t. the proportion of trusted support elements, requiring manual tuning of the parameters.

Thus, inspired by [8], [9], [13], [14] we make the following contributions:

- Utilizing soft-to-hard annealing [15], we propose a novel piecewise differentiable soft-thresholding with support selection operator that enables end-to-end training.
- We unroll N-STELA into the Learned STELA (LSTELA) DNN architecture and the computationally more efficient lightweight LSTELA (L-LSTELA) by integrating the piecewise differentiable support selection into the DNN as a bias-reducing technique and adopting an instance-adaptive parametrization.
- We empirically demonstrate improved achieved mean squared error (MSE) in case of noisy measurements as well as enhanced adaptation capabilities compared to state-of-the-art methods.

This work was financially supported by the Federal Ministry of Education and Research of Germany in the project "Open6GHub" (grant no. 16KISK014).

II. Sparse Signal Recovery

A. Data Model and Sparse Signal Recovery

We consider the linear forward model

$$\boldsymbol{b} = \boldsymbol{A}\boldsymbol{x}^\star + \boldsymbol{\varepsilon}, \tag{1}$$

where $\boldsymbol{b} \in \mathbb{R}^M$ is an observation vector, $\boldsymbol{x}^\star \in \mathbb{R}^N$ is the underlying sparse representation vector, $\boldsymbol{A} \in \mathbb{R}^{M \times N}$ a dictionary matrix, and $\boldsymbol{\varepsilon} \in \mathbb{R}^M$ is additive noise. The columns $\boldsymbol{a}_i \in \mathbb{R}^M$ of $\boldsymbol{A}$ are the kernels associated with entries $[\boldsymbol{x}^\star]_i$.

To recover $\boldsymbol{x}^\star$ from $\boldsymbol{b}$ for a known upper bound sparsity level $\|\boldsymbol{x}^\star\|_0 \le s$, under the assumption of zero-mean Gaussian noise $\boldsymbol{\varepsilon}$, the maximum likelihood estimator for $\boldsymbol{x}^\star$ [16]

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 \qquad \text{s.t.} \quad \|\boldsymbol{x}\|_0 \le s \tag{2}$$

can be formulated. Since the problem in (2) is nonconvex and finding a global optimum is NP-hard, the ℓ_0-"norm" is typically relaxed to the convex but non-differentiable ℓ_1-norm, which still promotes sparsity in the solution. The Lagrangian form of the resulting so-called LASSO is given as the convex optimization problem

$$\min_{\boldsymbol{x} \in \mathbb{R}^n} \quad \frac{1}{2}\|\boldsymbol{A}\boldsymbol{x} - \boldsymbol{b}\|_2^2 + \lambda\|\boldsymbol{x}\|_1, \tag{3}$$

where $\lambda > 0$ is a regularization parameter that controls the sparsity of the minimizer $\widehat{\boldsymbol{x}}(\lambda)$, which approximates the targeted sparse representation $\boldsymbol{x}^\star$.

B. STELA and N-STELA

The classical ISTA and FISTA perform a descent step based on the ℓ_2-component of (3) and subsequently apply a shrinkage, that accounts for the regularization, to find the minimizer $\widehat{\boldsymbol{x}}(\lambda)$ of (3) [5], [6]. In comparison, the STELA is based on SCA [8]. In each iteration $\ell + 1$, it first obtains a closed-form solution vector

$$\mathbb{B}\boldsymbol{x}^{(\ell)} = \boldsymbol{d}_{\boldsymbol{A}}^{-1} \odot \mathcal{S}_{\lambda\mathbf{1}}\left(\boldsymbol{d}_{\boldsymbol{A}} \odot \boldsymbol{x}^{(\ell)} - \boldsymbol{A}^{\mathrm{T}}\left(\boldsymbol{A}\boldsymbol{x}^{(\ell)} - \boldsymbol{b}\right)\right), \tag{4}$$

where each entry $\left[\mathbb{B}\boldsymbol{x}^{(\ell)}\right]_i$ for $i = 1, \dots, N$ minimizes a local approximation of the LASSO in (3) for fixed $\boldsymbol{x}_{-i} = \boldsymbol{x}_{-i}^{(\ell)}$:

$$\left[\mathbb{B}\boldsymbol{x}^{(\ell)}\right]_i = \underset{[\boldsymbol{x}]_i \in \mathbb{R}}{\arg\min}\, \frac{1}{2}\left\|\boldsymbol{a}_i[\boldsymbol{x}]_i + \boldsymbol{A}_{-i}\boldsymbol{x}_{-i}^{(\ell)} - \boldsymbol{b}\right\|_2^2 + \lambda|[\boldsymbol{x}]_i|. \tag{5}$$

Here, $[\boldsymbol{d}_{\boldsymbol{A}}]_i = \|\boldsymbol{a}_i\|_2^2$, $\boldsymbol{d}_{\boldsymbol{A}}^{-1}$ denotes the elementwise inverse of $\boldsymbol{d}_{\boldsymbol{A}}$ and $\mathcal{S}_{\boldsymbol{a}}(\boldsymbol{z})$ the soft-thresholding operator, defined as

$$[\mathcal{S}_{\boldsymbol{a}}(\boldsymbol{z})]_i = [|[\boldsymbol{z}]_i| - [\boldsymbol{a}]_i]_0^\infty \operatorname{sgn}([\boldsymbol{z}]_i). \tag{6}$$

In (5), $\boldsymbol{x}_{-i}$ denotes the vector containing all elements of $\boldsymbol{x}$ except $[\boldsymbol{x}]_i$ and $\boldsymbol{A}_{-i}$ refers to $\boldsymbol{A}$ without the kernel $\boldsymbol{a}_i$.

The STELA iterate $\boldsymbol{x}_{\text{stela}}^{(\ell)}$ is a step into the descent direction of the SCA minimizer

$$\boldsymbol{x}_{\text{stela}}^{(\ell)} = \boldsymbol{x}^{(\ell)} + \gamma_1^{(\ell)}\left(\mathbb{B}\boldsymbol{x}^{(\ell)} - \boldsymbol{x}^{(\ell)}\right), \tag{9}$$

where an exact line search on the majorization function

$$\gamma_1^{(\ell)} = \underset{\gamma \in [0,1]}{\arg\min}\left\{\frac{1}{2}\left\|\boldsymbol{A}\left(\boldsymbol{x}^{(\ell)} + \gamma\left(\mathbb{B}\boldsymbol{x}^{(\ell)} - \boldsymbol{x}^{(\ell)}\right)\right) - \boldsymbol{b}\right\|_2^2 + \gamma\lambda\left(\|\mathbb{B}\boldsymbol{x}^{(\ell)}\|_1 - \|\boldsymbol{x}^{(\ell)}\|_1\right)\right\} \tag{10}$$

yields $\gamma_1^{(\ell)}$ in (7) in closed form.

N-STELA [9] introduces a subsequent second descent step along the past trajectory $(\boldsymbol{x}_{\text{stela}}^{(\ell)} - \boldsymbol{x}^{(\ell-1)})$ similar to Nesterov momentum [17] as

$$\boldsymbol{x}^{(\ell+1)} = \boldsymbol{x}_{\text{stela}}^{(\ell)} + \gamma_2^{(\ell)}\left(\boldsymbol{x}_{\text{stela}}^{(\ell)} - \boldsymbol{x}^{(\ell-1)}\right) \tag{11}$$

to improve the convergence speed. The step size $\gamma_2^{(\ell)}$ can be derived by similarly minimizing a tight upper bound on the LASSO objective, resulting in the closed-form solution (8).

III. Unrolled STELA

When leveraging the LASSO for the task of sparse signal recovery, several problems arise. First, due to the convex relaxation of the ℓ_0 constraint, the LASSO introduces a bias towards $\mathbf{0}$ for the support entries of $\boldsymbol{x}^\star$ [18]. Secondly, the regularization parameter λ that obtains the best estimate is generally unknown. Thirdly, arriving at an estimate may require a substantial amount of iterations. To tackle these problems, we propose to apply deep unrolling to the N-STELA algorithm. We first discuss debiasing the N-STELA algorithm in III-A before unrolling it in Sec. III-B.

A. Differentiable Support Selection for Bias Reduction

Numerous approaches have been proposed to reduce the bias resulting from LASSO-based methods, thereunder tail-ℓ_1-minimization [19] and related methods such as (recursive) Tail-FISTA [20], [21], Tail-STELA [22], iterative support detection [23] or a bias correction for the final estimate [24].

To reduce the bias of the N-STELA, the local minimizer $\mathbb{B}\boldsymbol{x}^{(\ell)}$ and the step sizes $\gamma_1^{(\ell)}$ and $\gamma_2^{(\ell)}$ need to be considered.

Local Minimizer. Inspired by [13] and [14], we replace the soft-thresholding operator $\mathcal{S}_{\boldsymbol{a}}(\boldsymbol{z})$ by the support selection operator [25], which is related to tail-ℓ_1-minimization [20]–[22]. Let p be the proportion of elements which are "trusted" to be in the support set. Then, defining $\mathcal{Q}_{1-p}(\boldsymbol{z})$ as the $(1-p)$-quantile of the absolute values $|[\boldsymbol{z}]_i|$, the support selection operator $\mathcal{S}^p_{\mathbf{a}}(\boldsymbol{z})$ is defined as

$$[\mathcal{S}^p_{\mathbf{a}}(\boldsymbol{z})]_i = \begin{cases} [\boldsymbol{z}]_i, & |[\boldsymbol{z}]_i| > [\boldsymbol{a}]_i \wedge |[\boldsymbol{z}]_i| \ge \mathcal{Q}_{1-p}(\boldsymbol{z}) \\ [\mathcal{S}_{\boldsymbol{a}}(\boldsymbol{z})]_i & \text{otherwise.} \end{cases} \tag{12}$$

The support selection operator in (12) excludes all entries of $\boldsymbol{z}$ from regularization that both are among the pN entries largest in magnitude *and* exceed the threshold $\boldsymbol{a}$, while soft-thresholding is applied otherwise.

Similar to λ in (4), the proportion of trusted support elements p is a design choice that is a-priori unknown. However, a disadvantage of the support selection operator is that the partial derivative w.r.t. p does not exist, thus a "good" p must be found through manual tuning or a grid search [13], [14],

$$\gamma_1^{(\ell)} = \left[-\left(\left(\boldsymbol{A}\boldsymbol{x}^{(\ell)} - \boldsymbol{b} \right)^{\mathrm{T}} \boldsymbol{A} \left(\mathbb{B}\boldsymbol{x}^{(\ell)} - \boldsymbol{x}^{(\ell)} \right) + \lambda \left(\|\mathbb{B}\boldsymbol{x}^{(\ell)}\|_1 - \|\boldsymbol{x}^{(\ell)}\|_1 \right) \right) \Big/ \left\| \boldsymbol{A} \left(\mathbb{B}\boldsymbol{x}^{(\ell)} - \boldsymbol{x}^{(\ell)} \right) \right\|_2^2 \right]_0^1 \tag{7}$$

$$\gamma_2^{(\ell)} = \left[-\left(\left(\boldsymbol{A}\boldsymbol{x}_{\mathrm{stela}}^{(\ell)} - \boldsymbol{b} \right)^{\mathrm{T}} \boldsymbol{A} \left(\boldsymbol{x}_{\mathrm{stela}}^{(\ell)} - \boldsymbol{x}^{(\ell-1)} \right) + \lambda \left(\|2\boldsymbol{x}_{\mathrm{stela}}^{(\ell)} - \boldsymbol{x}^{(\ell-1)}\|_1 - \|\boldsymbol{x}_{\mathrm{stela}}^{(\ell)}\|_1 \right) \right) \Big/ \left\| \boldsymbol{A} \left(\boldsymbol{x}_{\mathrm{stela}}^{(\ell)} - \boldsymbol{x}^{(\ell-1)} \right) \right\|_2^2 \right]_0^1 \tag{8}$$

Algorithm 1 Learned STELA (LSTELA)

```
Input $\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{\theta} = (c_1, (c_2^{(\ell)})_{\ell=1}^L), L$
Initialize $\boldsymbol{x}^{(-1)} = \mathbf{0}, \boldsymbol{x}^{(0)} = \mathbf{0}$
for $\ell = 0, \dots, L-1$ do
    $\mathbb{B}\boldsymbol{x}_{\mathrm{ss}}^{(\ell)} \leftarrow$ Eq. (17)
    $\gamma_{\mathrm{ub},1}^{(\ell)} \leftarrow$ Eq. (15) using $\mathbb{B}\boldsymbol{x}_{\mathrm{ss}}^{(\ell)}$
    $\boldsymbol{x}_{\mathrm{stela,ss}}^{(\ell)} \leftarrow \boldsymbol{x}^{(\ell)} + \gamma_{\mathrm{ub},1}^{(\ell)} (\mathbb{B}\boldsymbol{x}_{\mathrm{ss}}^{(\ell)} - \boldsymbol{x}^{(\ell)})$
    $\gamma_{\mathrm{ub},2}^{(\ell)} \leftarrow$ Eq. (16) using $\boldsymbol{x}_{\mathrm{stela,ss}}^{(\ell)}$
    $\boldsymbol{x}^{(\ell+1)} = \boldsymbol{x}_{\mathrm{stela,ss}}^{(\ell)} + \gamma_{\mathrm{ub},2}^{(\ell)} (\boldsymbol{x}_{\mathrm{stela,ss}}^{(\ell)} - \boldsymbol{x}^{(\ell-1)})$
return $\boldsymbol{x}^{(L)}$
```

[25]. To remediate this problem, we propose a novel smoothed and piecewise differentiable support selection operator

$$\widehat{\mathcal{S}}_{\mathbf{a}}^{p}(\boldsymbol{z}) = \mathcal{S}_{\mathbf{a}}(\boldsymbol{z}) + \mathcal{C}_{\mathbf{a}}^{p}(\mathcal{S}_{\mathbf{a}}(\boldsymbol{z})) \tag{13}$$

as an approximation of (12), where the correction term $\mathcal{C}_{\mathbf{a}}^{p}(\cdot)$ mitigates the offset due to soft-thresholding in the support set. In particular, let $\boldsymbol{r}(\boldsymbol{z}) : \mathbb{R}^N \to \{1, \dots, N\}^N$ be a function which ranks the magnitude of the elements $[\boldsymbol{z}]_i$, i.e., for all $i \neq j$ we have $[\boldsymbol{r}(\boldsymbol{z})]_i \neq [\boldsymbol{r}(\boldsymbol{z})]_j$ and $[\boldsymbol{r}(\boldsymbol{z})]_i < [\boldsymbol{r}(\boldsymbol{z})]_j \Rightarrow |[\boldsymbol{z}]_i| \leq |[\boldsymbol{z}]_j|$. Then $\mathcal{C}_{\boldsymbol{a}}^{p}(\boldsymbol{z})$ is defined elementwise as

$$[\mathcal{C}_{\boldsymbol{a}}^{p}(\boldsymbol{z})]_i = \mathrm{sgn}([\boldsymbol{z}]_i)\, \mathrm{sig}\left(\frac{1}{\tau} \left(\frac{[\boldsymbol{r}(\boldsymbol{z})]_i}{N} - (1-p) \right) \right) [\boldsymbol{a}]_i, \tag{14}$$

where $\mathrm{sig}(z) = 1/(1 + \mathrm{e}^{-z})$ is the logistic function and $\tau > 0$ is an annealing parameter that determines how well $\mathrm{sig}(\cdot)$ approximates the unit step function. During training, τ can either be kept constant or decayed to gradually evolve the logistic function towards a hard threshold as $\tau \to 0$ [15]. Note that the smoothed support selection increases the computational cost during the training phase, requiring a sorting operation with $\mathcal{O}(n \log n)$ complexity, while the nondifferentiable support selection has $\mathcal{O}(n)$ complexity [26].

Step size. The bias-reduced local minimizer omits thresholding the largest residuals in (4). To keep the line searches consistent, we suggest to drop the regularization term in (10) and analogously for the N-STELA-step, leading to

$$\gamma_{\mathrm{ub},1}^{(\ell)} = \left[-\frac{\left(\boldsymbol{A}\boldsymbol{x}^{(\ell)} - \boldsymbol{b} \right)^{\mathrm{T}} \boldsymbol{A} \left(\mathbb{B}\boldsymbol{x}^{(\ell)} - \boldsymbol{x}^{(\ell)} \right)}{\left\| \boldsymbol{A} \left(\mathbb{B}\boldsymbol{x}^{(\ell)} - \boldsymbol{x}^{(\ell)} \right) \right\|_2^2} \right]_0^1, \tag{15}$$

$$\gamma_{\mathrm{ub},2}^{(\ell)} = \left[-\frac{\left(\boldsymbol{A}\boldsymbol{x}_{\mathrm{stela}}^{(\ell)} - \boldsymbol{b} \right)^{\mathrm{T}} \boldsymbol{A} \left(\boldsymbol{x}_{\mathrm{stela}}^{(\ell)} - \boldsymbol{x}^{(\ell-1)} \right)}{\left\| \boldsymbol{A} \left(\boldsymbol{x}_{\mathrm{stela}}^{(\ell)} - \boldsymbol{x}^{(\ell-1)} \right) \right\|_2^2} \right]_0^1. \tag{16}$$

B. Learned STELA

We now unroll L iterations of N-STELA to define the model-aided Learned STELA DNN architecture. Based on (4) and leveraging the piecewise differentiable support selection operator, we define a bias-reduced local minimizer

$$\mathbb{B}\boldsymbol{x}_{\mathrm{ss}}^{(\ell)} = \widehat{\mathcal{S}}_{\lambda^{(\ell)}\mathbf{1}}^{p^{(\ell)}} \left(\boldsymbol{x}^{(\ell)} - \boldsymbol{A}^{\mathrm{T}} \left(\boldsymbol{A}\boldsymbol{x}^{(\ell)} - \boldsymbol{b} \right) \right), \tag{17}$$

where we assumed normalized dictionary kernels, i.e., $\boldsymbol{d}_{\boldsymbol{A}} = \mathbf{1}$. In (17), $\lambda^{(\ell)}$ and $p^{(\ell)}$ become learnable parameters of layer ℓ. However, adopted from the work in [14], we select the thresholds and proportions of trusted support dependent on the instantiation of $(\boldsymbol{A}, \boldsymbol{b}, \boldsymbol{x}^{(\ell)})$. Specifically, we define

$$\lambda^{(\ell)} = c_1 \mu_A \left\| \boldsymbol{A}^{\dagger} \left(\boldsymbol{A}\boldsymbol{x}^{(\ell)} - \boldsymbol{b} \right) \right\|_1, \tag{18}$$

$$p^{(\ell)} = c_2^{(\ell)} \left[\log \left(\left\| \boldsymbol{A}^{\dagger}\boldsymbol{b} \right\|_1 \Big/ \left\| \boldsymbol{A}^{\dagger} \left(\boldsymbol{A}\boldsymbol{x}^{(\ell)} - \boldsymbol{b} \right) \right\|_1 \right) \right]_0^1 \tag{19}$$

with the hyperparameters c_1 and $c_2^{(\ell)}$. In (18), μ_A denotes the mutual coherence $\max_{i \neq j} \left| \boldsymbol{a}_i^{\mathrm{T}} \boldsymbol{a}_j \right|$. The intuition is to estimate the projected noise and residual error on a per-instance basis and, if noise and the residual error after layer ℓ is assumed to be low, decrease the threshold in (18) while simultaneously expanding the trust region in (19). In contrast to [14], we permit $c_2^{(\ell)}$ to differ over ℓ, which is feasible as $\boldsymbol{x}^{(L)}$ is piecewise differentiable in $p^{(\ell)}$ due to the smoothed support selection. The proposed LSTELA method is summarized in Alg. 1, where L is the number of layers and $\boldsymbol{\theta} = (c_1, (c_2^{(\ell)})_{\ell=1}^L)$ are the trainable weights.

Note that, compared to LSTELA, the methods proposed in [13], [14] require the iterative precomputation of a weight matrix to update the iterates for each dictionary matrix $\boldsymbol{A}$, which improves the convergence rate but can incur large computational costs depending on the dictionary's size. Moreover, the instance-adaptivity requires the precomputation of the pseudoinverse $\boldsymbol{A}^{\dagger}$, and additional matrix-vector products, which increases the computational cost of both LSTELA and HyperLISTA [14]. Hence, we additionally propose a computationally more efficient variant of LSTELA, denoted as lightweight LSTELA (L-LSTELA), which replaces $\boldsymbol{A}^{\dagger}$ with $\boldsymbol{A}^{\mathrm{T}}$, consequently reusing the matrix-vector product in (17) and thus not requiring any precomputations.

IV. Simulation Results

A. Simulation Setup

We compare the proposed architectures to the state-of-the-art ALISTA [13] and HyperLISTA [14]. All methods are implemented in PyTorch and trained end-to-end using minibatch stochastic gradient descent, thereby leveraging the Adam optimizer [27] with a supervised MSE loss

$$\mathcal{L}(\mathcal{M}; \boldsymbol{\theta}) = \frac{1}{|\mathcal{M}|} \sum\nolimits_{(\boldsymbol{x}^{\star}, \boldsymbol{b}) \in \mathcal{M}} \left\| \boldsymbol{x}^{(L)}(\boldsymbol{b}; \boldsymbol{\theta}) - \boldsymbol{x}^{\star} \right\|_2^2, \tag{20}$$

TABLE I
TRAINING AND SIGNAL MODEL HYPERPARAMETERS

Base Training Hyperparameters	
Number of Layers L	16
Adam(β_1, β_2)	$(0.9, 0.999)$
Learning Rate η	5×10^{-4}
Dataset Sizes $\lvert\mathcal{D}_{\text{train}}\rvert, \lvert\mathcal{D}_{\text{val}}\rvert, \lvert\mathcal{D}_{\text{test}}\rvert$	4096, 1024, 1024
Minibatch Size $\lvert\mathcal{M}\rvert$	64
Maximum Number of Training Steps $T_{\max}$	2×10^5
Ann. Constants $\tau_0, \tau_{\min}$	$10^{-1}, 5 \times 10^{-5}$
Ann. Decay Constants $c_{\text{LISTA}}; c_{\text{LSTELA}}$	5×10^{-4}; 2.5×10^{-4}
Base Data Model Parameters	
Dictionary Dimensions $M \times N$	250×500
Ground Truth Sparsity p_B	0.1

where $\mathcal{M} \subset \mathcal{D}_{\text{train}}$ is a minibatch. We reduce the learning rate by a factor of $1/\sqrt{10}$ if the validation loss has not improved over the last 10^3 training steps. After 10^4 steps of no improvement, training is terminated. During training of models employing the proposed piecewise differentiable support selection, we adopt an exponential annealing schedule, i.e., for each training step t, $[\tau(t) = \tau_0 \exp(-ct)]^{\infty}_{\tau_{\min}}$.

We train and evaluate the models on synthetic data generated as in [13], [25] using the model (1). The entries $[\boldsymbol{A}]_{i,j}$ are drawn from a standard normal distribution with subsequent ℓ_2-normalization of the kernels $\boldsymbol{a}_i$. The ground truths $\boldsymbol{x}^\star$ are generated as $\boldsymbol{x}^\star = \boldsymbol{z} \odot \boldsymbol{h}$ where $\boldsymbol{z} \sim \text{Bern}(p_B)$ and $\boldsymbol{h} \sim \mathcal{N}(0, 1)$. Additive noise $\boldsymbol{\varepsilon}$ is sampled from $\mathcal{N}(0, \sigma^2)$ with $\sigma^2 = 10^{-\text{SNR}/10} \left\lVert \boldsymbol{A}\boldsymbol{x}^\star \right\rVert_2^2 / M$ for a given SNR in dB. Unless stated otherwise, the training is conducted with the hyperparameters shown in Table I. The model performance is assessed using the normalized MSE w.r.t. the ground-truth $\boldsymbol{x}^\star$

$$\text{NMSE [dB]} = 10 \log_{10} \left(\lVert \boldsymbol{x} - \boldsymbol{x}^\star \rVert_2^2 \Big/ \lVert \boldsymbol{x}^\star \rVert_2^2 \right). \tag{21}$$

To validate the piecewise differentiable support selection, we consider ALISTA with manually tuned parameters $p^{(\ell)}$ as in [13], and ALISTA-d with learned parameters $p^{(\ell)}$. In ALISTA-d, an independent parameter $p^{(\ell)}$ is assigned to each layer and trained directly without instance adaptivity. All HyperLISTA parameters are learned end-to-end as well.

B. Simulation Results

Fig. 1 illustrates the NMSE after each model layer. ALISTA and ALISTA-d achieve a similar final MSE, showing that our proposed smoothed soft-thresholding with support selection operator enables end-to-end learning, thereby eliminating the need for time consuming manual tuning or grid searches. Note that we do not attain the same MSE for ALISTA and HyperLISTA as in [13], [14] in the noiseless case, which can be attributed to the significantly smaller training set compared to [13], [14] and the absence of a curriculum learning approach. In the absence of noise (Fig. 1 left), the proposed LSTELA performs similarly to ALISTA, but its convergence rate is slower than HyperLISTA. When considering the common case of measurement noise (Fig. 1 right), both LSTELA and L-LSTELA clearly achieve a lower noise floor than the ISTA-based unrolled DNNs. In both cases, LSTELA and L-LSTELA outperform the underlying iterative algorithm N-STELA.

C. Adaptivity Studies

One benefit of the instance-adaptivity is the increased robustness of the model against changes in the domain. We investigate the impact of deviations in the distribution of the test dataset w.r.t. the training and validation data in Fig. 2.

Both LSTELA and L-LSTELA adapt excellently to changes in the SNR, still achieving a lower MSE than HyperLISTA. If the ground truth sparsity is decreased, both proposed algorithms generalize well, whereas the performance significantly decreases for an increasing p_B, in particular for L-LSTELA. This suggests that replacing the pseudo-inverse $\boldsymbol{A}^\dagger$ with $\boldsymbol{A}^\text{T}$ yields a worse estimation of the sparsity level if the problem becomes less sparse. When adapting to a discrete cosine dictionary in Fig. 2 (bottom left), LSTELA performs comparable to a model that is trained for this data distribution, while the adaptation performance of L-LSTELA is slightly deteriorated. Note that HyperLISTA is significantly outperformed by both proposed methods in this experiment.

V. CONCLUSION

We propose two DNN architectures, LSTELA and L-LSTELA, based on unrolling an SCA algorithm, a novel piecewise differentiable soft-thresholding with support selection operator and instance-aware parameter adaptation. Comparing to state-of-the-art methods, the proposed LSTELA and L-LSTELA trade convergence rate for a significantly improved MSE in case of noisy measurements as well as enhanced adaptation to unseen distributions.

REFERENCES

[1] M. Rani, S. B. Dhok, and R. B. Deshmukh, "A Systematic Review of Compressive Sensing: Concepts, Implementations and Applications," *IEEE Access*, vol. 6, pp. 4875–4894, 2018.

[2] M. Elad and M. Aharon, "Image Denoising Via Sparse and Redundant Representations Over Learned Dictionaries," *IEEE Trans. on Image Process.*, vol. 15, no. 12, pp. 3736–3745, Dec. 2006.

[3] M. Lustig, D. Donoho, and J. M. Pauly, "Sparse MRI: The Application of Compressed Sensing for Rapid MR Imaging," *Magnetic Resonance in Med.*, vol. 58, no. 6, pp. 1182–1195, Dec. 2007.

[4] W. U. Bajwa, J. Haupt, A. M. Sayeed, and R. Nowak, "Compressed Channel Sensing: A New Approach to Estimating Sparse Multipath Channels," *Proc. IEEE*, vol. 98, no. 6, pp. 1058–1076, Jun. 2010.

[5] I. Daubechies, M. Defrise, and C. De Mol, "An Iterative Thresholding Algorithm for Linear Inverse Problems with a Sparsity Constraint," *Commun. Pure Appl. Math.*, vol. 57, no. 11, pp. 1413–1457, Nov. 2004.

[6] A. Beck and M. Teboulle, "A Fast Iterative Shrinkage-Thresholding Algorithm for Linear Inverse Problems," *SIAM J. Imaging Sci.*, vol. 2, no. 1, pp. 183–202, Jan. 2009.

[7] Bradley Efron, Trevor Hastie, Iain Johnstone, and Robert Tibshirani, "Least Angle Regression," *Ann. Statist.*, vol. 32, no. 2, pp. 407–499, Apr. 2004.

[8] Y. Yang and M. Pesavento, "A Unified Successive Pseudoconvex Approximation Framework," *IEEE Trans. Signal Process.*, vol. 65, no. 13, pp. 3313–3328, Jul. 2017.

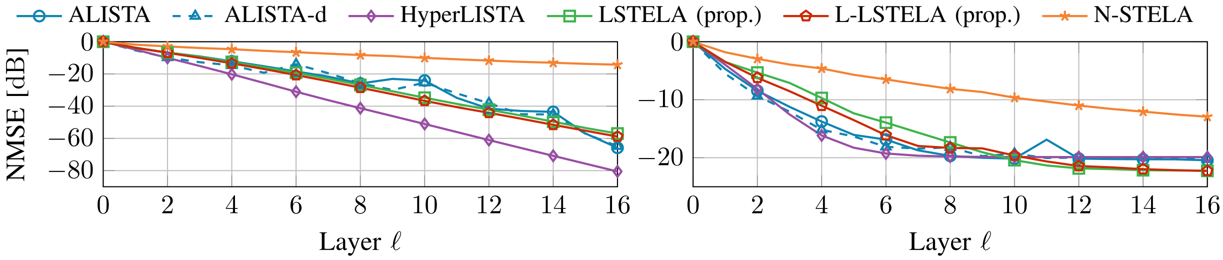

Fig. 1. Convergence of proposed LSTELA, L-LSTELA, ALISTA and HyperLISTA for the noiseless case (left) and SNR = 20 dB (right). N-STELA with manually tuned λ provided for reference.

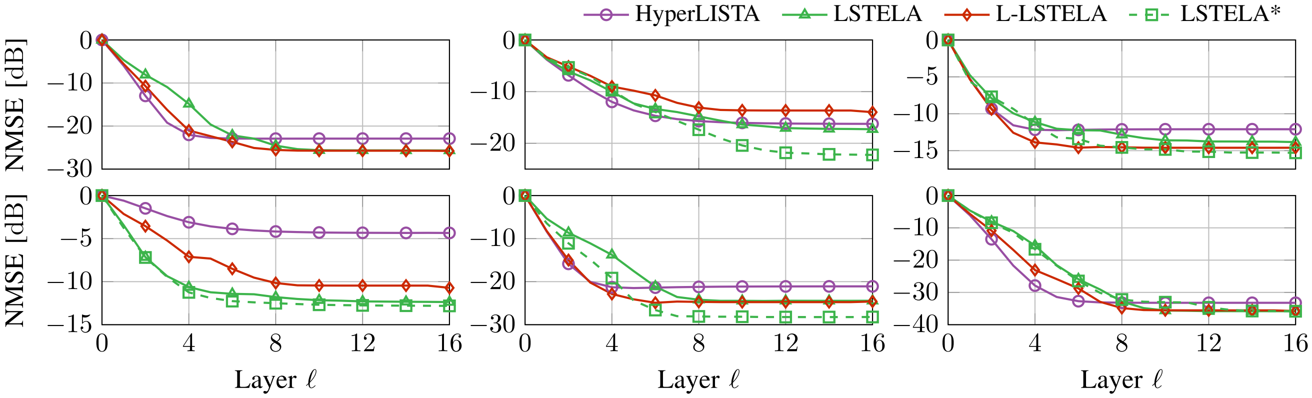

Fig. 2. Adaptivity Studies. Convergence of HyperLISTA and the proposed LSTELA and L-LSTELA for the hyperparameters used during training, i.e., SNR = 20 dB, $p_B = 0.05$ (top left); **Bottom Left:** adaptation to discrete cosine dictionary: $[\boldsymbol{A}]_{k,l} = \cos(j\pi(k-1)(l-1)/N)$ with subsequent kernel normalization; **Center:** changed ground truth sparsity: $p_B = 0.1$ (top), $p_B = 0.025$ (bottom); **Right:** changed noise level: SNR = 10 dB (top), SNR = 30 dB (bottom); LSTELA trained on the new data distribution provided for reference as LSTELA*.

[9] L. Schynol, M. Hemsing, and M. Pesavento, "An Accelerated Successive Convex Approximation Scheme With Exact Step Sizes for L1-Regression," *IEEE Open J. Signal Process.*, pp. 1–9, 2025.

[10] V. Monga, Y. Li, and Y. C. Eldar, "Algorithm Unrolling: Interpretable, Efficient Deep Learning for Signal and Image Processing," *IEEE Signal Process. Mag.*, vol. 38, no. 2, pp. 18–44, Mar. 2021.

[11] N. Shlezinger, Y. C. Eldar, and S. P. Boyd, "Model-Based Deep Learning: On the Intersection of Deep Learning and Optimization," *IEEE Access*, vol. 10, pp. 115 384–115 398, 2022.

[12] K. Gregor and Y. LeCun, "Learning Fast Approximations of Sparse Coding," in *Proceedings of the 27th International Conference on Machine Learning*, ser. ICML'10. Madison, WI, USA: Omnipress, 2010, pp. 399–406.

[13] J. Liu, X. Chen, Z. Wang, and W. Yin, "ALISTA: Analytic Weights Are As Good As Learned Weights in LISTA," in *International Conference on Learning Representations*, 2019.

[14] X. Chen, J. Liu, Z. Wang, and W. Yin, "Hyperparameter Tuning is All You Need for LISTA," in *Proceedings of the 35th International Conference on Neural Information Processing Systems*, ser. NIPS '21. Red Hook, NY, USA: Curran Associates Inc., 2024.

[15] E. Agustsson, F. Mentzer, M. Tschannen, L. Cavigelli, R. Timofte, L. Benini, and L. Van Gool, "Soft-to-Hard Vector Quantization for End-to-End Learning Compressible Representations," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, pp. 1141–1151.

[16] Z. Ben-Haim and Y. C. Eldar, "The Cramér-Rao Bound for Estimating a Sparse Parameter Vector," *IEEE Trans. Signal Process.*, vol. 58, no. 6, pp. 3384–3389, Jun. 2010.

[17] Y. Nesterov, "A Method for Solving the Convex Programming Problem with Convergence Rate O(1/k^2)," *Proceedings of the USSR Academy of Sciences*, vol. 269, pp. 543–547, 1983.

[18] T. Hastie, R. Tibshirani, and J. Friedman, "Linear Methods for Regression," in *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, 2nd ed., ser. Springer Series in Statistics, T. Hastie, R. Tibshirani, and J. Friedman, Eds. New York, NY: Springer New York, 2009, pp. 43–99.

[19] C.-K. Lai, S. Li, and D. Mondo, "Spark-Level Sparsity and the $\ell 1$ Tail Minimization," *Applied and Computational Harmonic Analysis*, vol. 45, no. 1, pp. 206–215, Jul. 2018.

[20] Q. Zhao, Y. Luo, C. Ma, and S. Li, "Sparse Signal Recovery via Tail-FISTA," in *2022 34th Chinese Control and Decision Conference (CCDC)*. Hefei, China: IEEE, Aug. 2022, pp. 1410–1415.

[21] P. Pradhan, S. B. Shah, R. Randhi, and Y. C. Eldar, "Recursive-Tail-Fista for Sparse Signal Recovery," in *ICASSP 2024 - 2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. Seoul, Korea, Republic of: IEEE, Apr. 2024, pp. 9726–9730.

[22] Y. Fan and M. Pesavento, "Tail-STELA for Fast Signal Recovery via Basis Pursuit," in *2024 IEEE 13rd Sensor Array and Multichannel Signal Processing Workshop (SAM)*, 2024, pp. 1–5.

[23] Y. Wang and W. Yin, "Sparse Signal Reconstruction via Iterative Support Detection," *SIAM J. Imaging Sci.*, vol. 3, no. 3, pp. 462–491, Jan. 2010.

[24] S. Van De Geer, P. Bühlmann, Y. Ritov, and R. Dezeure, "On Asymptotically Optimal Confidence Regions and Tests for High-Dimensional Models," *Ann. Statist.*, vol. 42, no. 3, Jun. 2014.

[25] X. Chen, J. Liu, Z. Wang, and W. Yin, "Theoretical Linear Convergence of Unfolded ISTA and its Practical Weights and Thresholds," in *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, ser. NIPS'18. Red Hook, NY, USA: Curran Associates Inc., 2018, pp. 9079–9089.

[26] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 4th ed. The MIT Press, 2022.

[27] D. P. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," in *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, San Diego, CA, USA, 2015.