Towards Neurally Augmented ALISTA

Freya Behrens
TU Berlin
f.behrens@campus.tu-berlin.de

Jonathan Sauder
TU Berlin
sauder@campus.tu-berlin.de

Peter Jung
TU Berlin
peter.jung@tu-berlin.de

Abstract

It is well-established that many iterative sparse reconstruction algorithms such as ISTA can be unrolled to yield a learnable neural network for improved empirical performance. Recently, ALISTA has been introduced, combining the strong empirical performance of a fully learned approach like LISTA, while retaining theoretical guarantees of classical compressed sensing algorithms and significantly reducing the number of parameters to learn. However, these parameters are trained to work in expectation, often leading to suboptimal reconstruction of individual targets. In this work we therefore introduce Neurally-Augmented-ALISTA, which computes step sizes and thresholds individually for each target vector during reconstruction. This adaptive approach is theoretically motivated by revisiting the recovery guarantees of ALISTA and is able to outperform existing algorithms in sparse reconstruction.

1 Introduction

Compressed sensing is concerned with recovering a sparse vector from far fewer compressive linear observations. Formally, consider the set of s-sparse vectors in $\mathbb{R}^N$, i.e. $\Sigma_s^N := \{x \in \mathbb{R}^N \big| \|x\|_0 \leq s\}$. Furthermore, let $\Phi \in \mathbb{R}^{M \times N}$ be the measurement matrix, with typically $M \ll N$. For a given noiseless observation $y = \Phi x^*$ of an unknown but s-sparse $x^* \in \Sigma_s^N$ we therefore wish to solve:

$$\underset{x}{\operatorname{argmin}} \|x\|_0 \quad \text{s.t.} \quad y = \Phi x \tag{1}$$

In [2] it has been shown, that under certain assumptions on Φ, the solution to the combinatorial problem in (1) can be instead obtained by solving the convex LASSO problem. A very popular approach for solving LASSO is the iterative shrinkage thresholding algorithm (ISTA) [4], in which a reconstruction $x^{(k)}$ is obtained after k iterations from initial $x^{(0)} = 0$ via the iteration:

$$x^{(k+1)} = \eta_{\lambda/L}\big(x^{(k)} + \frac{1}{L}\Phi^T(y - \Phi x^{(k)})\big) \tag{2}$$

where η_θ is the soft thresholding function given by $\eta_\theta(x) = \operatorname{sign}(x)\max(0, |x| - \theta)$ (applied coordinate-wise) and L is the Lipschitz constant (i.e. the largest eigenvalue) of $\Phi^T\Phi$. Famously, the computational graph of ISTA with K iterations can be unrolled to yield Learned ISTA (LISTA) [7], a K-layer neural network in which all parameters involved can be trained using backpropagation and gradient descent. LISTA achieves impressive empirical reconstruction performance for many sparse datasets but loses the theoretical guarantees of ISTA. Bridging the gap between LISTA's strong reconstruction quality and the theoretical guarantees for ISTA, ALISTA [11] was introduced. In ALISTA, all matrices are excluded from the learning process to retain desirable properties for compressed sensing, and $W \in \mathbb{R}^{M \times N}$ corresponding to Φ is instead computed by optimizing the generalized coherence:

$$\mu(W, \Phi) = \inf_{W \in \mathbb{R}^{M \times N}} \max_{i \neq j} W_{:,i}^T \Phi_{:,j} \ \text{ s.t. } \forall i \in \{1, \dots, N\} : W_{:,i}^T \Phi_{:,i} = 1 \tag{3}$$

For each layer of ALISTA, only a scalar step size parameter $\gamma^{(k)}$ and a scalar threshold $\theta^{(k)}$ is learned from the data, yielding the iteration:

$$x^{(k+1)} = \eta_{\theta^{(k)}}\Big(x^{(k)} - \gamma^{(k)} W^T(\Phi x^{(k)} - y)\Big) \tag{4}$$

NeurIPS 2020 Workshop on Deep Learning and Inverse Problems, virtual.

As in LISTA, the parameters for ALISTA are learned end-to-end using backpropagation and stochastic gradient descent by empirically minimizing the mean squared error between the reconstruction and the target vector. The authors rigorously upper-bound the reconstruction error of ALISTA in the noiseless case and demonstrate strong empirical reconstruction quality even in the noisy case. The empirical performance similar to LISTA, the retained theoretical guarantees, and the reduction of number of parameters to train from either $O(KM^2 + NM)$ in vanilla LISTA or $O(MNK)$ in the variant of LISTA-CPSS [3] to just $O(K)$, make ALISTA an appealing algorithm to study and extend.

Thresholds that are adaptive to the current target vector have been explored in ALISTA-AT [9], where thresholds are computed depending on the absolute values of the components from the current reconstruction, for which the authors demonstrate superior recovery over ALISTA for a specific setting of M, N and s. In a related approach [15] identify undershooting, meaning that reconstructed components are smaller than target components, as a shortcoming of LISTA and propose Gated-LISTA to address these issues. The authors show that adding their proposed gates to ALISTA, named AGLISTA, improves its performance in the same setting of M, N and s as ALISTA-AT.

In this paper, motivated by essential proof steps of ALISTA's recovery guarantee, we propose an alternative method for adaptively choosing thresholds and step sizes during reconstruction. Our method directly extends ALISTA by using a recurrent neural network to predict thresholds and step sizes depending on an estimate of the ℓ_1-error between the reconstruction and the unknown target vector after each iteration. We refer to our method as Neurally Augmented ALISTA (NA-ALISTA), as the method falls into the general framework of neural augmentation of unrolled algorithms [14, 12, 5]. To summarize, our main contributions are:

1. We introduce Neurally Augmented ALISTA (NA-ALISTA), an algorithm which learns to adaptively compute thresholds and step-sizes for individual target vectors during recovery.
2. We provide theoretical motivation inspired by guarantees for sparse reconstruction which show that NA-ALISTA can achieve tighter error bounds depending on the target x^*.
3. We show that NA-ALISTA empirically outperforms ALISTA and other state-of-the-art algorithms in all evaluated settings and that the gains increase with decreasing M/N.

2 Theoretical Motivation

The thresholds $\theta^{(k)}$ in (4) play an important role in the analysis of ALISTA. In this section we motivate the choice of adaptive thresholds - the key improvement in our proposed NA-ALISTA. More specifically, we repeat the conditions under which ALISTA guarantees no false positives and highlight an intermediate step in the error bound from [11], which tightens when the thresholds can adapt to specific instances of x^*.

Assumption (adapted from Assumption 1 in [11]):
Let $x^* \in \Sigma_s^N$ be a fixed s–sparse target vector. Let W be such that it attains the infimum (3) *yielding $\tilde{\mu} = \mu(W, \Phi)$. Let $s < (1 + 1/\tilde{\mu})/2$, $\{\gamma^{(k)}\}_{k=1}^{k} \subset (0, \frac{2}{2\tilde{\mu}s - \tilde{\mu} + 1})$ and $\{\theta^{(k)}\}_{k=1}^{K}$ with:*

$$\theta^{(k)} \geq \gamma^{(k)} \tilde{\mu} \|x^{(k)} - x^*\|_1 \tag{5}$$

Because in ALISTA, $\{\theta^{(k)}\}_{k=1}^{K}$ and $\{\gamma^{(k)}\}_{k=1}^{K}$ are optimized in expectation over the training data sampled from a compact set $\mathcal{X}$, (5) holds only if the thresholds are larger than the worst case ℓ_1-error committed by the algorithm over all training vectors, i.e. $\theta^{(k)} \geq \gamma^{(k)} \tilde{\mu} \sup_{x^* \in \mathcal{X}} \|x^{(k)} - x^*\|_1$, see eq. (9) in [11]. Only in this case it is guaranteed that no false positives are in the support of $x^{(k)}$, meaning that supp$(x^{(k)}) \subseteq$ supp(x^*), see Lemma 1 in [11]. However, the threshold $\theta^{(k)}$ also appears in the error upper bound. We employ an intermediate step of the error upper bound in Theorem 1 from [11]:

Reconstruction error bound: *Under the settings of the Assumption, it holds that:*

$$\|x^{(k+1)} - x^*\|_2 \leq \|x^{(k+1)} - x^*\|_1 \leq \tilde{\mu}\gamma^{(k)}(s-1)\|x^{(k)} - x^*\|_1 + \theta^{(k)}s + |1 - \gamma^{(k)}| \|x^{(k)} - x^*\|_1 \tag{6}$$

This inequality is derived in detail in Appendix A of [11]. Hence it is desirable that $\theta^{(k)}$ is as small as possible, but such that it still satisfies (5). This means that ALISTA has to learn thresholds at least proportional to the largest possible committed ℓ_1-error over all possible x^* in order to guarantee good reconstruction, for which it is in turn penalized in the error bound. However, the thresholds that make the error bound tighter vary depending on the x^* that is to be recovered. In fact, if an algorithm would have access to $\|x^{(k)} - x^*\|_1$ and were allowed to choose thresholds adaptively, depending on this quantity, the more relaxed inequality (5) could be employed directly, without taking the supremum.

3 Neurally Augmented ALISTA

In order to tighten the error upper bound in (6), we introduce Neurally Augmented ALISTA (NA-ALISTA), in which we adaptively predict thresholds $\theta^{(k,x^*)}$ depending on an estimate for the ℓ_1-error between $x^{(k)}$ and the unknown x^*. As can be observed from (6), such $\theta^{(k,x^*)}$ must be proportional to $\|x^{(k)} - x^*\|_1$. In theory, this true ℓ_1-error could be recovered exactly. This is because there are no false positives in $x^{(k)}$, making it s-sparse and for $\tilde{\mu} < 1/(2s-1)$ the column-normalized $W^T\Phi$ is restricted-invertible for any $2s$-sparse input [6] [Corollary 5.4, p.113]. However, it is infeasible to solve such an inverse problem at every iteration k. Furthermore, in practice the sparsity is often much larger than what is admissible via coherence bounds (see experiments of [7, 11, 15, 9], even assuming optimal admissible coherence by Welch bound [13]).

Therefore NA-ALISTA approximates the ℓ_1 error. For this, consider the ℓ_1-norms:

$$\begin{aligned} r^{(k)} &:= \|\Phi x^{(k)} - y\|_1 = \|\Phi(x^{(k)} - x^*)\|_1 \\ u^{(k)} &:= \|W^T(\Phi x^{(k)} - y)\|_1 = \|(W^T\Phi)(x^{(k)} - x^*)\|_1 \end{aligned} \tag{7}$$

of the residual and the iterative update quantity in (4). Both are known to the algorithm even though x^* is unknown and do not incur additional computational costs other than taking the norm. Since $W^T\Phi$ has low generalized mutual coherence, it is a restricted isometry for sparse vectors and therefore $u^{(k)}$ will be correlated with the true error for sparse vectors. This correlation is validated in Figure 1 for $u^{(k)}$, but also holds for $r^{(k)}$. Other useful quantities for approximating the ℓ_1-error are $\|x^{(0)} - x^*\|_1, \ldots, \|x^{(k-1)} - x^*\|_1$ (also shown in Figure 1. The latter suggests the use of a recurrent neural network in NA-ALISTA, leveraging the past estimations. We therefore propose to use an LSTM [8] which has two input neurons, receiving $u^{(k)}$ and $r^{(k)}$ at each iteration k. This is used to update the LSTM cell state $c \in \mathbb{R}^H$ and hidden state $h \in \mathbb{R}^H$ and produce the outputs $\theta^{(k,x^*)}$ and $\gamma^{(k,x^*)}$ via matrix multiplication with a learned matrix $U \in \mathbb{R}^{2\times H}$ and then passing through the softsign function:

$$c^{(k+1)}, h^{(k+1)} = \texttt{LSTM}(c^{(k)}, h^{(k)}, [r^{(k)}, u^{(k)}]) \tag{8}$$

$$\theta^{(k,x^*)}, \gamma^{(k,x^*)} = \texttt{Softsign}(Uc^{(k+1)}) \tag{9}$$

The threshold and stepsize are then used in the final step of one NA-ALISTA iteration:

$$x^{(k+1)} = \eta_{\theta^{(k,x^*)}}\left(x^{(k)} - \gamma^{(k,x^*)} W^T(\Phi x^{(k)} - y)\right). \tag{10}$$

4 Experiments

In this section, we evaluate NA-ALISTA in a sparse reconstruction task and compare it against ALISTA [11], ALISTA-AT [9], AGLISTA [15], as well as the classical ISTA [4] and FISTA [1]. To emphasize a fair and reproducible comparison between the models, the code for all experiments listed is available on Github[1].

Experimental Setup. Following the same experimental setup as [11, 15, 3, 9], the support of $x^* \in \mathbb{R}^N$ is determined via i.i.d. Bernoulli random variables with parameter S/N, leading to an

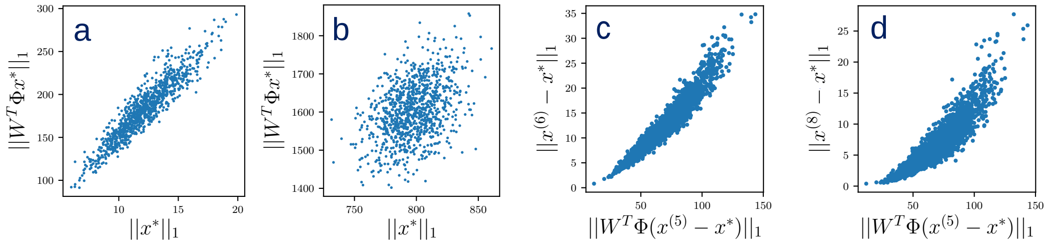


Figure 1: Correlation between $\|x^*\|_1$ and $u = \|W^T\Phi x^*\|_1$ for random Gaussian vectors $x^* \in \mathbb{R}^{1000}$ is strong for sparse ($\|x^*\|_0$=15 in (a)) and weak for non-sparse vectors ($\|x^*\|_0$=1000 in (b)). Also, the correlation between $u^{(i)}$ and the true error $\|x^{(j)} - x^*\|_1$ is even preserved over multiple layers for an instance of NA-ALISTA , e.g. for i=5 and j=6 in (c), and i=5 and j=8 in (d).

[1] https://github.com/feeds/na-alista

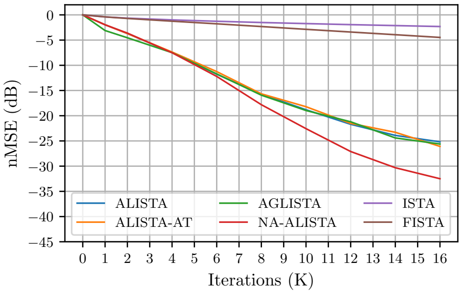


Figure 2: The reconstruction error over the number of iterations K for N=2000, SNR=40dB. NA-ALISTA outperforms all competitors.

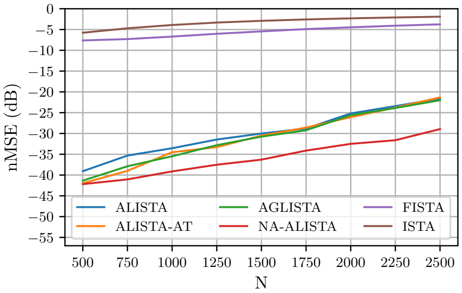


Figure 3: Reconstruction error over different compression ratios. NA-ALISTA outperforms competitors by an increasing margin as N increases.

expected sparsity of S. Its non-zero components are sampled according to $\mathcal{N}(0,1)$. The entries of Φ are also sampled from $\mathcal{N}(0,1)$ and each column is normalized to unit ℓ_2-norm. Then, the generalized coherence in (3) between W and Φ is minimized via the Frobenius-Norm approximation using projected gradient descent, identical to [11, 15, 9]. Adam [10] is used to minimize the reconstruction error for all algorithms. A test set of 10000 samples is fixed before training and recovery performance is measured with the normalized mean squared error. A support selection trick was introduced in [3] to speed up convergence and stabilize training and has been subsequently used extensively (see supplementary code of [11, 9, 15]). For a fair comparison, we employ support selection in all learned models compared in this paper. Our AGLISTA implementation follows the description from [15] and uses exponential gain gates and inverse-proportional-based overshoot gains. The λ parameter in ISTA and FISTA was tuned by hand, we found that $\lambda = 0.4$ led to the best performance in our tasks. When not otherwise indicated we use the following settings for experiments and algorithms: M=250, N=1000, S=50, K=16, H=128, and $y = \Phi x^* + z$ with additive white Gaussian noise z with a signal to noise ratio of 40 dB. We train all algorithms for 400 epochs, with each epoch containing 50,000 sparse vectors with a batch size of 512.

Comparison with Competitors. As shown in Figure 6, we first fix N and observe the reconstruction error for a varying amount of iterations. In Figure 3 we then decrease the compression ratio by increasing N and keeping everything else fixed. We observe that NA-ALISTA outperforms state-of-the-art adaptive methods in all evaluated scenarios, particularly by a larger margin as the compression ratio becomes more challenging.

Computational Cost. In Figure 4, we examine the effect of changing the size H of the LSTM network used. One can observe that an exponential increase in hidden neurons yields only a small error reduction for different N, suggesting that the size H=128 is a sufficient default value regardless of N. This verifies that the added computation in NA-ALISTA (adding the H^2 term in $O(MN + H^2)$ for each iteration) is negligible in practice.

Verification. As an empirical verification the Assumption 1 in (5) we need to check for every x^*, whether the ratio $\theta^{(k,x^*)}/\gamma^{(k,x^*)}$ is proportional to the ℓ_1-error $||x^* - x^{(k)}||_1$. Since it is infeasible to check the assumption for the whole space of sparse x^*, we empirically verify 5 for a sample of inputs from the training distribution. In Figure 5 the means of both values are proportional to each other for such a test sample, suggesting that the reconstruction bound from [11] holds for NA-ALISTA as well.

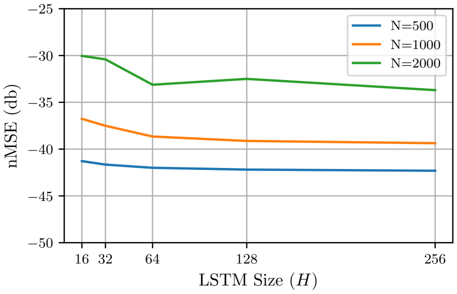


Figure 4: Reconstruction error for LSTM size H in NA-ALISTA. An exponential increase of the hidden layer size only yields a marginal improvement once H=64 is surpassed.

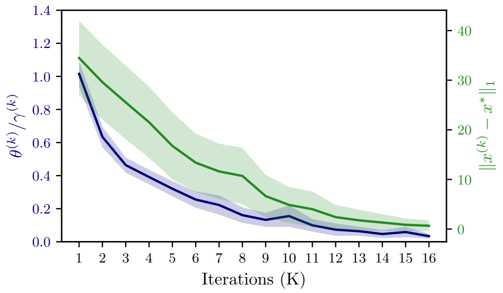


Figure 5: Comparison of the ratio $\theta^{(k)}/\gamma^{(k)}$ with the true ℓ_1-error at each iteration for NA-ALISTA. We report the mean and standard deviation over the test set.

References

[1] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2009.

[2] Emmanuel J Candès, Justin Romberg, and Terence Tao. Robust uncertainty principles: Exact signal reconstruction from highly incomplete frequency information. *IEEE Transactions on information theory*, 2006.

[3] Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ista and its practical weights and thresholds. In *Advances in Neural Information Processing Systems*, 2018.

[4] Ingrid Daubechies, Michel Defrise, and Christine De Mol. An iterative thresholding algorithm for linear inverse problems with a sparsity constraint. *Communications on Pure and Applied Mathematics: A Journal Issued by the Courant Institute of Mathematical Sciences*, 2003.

[5] Steven Diamond, Vincent Sitzmann, Felix Heide, and Gordon Wetzstein. Unrolled optimization with deep priors. *arXiv preprint arXiv:1705.08041*, 2017.

[6] Simon Foucart and Holger Rauhut. A mathematical introduction to compressive sensing. *Bull. Am. Math*, 2017.

[7] Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *Proceedings of the 27th international conference on international conference on machine learning*, 2010.

[8] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 1997.

[9] D. Kim and D. Park. Element-wise adaptive thresholds for learned iterative shrinkage thresholding algorithms. *IEEE Access*, 2020.

[10] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

[11] Jialin Liu, Xiaohan Chen, Zhangyang Wang, and Wotao Yin. Alista: Analytic weights are as good as learned weights in lista. In *International Conference on Learning Representations*, 2019.

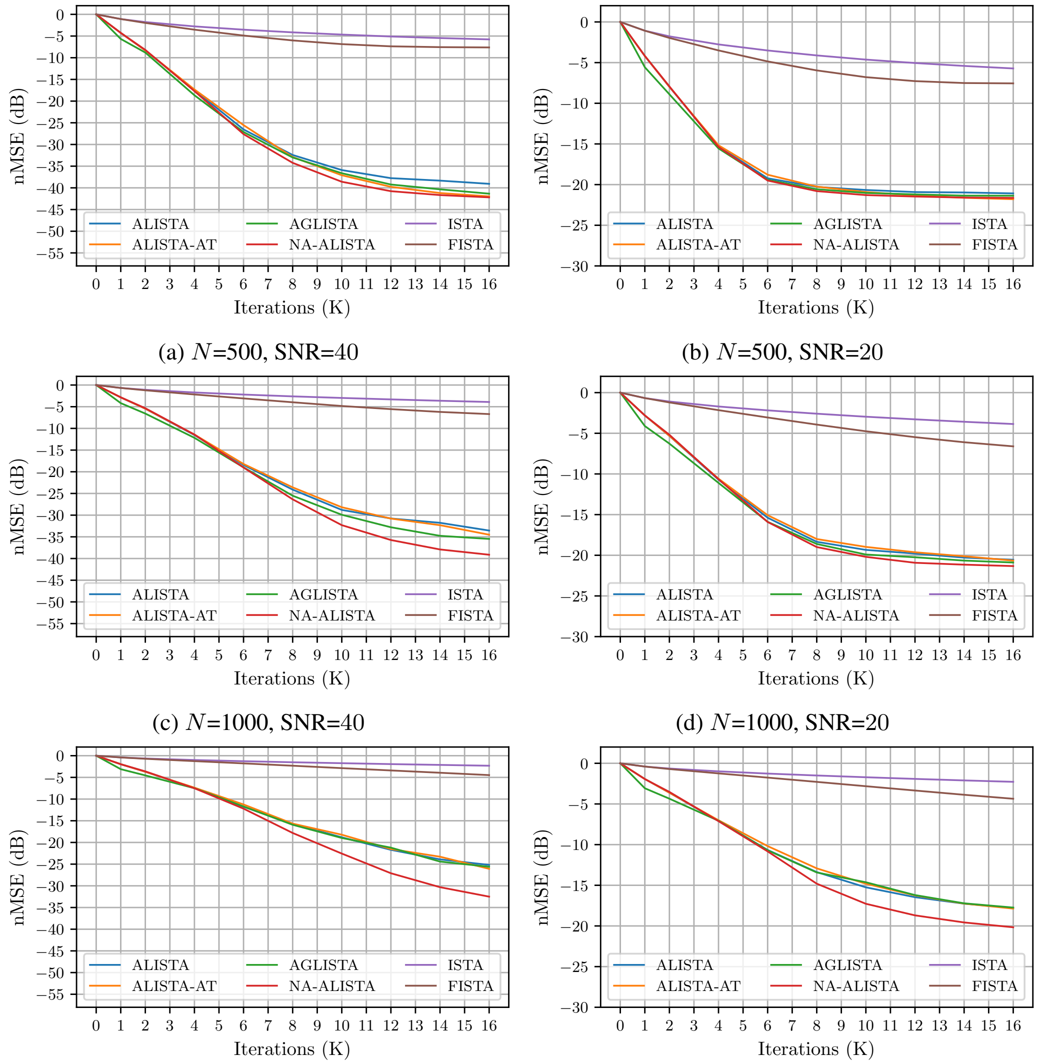
[12] Vishal Monga, Yuelong Li, and Yonina C Eldar. Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing. *arXiv preprint arXiv:1912.10557*, 2019.

[13] Lloyd Welch. Lower bounds on the maximum cross correlation of signals (corresp.). *IEEE Transactions on Information theory*, 1974.

[14] Max Welling. Neural augmentation in wireless communication. *Keynote at 2020 IEEE International Symposium on Information Theory*, 2020.

[15] Kailun Wu, Yiwen Guo, Ziang Li, and Changshui Zhang. Sparse coding with gated learned ista. In *International Conference on Learning Representations*, 2020.

A Supplementary Experiments



(a) N=500, SNR=40

(b) N=500, SNR=20

(c) N=1000, SNR=40

(d) N=1000, SNR=20

(e) N=2000, SNR=40

(f) N=2000, SNR=20

Figure 6: The reconstruction error for ALISTA, ALISTA-AT and NA-ALISTA over the number of iterations run for different SNR and N settings. In 6a, for the standard setting in the literature with N= 500 and a noise level of 40dB NA-ALISTA performs on par with competitors after 16 iterations. For an increased N=1000 under the same noise level in 6c, our algorithm outperforms the other methods clearly. For a noise level of 20dB all algorithms perform similarly for N =500 and N=1000 and NA-ALISTA outperforms the others at $N = 2000$.

B Algorithm

Algorithm 1: Neurally Augmented ALISTA

```
Learnable Parameters: initial cell state $c_0 \in \mathbb{R}^H$, initial hidden state $h_0 \in \mathbb{R}^H$,
cell state to output matrix $U \in \mathbb{R}^{2 \times H}$ and parameters of LSTM cell.
Input: $y$
$x \leftarrow 0; h \leftarrow h_0; c \leftarrow c_0$
for $\{1, \ldots, K\}$ do
    $r \leftarrow \|\Phi x - y\|_1$
    $u \leftarrow \|W^T(\Phi x - y)\|_1$
    $c, h \leftarrow \texttt{LSTM}(c, h, [r, u])$
    $\theta, \gamma \leftarrow \texttt{Softsign}(Uc)$
    $x \leftarrow \eta_\theta\Big(x - \gamma W^T(\Phi x - y)\Big)$
end
Return $x$;
```

Algorithm 1: An algorithmic expression of NA-ALISTA