Variational Learning ISTA

Anonymous authors
Paper under double-blind review

Abstract

Compressed sensing combines the power of convex optimization techniques with a sparsity-inducing prior on the signal space to solve an underdetermined system of equations. For many problems, the sparsifying dictionary is not directly given, nor its existence can be assumed. Besides, the sensing matrix can change across different scenarios. Addressing these issues requires solving a sparse representation learning problem, namely dictionary learning, taking into account the epistemic uncertainty of the learned dictionaries and, finally, jointly learning sparse representations and reconstructions under varying sensing matrix conditions. We address both concerns by proposing a variant of the LISTA architecture. First, we introduce Augmented Dictionary Learning ISTA (A-DLISTA), which incorporates an augmentation module to adapt parameters to the current measurement setup. Then, we propose to learn a distribution over dictionaries via a variational approach, dubbed Variational Learning ISTA (VLISTA). VLISTA exploits A-DLISTA as the likelihood model and approximates a posterior distribution over the dictionaries as part of an unfolded LISTA-based recovery algorithm. As a result, VLISTA provides a probabilistic way to jointly learn the dictionary distribution and the reconstruction algorithm with varying sensing matrices. We provide theoretical and experimental support for our architecture and show that our model learns calibrated uncertainties.

1 Introduction

To solve under-determined inverse problems, compressed sensing methods impose a prior on the signal structure. Sparsity and linear inverse problems are canonical examples of signal structure and sensing mediums (modeled by a linear transformation $\mathbf{\Phi}$). Several studies have been conducted in recent years to improve compressed sensing solvers performance and complexity. A typical approach involves unfolding iterative algorithms as layers of neural networks and learning parameters end-to-end (Gregor & LeCun, 2010). Some of the main challenges of data-driven approaches include varying sensing matrices and unknown sparsifying dictionaries. By learning a dictionary and including it in optimization iterations, the work in Aberdam et al. (2021); Schnoor et al. (2022) aims to address these issues. However, the data samples might not have any exact sparse representations, which means there is no ground truth dictionary. The issue can be more severe for heterogeneous datasets where the dictionary choice might vary from one sample to another. A principled approach to this problem would be to leverage a Bayesian framework and define a distribution over the learned dictionaries with proper uncertainty quantification. We follow two steps to accomplish this goal. First, we introduce Augmented Dictionary Learning ISTA (A-DLISTA), an augmented version of the Learning Iterative Soft-Thresholding Algorithm (LISTA)-like model, capable of adapting its parameters to the current measurement setup. We theoretically motivate its design and empirically prove its advantages compared to other non-adaptive LISTA-like models in a non-static measurement scenario, i.e., considering varying sensing matrices across data samples. Finally, to learn a distribution over dictionaries we introduce Variational Learning ISTA (VLISTA), a variational formulation that leverages A-DLISTA as the likelihood model. Specifically, VLISTA refines the dictionary iteratively after each iteration based on the outcome of the previous layer. Intuitively, our model can be understood as a form of a recurrent variational autoencoder, e.g., Chung et al. (2015), where at each iteration of the algorithm we have an approximate posterior distribution over the dictionaries conditioned on the outcome of the previous iteration. Moreover, VLISTA provides uncertainty estimation that can be used to detect Out-Of-Distribution (OOD) samples.

Concerning the A-DLISTA model, we know that an augmented version of LISTA, named Neurally Augmented ALISTA (NALISTA), was already proposed in Behrens et al. (2021). Nonetheless, there are some fundamental differences between NALISTA and A-DLISTA. First, our model takes as input the per-sample sensing matrix and the dictionary at the current layer. This means that A-DLISTA adapts the parameters to the current measurement setup and the learned dictionary. In contrast, NALISTA assumes to have a fixed sensing matrix to analytically evaluate its weight matrix, $\mathbf{W}$. Hypothetically, NALISTA could handle varying sensing matrices. However, that comes at the price of having to solve, for each data sample, the inner optimization step to evaluate the $\mathbf{W}$ matrix. Moreover, the architectures of the augmentation networks are profoundly different. Indeed, while NALISTA uses an LSTM, A-DLISTA employs a convolutional neural network, shared across all layers. Such a different choice reflects the different types of dependencies between layers and input data that networks try to model. We report in Appendix B and Appendix C detailed discussions about the theoretical motivation and architectural design for A-DLISTA.

Therefore, our work's main contributions can be summarized as follows:

- We design an augmented version of LISTA, dubbed A-DLISTA, that can handle non-static measurement setups, i.e. per-sample sensing matrices, and adapt parameters to the current data instance.
- We propose VLISTA that learns a distribution over sparsifying dictionaries. The model can be interpreted as a Bayesian LISTA model that leverages A-DLISTA as the likelihood model.
- VLISTA adapts the dictionary to optimization dynamics and therefore can be interpreted as a hierarchical representation learning approach, where the dictionary atoms gradually permit more refined signal recovery.
- The dictionary distributions can be used successfully for out-of-distribution sample detection.

The remaining part of the paper is organized as follows. In section 2 we briefly report related works relevant to the current research. In section 3 and section 4 we introduce some background notions and details of our model formulations, respectively. Datasets, baselines, and experimental results are described in section 5. Finally, we draw our conclusion in section 6.

2 Related Works

As part of the compressed sensing field, there are numerous theoretical and numerical analyses of recovery algorithms (Foucart & Rauhut, 2013), with iterative algorithms being one of the central approaches, for example: Iterative Soft-Thresholding Algorithm (ISTA) (Daubechies et al., 2004), Approximate message passing (AMP) (Donoho et al., 2009) Orthogonal Matching Pursuit (OMP) (Pati et al., 1993; Davis et al., 1994), and Iterative Hard-Thresholding Algorithm (IHTA) (Blumensath & Davies, 2009). There are a number of hyperparameters associated with the mentioned algorithms, such as the number of iterations and soft threshold, that can be adjusted in order to obtain a better trade-off between performance and complexity. With unfolding iterative algorithms as layers of neural networks, these parameters can be learned in an end-to-end fashion from a dataset, see for instance some variants Zhang & Ghanem (2018); Metzler et al. (2017); yang et al. (2016); Borgerding et al. (2017); Sprechmann et al. (2015).

A non-parametric Bayesian approach to dictionary learning has been introduced in Zhou et al. (2009; 2012), where the authors consider a fully Bayesian joint compressed sensing inversion and dictionary learning. Besides, their atoms are drawn and fixed a priori. Bayesian compressed sensing (Ji et al., 2008) leverages relevance vector machines (RVMs) (Tipping, 2001) and uses a hierarchical prior to model distributions of each entry. This line of work quantifies the uncertainty of recovered entries while assuming a fixed dictionary. In contrast, in our work, the source of uncertainty is the unknown dictionary over which we define a distribution.

Learning ISTA was first introduced in Gregor & LeCun (2010) with many follow-up variations. The follow-up works in Behrens et al. (2021); Liu et al. (2019); Chen et al. (2021); Wu et al. (2020) provides various guidelines for architecture change to improve LISTA for example in convergence, parameter efficiency, step size and threshold adaptation, and overshooting. These works assume fixed and known sparsifying dictionaries

and fixed sensing matrices. Steps toward relaxing these assumptions were taken in Aberdam et al. (2021); Behboodi et al. (2022); Schnoor et al. (2022). In Aberdam et al. (2021), the authors propose a model to deal with varying sensing matrices. The authors in Schnoor et al. (2022); Behboodi et al. (2022) provide an architecture that can both incorporate varying sensing matrices and learn dictionaries. However, their focus is on the theoretical analysis of the model. There are theoretical studies on the convergence and generalization of unfolded networks, see for example: Giryes et al. (2018); Pu et al. (2022); Aberdam et al. (2021); Pu et al. (2022); Chen et al. (2018); Behboodi et al. (2022); Schnoor et al. (2022). In our paper, not only do we consider varying sensing matrices and dictionaries, but we also model a distribution over dictionaries and thereby characterize epistemic uncertainty.

Variational autoencoders (VAEs) is a framework that learns a generative model over the data through latent variables (Kingma & Welling, 2013; Rezende et al., 2014). When there are data-sample-specific dictionaries in our proposed model, it reminisces extensions of VAEs to the recurrent setting (Chung et al., 2015; 2016), which assumes a sequential structure in the data and imposes temporal correlations between the latent variables. There are also connections and similarities to Markov state-space models, such as the ones described in Krishnan et al. (2017). When we employ global dictionaries in VLISTA, the model essentially becomes a variational Bayesian Recurrent Neural Network. Variational Bayesian neural networks have been introduced in Blundell et al. (2015), with independent priors and variational posteriors for each layer. This work has been further extended to recurrent settings in Fortunato et al. (2019). The main difference between these works and our setting is the prior and variational posterior; in our case, the prior and variational posterior for each step is conditioned on previous steps, instead of being fixed across steps.

3 Background

3.1 Sparse linear inverse problems

We consider the following linear inverse problem: $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{s}$, where we have access to a set of linear measurements $\boldsymbol{y} \in \mathbb{R}^m$ of an unknown signal $\boldsymbol{s} \in \mathbb{R}^n$, acquired through the forward operator $\boldsymbol{\Phi}$ represented by a matrix $\in \mathbb{R}^{m \times n}$. Generally speaking, in a compressed sensing context, $\boldsymbol{\Phi}$ is called the sensing, or measurements, matrix and it represents an underdetermined system of equations for $m < n$. The problem of reconstructing $\boldsymbol{s}$ from $\boldsymbol{y}$ and $\boldsymbol{\Phi}$ is ill-posed due to the shape of the forward operator. To uniquely solve for $\boldsymbol{s}$, a common assumption about the signal model is that it admits a sparse representation, $\boldsymbol{x} \in \mathbb{R}^b$, in a given basis, $\{e_i \in \mathbb{R}^n\}_{i=0}^b$. The e_i vectors are called *atoms* and are collected as the columns of a matrix $\boldsymbol{\Psi} \in \mathbb{R}^{n \times b}$ termed the *sparsifying dictionary*. Therefore, the problem of estimating $\boldsymbol{s}$ given a limited number of observations $\boldsymbol{y}$ through the operator $\boldsymbol{\Phi}$ is translated to a sparse recovery problem: $\boldsymbol{x}^* = \arg\min_{\boldsymbol{x}} \|\boldsymbol{x}\|_0$ s.t. $\boldsymbol{y} = \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{x}$. Given that the l_0 pseudo-norm requires solving an NP-hard problem, the l_1 norm is used instead as a convex relaxation of the problem.

A proximal gradient descent-based approach for solving such a problem yields the ISTA algorithm:

$$\boldsymbol{x}_t = \eta_{\theta_t}\left(\boldsymbol{x}_{t-1} + \gamma_t(\boldsymbol{\Phi}\boldsymbol{\Psi})^H(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}\boldsymbol{x}_{t-1})\right), \tag{1}$$

where t is the index of the current iteration, $\boldsymbol{x}_t$ ($\boldsymbol{x}_{t-1}$) is the reconstructed sparse vector at the current (previous) layer, and θ_t and γ_t are the *soft-threshold* and *step size* hyperparameters, respectively. Specifically, θ_t characterizes the *soft-threshold function* given by: $\eta_{\theta_t}(\boldsymbol{x}) = \text{sign}(\boldsymbol{x})(|\boldsymbol{x}| - \theta_t)_+$. In the ISTA formulation, those two parameters are shared across all the iterations: $\gamma_t, \theta_t \to \gamma, \theta$. In what follows we use the terms "layers" and "iterations" interchangeably when describing ISTA and its variations.

3.2 LISTA

LISTA (Gregor & LeCun, 2010) is an unfolded version of the ISTA algorithm in which each iteration is parametrized by learnable matrices. Specifically, LISTA reinterprets Equation 1 as defining the layer of a feed-forward neural network implemented as $S_{\theta_t}\left(\boldsymbol{V}_t\boldsymbol{x}_{t-1} + \boldsymbol{W}_t\boldsymbol{y}\right)$ where $\boldsymbol{V}_t, \boldsymbol{W}_t$ are learnt from a dataset. In that way, those weights implicitly contain information about $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}$ that are assumed to be known and fixed. As LISTA, also its variations, e.g., Analytic LISTA (ALISTA) (Liu et al., 2019), NALISTA (Behrens et al., 2021) and HyperLISTA (Chen et al., 2021), require similar constraints such as a fixed dictionary and

sensing matrix to reach the best performance. However, there are situations in which either one or none of the conditions is met. As an example, in wireless communication, and particularly millimeter wave channel estimation problems, the sensing matrix depends on the chosen transmit-receive antenna patterns and can vary in measurement rounds.

4 Method

4.1 Augmented Dictionary Learning ISTA (A-DLISTA)

To deal with situations where $\mathbf{\Psi}$ is not known, and $\mathbf{\Phi}$ is changing across samples, one can unfold the ISTA algorithm and re-parametrize the dictionary as a learnable matrix. Such an algorithm is termed Dictionary Learning ISTA (DLISTA) (Pezeshki et al., 2022; Behboodi et al., 2022; Aberdam et al., 2021) and, similarly to Equation 1, each layer is formulated as:

$$\boldsymbol{x}_t = \eta_{\theta_t}\left(\boldsymbol{x}_{t-1} + \gamma_t(\mathbf{\Phi}\mathbf{\Psi}_t)^\top(\boldsymbol{y} - \mathbf{\Phi}\mathbf{\Psi}_t\boldsymbol{x}_{t-1})\right), \tag{2}$$

with one last linear layer mapping $\boldsymbol{x}$ to the reconstructed signal $\boldsymbol{s}$. The model can be trained end-to-end to learn all $\theta_t, \gamma_t, \mathbf{\Psi}_t$.

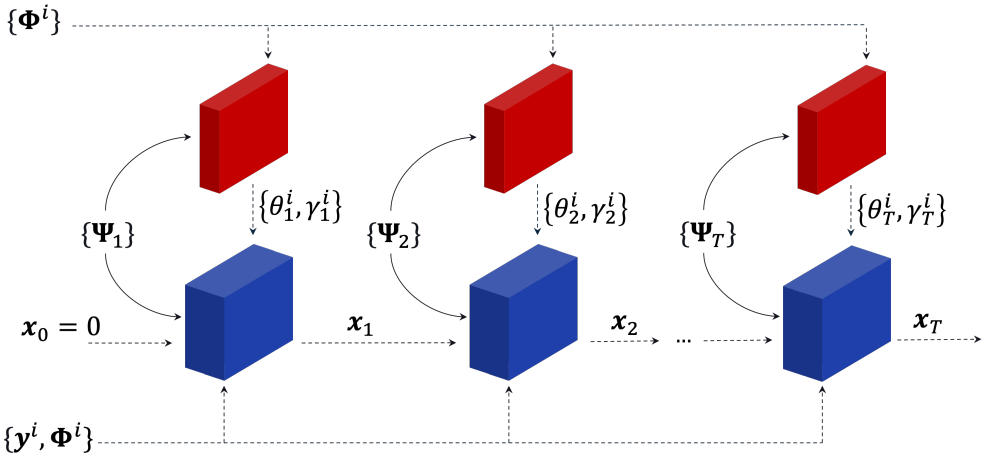

Figure 1: A-DLISTA architecture. Each blue block represents a single soft-thresholding operation parametrized by the dictionary $\mathbf{\Psi}_t$ together with threshold and step size $\{\theta_t, \gamma_t\}$ at layer t. The red blocks represent the augmentation network (with shared parameters across layers) that adapts $\{\theta_t, \gamma_t\}$ for layer t based on the dictionary $\mathbf{\Psi}_t$ and the current measurement setup $\mathbf{\Phi}^i$ for the $i-$th data sample. In such a setup, the dictionary is parametrized as a learnable matrix and can be shared across layers.

When the sensing matrix is fixed, the network can hopefully find good choices for parameters through end-to-end training. However, when the sensing matrix $\mathbf{\Phi}$ changes across different data samples, i.e., $\mathbf{\Phi} \to \mathbf{\Phi}^i$, it is not guaranteed anymore that there is a unique choice of γ_t and θ_t for all different $\mathbf{\Phi}^i$. This is supported by our theoretical analysis provided in Appendix B. Since these parameters can be determined for a given $\mathbf{\Phi}^i$ and $\mathbf{\Psi}_t$, we propose using an augmentation network that determines γ_t and θ_t from each pair of $\mathbf{\Phi}^i$ and $\mathbf{\Psi}_t$. We report in Appendix B the full theoretical discussion. We show in Figure 1 the resulting A-DLISTA model. At each layer, A-DLISTA relies on two basic operations, namely, soft-threshold (blue blocks in Figure 1) and augmentation (red blocks in Figure 1). The former represents an ISTA-like iteration parametrized by the set of weights: $\{\mathbf{\Psi}_t, \theta_t, \gamma_t\}$, whilst the latter is implemented using an encoder-decoder-like type of network. As shown in the figure, the augmentation network takes as input the sensing matrix for the given data sample, $\mathbf{\Phi}^i$, together with the dictionary learned at the layer for which the augmentation model will generate the θ and γ parameters. Through such an operation, the A-DLISTA adapts those last two parameters to the current data sample. We report more details about the augmentation network in the supplementary materials (Appendix C).

4.2 Variational Learning ISTA

Although A-DLISTA possesses adaptivity to data samples, it still assumes the existence of a ground truth dictionary. We relax such a hypothesis by defining a probability distribution over $\mathbf{\Psi}_t$ and formulate a variational approach, titled VLISTA, to jointly solve the dictionary learning and the sparse recovery problems. To forge our variational framework whilst retaining the helpful adaptivity property of A-DLISTA, we re-interpret the soft-thresholding layers of the latter as part of a likelihood model defining the output mean for the reconstructed signal. Given its recurrent-like structure (Chung et al., 2015), we equip VLISTA with a conditional trainable prior where the condition is given by the dictionary sampled at the previous iteration. Therefore, the full model comprises three components, namely, the conditional prior $p_\xi(\cdot)$, the variational posterior $q_\phi(\cdot)$, and the likelihood model, $p_\Theta(\cdot)$. All components are parametrized by neural networks whose outputs represent the parameters for the underlying probability distribution. In what follows, we describe more in detail the various building blocks of the VLISTA model.

4.2.1 Prior Model

The conditional prior, $p_\xi(\mathbf{\Psi}_t|\mathbf{\Psi}_{t-1})$, is modelled as a Gaussian distribution whose parameters are conditioned on the previously sampled dictionary. We implement $p_\xi(\cdot)$ as a neural network, $f_\xi(\cdot) = [f^\mu_{\xi_1} \circ g_{\xi_0}(\cdot), f^{\sigma^2}_{\xi_2} \circ g_{\xi_0}(\cdot)]$, with trainable parameters $\xi = \{\xi_0, \xi_1, \xi_2\}$. The model's architecture comprises a shared convolutional block followed by two different branches generating the mean and the standard deviation, respectively, of the Gaussian distribution. Therefore, at layer t, the prior conditional distribution is given by: $p_\xi(\mathbf{\Psi}_t|\mathbf{\Psi}_{t-1}) = \prod_{i,j} \mathcal{N}(\mathbf{\Psi}_{t;i,j}|\mu_{t;i,j} = f^\mu_{\xi_1}(g_{\xi_0}(\mathbf{\Psi}_{t-1}))_{i,j}; \sigma_{t;i,j} = f^{\sigma^2}_{\xi_2}(g_{\xi_0}(\mathbf{\Psi}_{t-1}))_{i,j})$, where the indices i, j run over the rows and columns of $\mathbf{\Psi}_t$. To simplify our expressions, we will abuse notation and refer to distributions like the former one as:

$$p_\xi(\mathbf{\Psi}_t|\mathbf{\Psi}_{t-1}) = \mathcal{N}(\mathbf{\Psi}_t|\boldsymbol{\mu}_t; \boldsymbol{\sigma}^2{}_t), \quad \text{where} \tag{3}$$
$$\boldsymbol{\mu}_t = f^\mu_{\xi_1}(g_{\xi_0}(\mathbf{\Psi}_{t-1})); \quad \boldsymbol{\sigma}^2{}_t = f^{\sigma^2}_{\xi_2}(g_{\xi_0}(\mathbf{\Psi}_{t-1}))$$

We will use the same type of notation throughout the rest of the manuscript to simplify formulas. The prior's design allows for enforcing a dependence of the dictionary at iteration t to the one sampled at the previous iteration. Thus, allowing us to refine $\mathbf{\Psi}_t$ as the iterations proceed. The only exception to such a process is the prior imposed over the dictionary at $t = 1$ since there is no previously sampled dictionary in this case. We handle such an exception by assuming a standard Gaussian distributed $\mathbf{\Psi}_1$. Finally, the joint prior distribution over the dictionaries for VLISTA is given by:

$$p_\xi(\mathbf{\Psi}_{1:T}) = \mathcal{N}(\mathbf{\Psi}_1|\mathbf{0}; \mathbf{1}) \prod_{t=2}^{T} \mathcal{N}(\mathbf{\Psi}_t|\boldsymbol{\mu}_t; \boldsymbol{\sigma}^2{}_t) \tag{4}$$

where $\boldsymbol{\mu}_t$ and $\boldsymbol{\sigma}^2{}_t$ are defined in Equation 3.

4.2.2 Posterior Model

Similarly to the prior model, also the variational posterior is modelled as a Gaussian distribution parametrized by a neural network $f_\phi(\cdot) = [f^\mu_{\phi_1} \circ h_{\phi_0}(\cdot), f^{\sigma^2}_{\phi_2} \circ h_{\phi_0}(\cdot)]$ which outputs the mean and variance for the underlying probability distribution

$$q_\phi(\mathbf{\Psi}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}^i, \mathbf{\Phi}^i) = \mathcal{N}(\mathbf{\Psi}_t|\boldsymbol{\mu}_t; \boldsymbol{\sigma}^2{}_t), \quad \text{where} \tag{5}$$
$$\boldsymbol{\mu}_t = f^\mu_{\phi_1}(h_{\phi_0}(\boldsymbol{x}_{t-1}, \boldsymbol{y}^i, \mathbf{\Phi}^i)); \quad \boldsymbol{\sigma}^2{}_t = f^{\sigma^2}_{\phi_2}(h_{\phi_0}(\boldsymbol{x}_{t-1}, \boldsymbol{y}^i, \mathbf{\Phi}^i))$$

The posterior distribution for the dictionary, $\mathbf{\Psi}_t$, at layer t is conditioned on the data, $\{\boldsymbol{y}^i, \mathbf{\Phi}^i\}$, as well as on the reconstructed signal at the previous layer, $\boldsymbol{x}_{t-1}$. Therefore, the joint posterior probability over the dictionaries at each layer is given by:

$$q_\phi(\mathbf{\Psi}_{1:T}|\boldsymbol{x}_{1:T}, \boldsymbol{y}^i, \mathbf{\Phi}^i) = \prod_{t=1}^{T} q_\phi(\mathbf{\Psi}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}^i, \mathbf{\Phi}^i) \tag{6}$$

Concerning both the prior and posterior models, we chose Gaussian distributions mostly for computational and implementation convenience. However, our framework is not at all restricted in that regard and it can support any flexible distribution family as long as the distributions involved are reparametrizable (Kingma & Welling, 2013) (such that we can obtain gradients of the random samples with respect to their parameters) and we can evaluate, and differentiate, their density. Some examples would be mixtures of Gaussians (to get heavier tails) or even distributions arising from normalizing flows (Rezende & Mohamed, 2015).

4.2.3 Likelihood Model

At the heart of the reconstruction module, there is the soft-thresholding block of A-DLISTA. Similarly to the prior and posterior, the likelihood distribution is modelled as a Gaussian parametrized by the output of an A-DLISTA block. Specifically, the likelihood network generates only the mean vector for the Gaussian distribution since we treat the standard deviation as a tunable hyperparameter. Therefore, we interpret the reconstructed sparse vector at a given layer as the mean of the likelihood distribution. The joint log-likelihood distribution can then be formulated as:

$$\log p_\Theta(\boldsymbol{x}_{1:T} = \boldsymbol{x}^i_{gt} | \boldsymbol{\Psi}_{1:T}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i) = \sum_{t=1}^{T} \log \mathcal{N}(\boldsymbol{x}^i_{gt} | \boldsymbol{\mu}_t, {\boldsymbol{\sigma}^2}_t), \quad \text{where} \tag{7}$$
$$\boldsymbol{\mu}_t = \text{A-DLISTA}(\boldsymbol{\Psi}_t, \boldsymbol{x}_{t-1}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i; \Theta); \quad {\boldsymbol{\sigma}^2}_t = \delta$$

where δ is a hyperparameter of the network, $\boldsymbol{x}^i_{gt}$ represents the ground truth value for the underlying unknown sparse signal for the i-th data sample, Θ is the set of A-DLISTA's parameters, and $p_\Theta(\boldsymbol{x}_{1:T} = \boldsymbol{x}^i_{gt} | \cdot)$ represents the likelihood for the ground truth x^i_{gt}, at each time step $t \in [1, T]$, under the likelihood model given the current reconstruction. Note that in Equation 7 we use the same $\boldsymbol{x}^i_{gt}$ through the entire sequence $t \in [1, T]$. To help visualize the formulation of VLISTA, we report in Figure 2 its graphical model. Moreover, we report more details about the models' architecture and the objective function in the supplementary material (Appendix C and Appendix D).

4.2.4 Training Objective

We train all the different components of VLISTA in an end-to-end fashion through maximization of the Evidence Lower Bound (ELBO):

$$\begin{aligned} \text{ELBO} = &\sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\Psi}_{1:t} \sim q_\phi(\boldsymbol{\Psi}_{1:t} | \boldsymbol{x}_{0:t-1}, \boldsymbol{D}^i)} \Big[\log p_\Theta(\boldsymbol{x}_t = \boldsymbol{x}^i_{gt} | \boldsymbol{\Psi}_{1:t}, \boldsymbol{D}^i) \Big] \\ &- \sum_{t=2}^{T} \mathbb{E}_{\boldsymbol{\Psi}_{1:t-1} \sim q_\phi(\boldsymbol{\Psi}_{1:t-1} | \boldsymbol{x}_{t-1}, \boldsymbol{D}^i)} \Big[D_{KL} \Big(q_\phi(\boldsymbol{\Psi}_t | \boldsymbol{x}_{t-1}, \boldsymbol{D}^i) \parallel p_\xi(\boldsymbol{\Psi}_t | \boldsymbol{\Psi}_{t-1}) \Big) \Big] \\ &- D_{KL} \Big(q_\phi(\boldsymbol{\Psi}_1 | \boldsymbol{x}_0, \boldsymbol{D}^i) \parallel p(\boldsymbol{\Psi}_1) \Big) \end{aligned} \tag{8}$$

where $\boldsymbol{D}^i = \{\boldsymbol{y}^i, \boldsymbol{\Phi}^i\}$, T is the number of layers (or iterations) of the model, and $\boldsymbol{x}_0 = \boldsymbol{0}$.

The likelihood contribution, the first term in Equation 8, is evaluated by marginalizing over the dictionaries sampled from the posterior $q_\phi(\boldsymbol{\Psi}_{1:t} | \boldsymbol{x}_{0:t-1}, \boldsymbol{D}^i)$. Instead, the last two terms in Equation 8 represent the KL divergence contribution evaluated between the prior and the posterior distributions. Specifically, the prior in the last term is not conditioned on the previously sampled dictionary given that $p_\xi(\boldsymbol{\Psi}_1) \rightarrow p(\boldsymbol{\Psi}_1) = \mathcal{N}(\boldsymbol{\Psi}_1 | \boldsymbol{0}; \boldsymbol{1})$ (see Equation 3 and Equation 4).

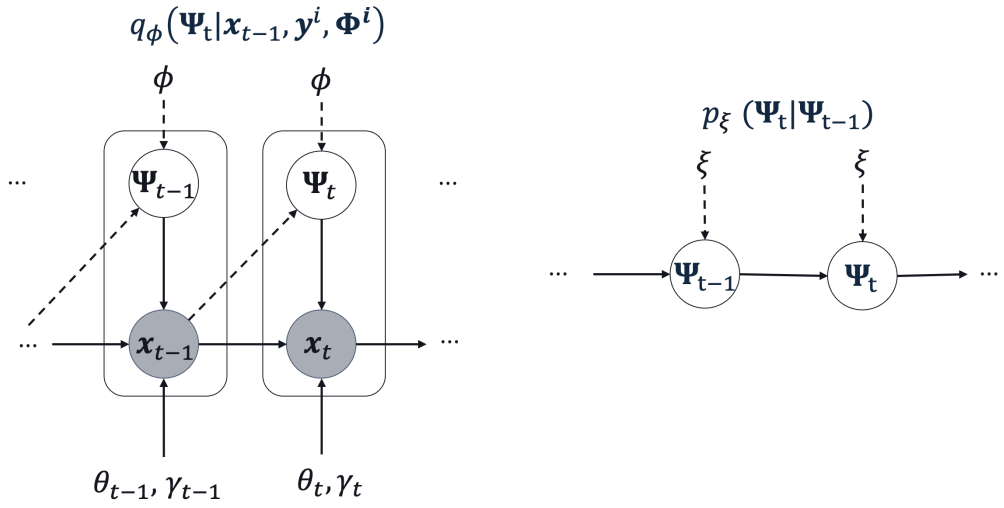

Figure 2: VLISTA graphical model. Dependencies on $\boldsymbol{y}^i, \boldsymbol{\Phi}^i$ are factored out for simplicity. The sampling is done only based on the posterior $q_{\boldsymbol{\phi}}(\boldsymbol{\Psi}_t|\boldsymbol{x}_{t-1}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i)$. Dashed lines represent variational approximations.

5 Experiments

5.1 Datasets and Baselines

To assess the performance of our models and compare them against classical and ML-based baselines, we leverage three datasets, namely MNIST, CIFAR10, and a synthetic one. Concerning the latter, we follow a similar prescription as in Chen et al. (2018); Liu & Chen (2019); Behrens et al. (2021). However, differently from the mentioned authors, we generate a different $\boldsymbol{\Phi}$ matrix for each datum by sampling i.i.d. entries from a standard Gaussian distribution, $\phi_{ij} \sim \mathcal{N}(0, 1/m)$, where m is the number of columns of $\boldsymbol{\Phi}$. To generate the ground truth sparse signals $\boldsymbol{x} \in \mathbb{R}^b$ we sample the entries from a standard Gaussian as well and then we set to zero elements as dictated by a Bernoulli distribution with $p = 0.1$. We generate 5K samples and use 3K for training, 1K for model selection and 1K for testing. Concerning the MNIST and CIFAR10 datasets, we train the models using the full images, i.e., no crop applied. Moreover, for CIFAR10, we grayscale and normalize images. To emulate a scenario with varying sensing matrices, we adopt the following procedure for both MNIST and CIFAR10. For each data sample, we generate a sensing matrix, $\boldsymbol{\Phi}^i$, by randomly sampling its entries from a standard Gaussian distribution. Subsequently, for each pair of sensing matrices, $\boldsymbol{\Phi}^i$, and ground truth image, $\boldsymbol{s}^i$, we generate the corresponding observations as $\boldsymbol{y}^i = \boldsymbol{\Phi}^i \boldsymbol{s}^i$. We compare A-DLISTA and VLISTA models against "classical" and "ML" baselines. By "classical" we refer to the ISTA algorithm for which we pre-compute the dictionary by either considering the canonical or the wavelet basis or using the SPCA algorithm. Instead, by "ML" we refer to different unfolded learning versions of ISTA (e.g., LISTA). To prove the benefit of adaptivity, we conduct an ablation study on A-DLISTA by removing its augmentation network and making the parameters θ_t, γ_t learnable through backpropagation only. We refer to the non-augmented version of A-DLISTA as DLISTA (see subsection 4.1 for more details). Hence, for DLISTA, θ_t and γ_t cannot be adapted to the specific input sensing matrix. Moreover, we consider BCS as a specific Bayesian baseline for VLISTA. Finally, we conduct Out-Of-Distribution (OOD) detection experiments. To compare the performance of the different models, we fixed the number of layers to three. Concerning the "classical" baselines, they do not possess learnable parameters. Therefore, we perform an extensive grid-search to find the best hyperparameters for them. More details about the training procedure and ablation studies can be found in Appendix D and Appendix E.

5.2 Synthetic Dataset

Concerning the synthetic dataset, we compare models' performance by reporting the median of the c.d.f. for the reconstruction NMSE (Figure 3).

Compared to the other models, the adaptivity of A-DLISTA seems to be beneficial. However, concerning VLISTA, we observe a drop in performance. Such a behaviour is consistent across the various experiments we perform and we can ground it to a few reasons. One contribution to such a drop might come from the noise naturally injected at training time due to the random sampling procedure used to generate the dictionary. Furthermore, the amortization gap that affects all models based on amortized variational inference (Cremer et al., 2018) can also contribute to such effect. Despite that, VLISTA still performs comparably to BCS. Finally, we notice that ALISTA and NALISTA do not perform as well as the other models. Such a result can be justified by recalling the optimization procedure those two models require to evaluate the weight matrix W. Indeed, the computation of the W matrix requires a fix sensing matrix. Such a requirement is not satisfied in the current setup (as far as non-static measurements are concerned, we averaged across multiple $\mathbf{\Phi}^i$, thus obtaining a non-optimal W matrix). To support our hypothesis, we report in Appendix E results considering a static measurement scenario for which ALISTA and NALISTA report very high performance.

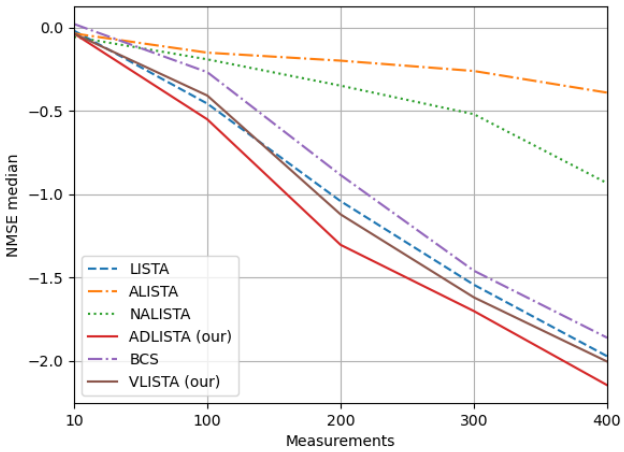

Figure 3: NMSE's median in dB (the lower the better) for a different number of measurements.

5.3 Image Reconstruction - MNIST & CIFAR10

Concerning the MNIST and CIFAR10 datasets, performance of different models are assessed by quantifying the Structural Similarity Index Measure (SSIM). As for the synthetic dataset, we experience strong instabilities in ALISTA and NALISTA training as a result of non-static measurement settings. We therefore do not report results for these models. However, as we already mentioned in the previous section, it is fundamental to emphasize that the poor performance we observe for the ALISTA and NALISTA models is a result of the particular setup we use in our experiments, which differs from the static case considered in the formulation of such models (see Appendix E for results using a static measurements scenario). Therefore, it is fundamental to take into account such a key element when interpreting our results. By looking at the results in Table 1 and Table 2, we can draw similar conclusions as for the synthetic dataset. Moreover, we report results from three classical baselines (subsection 5.1). As far as non-Bayesian models are concerned, A-DLISTA shows the best results. In addition, by comparing A-DLISTA with its non-augmented version, DLISTA, we can see the advantages of using an augmentation network to make the model adaptive. As far as Bayesian approaches are concerned, VLISTA performs better than BCS. It is important to note, however, that BC does not have trainable parameters, as opposed to VLISTA. Therefore, the higher performance of VLISTA comes at the price of an expensive training procedure. As with the synthetic dataset, also for MNIST and CIFAR10, VLISTA exhibits a drop in performance compared to A-DLISTA. In spite of the fact that the previous results seem to suggest that the variational model does not provide any advantage over A-DLISTA, this is not the case. It is noteworthy that compared to all other non-Bayesian approaches, VLISTA is capable of detecting OODs, a critical property for some domain-specific applications, such as wireless communication.

5.4 Out Of Distribution Detection

In this section, we focus on one of the most important differences between non-Bayesian models and VLISTA for solving inverse linear problems. Indeed, unlike any non-Bayesian approach to compressed sensing, VLISTA

Table 1: MNIST SSIM (the higher the better) for different number of measurements. First three rows correspond to “classical" baselines. We highlight in bold the best performance for Bayes and Non-Bayes models.

		SSIM ↑				
		number of measurements				
		1 ($\times e^{-1}$)	10 ($\times e^{-1}$)	100 ($\times e^{-1}$)	300 ($\times e^{-1}$)	500 ($\times e^{-1}$)
Non-Bayes	Canonical	$0.39_{\pm 0.12}$	$0.56_{\pm 0.04}$	$2.20_{\pm 0.04}$	$3.75_{\pm 0.05}$	$4.94_{\pm 0.06}$
	Wavelet	$0.40_{\pm 0.09}$	$0.56_{\pm 0.06}$	$2.30_{\pm 0.06}$	$3.90_{\pm 0.05}$	$5.05_{\pm 0.01}$
	SPCA	$0.45_{\pm 0.11}$	$0.65_{\pm 0.06}$	$2.72_{\pm 0.06}$	$3.52_{\pm 0.08}$	$4.98_{\pm 0.08}$
	LISTA	$\mathbf{0.96_{\pm 0.01}}$	$1.11_{\pm 0.01}$	$3.70_{\pm 0.01}$	$5.36_{\pm 0.01}$	$6.31_{\pm 0.01}$
	DLISTA	$\mathbf{0.96_{\pm 0.01}}$	$1.09_{\pm 0.01}$	$4.01_{\pm 0.02}$	$5.57_{\pm 0.01}$	$6.26_{\pm 0.01}$
	A-DLISTA (our)	$\mathbf{0.96_{\pm 0.01}}$	$\mathbf{1.17_{\pm 0.01}}$	$\mathbf{4.79_{\pm 0.01}}$	$\mathbf{6.15_{\pm 0.01}}$	$\mathbf{6.70_{\pm 0.01}}$
Bayes	BCS	$0.05_{\pm 0.01}$	$0.60_{\pm 0.01}$	$1.10_{\pm 0.01}$	$4.48_{\pm 0.02}$	$\mathbf{6.23_{\pm 0.02}}$
	VLISTA (our)	$\mathbf{0.80_{\pm 0.03}}$	$\mathbf{0.94_{\pm 0.02}}$	$\mathbf{3.29_{\pm 0.01}}$	$\mathbf{4.73_{\pm 0.01}}$	$6.02_{\pm 0.01}$

Table 2: CIFAR10 SSIM (the higher the better) for different number of measurements. First three rows correspond to “classical" baselines. We highlight in bold the best performance for Bayes and Non-Bayes models.

		SSIM ↑				
		number of measurements				
		1 ($\times e^{-1}$)	10 ($\times e^{-1}$)	100 ($\times e^{-1}$)	300 ($\times e^{-1}$)	500 ($\times e^{-1}$)
Non-Bayes	Canonical	$0.17_{\pm 0.10}$	$0.21_{\pm 0.02}$	$0.33_{\pm 0.02}$	$0.47_{\pm 0.02}$	$0.58_{\pm 0.03}$
	Wavelet	$0.23_{\pm 0.22}$	$0.42_{\pm 0.02}$	$1.44_{\pm 0.06}$	$2.52_{\pm 0.09}$	$3.43_{\pm 0.08}$
	SPCA	$0.31_{\pm 0.19}$	$0.43_{\pm 0.02}$	$1.53_{\pm 0.04}$	$2.66_{\pm 0.08}$	$3.58_{\pm 0.07}$
	LISTA	$\mathbf{1.34_{\pm 0.02}}$	$1.67_{\pm 0.02}$	$3.10_{\pm 0.01}$	$4.20_{\pm 0.01}$	$4.71_{\pm 0.01}$
	DLISTA	$1.16_{\pm 0.02}$	$1.96_{\pm 0.02}$	$4.50_{\pm 0.01}$	$5.15_{\pm 0.01}$	$5.42_{\pm 0.01}$
	A-DLISTA (our)	$\mathbf{1.34_{\pm 0.02}}$	$\mathbf{1.77_{\pm 0.02}}$	$\mathbf{4.74_{\pm 0.01}}$	$\mathbf{5.26_{\pm 0.01}}$	$\mathbf{5.83_{\pm 0.01}}$
Bayes	BCS	$0.04_{\pm 0.01}$	$0.48_{\pm 0.01}$	$0.59_{\pm 0.01}$	$1.29_{\pm 0.01}$	$1.91_{\pm 0.01}$
	VLISTA (our)	$\mathbf{0.86_{\pm 0.03}}$	$\mathbf{1.25_{\pm 0.03}}$	$\mathbf{3.59_{\pm 0.02}}$	$\mathbf{4.01_{\pm 0.01}}$	$\mathbf{4.36_{\pm 0.01}}$

allows quantifying uncertainties on the reconstructed signals. Thus, enabling OOD detection without the need to access ground truth data at inference time. Moreover, whilst other Bayesian approaches (Ji et al., 2008; Zhou et al., 2014) usually focus on designing specific priors to satisfy the sparsity constraint on the reconstructed signal after marginalization, VLISTA completely overcomes such an issue as the thresholding operations are not affected by the marginalization over dictionaries. To prove that VLISTA can detect OOD samples, we employ the MNIST dataset. First, we split the full dataset into two subsets named “Train”, or In-Distribution (ID), and OOD. The ID subset contains images from three digits only, namely, 0, 3, and 7 (randomly chosen). Instead, the OOD subset contains images from all the other digits. Then, we split the ID partition into training and test sets and train VLISTA on the former one. Once trained, we evaluate the model performance by considering reconstructions from the test set (ID) and the OOD partition. We reconstruct 100 times every single image, sampling every time a new dictionary. Subsequently, as a summarizing statistics, we compute the variance's c.d.f. of the per-pixel standard deviation ($var_{\sigma_{pp}}$) across reconstructions. Subsequently, to assess whether a given digit belongs to the ID or OOD distribution, we compute the p-value for $var_{\sigma_{pp}}$ by employing the two-sample t-test.

Moreover, to assess whether OOD detection is robust to measurement noise, we repeat the same test at different noise levels. As a baseline for the current task, we consider BCS. Due to the different nature of the BCS framework, we employ a slightly different procedure to evaluate its p-values. Specifically, we use the same ID and OOD splits as VLISTA. However, for BCS, we consider the c.d.f. of the reconstruction error that is evaluated by the model itself. The rest of the procedure is the same as for VLISTA. We report the results for OOD detection in Figure 4. As we can see from the figure, VLISTA outperforms BCS for each noise level showing a lower p-value than BCS which corresponds to a higher rejection power. As expected, at higher Signal-to-Noise Ratio (SNR) values, due to a larger overlap between ID and OOD samples distributions, we observe a larger p-value meaning that OOD rejection becomes more difficult. However, we can see that whilst VLISTA is still capable of detecting OOD samples, BCS fails when the SNR, expressed in decibels, is greater than 10. As a reference point to define whether the model is correctly rejecting OOD samples or not, we report in Figure 4 the 5% line for the p-value. Such a value is typically used as a reference in hypothesis testing to decide whether or not to reject the null hypothesis.

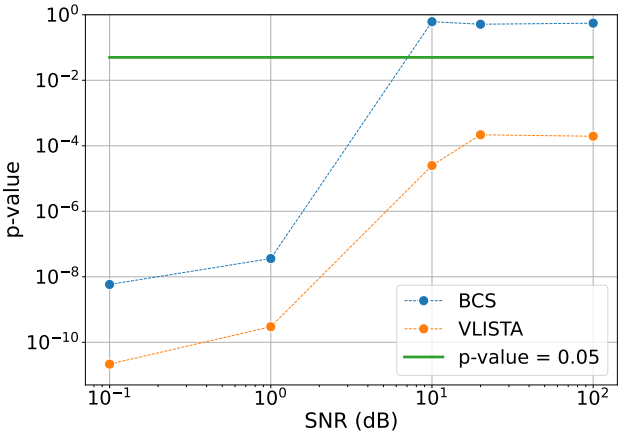

Figure 4: p-value for OOD rejection as a function of the noise level. The green line represents a reference p-value equal to 0.05.

6 Conclusion

Our study presents a variational approach, dubbed VLISTA, that can solve both dictionary learning and sparse recovery problems simultaneously. Typically, compressed sensing frameworks require a ground truth dictionary to reconstruct the signal. Furthermore, state-of-the-art LISTA-like models typically use a stationary measurement setup. In our work, we relax both assumptions. First, we show that it is possible to design a soft-thresholding algorithm, termed A-DLISTA, that can handle different sensing matrices and adapt its parameters to the given data instance. We theoretically justify the use of an augmentation network which adapts the threshold and step size for each layer based on the current input and the learned dictionary. As a final step, we introduce a probability distribution for the existence of a ground truth dictionary. Based on such an assumption, we propose the VLISTA variational framework for solving the compressed sensing problem. Our empirical results demonstrate that A-DLISTA is able to improve performance compared to classical and ML baselines in a non-static measurement scenario. Furthermore, while VLISTA does not outperform A-DLISTA, the variational framework allows to evaluate uncertainties over reconstructed signals useful for detecting OOD. In contrast, none of the non-Bayesian models are capable of performing such a task. Moreover, unlike other Bayesian approaches to compressed sensing, VLISTA does not require to design specific priors to preserve sparsity after marginalization over the reconstructed sparse signal; the averaging operation does not concern the sparse signal itself, but the sparsifying dictionary.

References

Aviad Aberdam, Alona Golts, and Michael Elad. Ada-LISTA: Learned Solvers Adaptive to Varying Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2021.

Arash Behboodi, Holger Rauhut, and Ekkehard Schnoor. Compressive Sensing and Neural Networks from a Statistical Learning Perspective. In Gitta Kutyniok, Holger Rauhut, and Robert J. Kunsch (eds.), *Compressed Sensing in Information Processing*, Applied and Numerical Harmonic Analysis, pp. 247–277. Springer International Publishing, Cham, 2022.

Freya Behrens, Jonathan Sauder, and Peter Jung. Neurally Augmented ALISTA. In *International Conference on Learning Representations*, 2021.

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

Charles Blundell, Julien Cornebise, Koray Kavukcuoglu, and Daan Wierstra. Weight uncertainty in neural network. In *International conference on machine learning*, pp. 1613–1622. PMLR, 2015.

Mark Borgerding, Philip Schniter, and Sundeep Rangan. AMP-inspired deep networks for sparse linear inverse problems. *IEEE Transactions on Signal Processing*, 65(16):4293–4308, 2017. Publisher: IEEE.

Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Theoretical linear convergence of unfolded ISTA and its practical weights and thresholds. *Advances in Neural Information Processing Systems*, 31, 2018.

Xiaohan Chen, Jialin Liu, Zhangyang Wang, and Wotao Yin. Hyperparameter tuning is all you need for lista. *Advances in Neural Information Processing Systems*, 34, 2021.

Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. *Advances in neural information processing systems*, 28, 2015.

Junyoung Chung, Sungjin Ahn, and Yoshua Bengio. Hierarchical multiscale recurrent neural networks. *arXiv preprint arXiv:1609.01704*, 2016.

Chris Cremer, Xuechen Li, and David Duvenaud. Inference suboptimality in variational autoencoders. In *International Conference on Machine Learning*, pp. 1078–1086. PMLR, 2018.

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*, 57(11):1413–1457, 2004.

Geoffrey M. Davis, Stephane G. Mallat, and Zhifeng Zhang. Adaptive time-frequency decompositions. *Optical Engineering*, 33(7):2183–2191, July 1994.

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

Meire Fortunato, Charles Blundell, and Oriol Vinyals. Bayesian Recurrent Neural Networks. *arXiv:1704.02798 [cs, stat]*, May 2019. URL `http://arxiv.org/abs/1704.02798`. arXiv: 1704.02798.

Simon Foucart and Holger Rauhut. *A Mathematical Introduction to Compressive Sensing*. Applied and Numerical Harmonic Analysis. Springer New York, New York, NY, 2013.

Raja Giryes, Yonina C. Eldar, Alex M. Bronstein, and Guillermo Sapiro. Tradeoffs between convergence speed and reconstruction accuracy in inverse problems. *IEEE Transactions on Signal Processing*, 66(7): 1676–1690, 2018. Publisher: IEEE.

Karol Gregor and Yann LeCun. Learning fast approximations of sparse coding. In *27th International Conference on Machine Learning, ICML 2010*, 2010.

Shihao Ji, Ya Xue, and Lawrence Carin. Bayesian Compressive Sensing. *IEEE Transactions on Signal Processing*, 56(6):2346–2356, June 2008. Conference Name: IEEE Transactions on Signal Processing.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Rahul Krishnan, Uri Shalit, and David Sontag. Structured inference networks for nonlinear state space models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017.

Jialin Liu and Xiaohan Chen. Alista: Analytic weights are as good as learned weights in lista. In *International Conference on Learning Representations (ICLR)*, 2019.

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.

Chris Metzler, Ali Mousavi, and Richard Baraniuk. Learned D-AMP: Principled Neural Network based Compressive Image Recovery. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett (eds.), *Advances in Neural Information Processing Systems 30*, pp. 1770–1781. Curran Associates, Inc., 2017.

Y.C. Pati, R. Rezaiifar, and P.S. Krishnaprasad. Orthogonal matching pursuit: recursive function approximation with applications to wavelet decomposition. In *Proceedings of 27th Asilomar Conference on Signals, Systems and Computers*, pp. 40–44 vol.1, November 1993. doi: 10.1109/ACSSC.1993.342465.

Hamed Pezeshki, Fabio Valerio Massoli, Arash Behboodi, Taesang Yoo, Arumugam Kannan, Mahmoud Taherzadeh Boroujeni, Qiaoyu Li, Tao Luo, and Joseph B Soriaga. Beyond codebook-based analog beamforming at mmwave: Compressed sensing and machine learning methods. In *GLOBECOM 2022-2022 IEEE Global Communications Conference*, pp. 776–781. IEEE, 2022.

Wei Pu, Yonina C. Eldar, and Miguel R. D. Rodrigues. Optimization Guarantees for ISTA and ADMM Based Unfolded Networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8687–8691, May 2022.

Danilo Rezende and Shakir Mohamed. Variational inference with normalizing flows. In *International conference on machine learning*, pp. 1530–1538. PMLR, 2015.

Danilo Jimenez Rezende, Shakir Mohamed, and Daan Wierstra. Stochastic backpropagation and approximate inference in deep generative models. In *International conference on machine learning*, pp. 1278–1286. PMLR, 2014.

Ekkehard Schnoor, Arash Behboodi, and Holger Rauhut. Generalization Error Bounds for Iterative Recovery Algorithms Unfolded as Neural Networks. *arXiv:2112.04364*, January 2022.

P. Sprechmann, A. M. Bronstein, and G. Sapiro. Learning Efficient Sparse and Low Rank Models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 37(9):1821–1833, September 2015. ISSN 0162-8828.

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

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

yan yang, Jian Sun, Huibin Li, and Zongben Xu. Deep ADMM-Net for Compressive Sensing MRI. In D. Lee, M. Sugiyama, U. Luxburg, I. Guyon, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 29. Curran Associates, Inc., 2016.

Jian Zhang and Bernard Ghanem. ISTA-Net: Interpretable optimization-inspired deep network for image compressive sensing. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 1828–1837, 2018.

Mingyuan Zhou, Haojun Chen, Lu Ren, Guillermo Sapiro, Lawrence Carin, and John Paisley. Non-Parametric Bayesian Dictionary Learning for Sparse Image Representations. In *Advances in Neural Information Processing Systems*, volume 22. Curran Associates, Inc., 2009.

Mingyuan Zhou, Haojun Chen, John Paisley, Lu Ren, Lingbo Li, Zhengming Xing, David Dunson, Guillermo Sapiro, and Lawrence Carin. Nonparametric Bayesian Dictionary Learning for Analysis of Noisy and Incomplete Images. *IEEE Transactions on Image Processing*, 21(1):130–144, January 2012.

Zhou Zhou, Kaihui Liu, and Jun Fang. Bayesian compressive sensing using normal product priors. *IEEE Signal Processing Letters*, 22(5):583–587, 2014.

A Appendix

B Theoretical Analysis

In this section, we report about theoretical motivations for the A-DLISTA design choices. The design is motivated by considering the convergence analysis of LISTA method. We start by recalling a result from Chen et al. (2018), upon which our analysis relies. The authors of Chen et al. (2018) consider the inverse problem $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}_*$, with $\boldsymbol{x}_*$ as the ground truth sparse vector, and use the model with each layer given by:

$$\boldsymbol{x}_t = \eta_{\theta_t}\left(\boldsymbol{x}_{t-1} + \boldsymbol{W}_t^\top(\boldsymbol{y} - \boldsymbol{A}\boldsymbol{x}_{t-1})\right), \tag{9}$$

where $(\boldsymbol{W}_t, \theta_t)$ are learnable parameters.

The following result from Chen et al. (2018, Theorem 2) is adapted to noiseless setting.

Theorem B.1. *Suppose that the iterations of LISTA are given by equation 9, and assume* $\|\boldsymbol{x}_*\|_\infty \leq B$ *and* $\|\boldsymbol{x}_*\|_0 \leq s$. *There exists a sequence of parameters* $\{\boldsymbol{W}_t, \theta_t\}$ *such that*

$$\|\boldsymbol{x}_t - \boldsymbol{x}_*\|_2 \leq sB\exp(-ct), \quad \forall t = 1, 2, \ldots,$$

for constant $c > 0$ *that depend only on the sensing matrix* $\boldsymbol{A}$ *and the sparsity level* s.

It is important to note that the above convergence result only assures the *existence* of the parameters that are good for convergence but does not guarantee that the training would necessarily find them. The latter result is in general difficult to obtain.

The proof in Chen et al. (2018) has two main steps:

1. No false positive: the thresholds are chosen such that the entries outside the support of $\boldsymbol{x}_*$ remain zero. The choice of threshold, among others, depend on the coherence value of $\boldsymbol{W}_t$ and $\boldsymbol{A}$. We will provide more details below.
2. Error bounds for $\boldsymbol{x}_*$: assuming proper choice of thresholds, the authors derive bounds on the recovery error.

We focus on adapting these steps to our setup. Note that to assure there is no false positive, it is common in classical ISTA literature to start from large thresholds, so the soft thresholding function aggressively maps many entries to zero, and then gradually reduce the threshold value as the iterations progress.

B.1 Analysis with known ground-truth dictionary

Let's consider the extension of Theorem B.1 to our setup:

$$\boldsymbol{x}_t = \eta_{\theta_t}\left(\boldsymbol{x}_{t-1} + \gamma_t(\boldsymbol{\Phi}_k\boldsymbol{\Psi}_t)^H(\boldsymbol{y} - \boldsymbol{\Phi}_k\boldsymbol{\Psi}_t\boldsymbol{x}_{t-1})\right). \tag{10}$$

Note that in our case, the weight $\boldsymbol{W}_t$ is replaced with $\gamma_t(\boldsymbol{\Phi}_k\boldsymbol{\Psi}_t)$ with learnable $\boldsymbol{\Psi}_t$ and γ_t. Besides, the matrix $\boldsymbol{A}$ is replaced by $\boldsymbol{\Phi}_k\boldsymbol{\Psi}_t$, and the forward model is given by $\boldsymbol{y}_k = \boldsymbol{\Phi}_k\boldsymbol{\Psi}_o\boldsymbol{x}_*$. The sensing matrix $\boldsymbol{\Phi}_k$ can change across samples, hence the dependence on the sample index k.

If the learned dictionary $\boldsymbol{\Psi}_t$ is equal to $\boldsymbol{\Psi}_o$, the layers of our model are equal to classical iterative soft-thresholding algorithms with learnable step-size γ_t and threshold θ_t.

There are many convergence results in the literature, for example see Daubechies et al. (2004). We can use convergence analysis of iterative soft thresholding algorithms using the mutual coherence similar to Chen et al. (2018); Behrens et al. (2021). As a reminder, the mutual coherence of the matrix $\boldsymbol{M}$ is defined as:

$$\mu(\boldsymbol{M}) := \max_{1 \leq i \neq j \leq N} \left|\boldsymbol{M}_i^\top \boldsymbol{M}_j\right|, \tag{11}$$

where $\boldsymbol{M}_i$ is the i'th column of $\boldsymbol{M}$.

The convergence result requires that the mutual coherence $\mu(\boldsymbol{\Phi}_k\boldsymbol{\Psi}_o)$ be sufficiently small, for example in order of $1/(2s)$ with s the sparsity, and the matrix $\boldsymbol{\Phi}_k\boldsymbol{\Psi}_o$ is column normalized, i.e., $\|(\boldsymbol{\Phi}_k\boldsymbol{\Psi}_o)\|_2 = 1$. Then the step size can be chosen equal to one, i.e., $\gamma_t = 1$. The thresholds θ_t are chosen to avoid false positive using a similar schedule mentioned above, that is, first starting with a large threshold θ_0 and then gradually decreasing it to a certain limit. We do not repeat the derivations, and interested readers can refer to Daubechies et al. (2004); Behrens et al. (2021); Chen et al. (2018) and references therein.

Remark B.2. When the dictionary $\boldsymbol{\Psi}_0$ is known, we can adapt the algorithm to the varying sensing matrix $\boldsymbol{\Phi}_k$ by first normalizing the column $\boldsymbol{\Phi}_k\boldsymbol{\Psi}_0$. What is important to note is that the threshold choice is a function the mutual coherence of the sensing matrix. So with each new sensing matrix, the thresholds should be adapted following the mutual coherence value. This observation partially justifies the choice of thresholds as a function of the dictionary and the sensing matrix, hence the augmentation network.

B.2 Analysis with unknown dictionary

We now move to the scenario where the dictionary is itself learned, and not known in advance.

Consider the layer t of DLISTA with the sensing matrix $\boldsymbol{\Phi}_k$, and define the following parameters:

$$\tilde{\mu}(t, \boldsymbol{\Phi}_k) := \max_{1\leq i\neq j\leq N} \left|((\boldsymbol{\Phi}_k\boldsymbol{\Psi}_t)_i)^\top(\boldsymbol{\Phi}_k\boldsymbol{\Psi}_t)_j\right| \tag{12}$$

$$\tilde{\mu}_2(t, \boldsymbol{\Phi}_k) := \max_{1\leq i,j\leq N} \left|((\boldsymbol{\Phi}_k\boldsymbol{\Psi}_t)_i)^\top(\boldsymbol{\Phi}_k(\boldsymbol{\Psi}_t - \boldsymbol{\Psi}_o))_j\right| \tag{13}$$

$$\delta(\gamma, t, \boldsymbol{\Phi}_k) := \max_i \left|1 - \gamma\,\|(\boldsymbol{\Phi}_k\boldsymbol{\Psi}_t)_i\|_2^2\right| \tag{14}$$

Some comments are in order:

- The term $\tilde{\mu}$ is the **mutual coherence** of the matrix $\boldsymbol{\Phi}\boldsymbol{\Psi}_t$.
- The term $\tilde{\mu}_2$ is closely connected to **generalized mutual coherence**, however, it differs in that unlike generalized mutual coherence, it includes the diagonal inner product for $i = j$. It captures the effect of mismatch with ground-truth dictionary.
- Finally, the term $\delta(\cdot)$ is reminiscent of restricted isometry property (RIP) constant (Foucart & Rauhut, 2013), a key condition for many recovery guarantees in compressed sensing. When the columns of the matrix $\boldsymbol{\Phi}_k\boldsymbol{\Psi}_t$ is normalized, the choice of $\gamma = 1$ yield $\delta(\gamma, t, \boldsymbol{\Phi}_k) = 0$.

For the rest of the paper, for simplicity, we only kept the dependence on γ in the notation and dropped the dependence of $\tilde{\mu}, \tilde{\mu}_2$ and δ on t, $\boldsymbol{\Phi}$ and $\boldsymbol{\Psi}_t$.

Proposition B.3. *Suppose that $\boldsymbol{y} = \boldsymbol{\Phi}_k\boldsymbol{\Psi}_o\boldsymbol{x}_*$ with the support $supp(\boldsymbol{x}_*) = S$. For DLISTA iterations give as*

$$\boldsymbol{x}_t = \eta_{\theta_t}\left(\boldsymbol{x}_{t-1} + \gamma_t(\boldsymbol{\Phi}_k\boldsymbol{\Psi}_t)^H(\boldsymbol{y} - \boldsymbol{\Phi}_k\boldsymbol{\Psi}_t\boldsymbol{x}_{t-1})\right), \tag{15}$$

we have:

1. *If for all t, the pairs $(\theta_t, \gamma_t, \boldsymbol{\Psi}_t)$ satisfy*

$$\gamma_t\left(\tilde{\mu}\,\|\boldsymbol{x}_* - \boldsymbol{x}_{t-1}\|_1 + \tilde{\mu}_2\,\|\boldsymbol{x}_*\|_1\right) \leq \theta_t, \tag{16}$$

   then there is no false positive in each iteration. In other words, for all t, we have $supp(\boldsymbol{x}_t) \subseteq supp(\boldsymbol{x}_*)$.

2. *Assuming that the conditions of the last step hold, then we get the following bound on the error:*

$$\|\boldsymbol{x}_t - \boldsymbol{x}_*\|_1 \leq (\delta(\gamma_t) + \gamma_t\tilde{\mu}(|S| - 1))\,\|\boldsymbol{x}_{t-1} - \boldsymbol{x}_*\|_1 + \gamma_t\tilde{\mu}_2|S|\,\|\boldsymbol{x}_*\|_1 + |S|\theta_t.$$

B.2.1 Guidelines from Proposition.

We remark on some of the guidelines we can get from the above result.

- **Thresholds.** Similar to the discussion in previous sections, there are thresholds such that, there is no false positive at each layer. The choice of θ_t is a function of γ_t and, through coherence terms, $\mathbf{\Phi}_k$ and $\mathbf{\Psi}_t$. Since $\mathbf{\Phi}_k$ changes for each sample k, we learn a neural network that yields this parameter as a function of $\mathbf{\Phi}_k$ and $\mathbf{\Psi}_t$.
- **Step size.** The step size γ_t can be chosen to control the error decay. Ideally, we would like to have the term $(\delta(\gamma_t) + \gamma_t\tilde{\mu}(|S|-1))$ to be strictly smaller than one. In particular, γ_t directly impacts $\delta(\gamma_t)$, also a function of $\mathbf{\Phi}_k$ and $\mathbf{\Psi}_t$. We can therefore consider γ_t as a function of $\mathbf{\Phi}_k$ and $\mathbf{\Psi}_t$, which hints at the augmentation neural network we introduced for giving γ_t as a function of those parameters.

Remarks on Convergence. One might wonder if the convergence is possible given the bound on the error. We try to sketch a scenario where this can happen. First, note that once we have chosen γ_t, and given Φ_k and Ψ_t, we can select θ_t using the condition 16. Also, if the network gradually learns the ground truth dictionary at later stages, the term $\tilde{\mu}_2$ vanishes. We need to choose the term γ_t carefully such that the term $(\delta(\gamma_t) + \gamma_t\tilde{\mu}(|S|-1))$ is smaller than one. Similar to ISTA analysis, we would need to assume bounds on the mutual coherence $\tilde{\mu}$ and the column norm for $\mathbf{\Phi}_k\mathbf{\Psi}_o$. With standard assumptions, sketched above as well, the error gradually decreases per iteration, and we can reuse the convergence results of ISTA. We would like to emphasize that this is a heuristic argument, and there is no guarantee that the training yields a model with the parameters in accordance with these guidelines. Although we show experimentally that the proposed methods provide the promised improvements.

B.3 Proof of Proposition B.3

In what follows, we provide the derivations for Proposition B.3.

Convergence proofs of ISTA type models involve two steps in general. First, it is investigated how the support is found and locked in, and second how the error shrinks at each step. We focus on these two steps, which matter mainly for our architecture design. Our analysis is similar in nature to Chen et al. (2018); Aberdam et al. (2021), however it differs from Aberdam et al. (2021) in considering unknown dictionaries and from Chen et al. (2018) in both considered architecture and varying sensing matrix. In what follows, we consider noiseless setting. However, the results can be extended to noisy setups by adding additional terms containing noise norm similar to Chen et al. (2018). We make following assumptions:

1. There is a ground-truth (unknown) dictionary $\mathbf{\Psi}_o$ such that $\boldsymbol{s}_* = \mathbf{\Psi}_o\boldsymbol{x}_*$.
2. As a consequence, $\boldsymbol{y} = \mathbf{\Phi}_k\mathbf{\Psi}_o\boldsymbol{x}_*$.
3. We assume that $\boldsymbol{x}_*$ is sparse with its support contained in S. In other words: $x_{i,*} = 0$ for $i \notin S$.

To simplify the notation, we drop the index k, which indicates the varying sensing matrix, from $\mathbf{\Phi}_k$ and use $\mathbf{\Phi}$ for the rest. We break the proof to two lemma, each proving one part of Proposition B.3.

B.3.1 Proof - step 1: no false positive condition

The following lemma focuses on assuring that we do not have false positive in support recovery after each iteration of our model. In other words, the models continues updating only the entries in the support and keep the entries outside the support zero.

Lemma B.4. *Suppose that the support of $\boldsymbol{x}_*$ is given as supp$(\boldsymbol{x}_*) = S$. Consider iterations given by:*

$$\boldsymbol{x}_t = \eta_{\theta_t}\left(\boldsymbol{x}_{t-1} + \gamma_t(\mathbf{\Phi}\mathbf{\Psi}_t)^\top(\boldsymbol{y} - \mathbf{\Phi}\mathbf{\Psi}_t\boldsymbol{x}_{t-1})\right),$$

with $\boldsymbol{x}_0 = 0$. *If we have for all* $t = 1, 2, \ldots$:

$$\gamma_t \left(\tilde{\mu} \left\| \boldsymbol{x}_* - \boldsymbol{x}_{t-1} \right\|_1 + \tilde{\mu}_2 \left\| \boldsymbol{x}_* \right\|_1 \right) \leq \theta_t,$$

then there will be no false positive, i.e., $x_{t,i} = 0$ *for* $\forall i \notin S, \forall t$.

Proof. We prove this by induction. Since $\boldsymbol{x}_0 = 0$, the induction base is trivial. Suppose that the support of $\boldsymbol{x}_{t-1}$ is already included in that of $\boldsymbol{x}_*$, namely $\text{supp}(\boldsymbol{x}_{t-1}) \subseteq \text{supp}(\boldsymbol{x}_*) = S$. Consider $i \in S^c$. We have

$$x_{t,i} = \eta_{\theta_t} \left(\gamma_t ((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}_t \boldsymbol{x}_{t-1}) \right). \tag{17}$$

To avoid false positives, we need to guarantee that for $i \notin S$:

$$\eta_{\theta_t} \left(\gamma_t ((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}_t \boldsymbol{x}_{t-1}) \right) = 0 \implies \left| \gamma_t ((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}_t \boldsymbol{x}_{t-1}) \right| \leq \theta_t, \tag{18}$$

which means that the soft-thresholding function will have zero output for these entries. First note that:

$$\begin{aligned} \left| ((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top \boldsymbol{\Phi} (\boldsymbol{\Psi}_o \boldsymbol{x}_* - \boldsymbol{\Psi}_t \boldsymbol{x}_{t-1}) \right| &\leq \left| ((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top \boldsymbol{\Phi} (\boldsymbol{\Psi}_t \boldsymbol{x}_* - \boldsymbol{\Psi}_t \boldsymbol{x}_{t-1}) \right| \\ &\quad + \left| ((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top \boldsymbol{\Phi} (\boldsymbol{\Psi}_o \boldsymbol{x}_* - \boldsymbol{\Psi}_t \boldsymbol{x}_*) \right| \qquad (19) \\ &= \left| \sum_{j \in S} ((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top (\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_j (\boldsymbol{x}_{*,j} - \boldsymbol{x}_{t-1,j}) \right| + \left| ((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top \boldsymbol{\Phi} (\boldsymbol{\Psi}_o \boldsymbol{x}_* - \boldsymbol{\Psi}_t \boldsymbol{x}_*) \right| \qquad (20) \end{aligned}$$

We can bound the first term by:

$$\begin{aligned} \left| \sum_{j \in S} ((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top (\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_j (\boldsymbol{x}_{*,j} - \boldsymbol{x}_{t-1,j}) \right| &\leq \sum_{j \in S} \left| ((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top (\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_j \right| \left| (\boldsymbol{x}_{*,j} - \boldsymbol{x}_{t-1,j}) \right| \\ &\leq \tilde{\mu} \left\| \boldsymbol{x}_* - \boldsymbol{x}_{t-1} \right\|_1, \end{aligned}$$

where we use the definition of mutual coherence for the upper bound. The last term is bounded by

$$\begin{aligned} \left| ((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top \boldsymbol{\Phi} (\boldsymbol{\Psi}_o \boldsymbol{x}_* - \boldsymbol{\Psi}_t \boldsymbol{x}_*) \right| &= \left| \sum_{j \in S} ((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top (\boldsymbol{\Phi}(\boldsymbol{\Psi}_o - \boldsymbol{\Psi}_t))_j x_{j,*} \right| \qquad (21) \\ &\leq \sum_{j \in S} \left| ((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top (\boldsymbol{\Phi}(\boldsymbol{\Psi}_o - \boldsymbol{\Psi}_t))_j \right| |x_{j,*}| \qquad (22) \\ &\leq \tilde{\mu}_2 \left\| \boldsymbol{x}_* \right\|_1. \qquad (23) \end{aligned}$$

Therefore, we get

$$\left| \gamma_t ((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top (\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}_t \boldsymbol{x}_{t-1}) \right| \leq \gamma_t \left(\tilde{\mu} \left\| \boldsymbol{x}_* - \boldsymbol{x}_{t-1} \right\|_1 + \tilde{\mu}_2 \left\| \boldsymbol{x}_* \right\|_1 \right)$$

The following choice guarantees that there is no false positive:

$$\gamma_t \left(\tilde{\mu} \left\| \boldsymbol{x}_* - \boldsymbol{x}_{t-1} \right\|_1 + \tilde{\mu}_2 \left\| \boldsymbol{x}_* \right\|_1 \right) \leq \theta_t. \tag{24}$$

□

B.3.2 Proof - step 2: controlling the recovery error

The previous lemma provided the conditions such that there is no false positive. We see under which conditions the model can reduce the error inside the support S.

Lemma B.5. *Suppose that the threshold parameter* θ_t *has been chosen such that there is no false positive after each iteration. We have:*

$$\left\| \boldsymbol{x}_t - \boldsymbol{x}_* \right\|_1 \leq \left(\delta(\gamma_t) + \gamma_t \tilde{\mu} (|S| - 1) \right) \left\| \boldsymbol{x}_{t-1} - \boldsymbol{x}_* \right\|_1 + \gamma_t \tilde{\mu}_2 |S| \left\| \boldsymbol{x}_* \right\|_1 + |S| \theta_t.$$

Proof. For $i \in S$, we have:

$$|x_{t,i} - x_{*,i}| \leq \left|x_{t-1,i} + \gamma_t((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}_t\boldsymbol{x}_{t-1}) - x_{*,i}\right| + \theta_t. \tag{25}$$

At the iteration t for $i \in S$, we can separate the dictionary mismatch and the rest of the error as follows:

$$\begin{aligned} &x_{t-1,i} + \gamma_t((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}_t\boldsymbol{x}_{t-1}) = \\ &\quad x_{t-1,i} + \gamma_t(\sum_{j\in S}((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top(\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_j(\boldsymbol{x}_{*,j} - \boldsymbol{x}_{t-1,j}) + ((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top\boldsymbol{\Phi}(\boldsymbol{\Psi}_o\boldsymbol{x}_* - \boldsymbol{\Psi}_t\boldsymbol{x}_*)). \end{aligned}$$

We can decompose the first part further as:

$$\begin{aligned} &x_{t-1,i} + \gamma_t \sum_{j\in S}((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top(\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_j(x_{*,j} - \boldsymbol{x}_{t-1,j}) = \\ &\qquad (\boldsymbol{I} - \gamma_t(\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top(\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i))\boldsymbol{x}_{t-1,i} + \gamma_t(\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top(\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)\boldsymbol{x}_{*,i} \\ &\qquad + \gamma_t \sum_{j\in S, j\neq i}((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top(\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_j(\boldsymbol{x}_{*,j} - \boldsymbol{x}_{t-1,j}). \end{aligned}$$

Using triangle inequality for the previous decomposition we get:

$$\begin{aligned} \left|x_{t-1,i} + \gamma_t((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top(\boldsymbol{y} - \boldsymbol{\Phi}\boldsymbol{\Psi}_t\boldsymbol{x}_{t-1}) - x_{*,i}\right| &\leq \left|(1 - \gamma_t(\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top(\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i))(x_{t-1,i} - x_{*,i})\right| \\ &\quad + \gamma_t\left|\sum_{j\in S, j\neq i}((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top(\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_j(\boldsymbol{x}_{*,j} - \boldsymbol{x}_{t-1,j})\right| \\ &\quad + \gamma_t\left|((\boldsymbol{\Phi}\boldsymbol{\Psi}_t)_i)^\top\boldsymbol{\Phi}(\boldsymbol{\Psi}_o\boldsymbol{x}_* - \boldsymbol{\Psi}_t\boldsymbol{x}_*))\right| \\ &\quad \leq \delta(\gamma_t)\left|(z_{t-1,i} - z_{*,i})\right| \\ &\quad + \gamma_t\sum_{j\in S, j\neq i}\tilde{\mu}\left|x_{*,j} - x_{t-1,j}\right| + \gamma_t\tilde{\mu}_2\left\|\boldsymbol{x}_*\right\|_1 \end{aligned}$$

It suffices to sum up the errors and combine previous inequalities to get:

$$\begin{aligned} \left\|\boldsymbol{x}_{S,t} - \boldsymbol{x}_*\right\|_1 &= \sum_{i\in S}\left|\boldsymbol{x}_{t,i} - \boldsymbol{x}_{*,i}\right| \leq \\ &\leq (\delta(\gamma_t) + \gamma_t\tilde{\mu}(|S| - 1))\left\|\boldsymbol{x}_{S,t-1} - \boldsymbol{x}_*\right\|_1 + \gamma_t\tilde{\mu}_2|S|\left\|\boldsymbol{x}_*\right\|_1 + |S|\theta_t. \end{aligned}$$

Since we assumed there is no false positive, we get the final result:

$$\left\|\boldsymbol{x}_t - \boldsymbol{x}_*\right\|_1 = \sum_{i\in S}\left|\boldsymbol{x}_{t,i} - \boldsymbol{x}_{*,i}\right| \leq (\delta(\gamma_t) + \gamma_t\tilde{\mu}(|S| - 1))\left\|\boldsymbol{x}_{t-1} - \boldsymbol{x}_*\right\|_1 + \gamma_t\tilde{\mu}_2|S|\left\|\boldsymbol{x}_*\right\|_1 + |S|\theta_t.$$

□

C Implementation Details

In this section we report details concerning the architecture of A-DLISTA and VLISTA.

C.1 A-DLISTA (Augmentation Network)

As already reported in the main paper (subsection 4.1), A-DLISTA consists of two architectures: the DLISTA model (blue blocks in Figure 1) representing the unfolded version of the ISTA algorithm with parametrized $\boldsymbol{\Psi}$, and the augmentation (or adaptation) network (red blocks in Figure 1). The augmentation model takes as input the measurement matrix, $\boldsymbol{\Phi}^i$, and the dictionary at a given reconstruction layer t, $\boldsymbol{\Psi}_t$, and generates the parameters $\{\gamma_t, \theta_t\}$ for the current iteration. We show the architecture for the augmentation network in Figure 5. According to the figure, the model is characterized by a features extraction section and two output branches, one for each parameter to be generated. In order to ensure that the estimated $\{\gamma_t, \theta_t\}$ parameters are positive, the architecture provides each branch with a softplus function. As mentioned in the main paper, the weights of the augmentation model are shared across all the A-DLISTA layers.

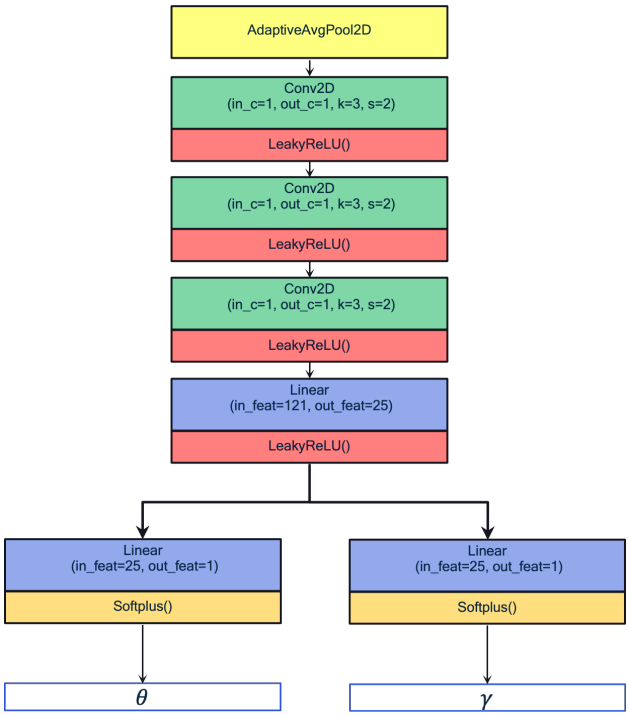

Figure 5: Augmentation model's architecture for A-DLISTA.

C.2 VLISTA

As described in subsection 4.2 of the meain paper, VLISTA comprises three different components: the likelihood, and the prior and posterior models.

C.2.1 VLISTA - Likelihood model

The likelihood model (subsection 4.2) represents a Gaussian distribution with mean value parametrized by using the A-DLISTA model. There is, however, a fundamental difference among the likelihood model and the A-DLISTA architecture presented in subsection 4.1. Indeed, differently from the latter, the likelihood model of VLISTA does not learn the dictionary using backpropagation. Rather, it uses the dictionary sampled from the posterior distribution.

C.2.2 VLISTA - Posterior & Prior models

We report in Figure 6 the prior (left image) and the posterior (right image) architectures. We implement both models using an encoder-decoder scheme based on convolutional layers. The prior network is made by two convolutional layers followed by two separate branches dedicated to generate the mean and variance of the Gaussian distribution subsection 4.2. As input for the prior, we use the dictionary sampled at the previous iteration. In contrast to the prior, the posterior network accepts three different quantities as input: the sensing matrix, the observations, and the reconstructed sparse vector from the previous iteration. To process the three inputs together, the posterior accounts for three separated "input" layers followed by an aggregation step. Subsequently, two branches are used to generate the mean and the standard deviation of the Gaussian distribution of the dictionary subsection 4.2.
We offer the reader with a unified overview of our variational model in Figure 7.

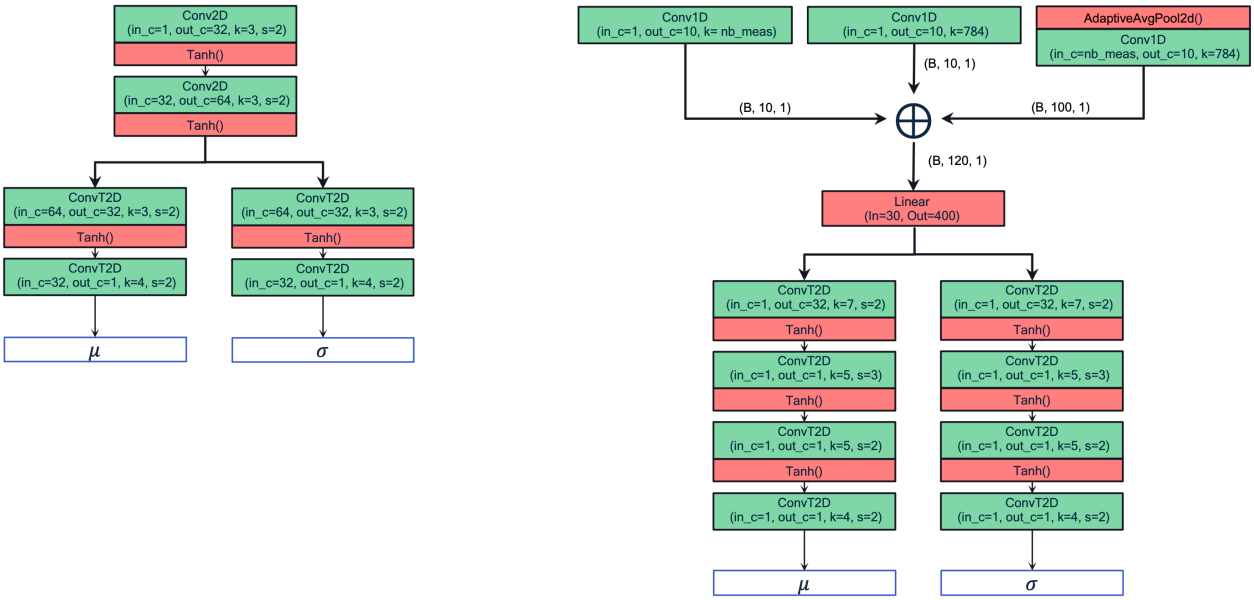

Figure 6: Left: prior network architecture. Right: posterior network architecture. For the posterior model we show in the figure the output shape from each of the three input heads. Such a structure is necessary since the posterior model accepts three quantities as input, namely, observations, the sensing matrix, and the reconstruction from the previous layer. These quantities are characterized by different shapes. The letter "B" indicates batch size.

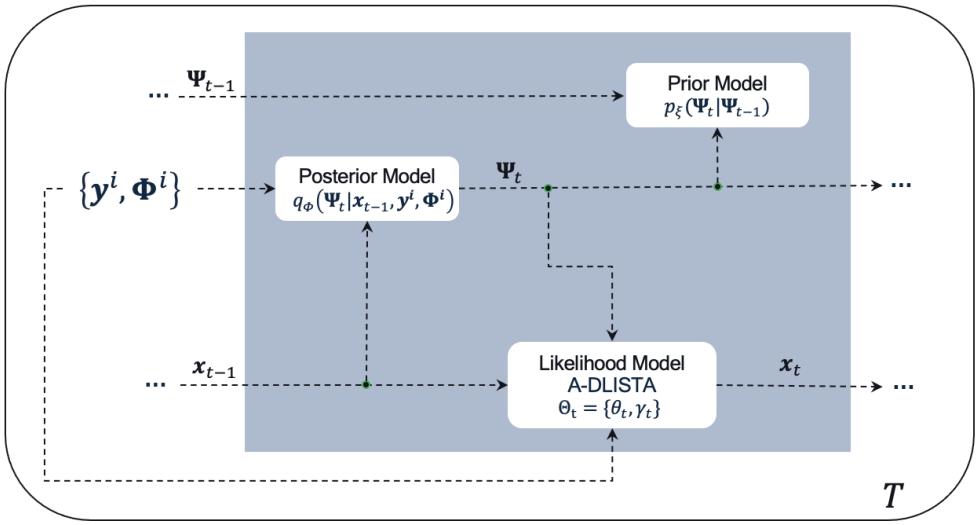

Figure 7: VLISTA iterations' schematic view.

D Training Details

The purpose of this section is to provide details regarding the training of the A-DLISTA and VLISTA models. Using the Adam optimizer, we train the reconstruction and augmentation models for A-DLISTA jointly. We set the initial learning rate to $1.e^{-2}$ and $1.e^{-3}$ for the reconstruction and augmentation network, respectively, and we drop its value by a factor 10 every time the loss stops improving. Additionally, we set the weight decay to $5.e^{-4}$ and the batch size to 128. For all datasets, we use Mean Squared Error (MSE) as the objective function. We train all the components of VLISTA using the Adam optimizer, similar to A-DLISTA. We set the learning rate to $1.e^{-3}$ and drop its value by a factor 10 every time the loss stops improving. Regarding the objective function, we maximize the ELBO and set the weight for KL divergence to $1.e^{-3}$. We report in Equation 26 the ELBO derivation.

$$
\begin{aligned}
\log p(\boldsymbol{x}_{1:T} = \boldsymbol{x}^i_{gt} | \boldsymbol{y}^i, \boldsymbol{\Phi}^i) &= \log \int p(\boldsymbol{x}_{1:T} = \boldsymbol{x}^i_{gt} | \boldsymbol{\Psi}_{1:T}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i) p(\boldsymbol{\Psi}_{1:T}) d\boldsymbol{\Psi}_{1:T} \\
&= \log \int \frac{p(\boldsymbol{x}_{1:T} = \boldsymbol{x}^i_{gt} | \boldsymbol{\Psi}_{1:T}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i) p(\boldsymbol{\Psi}_{1:T}) q(\boldsymbol{\Psi}_{1:T} | \boldsymbol{x}_{1:T}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i)}{q(\boldsymbol{\Psi}_{1:T} | \boldsymbol{x}_{1:T}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i)} d\boldsymbol{\Psi}_{1:T} \\
&\geq \int q(\boldsymbol{\Psi}_{1:T} | \boldsymbol{x}_{1:T}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i) \log \frac{p(\boldsymbol{x}_{1:T} = \boldsymbol{x}^i_{gt} | \boldsymbol{\Psi}_{1:T}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i) p(\boldsymbol{\Psi}_{1:T})}{q(\boldsymbol{\Psi}_{1:T} | \boldsymbol{x}_{1:T}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i)} d\boldsymbol{\Psi}_{1:T} \\
&= \int q(\boldsymbol{\Psi}_{1:T} | \boldsymbol{x}_{1:T}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i) \log p(\boldsymbol{x}_{1:T} = \boldsymbol{x}^i_{gt} | \boldsymbol{\Psi}_{1:T}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i) d\boldsymbol{\Psi}_{1:T} \\
&+ \int q(\boldsymbol{\Psi}_{1:T} | \boldsymbol{x}_{1:T}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i) \log \frac{p(\boldsymbol{\Psi}_{1:T})}{q(\boldsymbol{\Psi}_{1:T} | \boldsymbol{x}_{1:T}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i)} d\boldsymbol{\Psi}_{1:T} \\
&= \sum_{t=1}^{T} \mathbb{E}_{\boldsymbol{\Psi}_{1:t} \sim q(\boldsymbol{\Psi}_{1:t} | \boldsymbol{x}_{0:t-1}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i)} \Big[\log p(\boldsymbol{x}_t = \boldsymbol{x}^i_{gt} | \boldsymbol{\Psi}_{1:t}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i) \Big] \\
&- \sum_{t=2}^{T} \mathbb{E}_{\boldsymbol{\Psi}_{1:t-1} \sim q(\boldsymbol{\Psi}_{1:t-1} | \boldsymbol{x}_{t-1}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i)} \Big[D_{KL} \Big(q(\boldsymbol{\Psi}_t | \boldsymbol{x}_{t-1}, \boldsymbol{y}^i, \boldsymbol{\Phi}^i) \parallel p(\boldsymbol{\Psi}_t | \boldsymbol{\Psi}_{t-1}) \Big) \Big] \\
&- D_{KL} \Big(q(\boldsymbol{\Psi}_1 | \boldsymbol{x}_0) \parallel p(\boldsymbol{\Psi}_1) \Big)
\end{aligned}
\tag{26}
$$

Note that in Equation 26, we consider the same ground truth, $\boldsymbol{x}^i_{gt}$, for each iteration $t \in [1, T]$.

E Additional Results

In this section we report additional experimental results. In subsection E.1 we report results concerning a fix measurement setup, i.e., $\boldsymbol{\Phi}^i \rightarrow \boldsymbol{\Phi}$. Finally, we report ablation studies concerning classical baselines in subsection E.2.

E.1 Fixed Sensing Matrix

We provide in Table 3 and Table 4 results considering a fixed measurement scenario, i.e. using a single sensing matrix $\boldsymbol{\Phi}$. Comparing these results to Table 1 and Table 2, we notice the following. To begin with, LISTA and A-DLISTA perform better compared to the setup in which we use a varying sensing matrix (see section 5). We should expect such a behavior given that we simplified the problem by fixing the $\boldsymbol{\Phi}$ matrix. Additionally, as we mentioned in the main paper, ALISTA and NALISTA exhibit high performances (superior to other models when 300 and 500 measurements are considered). Such a result is expected given that these two models were designed for solving inverse problems in a fixed measurement scenario. Furthermore, the results in Table 3 and Table 4 support our hypothesis that the convergence issues we observe in the varying sensing matrix setup are likely related to the "inner" optimization that ALISTA and NALISTA require to evaluate the "W" matrix.

Table 3: MNIST SSIM (the higher the better) for a different number of measurements with **fixed sensing matrix**, i.e., $\mathbf{\Phi}^i \to \mathbf{\Phi}$. We highlight in bold the best performance. Note that whenever there is agreement within the error for the best performances, we highlight all of them.

	SSIM ↑				
	number of measurements				
	1 ($\times e^{-1}$)	10 ($\times e^{-1}$)	100 ($\times e^{-1}$)	300 ($\times e^{-1}$)	500 ($\times e^{-1}$)
LISTA	$\mathbf{1.34_{\pm 0.02}}$	$3.12_{\pm 0.02}$	$\mathbf{5.98_{\pm 0.01}}$	$6.74_{\pm 0.01}$	$6.96_{\pm 0.01}$
ALISTA	$0.84_{\pm 0.01}$	$0.94_{\pm 0.01}$	$1.70_{\pm 0.01}$	$5.71_{\pm 0.01}$	$6.65_{\pm 0.01}$
NALISTA	$0.91_{\pm 0.01}$	$1.12_{\pm 0.01}$	$2.46_{\pm 0.01}$	$\mathbf{7.03_{\pm 0.01}}$	$\mathbf{8.22_{\pm 0.02}}$
A-DLISTA (our)	$1.21_{\pm 0.02}$	$\mathbf{3.58_{\pm 0.01}}$	$5.66_{\pm 0.01}$	$6.47_{\pm 0.01}$	$6.84_{\pm 0.01}$

Table 4: CIFAR10 SSIM (the higher the better) for a different number of measurements with **fixed sensing matrix**, i.e., $\mathbf{\Phi}^i \to \mathbf{\Phi}$. Note that whenever there is agreement within the error for the best performances, we highlight all of them.

	SSIM ↑				
	number of measurements				
	1 ($\times e^{-1}$)	10 ($\times e^{-1}$)	100 ($\times e^{-1}$)	300 ($\times e^{-1}$)	500 ($\times e^{-1}$)
LISTA	$2.52_{\pm 0.01}$	$\mathbf{3.19_{\pm 0.01}}$	$\mathbf{4.48_{\pm 0.01}}$	$\mathbf{6.29_{\pm 0.01}}$	$6.74_{\pm 0.01}$
ALISTA	$0.21_{\pm 0.03}$	$0.54_{\pm 0.02}$	$0.88_{\pm 0.01}$	$3.54_{\pm 0.01}$	$5.52_{\pm 0.01}$
NALISTA	$1.32_{\pm 0.02}$	$1.32_{\pm 0.02}$	$1.06_{\pm 0.02}$	$4.59_{\pm 0.01}$	$\mathbf{6.88_{\pm 0.01}}$
A-DLISTA (our)	$\mathbf{2.91_{\pm 0.02}}$	$3.07_{\pm 0.01}$	$4.26_{\pm 0.01}$	$5.89_{\pm 0.01}$	$6.56_{\pm 0.01}$

E.2 Classical baselines

The purpose of this section is to report additional results concerning classical dictionary learning methods tested on the MNIST and CIFAR10 datasets. Specifically, we find interesting to show that classical baselines can reach very high performance if we assume to have neither any computational nor time constraints (although this would correspond to an unrealistic scenario concerning real-world applications). Therefore, while tuning hyperparameters, we consider a number of layers (or iterations) up to a few thousands. In Figure 8 and Figure 9, we report results in terms of SSIM on the MNIST and CIFAR10 datasets concerning the following baselines: Canonical, Wavelet, and SPCA. Moreover, whenever in the figures we indicate the number of layers as "≥ 500", it means that we let the models use as many iterations as needed (specifically, more than 500) to reach the best reconstruction possible. In Figure 8, we focus on comparing different baselines for the same number of layers, while in Figure 9, we show the improvement for each algorithm as we increase the number of iterations.

Finally, from Figure 10 to Figure 15, we report some example of reconstructed images for different baselines. Note that to reconstruct images, we consider the best overall hyperparameters configuration for each model. Moreover, we use 500 measurements for each image.

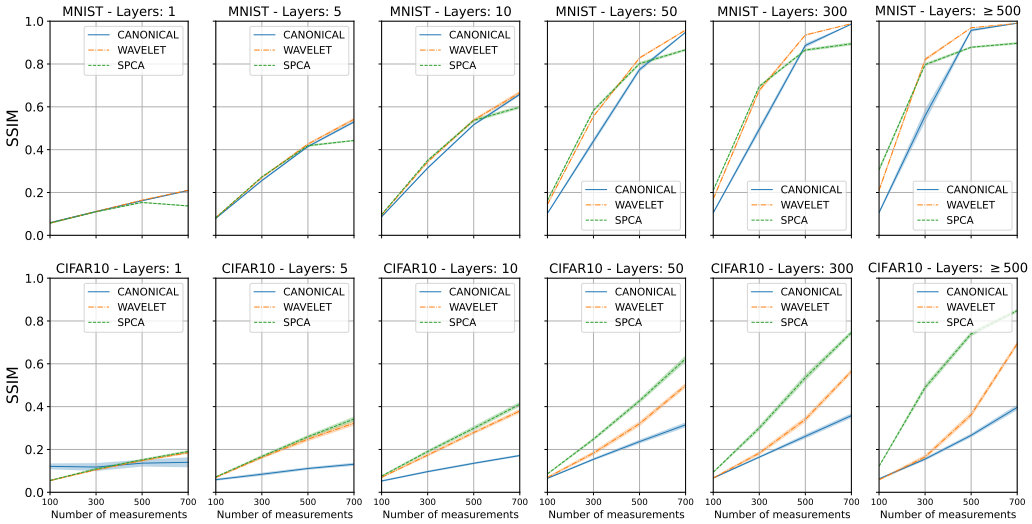

Figure 8: SSIM mean value (solild lines) and its uncertainty (shadow ares) for a different number of measurements. Top row: MNIST dataset. Bottom row: CIFAR10 dataset. Each column corresponds to a different number of layers used for each baseline.

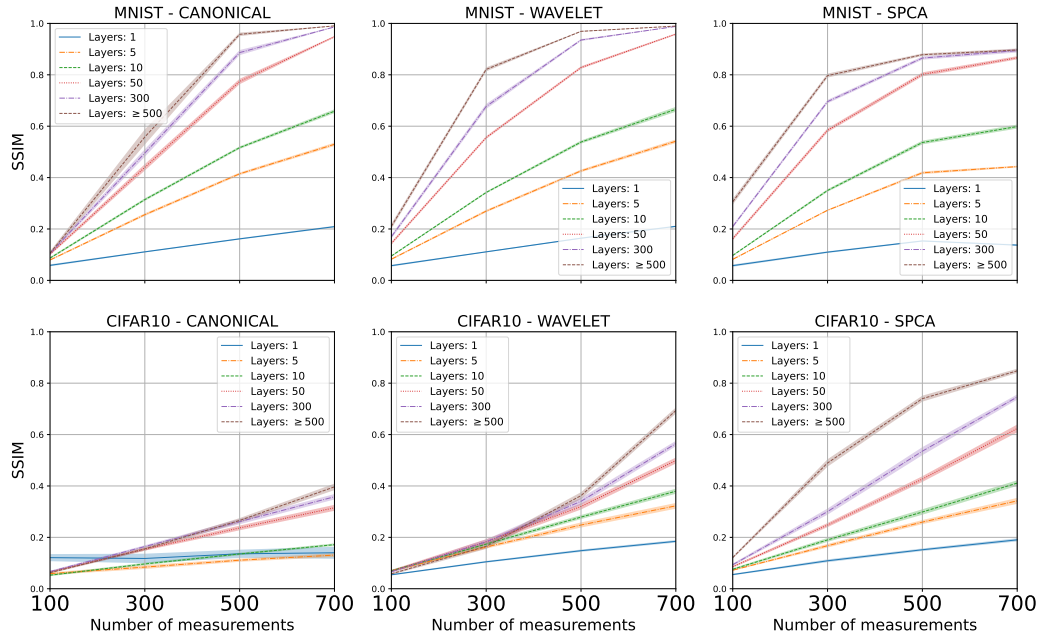

Figure 9: SSIM mean value (solild lines) and its uncertainty (shadow ares) for a different number of measurements. Top row: MNIST dataset. Bottom row: CIFAR10 dataset. Each column corresponds to a different baseline.

Figure 10: Example of reconstructed MNIST images using the canonical basis. Top row: reconstructed images. Bottom row: ground truth images. To reconstruct images we use 500 measurements and the number of layers optimized to get the best reconstruction possible.

Figure 11: Example of reconstructed MNIST images using the wavelet basis. Top row: reconstructed images. Bottom row: ground truth images. To reconstruct images we use 500 measurements and the number of layers optimized to get the best reconstruction possible.

Figure 12: Example of reconstructed MNIST images using SPCA. Top row: reconstructed images. Bottom row: ground truth images. To reconstruct images we use 500 measurements and the number of layers optimized to get the best reconstruction possible.

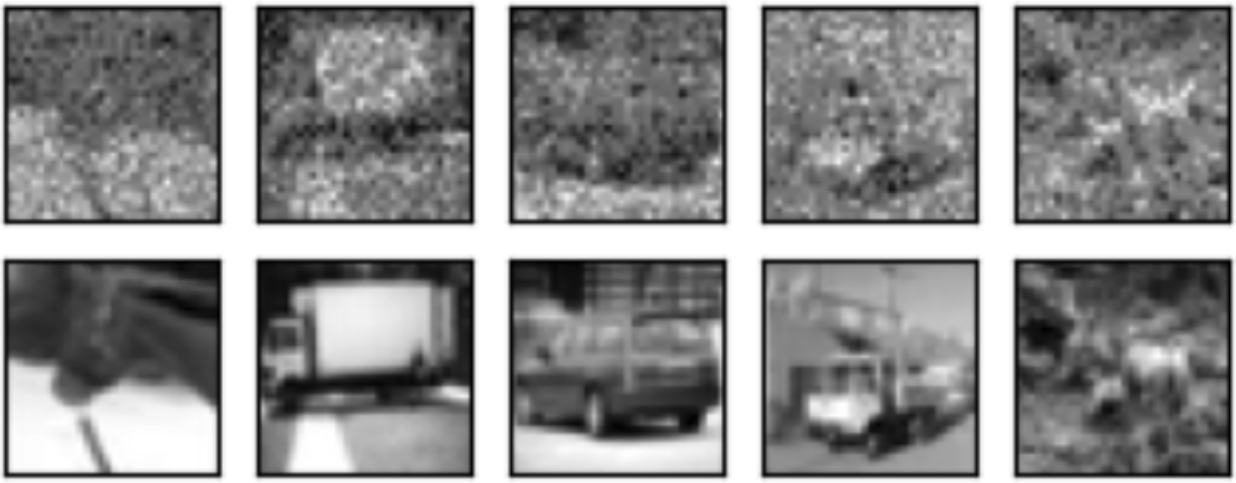

Figure 13: Example of reconstructed CIFAR10 images using the canonical basis. Top row: reconstructed images. Bottom row: ground truth images. To reconstruct images we use 500 measurements and the number of layers optimized to get the best reconstruction possible.

Figure 14: Example of reconstructed CIFAR10 images using the wavelet basis. Top row: reconstructed images. Bottom row: ground truth images. To reconstruct images we use 500 measurements and the number of layers optimized to get the best reconstruction possible.

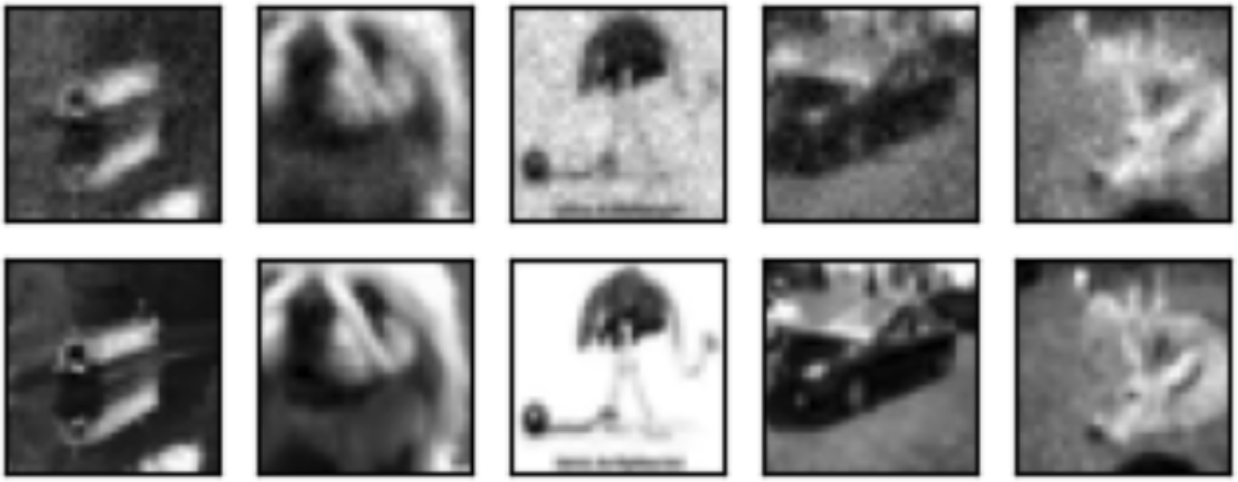

Figure 15: Example of reconstructed CIFAR10 images using SPCA. Top row: reconstructed images. Bottom row: ground truth images. To reconstruct images we use 500 measurements and the number of layers optimized to get the best reconstruction possible.