A Statistical Framework to Investigate the Optimality of Neural Networks for Inverse Problems

Pakshal Bohra, Pol del Aguila Pla, Member, IEEE, Jean-François Giovannelli, and Michael Unser, Fellow, IEEE

Abstract—We present a statistical framework to benchmark the performance of neural-network-based reconstruction algorithms for linear inverse problems. We generate synthetic signals as realizations of sparse stochastic processes, which makes them ideally matched to variational sparsity-promoting techniques. We derive Gibbs sampling schemes to compute the minimum mean-square error estimators for processes with Laplace, Student’s $t$, and Bernoulli-Laplace innovations. These allow our framework to provide quantitative measures of the degree of optimality (in the mean-square-error sense) for any given reconstruction method. We showcase our framework by benchmarking the performance of convolutional neural-network architectures in the context of deconvolution and Fourier sampling. Our experimental results suggest that, while these architectures achieve near-optimal results in many settings, their performance deteriorates severely for signals associated with heavy-tailed distributions.

Index Terms—Inverse problems, CNNs, minimum mean-square error, sparse stochastic processes.

I. INTRODUCTION

Inverse problems are often encountered in biomedical imaging [1], particularly in modalities such as computed tomography (CT), magnetic resonance imaging (MRI), or deconvolution microscopy. Their goal is to reconstruct an unknown signal from its measurements. Often, these are hard to solve due to their ill-posedness, which implies that the underlying signal cannot be determined uniquely by the acquired measurements, unless one introduces some form of regularization. Therefore, prior knowledge about the signal of interest is required for the resolution of such problems.

A. Model-Based Methods

Model-based methods rely on the mathematical modeling of the signal of interest to counteract the ill-posedness of the inverse problem. We organize them in two categories.

The first category is composed of linear reconstruction methods (e.g., filtered back-projection), which are fast, well understood, and come with performance and stability guarantees [2], [3]. From a variational standpoint, they can be interpreted as minimizers of a cost functional that consists of a quadratic data term to ensure consistency with the measurements, along with an additive quadratic (Tikhonov) regularization term that imposes some smoothness on the solution. Interestingly, these methods can also be derived from a statistical perspective as optimal linear reconstructors under the Gaussian hypothesis [4].

The second category is composed of methods that exploit sparsity—the property that a signal admits a concise representation in some transform domain (e.g., wavelets) [5]–[8]. This powerful concept supports the theory of compressed sensing, which gives conditions under which the reconstruction of an image from a limited set of measurements is feasible [9]–[11]. To obtain a sparse reconstruction, one typically uses $\ell_1$-norm regularization and solves the corresponding convex optimization problem using iterative algorithms such as the fast iterative shrinkage-thresholding algorithm (FISTA) [12] or the alternating direction method of multipliers (ADMM) [13]. In practice, sparsity-promoting regularizers such as total variation (TV) [14] generally improve the quality of the image.

B. Learning-Based Methods

Neural-network-based methods that make use of prior information learned from a large collection of training data are now the focus of much of the current research in image reconstruction [15], [16]. They shine in extreme imaging scenarios where one wishes to achieve more with fewer data, for instance when operating with short integration times, which leads to an abundance of noise, or when collecting fewer measurements to reduce either the acquisition duration and/or the radiation exposure [17]. Here, we focus on two classes of neural-network-based methods and classify them as the counterparts of the model-based ones.

The first successful applications of deep convolutional neural networks (CNNs) in imaging build upon the classical linear-reconstruction algorithms, training a CNN to correct for reconstruction artifacts in extreme imaging conditions [17]–[21]. Unrolling methods [22]–[27] also fall into this class of direct, nonlinear reconstructions. Examples of successful applications include MRI, CT, optical imaging, and ultrasound. Their gain over the state-of-the-art is impressive and comparable in magnitude to the one afforded by a decade of refinement of the sparsity-promoting techniques.

The second class includes methods that attempt to reconstruct an image that is consistent with the measurements by replacing the proximal operator that is typically involved in the iterative sparsity-promoting methods by an appropriate denoising CNN, which then plays the role of the regularizer.
They come in a variety of flavors, including plug-and-play (PnP) \cite{28}–\cite{31}, regularization-by-denoising (RED) \cite{32}–\cite{34}, and projected-gradient-descent \cite{35}, \cite{36} methods. Despite their remarkable performance, CNN-based imaging methods have limitations that currently hinder their further development. Unlike the model-based methods, which are backed by sound mathematics, the development of CNN-based approaches is empirical. Expressivity is obtained through the composition of simple units, but the working of the whole is hard to comprehend and the architectural options are overwhelming (e.g., depth, number of channels, size of the filters). In practice, one usually proceeds by trial and error using the training, validation, and testing errors as quantitative criteria. Further, the training of CNNs is poorly understood and often difficult because of the underlying over-parameterization: getting a stochastic optimization algorithm to perform properly for a specific application typically requires a lot of adjustments and experimentation.

Beside the strain that this empirical approach exerts on developers, the performance greatly depends on the quality, cardinality, and representability of the training dataset, while the outcome is not necessarily transposable to other applications. The bottleneck with biomedical imaging is often a limited access to large, representative datasets. This is mostly because of legal issues in medical imaging and because of the lack of standardized protocols in biomicroscopy. Another issue is the chicken-and-egg nature of the training process because the desired image (the physical object that corresponds to the measurements) is not known precisely—in practice, the goldstandard is an image produced by a state-of-the-art model-based method with high-density/low-noise measurements. This is adequate for developing methods for compressed sensing, but not otherwise. This explains why the works that demonstrate the superiority of the CNN-based approaches over the more traditional model-based methods for image reconstruction have used limited benchmarks so far.

C. Contribution

The purpose of this paper is to provide an objective benchmarking environment that can assist in the design of neural-network architectures for robust signal reconstruction in linear inverse problems. In addition to training data, our proposed framework offers quantitative measures of the degree of optimality (in the mean-square-error sense) of CNN-based approaches.

We synthesize ground-truth signals and then simulate the measurement process (e.g., convolution for deconvolution microscopy, Fourier sampling for MRI) in the presence of noise. Specifically, we consider a statistical framework where the underlying signals are realizations of 1D sparse stochastic processes (SSPs) \cite{37}. The motivation there is that these processes are ideally matched to model-based methods, the most prominent of which can be interpreted as their maximum a posteriori (MAP) estimators \cite{38}. This makes our benchmark all the more challenging for CNNs. It offers a good ground for the tuning of CNN architectures and for the demonstration of their superiority in a tightly controlled environment. We can produce any desired number of training pairs for a given reconstruction task and some chosen stochastic signal model, which allows for an informed comparison of network architectures. Further, since the true statistical distribution of the signal is known exactly in our framework, the minimum-mean-square-error (MMSE) estimator is indeed optimal in the mean-square-error (MSE) sense. Therefore, we are able to provide statistical guarantees of optimality by specifying an upper limit on the reconstruction performance.

The MAP estimates of SSPs are solutions of optimization problems that resemble the ones used in model-based methods, and can be computed efficiently. However, it has been observed that these MAP estimators are suboptimal in the MSE sense \cite{38}, \cite{59}, except in the Gaussian scenario where the MAP and MMSE estimators (generalized Wiener filter) coincide \cite{4}. In this work, we focus on non-Gaussian signal models. In principle, the MMSE estimator involves the calculation of high-dimensional integrals, which are not numerically tractable in general. Thus, we develop efficient Gibbs-sampling-based algorithms to compute the MMSE estimator for specific classes of SSPs, with innovations following the Laplace, Student’s t, and Bernoulli-Laplace distributions.

To the best of our knowledge, no such working solution for generic linear inverse problems with SSPs has been presented in the literature.

Finally, we present experimental results that illustrate the usefulness of our framework. Specifically, we benchmark the performance of CNNs that perform direct nonlinear reconstructions, in the context of deconvolution and Fourier sampling for first-order SSPs. On one hand, when the innovations follow a Bernoulli-Laplace distribution, we observe that the CNNs achieve near-optimal mean-square-error performance and outperform the sparsity-promoting methods, even though they are well-suited to these piecewise-constant signals. On the other hand, our experiments with Student’s t innovations indicate regimes where CNNs fail to reconstruct the signals well. More specifically, we observe that, when the tails of the Student’s t distribution are made heavier (i.e., when we move towards a Cauchy distribution), CNNs perform rather poorly.

D. Roadmap

In Section \[II\] we describe a continuous-domain model for the measurement process along with a way to discretize it. In Section \[III\] we introduce Lévy processes as stochastic models for our signals and we derive the probability distribution for samples of such processes. We then discuss MAP and MMSE estimation in Section \[IV\] before we develop Gibbs samplers for Lévy processes associated with Laplace, Student’s t, and Bernoulli-Laplace distributions in Section \[V\]. Finally, we present experimental results in Section \[VI\].

II. MEASUREMENT MODEL

In the proposed framework, we consider the recovery of a continuous-domain signal \(s : \mathbb{R} \rightarrow \mathbb{R}\) from a finite number \(M\) of measurements \(y = (y_m)_{m=1}^M\).
A. Continuous-Domain Measurement Model

We model the measurements $y = (y_m)_{m=1}^M$ as

$$y_m = \int_{\mathbb{R}} s(t) \nu_m(t) \, dt + n[m], \quad (1)$$

where $(\nu_m)_{m=1}^M$ are linear functionals that describe the physics of the acquisition process and $n[.]$ is an additive white Gaussian noise (AWGN) with variance $\sigma_n^2$. By choosing appropriate functionals $(\nu_m)_{m=1}^M$, we can study a variety of linear inverse problems such as denoising, deconvolution, inpainting, and Fourier sampling.

B. Discrete Measurement Model

We need to discretize (1) to obtain a computationally feasible model for the measurements. To that end, we consider a finite region of interest $\Omega = (0, T)$ of the signal and approximate it with

$$s_h(t) = \sum_{k=1}^K s(kh) \text{sinc} \left( \frac{t}{h} - k \right), \quad (2)$$

where $h$ is the sampling step and $K = \left\lceil \frac{T}{h} \right\rceil - 1$. When $h$ is small enough, $s_h$ is a good approximation of $s$ within the interval $\Omega$. On introducing (2) into (1), we get that

$$y = Hs + n, \quad (3)$$

where $s = (s(kh))_{k=1}^K \in \mathbb{R}^K$ contains equidistant samples of the signal, $H : \mathbb{R}^K \rightarrow \mathbb{R}^M$ is the discrete system matrix with

$$[H]_{m,k} = \int_{\mathbb{R}} \text{sinc} \left( \frac{t}{h} - k \right) \nu_m(t) \, dt, \quad (4)$$

and $n \in \mathbb{R}^M$ is the noise.

Thus, for any signal samples $s \in \mathbb{R}^K$, we can simulate noisy measurements using (3). Next, we derive the discrete system matrices for deconvolution and Fourier sampling. Hereafter, we assume for simplicity that $h = 1$.

C. Deconvolution

In deconvolution, the measurements are acquired by sampling the result of the convolution between the signal and the point-spread function (PSF) $\psi$ of the acquisition system, which we model by letting the measurement functionals be $\nu_m = \psi(m - \cdot)$. We assume that the cutoff frequency of $\psi$ is $\omega_0 \leq \pi$, as this allows us to sample $(s * \psi)$ on an integer grid without aliasing effects. In this case, The entries of the resulting system matrix $H$ are given by

$$[H]_{m,k} = \int_{\mathbb{R}} \text{sinc}(t-k)\psi(m-t) \, dt = \psi(m-k). \quad (5)$$

Here, $H$ is a discrete convolution matrix whose entries are samples of the bandlimited PSF $\psi$.

D. Fourier Sampling

In Fourier sampling, the measurements are acquired by sampling the Fourier transform of the signal at arbitrary frequencies $\{\omega_m\}_{m=1}^M$. Accordingly, the measurement functionals are the complex exponentials $\nu_m = e^{-j\omega_m \cdot}$. Assuming that $|\omega_m| \leq \pi$, we get that

$$[H]_{m,k} = \int_{\mathbb{R}} \text{sinc}(t-k)e^{-j\omega_m t} \, dt = e^{-j\omega_mk}. \quad (6)$$

Here, $H$ is a discrete Fourier-like matrix, except that the frequencies $\omega_m$ do not necessarily lie on an uniform grid.

III. Stochastic Signal Model

In this section, we describe a continuous-domain stochastic model for the signal. We also derive the probability distribution for the discrete signal vector $s = (s(k))_{k=1}^K$.

A. Lévy Processes

In our framework, the underlying signals are realizations of a well-known class of first-order sparse stochastic processes: the Lévy processes [37, 41].

**Definition 1** (Lévy process). A stochastic process $s = \{s(t) : t \in \mathbb{R}^+\}$ is a Lévy process if

1) $s(0) = 0$ almost surely;
2) (independent increments) for any $N \in \mathbb{N} \setminus \{0,1\}$ and $0 \leq t_1 < t_2 \cdots < t_N < \infty$, the increments $(s(t_2) - s(t_1)), (s(t_3) - s(t_2)), \ldots, (s(t_N) - s(t_{N-1}))$ are mutually independent;
3) (stationary increments) for any given step $h$, the increment process $u_h = \{s(t) - s(t-h) : t \in \mathbb{R}^+\}$ is stationary;
4) (stochastic continuity) for any $\epsilon > 0$ and $t \geq 0$

$$\lim_{h \to 0} \Pr\{|s(t+h) - s(t)| > \epsilon\} = 0.$$

Lévy processes are closely linked to infinitely divisible (id) distributions.

**Definition 2** (Infinite divisibility). A random variable $X$ is infinitely divisible if, for any $N \in \mathbb{N} \setminus \{0\}$, there exist independent and identically distributed (i.i.d.) random variables $X_1, \ldots, X_N$ such that $X = X_1 + \cdots + X_N$.

For any Lévy process $s$, the random variable $s(t)$ for some $t > 0$ is infinitely divisible. Moreover, its probability density function (pdf) is given by

$$p_{s(t)}(x) = \int_{\mathbb{R}} \left( \int_{\mathbb{R}} p_s(1)(y)e^{iyx} \, dy \right) \frac{e^{-t|\omega|^2}}{2\pi}. \quad (7)$$

Conversely, for any id distribution with pdf $p_{id}$, it is possible to construct a Lévy process $s$ such that $p_s(1) = p_{id}$. Thus, there is a one-to-one correspondence between Lévy processes and id distributions [41].

Among all id distributions, the pdf of the Gaussian distribution exhibits the fastest rate of decay at infinity. In this sense, we refer to the non-Gaussian, heavier-tailed members (e.g., Laplace, Bernoulli-Laplace, Student’s t, symmetric-alpha-stable) of the class of id distributions as sparse [42]. Indeed, some of these sparse distributions have a mass at the origin in their probability distribution (e.g., Bernoulli-Laplace)}


and some of them are strongly compressible (e.g., Student’s t, symmetric-alpha-stable) \cite{43}. The stochastic model of Lévy processes allows us to consider a variety of signals with different types of sparsity. Some realizations of such processes are shown in Figure \ref{fig:realizations}.

\begin{figure}[h]
\centering
\includegraphics[width=0.8\textwidth]{realizations}
\caption{Realizations of different Lévy processes as characterized by the corresponding infinitely divisible pdfs.}
\end{figure}

\section{Discrete Stochastic Model}

Now, we derive the pdf of the random vector $s = (s(k))_{k=1}^K$, which contains uniform samples of a Lévy process. Consider the stationary increment process $u(t) = \{s(t) - s(t-1) : t \in \mathbb{R}^+\}$ whose first-order pdf $p_u$ is the same as $p_s(1)$ and so is infinitely divisible. Its samples $u = (u(k))_{k=1}^K$ can be expressed as

$$u = Ds,$$

where $D$ is a finite-difference matrix of the form

$$D = \begin{bmatrix}
1 & 0 & 0 & \cdots & 0 \\
-1 & 1 & 0 & \cdots & 0 \\
0 & -1 & 1 & \cdots & 0 \\
\vdots & \vdots & \vdots & \ddots & \vdots \\
0 & 0 & \cdots & -1 & 1
\end{bmatrix}.$$  

(9)

Using (8) and the fact that the increments are independent, we obtain the pdf of the discrete signal as

$$p_u(s) = \prod_{k=1}^K p_u(|Ds|_k).$$  

(10)

Note that (8) can also be written as

$$|s|_k = \sum_{n=1}^k |u|_n, \quad k = 1, \ldots, K,$$

which gives us a direct way to generate samples of Lévy processes.

\section{Bayesian Inference}

So far, we have introduced the signal and measurement models that allow us to generate our ground-truth signals and simulate their noisy measurements for a certain acquisition setup. Next, we focus on statistical estimators for the reconstruction problem at hand, which is to recover the signal $s$ from the measurements $y$.

In Bayesian inference, the goal is to characterize the posterior distribution $p_{s|y}$ and derive estimators based on it. Using Bayes’ rule and (10), we get

$$p_{s|y}(s|y) = \frac{p_{y|s}(y|s)p_u(s)}{\int_{\mathbb{R}^K} p_{y|s}(y|s)p_u(s) \, ds} \propto \exp\left(-\frac{\|y - Hs\|^2_2}{2\sigma^2_n}\right) \prod_{k=1}^K p_u(|Ds|_k).$$  

(12)

\subsection{Maximum a Posteriori Estimator}

The MAP estimator calculates the mode of the posterior distribution $p_{s|y}$ and is given by

$$\hat{s}_{\text{MAP}}(y) = \arg \max_{s \in \mathbb{R}^K} p_{s|y}(s|y)$$

$$= \arg \min_{s \in \mathbb{R}^K} \left(\frac{1}{2\sigma^2_n} \|y - Hs\|^2_2 + \sum_{k=1}^K \Phi_u(|Ds|_k)\right),$$  

(13)

where $\Phi_u(x) = -\log(p_u(x))$. The cost functional in (13) consists of a quadratic data-fidelity term and a penalty term that encodes the prior signal model. The optimization task in (13) resembles the one formulated in the variational model-based methods. For instance, if $p_u$ is a Gaussian pdf, then the penalty term is proportional to $\|Ds\|^2_2$, which is a classical Tikhonov regularizer \cite{2}. However, if $p_u$ is a Laplace pdf, then we have a sparsity-promoting $\ell_1$-norm penalty term $\|Ds\|_1$, which corresponds to the popular TV regularizer \cite{14}.
These MAP estimators can be computed efficiently with the help of iterative algorithms such as gradient descent, FISTA [12], and ADMM [13].

B. Minimum Mean-Square Error Estimator

The MMSE estimator is given by

$$\hat{s}_{\text{MMSE}}(y) = \arg \min_{s \in \mathbb{R}^n} \left( \int_{\mathbb{R}^K} \| s - \hat{s} \|^2 p_s|y(s|y) \, ds \right)$$

$$= \int_{\mathbb{R}^K} s \, p_s|y(s|y) \, ds,$$

(14)

which is the mean of the posterior distribution $p_{s|y}$. For a fixed stochastic model, the MMSE estimator is the optimal reconstructor in the MSE sense and thus serves as the goldstandard in our benchmarking framework.

In the Gaussian case, the MMSE estimator is known to coincide with the MAP estimator and is straightforward to calculate [1], [4]. However, in the non-Gaussian case, we need to numerically evaluate the high-dimensional integral in (14), which is computationally challenging.

V. MMSE ESTIMATORS FOR SPARSE LÉVY PROCESSES

In this section, we present efficient methods to compute the MMSE estimator for sparse Lévy processes with increments that follow the Laplace, Student’s t, and Bernoulli-Laplace distributions, which constitutes a key contribution of this paper.

A. Markov Chain Monte Carlo Methods

The MMSE estimator $\hat{s}_{\text{MMSE}}$ involves the calculation of the integral (14). The high dimensionality of this integral makes its approximation by simple techniques such as uniform-grid-based Riemann sums infeasible. Instead, one can use Markov Chain Monte Carlo (MCMC) methods [44]–[47] for the numerical approximation of (14) in a tractable manner.

MCMC methods are designed for generating random samples from nontrivial high-dimensional probability distributions. Broadly speaking, the idea in MCMC is to design a Markov chain such that the distribution that one wishes to draw samples from is its stationary distribution. The desired samples can be obtained by simulating the Markov chain and recording its states after convergence.

In order to compute the integral in (14), we first generate samples $\{s(q)\}_{q=1}^Q$ from $p_{s|y}$ using an MCMC method. We then approximate $\hat{s}_{\text{MMSE}}$ by the empirical mean $\hat{s}_Q = \frac{1}{Q} \sum_{q=1}^Q s(q)$. Although the collected samples are correlated, the Markov chain central limit theorem [48] guarantees that $\hat{s}_Q$ is a good approximation of $\hat{s}_{\text{MMSE}}$ for a large-enough $Q$.

B. Gibbs Sampling

In this work, we propose to use the MCMC method called Gibbs sampling [49], [50] to generate samples $\{u(q)\}_{q=1}^Q$ from the posterior distribution $p_{u|y}$. These can then be transformed in accordance with [1] to obtain samples $\{\mathbf{D}^{-1} u(q)\}_{q=1}^Q$ from $p_{u|y}$. We now give the gist of this algorithm.

Algorithm 1 Gibbs sampling

1: Input: $Q$ (number of samples), $B$ (burn-in period)
2: Initialization: $\{\bar{x}(0), \bar{y}(0)\}$
3: for $q = 1, \ldots, B + Q$ do
4:   Generate $\bar{x}(q) \sim p_{x|y}(x|y(q-1))$
5:   Generate $\bar{y}(q) \sim p_{y|x}(y|x(\bar{q}))$
6: end for
7: Output: $\{\left( x(q), y(q) \right) \}_{q=1}^Q = \left\{ \left( \bar{x}(q+B), \bar{y}(q+B) \right) \right\}_{q=1}^Q$

Let $x$ and $y$ be two random variables. Consider the task of generating samples from their joint distribution $p_{x,y}$. Gibbs sampling is advantageous whenever it is computationally difficult to sample from the joint distribution directly but the conditional distributions $p_{x|y}$ and $p_{y|x}$ are easy to sample from. The steps involved in this method are presented in Algorithm 1. They yield a Markov chain whose stationary distribution is indeed $p_{x,y}$ [50]. In practice, one discards some of the initial samples (burn-in period) to allow the chain to converge. Moreover, quantities (expectation integrals) based on the marginal distributions $p_x$ and $p_y$ can be computed from the individual samples $\{x(q)\}_{q=1}^Q$ and $\{y(q)\}_{q=1}^Q$, respectively.

Next, we present Gibbs sampling schemes for Lévy processes with Laplace, Student’s t, and Bernoulli-Laplace increments. Our strategy is to introduce an auxiliary vector $w$ and perform Gibbs sampling for the joint distribution $p_{u,w|y}$ [51], [52]. The key is to choose $w$ such that the conditional distributions $p_{u|w,y}$ and $p_{w|u,y}$ can be sampled from in an efficient manner.

Hereafter, we assume that the noise variance $\sigma^2_0$ and the parameters of the signal model are known.

C. Laplace Increments

For Lévy processes with Laplace increments, we adapt the approach that was developed in [53].

The pdf for the Laplace distribution is

$$p_u(u) = \frac{b}{2} \exp \left( -b |u| \right),$$

(15)

where $b$ is the scale parameter. The density in (15) can be expressed as a scale mixture of normal distributions [54], as

$$p_u(u) = \int_{\mathbb{R}} p_{u|w}(u|w)p_{w}(w) \, dw,$$

(16)

where

$$p_{u|w}(u|w) = \frac{1}{\sqrt{2\pi w}} \exp \left( -\frac{u^2}{2w} \right)$$

(17)

is the Gaussian pdf and

$$p_w(w) = \frac{b^2}{2} \exp \left( -\frac{b^2 w}{2} \right) \mathbb{I}_+(w)$$

(18)

is a mixing exponential pdf with $\lambda = 2/b^2$. This property

\footnotetext{The pdf of the exponential distribution is $p_{exp}(x) = (1/\lambda)e^{-x/\lambda} \mathbb{I}_+(x)$, where $\lambda > 0$ is the scale parameter.}
allows us to define an auxiliary random vector \( w \in \mathbb{R}^K \) with i.i.d. entries following the distribution \( p_w \) in (18), such that
\[
p_{u|w}(u|w) = \prod_{k=1}^{K} p_{u|w}(|u|_k | |w|_k), \tag{19}
\]
where \( p_{u|w} \) is shown in (17).

Due to the chain rule of probability (or the general product rule), the full joint distribution \( p_{y,u,w} \) can be written as
\[
p_{y,u,w}(y,u,w) = p_{y|u,w}(y|u,w)p_{u|w}(u|w)p_w(w).	ag{20}
\]

Consequently, the distribution \( p_{u,w|y} \) takes the form
\[
p_{u,w|y}(u,w|y) \propto \exp\left(-\frac{1}{2\sigma_n^2} \| y - Au \|_2^2 \right) \\
\times \prod_{k=1}^{K} |w|_k^{-\frac{1}{2}} \exp\left(-\frac{|u|_k^2}{2|w|_k^2} \right) \\
\times \prod_{k=1}^{K} b^2_2 \exp\left(-\frac{b_2^2|w|_k^2}{2} \right) \mathbb{I}_+ (|w|_k),
\tag{21}
\]
where \( A := HD^{-1} \).

Based on (21), the conditional distribution \( p_{u|w,y} \) is then obtained as
\[
p_{u|w,y}(u|w,y) \propto \exp\left(-\frac{1}{2\sigma_n^2} \| y - Au \|_2^2 \right) \\
\times \prod_{k=1}^{K} |w|_k^{-\frac{1}{2}} \exp\left(-\frac{|u|_k^2}{2|w|_k^2} \right) \\
+ u^T C_L(w) u),
\tag{22}
\]
where \( C_L(w) \) is a diagonal matrix with entries \((|w|_k^{-1})_{k=1}^{K}\).

Specifically, \( p_{u|w,y} \) is a multivariate Gaussian pdf with mean \( \mu = \sigma_n^{-2} (\sigma_n^{-2} A^T \Sigma + C_L(w))^{-1} A^T \Sigma y \) and covariance matrix \( \Sigma = (\sigma_n^{-2} A^T \Sigma + C_L(w))^{-1} \). There exist several methods for the efficient generation of samples from a multivariate Gaussian density [55]–[58].

The conditional distribution \( p_{w|u,y} \) is
\[
p_{w|u,y}(w|u,y) \propto \prod_{k=1}^{K} p_{w|u,y}(|w|_k | |u|_k,y),
\tag{23}
\]
where
\[
p_{w|u,y}(w|u,y) \propto \exp\left(-\frac{1}{2} \left( \frac{u^2}{w} + b^2 w \right) \right) \\
\times w^{-\frac{1}{2}} \mathbb{I}_+ (w),
\tag{24}
\]
belongs to the family of generalized inverse Gaussian distributions\footnote{The pdf of the generalized inverse Gaussian distribution is
\[ p_{gig}(x) = \frac{(\lambda_1/\lambda_2)^{\lambda_2/2} x^{\lambda_1 - 1} e^{-(\lambda_1 x + \lambda_2 x^2)/2}}{2\pi K_0(\sqrt{\lambda_1 \lambda_2})} \mathbb{I}_{+}(x), \]
where \( K_0 \) is the modified Bessel function of the second kind, \( \lambda_1 > 0 \), \( \lambda_2 > 0 \), and \( a \in \mathbb{R} \).}

Now, the conditional distribution \( p_{u|w,y}(u|w,y) \) turns out to be
\[
p_{u|w,y}(u|w,y) \propto \exp\left(-\frac{1}{2\sigma_n^2} \| y - Au \|_2^2 \right) \\
+ u^T C_T(w) u),
\tag{31}
\]
where \( C_T(w) \) is a diagonal matrix with entries \((|w|_k)^{-1})_{k=1}^{K}\). Similar to the Laplace case, \( p_{u|w,y} \) is a multivariate Gaussian
\footnote{The pdf of the gamma distribution is
\[ p_{gam}(x) = \frac{1}{\lambda_2^\lambda_1 \Gamma(\lambda_1)} x^{\lambda_1 - 1} e^{-x/\lambda_2} \mathbb{I}_{+}(x), \]
where \( \lambda_1 > 0 \) and \( \lambda_2 > 0 \) are the shape and scale parameters, respectively.}

### D. Student’s t Increments

The case of Student’s t increments can be handled by adapting the method shown in [60], which is in fact similar to the one we described for Laplace increments.

The Student’s t pdf is given by
\[
p_u(u) = \frac{\Gamma(\frac{\alpha+1}{2})}{\Gamma(\frac{\alpha}{2})} \frac{1}{\sqrt{\pi(1+u^2)^\frac{\alpha}{2}}},
\tag{25}
\]
where \( \alpha \) is the number of degrees of freedom and controls the tail of the distribution, and where \( \Gamma \) denotes the gamma function. It can also be expressed as
\[
p_u(u) = \int_{\mathbb{R}} p_{w|u}(w|u) p_w(w) \, dw,
\tag{26}
\]
where
\[
p_w(w) = (\frac{0.5}{\Gamma(\frac{\alpha}{2})}) w^{-\frac{\alpha}{2} - 1} \exp\left(-\frac{w}{2} \right) \mathbb{I}_+(w)
\tag{28}
\]
is the pdf of a gamma\footnote{The pdf of the gamma distribution is
\[ p_{gam}(x) = \frac{1}{\lambda_2^\lambda_1 \Gamma(\lambda_1)} x^{\lambda_1 - 1} e^{-x/\lambda_2} \mathbb{I}_{+}(x), \]
where \( \lambda_1 > 0 \) and \( \lambda_2 > 0 \) are the shape and scale parameters, respectively.} distribution. Again, we introduce an auxiliary vector \( w \in \mathbb{R}^K \) whose i.i.d. entries follow \( p_w \) defined in (28). It is such that
\[
p_{u|w} (u|w) = \prod_{k=1}^{K} p_{u|w} (|u|_k | |w|_k),
\tag{29}
\]
where \( p_{u|w} \) is defined in (27).

Here, the distribution \( p_{u|w,y} \) is given by
\[
p_{u|w,y}(u|w,y) \propto \exp\left(-\frac{1}{2\sigma_n^2} \| y - Au \|_2^2 \right) \\
\times \prod_{k=1}^{K} |w|_k^{-\frac{1}{2}} \exp\left(-\frac{|w|_k^2}{2} \right) \\
\times \prod_{k=1}^{K} |w|_k^{-\frac{1}{2}} \mathbb{I}_+ (|w|_k),
\tag{30}
\]
density with mean \( \mu = \sigma_n^{-2}(\sigma_n^{-2}A^TA + C_T(w))^{-1}A^TY \) and covariance matrix \( \mathbf{K} = (\sigma_n^{-2}A^TA + C_T(w))^{-1} \).

The distribution \( p_{w|u,y} \) is again separable and takes the form

\[
p_{w|u,y}(w|u,y) \propto \prod_{k=1}^{K} p_{w|u,y}(|w|_k|u_k,y),
\] (32)

where

\[
p_{w|u,y}(w|u,y) \propto \exp\left(-\frac{(1 + |u|^2)w}{2}\right) \times w^{\alpha + \frac{n}{2}} \mathbb{1}_+(w).
\] (33)

is a gamma distribution with \( \lambda_1 = \frac{\alpha + \frac{n}{2}}{2} \) and \( \lambda_2 = \frac{2}{(1 + u^2)^2} \), which can easily be sampled from.

### E. Bernoulli-Laplace Increments

In [61], Gibbs sampling schemes have been designed for a deconvolution problem where the underlying signal is an i.i.d. spike train that follows the Bernoulli-Gaussian distribution. Unfortunately, the Bernoulli-Gaussian distribution is not infinitely divisible and so is not compatible with our framework of Lévy processes. While there exists some work [62] on finitely divisible and so is not compatible with our framework

Based on this representation, we introduce two independent auxiliary vectors \( v \in \mathbb{R}^K \) and \( w \in \mathbb{R}^K \). Their elements are i.i.d. and follow the distributions \( p_v \) and \( p_w \), as defined in (36) and (37), respectively. Further, these vectors satisfy

\[
p_{u|v,w}(u|v,w) = \prod_{k=1}^{K} p_{u|v,w}(|u_k|v_k,|w_k|),
\] (40)

where \( p_{u|v,w} \) is defined in (38) and (39).

Here, the full joint distribution \( p_{Y,u,v,w} \) is given by

\[
p_{y,u,v,w}(y,u,v,w) = p_{y|u,v,w}(y|u,v,w)p_{u,v,w}(u,v,w)
= p_{y|u}(y|u)p_{u|v,w}(u|v,w)
\times p_v(v)p_w(w).
\] (41)

As a result, the distribution \( p_{u,v,w|y} \) takes the form

\[
p_{u,v,w|y}(u,v,w|y) \propto \exp \left(-\frac{1}{2\sigma_n^2} \|y - Au\|_2^2 \right)
\times \prod_{k=1}^{K} p_{u|v,w}(|u_k|v_k,|w_k|)
\times \prod_{k=1}^{K} \lambda^{-1} [v_k - (1 - \lambda) y_k]
\times \prod_{k=1}^{K} b^2 \exp \left(-\frac{b^2[w_k]}{2}\right) \mathbb{1}_+(|w_k|),
\] (42)

where \( A = \mathbf{HD}^{-1} \).

Let us now introduce some notations. For any binary vector \( q \in \mathbb{R}^K \), let \( I_{q,0} \) and \( I_{q,1} \) denote sets of indices such that \( q_k = 0 \) for \( k \in I_{q,0} \) and \( q_k = 1 \) for \( k \in I_{q,1} \). Further, let \( A(q) \) be the matrix constructed by taking the columns of \( A \) corresponding to the indices in \( I_{q,1} \). We then define the matrix

\[
B(q,r) = \sigma_n^2 I + A(q)C_{BL}(q,r)A(q)^T,
\]

where \( r \in \mathbb{R}^K \) is a vector with positive entries and \( C_{BL}(q,r) \) is a diagonal matrix with entries \( (r_k)_{k \in I_{q,1}} \). Here, we also introduce the vector \( q_{(-k)} = ([q_1], \ldots, [q_{k-1}, 1, q_{k+1}, \ldots, [q_K])^T \).

Lastly, for \( q \in \{0,1\} \), we define the vector \( q_{(-k)} \in \mathbb{R}^{K-1} \) such that \( q_{(-k)} = ([q_1], \ldots, [q_{k-1}, 1, q_{k+1}, \ldots, [q_K]^T \).

First, we look at the conditional distribution \( p_{u,v|w,y} \). From (38) and (37), we deduce that any sample from \( p_{u,v|w,y} \) takes the value of zero at the indices in \( I_{v,0} \). If we define \( u_1 = ([u_k])_{k \in I_{v,1}} \), then we get

\[
p_{u_1|v,w,y}(u_1|v,w,y) \propto \exp \left(-\frac{1}{2\sigma_n^2} \|y - A(v)u_1\|_2^2 \right)
\times u_1^T C_{BL}(v,w)u_1.
\] (43)

Thus, \( p_{u|v,w,y} \) is a multivariate Gaussian density with mean

\[
\mu = \sigma_n^{-2}(\sigma_n^{-2}A(v)^TA(v) + C_{BL}(v,w))^{-1}A(v)^TY
\]

and covariance matrix \( \mathbf{K} = (\sigma_n^{-2}A(v)^TA(v) + C_{BL}(v,w))^{-1} \).

The conditional distribution \( p_{w|u,v,y} \) takes the form

\[
p_{w|u,v,y}(w|u,v,y) \propto \prod_{k=1}^{K} p_{w|u,v,y}(|w_k|u_k,|v_k|,|v_k|,|y_k|),
\] (44)

where \( p_{w|u,v,y} \) is given by

\[
p_{w|u,v,y}(w|u,v,y) \propto \frac{b^2}{2} \exp \left(-\frac{b^2[w]}{2}\right) \mathbb{1}_+(w),
\] (45)

\[
p_{w|u,v,y}(w|u,v,y) \propto \exp \left(-\frac{1}{2\sigma_n^2} \|y - A(v)u_1\|_2^2 \right)
\times u_1^T C_{BL}(v,w)u_1.
\] (46)
The densities in (45) and (46) correspond to the exponential distribution with $\lambda = 2/b^2$ and the generalized inverse Gaussian distribution with $\lambda_1 = b^2$, $\lambda_2 = u^2$, and $a = 0.5$.

Next, inspired by the work in [61], we consider sampling from the marginalized conditional distribution of $v$ in a sequential manner as this can allow for a more efficient exploration of configurations of $v$. More specifically, at each iteration $q$, we draw $[v^{(q)}]_k$ from the distribution $p[v^{(q)}]_k|v^{(-k)}, w, y] = \mathbf{v}^{(q)}_k, u^{(q-1)}$, where $v^{(q)}_k = ([v^{(q)}]_1, \ldots, [v^{(q)}]_{k-1}, [v^{(q-1)}]_{k+1}, \ldots, [v^{(q-1)}]_K)$ and $k \in \{1, \ldots, K\}$.

The marginalized posterior distribution $p_v, w, y$ is given by

$$p_v, w, y(v, w|y) \propto p_v|v, w(y|v, w)p_v(v)p_w(w),$$

where

$$p_y|v, w(y|v, w) = \int_{\mathbb{R}^K} p_y|u, v, w(y|u, v, w)p_u|v, w(u|v, w)v dv du.$$ 

It can be shown that (47) and (48) lead to

$$p_v, w, y(v, w|y) \propto [B(v, w)]^{-\frac{1}{2}} \exp \left( -\frac{1}{2} y^T B(v, w)^{-1} y \right)$$

$$\times \prod_{k=1}^K \lambda^{-[v_k]} (1 - \lambda) [v_k]$$

$$\times \prod_{k=1}^K \frac{b^2}{2} \exp \left( -\frac{b^2 [w_k]}{2} \right) \mathbb{I}([w_k]).$$

From (49), we see that $p[v|v^{(-k)}, w, y]$ is a Bernoulli distribution with

$$p[v|v^{(-k)}, w, y(v|v^{(-k)}, w, y) = \left( 1 + \exp \left( -\frac{1}{\lambda} h(v^{(-k)}, w, y) \right) \right)^{-1},$$

where

$$h(v^{(-k)}, w, y) = y^T B(v^{(-k)}, w)^{-1} y + \log \left( |B(v^{(-k)}, w)| \right) + 2v \log \left( \frac{\lambda}{1 - \lambda} \right).$$

To summarize, in each iteration $q$ of the above-described sampler, we generate $w^{(q)} \sim p_w|u, v, y(w|u^{(q-1)}, v^{(q-1)}, y)$, $v^{(q)}_k \sim p[v^{(q)}|v^{(-k)}, w, y(v^{(q)}_k|v^{(-k)}, w, y(q)_k, u^{(q-1)}_k)$ for all $k$ and $u^{(q)} \sim p_u|v, w, y(u|v^{(q)}_k, w(q))$. This particular order of updates is important as it yields a partially collapsed Gibbs sampler [63] where the stationary distribution is still $p_u, v, w, y$.

VI. EXPERIMENTAL RESULTS

A. Methods

In our experiments, we benchmark the performance of a CNN-based direct nonlinear reconstruction scheme on deconvolution and Fourier sampling problems with Lévy processes associated with the Bernoulli-Laplace and Student’s $t$ distributions, respectively. This method involves the training of the CNN as a regression model that maps an initial low-quality reconstruction $\hat{s}_0$ to a high-quality one $\hat{s}_{\text{CNN}}$ [17–[21]. The architecture of our CNN is based on the well-known denoising network DnCNN [64] and is described in Figure 2 and Table 1. Given a dataset $\{s_m, y_m\}_{m=1}^{M_F}$ of ground-truth signals and their corresponding measurements, we train the model by minimizing the MSE loss function

$$L(\theta) = \frac{1}{M_F} \sum_{m=1}^{M_F} \|s_m - \hat{s}_{\text{CNN}}(\theta; \hat{s}_0(y_m))\|_2^2,$$

where $\theta$ represents the learnable parameters of the network, with the help of the ADAM optimizer [65]. This scheme is implemented in PyTorch.

We compare the CNN-based method with the MMSE estimators for the underlying signal models and with the model-based techniques

$$\hat{s}_{\ell_2} = \mathbf{arg min}_{a \in \mathbb{R}^K} \left( \|y - Ha\|_2^2 + \tau \|Da\|_2^2 \right),$$

$$\hat{s}_{\ell_1} = \mathbf{arg min}_{a \in \mathbb{R}^K} \left( \|y - Ha\|_2^2 + \tau \|Da\|_1 \right),$$

and

$$\hat{s}_{\text{log}} = \mathbf{arg min}_{a \in \mathbb{R}^K} \left( \|y - Ha\|_2^2 + \tau \sum_{k=1}^K \log \left( 1 + ([Da]_k)^2 \right) \right),$$

where $\tau \in \mathbb{R}_+$. Equations (53), (54) and (55) resemble the MAP estimators of Lévy processes associated with Gaussian, Laplace, and Student’s $t$ distributions, respectively. However, unlike the MAP estimators, these include an adjustable hyperparameter $\tau$, which we tune to optimize the performance. The $\ell_2$ estimator can be expressed in closed-form as

$$\hat{s}_{\ell_2} = (H^T H + \tau D^T D)^{-1} H^T y.$$
Meanwhile, the $\ell_1$ and log estimators are computed using ADMM. Since the cost functional in (55) is non-convex, we initialize ADMM for $\hat{s}_{\log}$ with $\hat{s}_{\ell_1}$ so that it can reach a better local minimum. The MMSE and variational estimators are implemented in MATLAB. For the variational ones, we use GlobalBioIm [66]—a library for solving inverse problems.

B. Deconvolution

We first consider a deconvolution problem where the signal vector $s \in \mathbb{R}^{100}$ contains samples of a Lévy process whose increments follow the Bernoulli-Laplace distribution. As shown in Section II-C, the system matrix $H$ for this case turns out to be a discrete convolution-like matrix. Accordingly, we construct $H: \mathbb{R}^{100} \rightarrow \mathbb{R}^{88}$ such that

$$H = \begin{bmatrix} [h]_{13} & \cdots & [h]_1 & 0 & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 & \ddots & \vdots \\ 0 & \cdots & 0 & [h]_{13} & \cdots & [h]_1 \end{bmatrix},$$

(57)

where $h \in \mathbb{R}^{13}$ consists of the central samples of a truncated Gaussian PSF with variance $\sigma_0^2 = 4$. The Bernoulli-Laplace pdf [34] is characterized by the parameters $\lambda$ and $b$, where $\lambda$ determines the mass probability at the origin and $b$ represents the scale of the Laplace component. For a specific set of values of $\lambda$, $b$, and of the AWGN variance $\sigma_n^2$, we generate training, validation, and test datasets consisting of 200,000, 1,000 and 1,000 pairs of ground-truth signals and their noisy measurements, respectively.

For the CNN-based approach, we set $F = 5$, $C = 64$, and $L = 10$. We choose the initial low-quality reconstruction to be $\hat{s}_0(y) = H^T y$. The CNN is trained for 1,000 epochs with a batch size of 256 and a weight decay of $5 \times 10^{-4}$. The initial learning rate is set as $10^{-2}$, and during the first 600 epochs, it is decreased by a factor of 0.5 every 50 epochs. For each variational method, the same regularization parameter $\tau$ is used for the entire test dataset. This particular value of $\tau$ is the one that yields the lowest MSE for the validation dataset. For the Gibbs-sampling-based MMSE estimator, we set the number of samples as $Q = 8,000$ and the burn-in period as $B = 3,000$.

We perform the above-described deconvolution experiment for $\lambda \in \{0.6, 0.7, 0.8, 0.9\}$. In each case, the scale parameter is set to $b = 1$ and $\sigma_n^2$ is chosen such that the (average) measurement SNR is around 30 dB. The results for the test dataset are shown in Figure 3.

The sparsity-promoting $\ell_1$ estimator, which corresponds to the popular TV regularization, is known to be well-suited to piecewise-constant Lévy processes with Bernoulli-Laplace increments. As the scale of $\lambda$ increases, these signals become sparser and exhibit fewer jumps. Consequently, we observe that the $\ell_1$ estimator performs better than the $\ell_2$ estimator. The log estimator also promotes sparse solutions [67] and we see that it performs well for these piecewise-constant signals. However, despite the good fit, there is still some gap between the MSE attained by the $\ell_1$ and log, and MMSE estimators. Remarkably, the CNN-based method consistently outperforms them and achieves a near-optimal MSE.

C. Fourier Sampling

Next, we look at Fourier sampling in one dimension, which is reminiscent of MRI. Here, the signal vector $s \in \mathbb{R}^{100}$ represents samples of a Lévy process associated with the Student’s t distribution. In this case, the forward model $H$ resembles a discrete Fourier matrix (see Section II-D). Thus, in order to construct $H$, we first sample $M' = 16$ rows of the DFT matrix. Except for the DC component, which we always keep, the remaining $(M' - 1)$ rows are selected in a quasi-random fashion such that there is a denser sampling at low frequencies. We then create the real system matrix $H: \mathbb{R}^{100} \rightarrow \mathbb{R}^M$, where $M = 2M' - 1$, by treating separately the real and imaginary parts. The Student’s t distribution [25] is parameterized by $\alpha$, which controls the tails of the distribution. For chosen values of $\alpha$ and the AWGN variance $\sigma_n^2$, we create training, validation, and test datasets with 200,000, 1,000 and 1,000 samples, respectively.

The parameters for our CNN architecture are set as $F = 7$, $C = 64$, and $L = 15$. Here, the initial estimate is $\hat{s}_0(y) = (H^T y/K)$, and the CNN is trained for 1,000 epochs with a batch size of 256. The weight decay is fixed as $10^3$ when $\alpha = 1$ and as $10^{-3}$ for other values of $\alpha$. The initial learning rate is set as $10^{-2}$, and during the first 600 epochs, it is decreased by a factor of 0.5 every 50 epochs. The regularization parameters for the variational methods are determined in the same way as for the deconvolution experiments. For the MMSE estimator, the number of samples is $Q = 15,000$ and the burn-in period is $B = 5,000$.

We perform the Fourier-sampling experiment for $\alpha \in \{1, 3, 5, 39\}$. For each value of $\alpha$, the noise variance $\sigma_n^2$ is such that the (average) measurement SNR is around 30 dB. The results for the test dataset are shown in Figure 4.

The parameter $\alpha$ allows us to consider a wide range of signals. On one hand, as $\alpha \rightarrow \infty$, we approach the
Gaussian regime. Thus, we observe that the $\ell_2$ estimator is optimal for a large value of $\alpha$. On the other hand, $\alpha = 1$ corresponds to an extremely heavy-tailed (sparse) Cauchy distribution. Consequently, as the value of $\alpha$ decreases, the relative performance of the $\ell_1$ estimator improves while that of the $\ell_2$ estimator deteriorates. For all the cases, the log estimator, which corresponds to a tunable MAP estimator for the Student’s t distribution, attains reasonable MSE values. Interestingly, the CNN-based method performs well up to $\alpha = 3$, after which there seems to be a steep transition and its performance drops sharply. In fact, for Cauchy signals, the CNN-based method performs worse than the $\ell_1$ estimator. A possible explanation for the poor performance of CNNs is that, as the distribution becomes more heavy-tailed, the signals exhibit a vastly different range of values and the training becomes very difficult.

VII. Conclusion

We have introduced a controlled environment, based on the $\ell_1$ estimator improves while that of the $\ell_2$ estimator deteriorates. For all the cases, the log estimator, which corresponds to a tunable MAP estimator for the Student’s t distribution, attains reasonable MSE values. Interestingly, the CNN-based method performs well up to $\alpha = 3$, after which there seems to be a steep transition and its performance drops sharply. In fact, for Cauchy signals, the CNN-based method performs worse than the $\ell_2$ estimator. A possible explanation for the poor performance of CNNs is that, as the distribution becomes more heavy-tailed, the signals exhibit a vastly different range of values and the training becomes very difficult.

VII. Conclusion

We have introduced a controlled environment, based on sparse stochastic processes (SSPs), for the objective benchmarking of neural-network methods in the context of linear inverse problems. We have developed efficient posterior sampling schemes to compute the minimum-mean-square-error estimators for specific classes of SSPs. These yield the upper limit on reconstruction performance and allow us to provide a measure of statistical optimality. We have highlighted the abilities of our framework by benchmarking some convolutional neural-network (CNN) architectures for deconvolution and Fourier-sampling problems. In particular, we have shown that, while CNNs achieve a near-optimal performance in terms of mean-square error for a wide range of conditions, they can sometimes fail too, especially for signals with heavy-tailed innovations.

VIII. Acknowledgements

We acknowledge access to the facilities and expertise of the CIBM Center for Biomedical Imaging, a Swiss research center of excellence founded and supported by Lausanne University Hospital (CHUV), University of Lausanne (UNIL), École polytechnique fédérale de Lausanne (EPFL), University of Geneva (UNIGE), and Geneva University Hospitals (HUG).

REFERENCES

[1] M. Unser and M. T. McCann, “Biomedical image reconstruction: From the foundations to deep neural networks,” Foundations and Trends in Signal Processing, vol. 13, pp. 280–359, 2019.
[2] A. N. Tikhonov, “Solution of incorrectly formulated problems and the regularization method,” Soviet Mathematics, vol. 4, pp. 1035–1038, 1963.
[3] M. Bertero and P. Boccacci, Introduction to Inverse Problems in Imaging. CRC Press, 1998.
[4] S. M. Kay, Fundamentals of Statistical Signal Processing: Estimation Theory. Prentice-Hall, Inc., 1993.
[5] S. Mallat, A Wavelet Tour of Signal Processing. Elsevier, 1999.
[6] A. M. Bruckstein, D. L. Donoho, and M. Elad, “From sparse solutions of systems of equations to sparse modeling of signals and images,” SIAM Review, vol. 51, no. 1, pp. 34–81, 2009.
[7] R. G. Baraniuk, E. Candès, M. Elad, and Y. Ma, “Applications of sparse representation and compressive sensing [scanning the issue],” Proceedings of the IEEE, vol. 98, no. 6, pp. 906–909, 2010.
[8] M. Elad, Sparse and Redundant Representations: From Theory to Applications in Signal and Image Processing. Springer, 2010, vol. 2, no. 1.
[9] D. L. Donoho, “Compressed sensing.” IEEE Transactions on Information Theory, vol. 52, no. 4, pp. 1289–1306, 2006.
[10] E. J. Candès and M. B. Wakin, “An introduction to compressive sampling,” IEEE Signal Processing Magazine, vol. 25, no. 2, pp. 21–30, 2008.
[11] S. Foucart and H. Rauhut, “An invitation to compressive sensing,” in A Mathematical Introduction to Compressive Sensing. Springer, 2013, pp. 1–39.
[12] A. Beck and M. Teboulle, “A fast iterative shrinkage-thresholding algorithm for linear inverse problems,” SIAM Journal on Imaging Sciences, vol. 2, no. 1, pp. 183–202, 2009.
[13] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, “Distributed optimization and statistical learning via the alternating direction method of multipliers,” Foundations and Trends in Machine Learning, vol. 3, no. 1, pp. 1–122, 2011.
[14] L. I. Rudin, S. Osher, and E. Fatemi, “Nonlinear total variation based noise removal algorithms,” Physica D: Nonlinear Phenomena, vol. 60, no. 1-4, pp. 259–268, 1992.
[15] M. T. McCann, K. H. Jin, and M. Unser, “Convolutional neural networks for inverse problems in imaging: A
review,” *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 85–95, 2017.

[16] G. Ongie, A. Jalal, C. A. Metzler, R. G. Baraniuk, A. G. Dimakis, and R. Willett, “Deep learning techniques for inverse problems in imaging,” *IEEE Journal on Selected Areas in Information Theory*, vol. 1, no. 1, pp. 39–56, 2020.

[17] K. H. Jin, M. T. McCann, E. Froustey, and M. Unser, “Deep convolutional neural network for inverse problems in imaging,” *IEEE Transactions on Image Processing*, vol. 26, no. 9, pp. 4509–4522, 2017.

[18] H. Chen, Y. Zhang, M. K. Kalra, F. Lin, Y. Chen, P. Liao, J. Zhou, and G. Wang, “Low-dose CT with a residual encoder-decoder convolutional neural network,” *IEEE Transactions on Medical Imaging*, vol. 36, no. 12, pp. 2524–2535, 2017.

[19] C. M. Hyun, H. P. Kim, S. M. Lee, S. Lee, and J. K. Seo, “Deep learning for undersampled MRI reconstruction,” *Physics in Medicine & Biology*, vol. 63, no. 13, p. 135007, 2018.

[20] K. Monakhova, J. Yurtsever, G. Kuo, N. Antipa, K. Yanny, and L. Waller, “Learned reconstructions for practical mask-based lensless imaging,” *Optics Express*, vol. 27, no. 20, pp. 28075–28090, 2019.

[21] D. Perdios, M. Vonlanthen, F. Martinez, M. Arditi, and J.-P. Thiran, “CNN-based image reconstruction method for ultrafast ultrasound imaging,” *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency Control*, vol. 69, no. 4, pp. 1154–1168, 2021.

[22] K. Gregor and Y. LeCun, “Learning fast approximations of sparse coding,” in *Proceedings of the 27th International Conference on Machine Learning*, 2010, pp. 399–406.

[23] Y. Chen and T. Pock, “Trainable nonlinear reaction diffusion: A flexible framework for fast and effective image restoration,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1256–1272, 2016.

[24] Y. Yang, J. Sun, H. Li, and Z. Xu, “Deep ADMM-Net for compressive sensing MRI,” *Advances in Neural Information Processing Systems*, vol. 29, 2016.

[25] H. K. Aggarwal, M. P. Mani, and M. Jacob, “MoDL: Model-based deep learning architecture for inverse problems,” *IEEE Transactions on Medical Imaging*, vol. 38, no. 2, pp. 394–405, 2018.

[26] J. Adler and O. Öktem, “Learned primal-dual reconstruction,” *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1322–1332, 2018.

[27] V. Monga, Y. Li, and Y. C. Eldar, “Algorithm unrolling: Interpretable, efficient deep learning for signal and image processing,” *IEEE Signal Processing Magazine*, vol. 38, no. 2, pp. 18–44, 2021.

[28] S. V. Venkatakrishnan, C. A. Bouman, and B. Wohlberg, “Plug-and-play priors for model based reconstruction,” in *2013 IEEE Global Conference on Signal and Information Processing*, 2013, pp. 945–948.

[29] E. Ryu, J. Liu, S. Wang, X. Chen, Z. Wang, and W. Yin, “Plug-and-play methods provably converge with properly trained denoisers,” in *International Conference on Machine Learning*, 2019, pp. 5546–5557.

[30] K. Zhang, Y. Li, W. Zuo, L. Zhang, L. Van Gool, and R. Timofte, “Plug-and-play image restoration with deep denoiser prior,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2021.

[31] Y. Sun, Z. Wu, X. Xu, B. Wohlberg, and U. S. Kamilov, “Scalable plug-and-play admr with convergence guarantees,” *IEEE Transactions on Computational Imaging*, vol. 7, pp. 849–863, 2021.

[32] Y. Romano, M. Elad, and P. Milanfar, “The little engine that could: Regularization by denoising (RED),” *SIAM Journal on Imaging Sciences*, vol. 10, no. 4, pp. 1804–1844, 2017.

[33] Y. Sun, J. Liu, and U. S. Kamilov, “Block coordinate regularization by denoising,” in *Advances in Neural Information Processing Systems*, 2019, pp. 382–392.

[34] Z. Wu, Y. Sun, A. Matlock, J. Liu, L. Tian, and U. S. Kamilov, “SIMBA: Scalable inversion in optical tomography using deep denoising priors,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 14, no. 6, pp. 1163–1175, 2020.

[35] J. Rick Chang, C.-L. Li, B. Poczos, B. Vijaya Kumar, and A. C. Sankaranarayanan, “One network to solve them all—Solving linear inverse problems using deep projection models,” in *IEEE International Conference on Computer Vision*, 2017, pp. 5888–5897.

[36] H. Gupta, K. H. Jin, H. Q. Nguyen, M. T. McCann, and M. Unser, “CNN-based projected gradient descent for consistent CT image reconstruction,” *IEEE Transactions on Medical Imaging*, vol. 37, no. 6, pp. 1440–1453, 2018.

[37] M. Unser and P. D. Tafti, *An Introduction to Sparse Stochastic Processes*. Cambridge, United Kingdom: Cambridge University Press, 2014, 367 p.

[38] U. S. Kamilov, P. Pad, A. Amini, and M. Unser, “MMSE estimation of sparse Lévy processes,” *IEEE Transactions on Signal Processing*, vol. 61, no. 1, pp. 137–147, 2012.

[39] A. Amini, U. S. Kamilov, E. Bostan, and M. Unser, “Bayesian estimation for continuous-time sparse stochastic processes,” *IEEE Transactions on Signal Processing*, vol. 61, no. 4, pp. 907–920, 2012.

[40] M. Unser, “Sampling—50 Years after Shannon,” *Proceedings of the IEEE*, vol. 88, no. 4, pp. 569–587, 2000.

[41] S. Ken-Iti, *Lévy Processes and Infinitely Divisible Distributions*. Cambridge University Press, 1999.

[42] A. Amini and M. Unser, “Sparsity and infinite divisibility,” *IEEE Transactions on Information Theory*, vol. 60, no. 4, pp. 2346–2358, 2014.

[43] A. Amini, M. Unser, and F. Marvasti, “Compressibility of deterministic and random infinite sequences,” *IEEE Transactions on Signal Processing*, vol. 59, no. 11, pp. 5193–5201, November 2011.

[44] W. K. Hastings, “Monte Carlo sampling methods using Markov chains and their applications,” 1970.

[45] C. J. Geyer, “Practical Markov chain Monte Carlo,” *Statistical Science*, pp. 473–483, 1992.

[46] W. R. Gilks, S. Richardson, and D. Spiegelhalter, *Markov chain Monte Carlo in Practice*. CRC press, 1995.
[47] D. Gamerman and H. F. Lopes, *Markov chain Monte Carlo: Stochastic Simulation for Bayesian Inference*. CRC press, 2006.

[48] M. I. Gordin and B. A. Lifšic, “The central limit theorem for stationary Markov processes,” in *Doklady Akademii Nauk*, vol. 239, no. 4, 1978, pp. 766–767.

[49] S. Geman and D. Geman, “Stochastic relaxation, Gibbs distributions, and the Bayesian restoration of images,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, no. 6, pp. 721–741, 1984.

[50] G. Casella and E. I. George, “Explaining the Gibbs sampler,” *The American Statistician*, vol. 46, no. 3, pp. 167–174, 1992.

[51] M. A. Tanner and W. H. Wong, “The calculation of posterior distributions by data augmentation,” *Journal of the American Statistical Association*, vol. 82, no. 398, pp. 528–540, 1987.

[52] A. Mira and L. Tierney, “On the use of auxiliary variables in Markov chain Monte Carlo sampling,” *Scandinavian Journal of Statistics*, 1997.

[53] T. Park and G. Casella, “The Bayesian LASSO,” *Journal of the American Statistical Association*, vol. 103, no. 482, pp. 681–686, 2008.

[54] D. F. Andrews and C. L. Mallows, “Scale mixtures of normal distributions,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 36, no. 1, pp. 99–102, 1974.

[55] H. Rue, “Fast sampling of Gaussian Markov random fields,” *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, vol. 63, no. 2, pp. 325–338, 2001.

[56] F. Orieux, O. Féron, and J.-F. Giovannelli, “Sampling high-dimensional Gaussian distributions for general linear inverse problems,” *IEEE Signal Processing Letters*, vol. 19, no. 5, pp. 251–254, 2012.

[57] C. Gilavert, S. Moussaoui, and J. Idier, “Efficient Gaussian sampling for solving large-scale inverse problems using MCMC,” *IEEE Transactions on Signal Processing*, vol. 63, no. 1, pp. 70–80, 2014.

[58] M. Vono, N. Dobigeon, and P. Chainais, “High-dimensional Gaussian sampling: A review and a unifying approach based on a stochastic proximal point algorithm,” *SIAM Review*, vol. 64, no. 1, pp. 3–56, 2022.

[59] L. Devroye, “Random variate generation for the generalized inverse Gaussian distribution,” *Statistics and Computing*, vol. 24, no. 2, pp. 239–246, 2014.

[60] C. Fevotte and S. J. Godsill, “A Bayesian approach for blind separation of sparse sources,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 6, pp. 2174–2188, 2006.

[61] D. Ge, J. Idier, and E. Le Carpentier, “Enhanced sampling schemes for MCMC based blind Bernoulli–Gaussian deconvolution,” *Signal Processing*, vol. 91, no. 4, pp. 759–772, 2011.

[62] L. Chaari, J.-Y. Tourneret, and H. Batatia, “Sparse Bayesian regularization using Bernoulli-Laplacian priors,” in *21st European Signal Processing Conference (EUSIPCO 2013)*, 2013, pp. 1–5.